Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής & Υπολογιστών

Χρήση των Generative Adversarial Networks για επιλογή πόζας και στιλ σε εφαρμογές σχεδιασμού μόδας

Διπλωματική Εργασία

του

Αθανάσιου Χαρισούδη

Επιβλέποντες: κ. Περικλής Μήτκας
Καθηγητής Α.Π.Θ., ISSEL
κ. Αντώνης Χρυσόπουλος
Μεταδιδακτορικός Ερευνητής, ISSEL

22 Ιουλίου 2021

Περίληψη

Η Παραγωγική Μοντελοποίηση αποτελεί τον κλάδο της Μηχανικής Μάθησης ο οποίος επικεντρώνεται στην παραγωγή νέων ρεαλιστικών δεδομένων και ο οποίος αποτελούσε παραδοσιακά το ανώφλι των δυνατοτήτων των μοντέλων Μηχανικής και Βαθιάς Μάθησης. Το σκηνικό έχει αλλάξει τα τελευταία χρόνια, κυρίως από το 2014, όταν ο I. Goodfellow παρουσίασε την ιδέα του για ένα παραγωγικό μοντέλο αποτελούμενο από δύο αντιμαχόμενα νευρωνικά δίκτυα, το οποίο ονομάστηκε Generative Adversarial Network. Ακολούθως, προέκυψε πληθώρα μοντέλων βασιζόμενα σε αυτό, με εντυπωσιακά αποτελέσματα που, ειδικά στο πλαίσια της παραγωγής εικόνων, εκπλήσσουνε ακόμα και ένα έμπειρο ανθρώπινο σύστημα όρασης.

Ταυτόχρονα, τις τελευταίες δεκαετίες, γίνεται όλο και περισσότερη έρευνα γύρω από την ανάπτυξη τεχνικών κατανόησης της έννοιας της μόδας και της τάσης αυτής, είτε για συνδρομή κατά τη φάση σχεδίασης ρούχων ή για καλύτερες και πιο στοχευμένες αγορές ή για άλλους σκοπούς. Στην προσπάθειά μας να εφαρμόσουμε σύγχρονες τεχνικές μηχανικής μάθησης για αυτοματοποιημένη παραγωγή και επεξεργασία εικόνων μόδας, στην παρούσα εργασία, χρησιμοποιούμε Generative Adversarial Networks. Συγκεκριμένα, σχεδιάζουμε και υλοποιούμε ένα πολυ-εργαλείο αυτοματοποιημένης επεξεργασίας εικόνων μόδας το οποίο οπλίζουμε με τέσσερις (4) βασικές λειτουργίες: αλλαγή πόζας, εξαγωγή ρούχου, ταίριασμα στιλ και παραγωγή ρεαλιστικών εικόνων κατ΄ απαίτηση.

Στην προσπάθειά μας αυτή, εκπαιδεύουμε ισάριθμα μοντέλα τύπου Generative Adversarial Network σε σύνολα δεδομένων αποτελούμενα από εικόνες μόδας (ενν. ρούχων και μοντέλων που τα διαφημίζουν), παραθέτοντας στο τέλος σχετικά αποτελέσματα. Αποτελεί βαθιά πεποίθησή μας ότι εξελίξεις παρόμοιων μοντέλων θα κατέχουν κεντρικό ρόλο στη σχεδίαση ρούχων και κυρίως στη διάθεσή τους μέσω συστημάτων ηλεκτρονικού εμπορίου στο εγγύς μέλλον, κάτι που μας έκανε να προσηλωθούμε με ζήλο στην υλοποίηση ενός αποτελεσματικού νοήμονος εργαλείου για επεξεργασία εικόνων μόδας στην παρούσα εργασία.

Λέξεις κλειδιά— Παραγωγική Μοντελοποίηση, Νοήμονα συστήματα μόδας, Generative Adversarial Networks, Παραγωγή εικόνων από θόρυβο, Μετατροπή εικόνας-σε-εικόνα, DeepFashion, StyleGAN, CycleGAN, Μηχανική Όραση, Τεχνητά Νευρωνικά Δίκτυα

Abstract

Generative Modelling, a branch of Machine Learning that focuses on generating realistic-looking samples, has traditionally constituted the upper bound of what Machine and Deep Learning models can achieve. This regime has completely changed the past years, especially after 2014, when I. Goodfellow presented his idea for a generative model comprising two competing neural networks: the Generative Adversarial Network of GAN for short. Subsequently, a plethora of models based on GAN have been proposed with impressive results, some of which, principally in the context of image generation, surprise even an experienced human vision system.

Concurrently, more and more research is devoted during the last decades around the development of techniques for demystifying the notion of fashion and fashion trends. Among its purposes, is creating artificial intelligence systems that provide help in the process of designing new garments as well as in the process of conducting better and more well-targeted purchases. In an endeavour to apply modern machine learning techniques to automate generation and editing of fashion images, in this project we employ Generative Adversarial Networks. In particular, we design and utilize a multi-tool for automatic editing of fashion images, equipped with four (4) fundamental operations: pose change, cloth extraction, style matching and on-demand realistic fashion images generation.

In order to achieve our goals, we train four models based on the Generative Adversarial Network in fashion image (i.e. images of garments as well as human models advertising them) datasets, giving the corresponding outcomes at the end. It is our firm belief that further developments of such models will play a central role in fashion design and especially in clothes distribution through e-commerce systems in the near future, which has made us focus zealously on implementing an effective intelligent tool for fashion image editing in this work.

Keywords— Generative Modelling, Intelligent fashion systems, Generative Adversarial Networks, Noise-to-image generation, Image-to-image translation, DeepFashion, StyleGAN, CycleGAN, Computer Vision, Artificial Neural Networks

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον κ. Μήτκα Περικλή και τον κ. Αντώνη Χρυσόπουλο για την εμπιστοσύνη που μου έδειξαν τόσο κατά την ανάθεση της παρούσας διπλωματικής εργασίας όσο και κατά την εκπόνησή της. Επίσης, τους ευχαριστώ θερμά για την αμέριστη υποστήριξη και καθοδήγηση που μου παρείχαν έως και την τελευταία ημέρα εκπόνησης αυτής.

Επίσης, θέλω να ευχαριστήσω τον κ. Αλέξανδρο Κυπριανίδη και τον κ. Σωτήρη Τσαρούχη για την πολύτιμη βοήθεια που μου παρείχαν, και τη γενικότερη συμβουλευτική τους παρουσία.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, την αδελφή και τον αδελφό μου, που είναι πάντα δίπλα μου και με στηρίζουν σε όλη τη διάρκεια της ζωής μου, καθώς και τους φίλους μου, τους συμφοιτητές μου και ένα δικό μου πρόσωπο, που γέμισαν τα φοιτητικά μου χρόνια με αξέχαστες εμπειρίες.

Χρήση των Generative Adversarial Networks για επιλογή πόζας και στιλ σε εφαρμογές σχεδιασμού μόδας

Αθανάσιος Χαρισούδης
achariso@ece.auth.gr

22 Ιουλίου 2021

Περιεχόμενα

Κατάλογος Πινάκων

Κατάλογος Σχημάτων

Κεφάλαιο 1

Σχετικά με τη Διπλωματική Εργασία

Αντικείμενο και Σκοπός

Στην παρούσα διπλωματική εργασία ασχολούμαστε με Παραγωγική Μοντελοποίηση εικόνων, δηλαδή προσπαθούμε να εκπαιδεύσουμε μοντέλα Μηχανικής/Βαθιάς Μάθησης ώστε να παράγουν ρεαλιστικές εικόνες. Για να το πετύχουμε αυτό, σχεδιάζουμε τα μοντέλα με τέτοιο τρόπο ώστε να μπορούν να αφομοιώσουν τα βασικά χαρακτηριστικά και τη δομή των εικόνων ενός συνόλου δεδομένων εκπαίδευσης. Στα πλαίσια αυτά, χρησιμοποιούμε Βαθιά Παραγωγικά Μοντέλα τύπου Generative Adversarial Network (GAN), τα οποία αποτελούνται από δύο νευρωνικά δίκτυα και εκπαιδεύονται αντιπαραθετικά: όσο το ένα δίκτυο γίνεται καλύτερο τόσο το άλλο μεταβάλλει τις παραμέτρους του για να ανακτήσει την κυριαρχία.

Έτσι, σχεδιάσαμε και υλοποιήσαμε μοντέλα GAN, τα οποία εκπαιδεύσαμε σε σύνολα δεδομένων αποτελούμενα από εικόνες μόδας. Σε μία γενική εικόνα, αυτό που προσπαθήσαμε να υλοποιήσουμε στα πλαίσια της παρούσας διπλωματικής εργασίας είναι *ένα πολυεργαλείο μηχανικής μάθησης με δυνατότητες παραγωγής ρεαλιστικών εικόνων μόδας αλλά και εφαρμογής ρεαλιστικών μετασχηματισμών στις εικόνες αυτές*. Το εργαλείο αυτό το οπλίσαμε με τις ακόλουθες δυνατότητες:

1. **Αλλαγή Πόζας**: δοθείσης μίας εικόνας εισόδου και μιας πόζας της εικόνας εξόδου, το αντίστοιχο μοντέλο του εργαλείου μας προσπαθεί να παράξει μία ρεαλιστική εικόνα που να μοιάζει όσο το δυνατόν περισσότερο στην πραγματική εικόνα εξόδου (ενν. με τη νέα πόζα)

2. **Εξαγωγή Ρούχου**: δοθείσης μιας εικόνας ενός ανθρώπου (σε αυθαίρετο παρασκήνιο), το αντίστοιχο μοντέλο του εργαλείου μας προσπαθεί να παράξει μία εικόνα στην οποία να απεικονίζεται μόνο το ρούχο του ανθρώπου σε ουδέτερο παρασκήνιο

3. **Ταίριασμα Στιλ**: δοθείσης μίας εικόνας υποδήματος (ή αντίστροφα μίας τσάντας χεριού), το αντίστοιχο μοντέλο του εργαλείου μας προσπαθεί να παράξει μία εικόνα τσάντα (ή αντίστροφα υποδήματος) η οποία να ταιριάζει στιλιστικά (π.χ. στο χρώμα ή στην επισημότητα) στην αρχική εικόνα

4. **Παραγωγή ρεαλιστικών εικόνων μόδας**: τέλος, οπλίσαμε το πολυ-εργαλείο μας με ένα από τα πλέον εξελιγμένα μοντέλα GAN, το StyleGAN, με σκοπό αυτό να έχει τη δυνατότητα παραγωγής ρεαλιστικών εικόνων μόδας όπου απεικονίζονται άνθρωποι-μοντέλα σε διάφορες πόζες και φορώντας διάφορα σύνολα ρούχων, ενώ σε μελλοντική επέκταση θα είναι δυνατή η ελεγχόμενη μείξη μεταξύ των παραγόμενων εικόνων για αλλαγή στιλ/ρούχων/πόζας κ.ο.κ.

Διάρθρωση

Η εργασία διαρθρώθηκε σε πέντε κεφάλαια (κεφάλαια 2-6) πλην του παρόντος και του τελευταίου κεφαλαίου με μελλοντικές προεκτάσεις. Σε κάθε ένα από τα κεφάλαια αυτά περιλαμβάνονται τα εξής:

- **Κεφάλαιο 2**: στο κεφάλαιο αυτό περιλαμβάνονται γενικές βιβλιογραφικές αναφορές σχετικές αναλύσεις που εισάγουν τον αναγνώστη στην Παραγωγική Μοντελοποίηση εικόνων και στα Generative Adversarial Networks (GANs) που αποτελούν και τον πυρήνα της παρούσας εργασίας.

- **Κεφάλαιο 3**: ακολουθεί ένα κεφάλαιο που εστιάζει στα GANs, σε ότι αφορά τη δόμή και τον τρόπο εκπαίδευσής τους ενώ περιλαμβάνει και τις κατεξοχήν μετρικές για αξιολόγησης της απόδοσής τους.

- **Κεφάλαιο 4**: κατόπιν περνάμε στις εφαρμογές των GANs καθώς και σε αρχιτεκτονικές τέτοιων μοντέλων που έχουν προταθεί στη βιβλιογραφία και έχουν χρησιμοποιηθεί στη πράξη στα πλαίσια Παραγωγικής Μοντελοποίησης εικόνων. Εκεί περιγράφουμε και τα μοντέλα στα οποία βασιστήκαμε για την υλοποίηση του πολυ-εργαλείου μας δίνοντας αντίστοιχα αποτελέσματα από τη σχετική βιβλιογραφία.

- **Κεφάλαιο 5**: αυτό και το επόμενο κεφάλαιο αποτελούν το επίκεντρο της υλοποίησής μας με το κεφάλαιο 5 να περιλαμβάνει πληροφορίες για τα σύνολα δεδομένων που χρησιμοποιήθηκαν, τη μεθοδολογία που ακολουθήθηκε για την προ-επεξεργασία αυτών, τα μοντέλα που σχεδιάστηκαν και υλοποιήθηκαν καθώς και παραμέτρους εκπαίδευσης των μοντέλων αυτών.

- **Κεφάλαιο 6**: στο κεφάλαιο 6 προχωρούμε στην παράθεση μετρικών κατά την εκπαίδευση των μοντέλων μας καθώς και παράθεση μετρικών και αποτελεσμάτων αφότου αυτή έχει ολοκληρωθεί.

- **Κεφάλαιο 7**: στο σύντομο αυτό κεφάλαιο δίνουμε σειρά μελλοντικών προεκτάσεων που τόσο ο εκπονητής όσο και ο ενδιαφερόμενος αναγνώστης μπορούν να χρησιμοποιήσουν για τη συνέχιση της παρούσας διπλωματικής εργασίας.

Πριν προχωρήσουμε, αναφέρουμε στο σημείο αυτό πως η όλη ανάπτυξη και εκπαίδευση των μοντέλων έγινε στην προγραμματιστική γλώσσα Python (ελάχ. εκδ. 3.7), ενώ χρησιμοποιήθηκε το σύνολο βιβλιοθηκών PyTorch (ελαχ. εκδ. 1.7.2). Ο κώδικας της παρούσας διπλωματικής εργασίας, ο οποίος ξεπερνάει τις 15.5K LoC, δίνεται ανοιχτά αλλά χωρίς ευθύνη στο ακόλουθο αποθετήριο κώδικα του GitHub:

`github.com/achariso/gans-thesis`

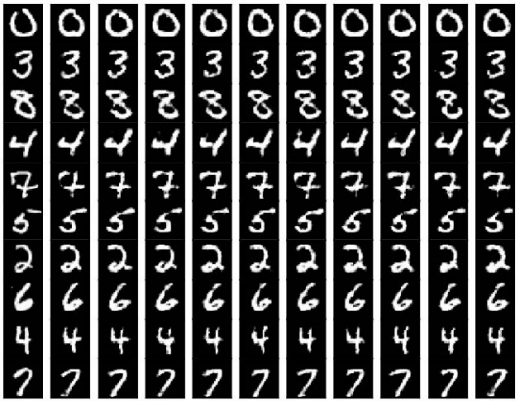
Κεφάλαιο 2

Εισαγωγή στα Generative Adversarial Networks (GANs) και στην Παραγωγική Μοντελοποίηση

Σύμφωνα με τον ορισμό της Μηχανικής Ευφυΐας που δόθηκε από τον Alan M. Turing στο άρθρο του *Computing Machinery and Intelligence* [3], αυτή μπορεί να στοιχειοθετηθεί για μια μηχανή όταν οι αποκρίσεις της είναι δύσκολο να διακριθούν από αυτές ενός ανθρώπου (και έτσι η μηχανή θα περνούσε από το τεστ γνωστό ως *imitation game*). Παρόλο που σε κάποιους τομείς της Μηχανικής Μάθησης όπως η Αλληλεπίδραση Φυσικής Γλώσσας (Conversational AI) υπάρχει ακόμα σημαντική απόσταση μέχρι οι μηχανές να αποκρίνονται με ρεαλιστικό και ανθρώπινο τρόπο, σε άλλους τομείς όπως η Αναγνώριση Προτύπων και κατ΄ επέκταση η Ταξινόμηση ή Πρόβλεψη, οι υπολογιστικές μηχανές επιδεικνύουν ακόμη και υπεροχή σε σύγκριση με τους ανθρώπους.

Στην παρούσα διπλωματική εργασία επικεντρωνόμαστε στη μελέτη τεχνικών και αλγορίθμων Παραγωγικής Μοντελοποίησης (Generative Modeling) στα πλαίσια της Μηχανικής Όρασης (Computer Vision), δηλαδή Παραγωγική Μοντελοποίηση εικόνων. Η Παραγωγική Μοντελοποίηση, εστιάζοντας στην παραγωγή τεχνητών πολυμεσικών δεδομένων που παρουσιάζουν ποικιλομορφία και ρεαλισμό, παραδοσιακά αποτελούσε την «Αχίλλειο πτέρνα» της Τεχνητής Νοημοσύνης. Πρόσφατες εξελίξεις, ωστόσο, στον τομέα της Βαθιάς Μάθησης (Deep Learning), σε συνδυασμό με τη συνεχή βελτίωση του υλικού (hardware) των υπολογιστών, έχουν οδηγήσει την ερευνητική κοινότητα στην αναζήτηση

νέων μεθόδων Παραγωγικής Μοντελοποίησης, ιδιαίτερα σε ότι αφορά τη δημιουργία ρεαλιστικών δισδιάστατων και τρισδιάστατων απεικονίσεων. Αρχικά, η Παραγωγική Μοντελοποίηση για Μηχανική Όραση βασίζονταν στους Αυτόματους Κωδικοποιητές (Autoencoders), η εφαρμογή των οποίων ως ενδιάμεσες στρώσεις (layers) εξαγωγής χαρακτηριστικών σε Βαθιά Νευρωνικά Δίκτυα (όπως στο [18]), οδήγησε στην παραγωγή των πρώτων αξιόλογων και αρκετά ρεαλιστικών εικόνων στις αρχές του αιώνα.



Σχήμα 1: Ποικιλία (diversity) από δείγματα χειρόγραφων ψηφίων που έχουν παραχθεί από Δίκτυο Stacked Denoising Autoencoder (SDAE) με τρεις (3) κρυφές στρώσεις.
Πηγή: «Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion», Pascal Vincent et al., 2010 [18]

Ίσως η πιο αξιοσημείωτη προσθήκη στις τεχνικές Παραγωγικής Μοντελοποίησης εικόνων ήρθε το 2014 όταν ο Ian Goodfellow, κατά τη διάρκεια της διδακτορικής του διατριβής στο Πανεπιστήμιο του Montreal, εφηύρε τα Παραγωγικά Αντιπαραθετικά Δίκτυα ή Generative Adversarial Networks (GANs). Η τεχνική αυτή βασιζόμενη στην ταυτόχρονη εκπαίδευση δύο (2) ξεχωριστών Βαθιών Νευρωνικών Δικτύων (DNNs), οδήγησε στην παραγωγή εικόνων ο βαθμός ποιότητας και ρεαλισμού των οποίων διέκρινε αμέσως την τεχνική αυτή από όλες τις προηγούμενες. Για παράδειγμα, η χρήση των GANs επέτρεψε την εκτέλεση εργασιών που έως τότε θεωρούνταν αδύνατες, όπως την παραγωγή τεχνητών (fake) εικόνων με ποιότητα εφάμιλλη εικόνων του πραγματικού κόσμου, τη μετατροπή ενός σκίτσου σε μια εικόνα σημαντικά όμοια με φυσικές εικόνες, ή τη μετατροπή ενός βίντεο όπου εικονίζεται ένα άλογο σε ένα άλλο βίντεο, όπου το άλογο έχει αντικατασταθεί με ζέβρα — και όλα αυτά χωρίς την ανάγκη ύπαρξης τεράστιων σετ από επιμελώς επισημασμένα (annotated) δεδομένα εκπαίδευσης. Στο σχήμα 2 που παρατίθεται ακολούθως, φαίνεται αφενός η ικανότητα των GANs να παράγουν ρεαλιστικές εικόνες και αφετέρου ο ραγδαίος ρυθμός με τον οποίο προχωράει και αναπτύσσεται η ερεύνα

γύρω από αυτή την επαναστατική τεχνική Παραγωγικής Μοντελοποίησης. Αυτή πρόοδος αφενός επιβεβαιώνει την εστίαση της ερευνητικής κοινότητας τα τελευταία χρόνια στην υιοθέτηση και ανάπτυξη GANs ενώ αφετέρου αποτελεί πειστήριο της υπεροχής τους μεταξύ των τεχνικών Παραγωγικής Μοντελοποίησης.

Σχήμα 2: Πρόοδος στην παραγωγή τεχνητών εικόνων με πρόσωπα ανθρώπων χρησιμοποιώντας Generative Adversarial Networks (GANs): από ασπρόμαυρες εικόνες το 2014 σε έγχρωμες φωτορεαλιστκές εικόνες το 2018.
Πηγή: «Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments», Salehi et al., 2020 [114]

Πριν προχωρήσουμε, ωστόσο, στην παρουσίαση της βασικής δομής και χαρακτηριστικών των GANs, θεωρούμε σκόπιμο να ορίσουμε σαφώς την Παραγωγική Μοντελοποίηση για Μηχανική Όραση χρησιμοποιώντας τεχνικές Βαθιάς Μάθησης, να συγκρίνουμε τα Διακριτικά με τα Παραγωγικά Μοντέλα, καθώς και να αναφερθούμε σε κομβικές τεχνικές Παραγωγικής Μοντελοποίησης, όπως τα Αυτοπαλινδρονούμενα Παραγωγικά Μοντέλα και οι Μεταβλητοί Αυτόματοι Κωδικοποιητές (Variational Autoencoders). Οι Variational Autoencoders (VAEs), συγκεκριμένα, ουσιαστικά αποτελούν άμεσο πρόγονο των GANs και μοιράζονται πολλά κοινά στον τρόπο κατασκευής και εκπαίδευσης με αυτά, με την πιο σημαντική διαφοροποίηση τους να εντοπίζεται στο σχηματισμό της συνάρτηση κόστους.

2.1 Παραγωγική Μοντελοποίηση

Σε επέκταση της διαδεδομένης ρήσης του Αμερικανού φυσικού Richard Feynman, *"What I cannot create, I do not understand"*, θα μπορούσαμε να κινητροδοτήσουμε τη δημιουργία

τεχνικών Παραγωγικής Μοντελοποίησης με τη φράση *"What I understand, I can create"* [100]. Μια ερμηνεία αυτής στα πλαίσια της Μηχανικής Όρασης θα μπορούσε να είναι ότι εφόσον η λανθάνουσα «δομή» ή πιθανοτική κατανομή ενός συνόλου πραγματικών εικόνων μπορεί επαρκώς να «αιχμαλωτιστεί» από κάποιο μοντέλο Παραγωγικής Μοντελοποίησης με Μηχανική/Βαθιά Μάθηση, τότε το μοντέλο αυτό θα πρέπει να είναι ικανό να παράγει νέες εικόνες ή δείγματα, μη-διακρίσιμα από τις εικόνες ή δείγματα που προέρχονται από το σύνολο δεδομένων εκπαίδευσης.

Στην παρούσα εργασία, επικεντρωνόμαστε σε μοντέλα **Στατιστικής Παραγωγικής Μοντελοποίησης** (Statistical Generative Modeling) με Δίκτυα Βαθιάς Μάθησης, τα οποία και αναλύονται ακολούθως. Υπάρχουν αρκετοί τύποι τέτοιων μοντέλων που έχουν αναπτυχθεί στη βιβλιογραφία αλλά και στη πράξη. Ένας από αυτούς είναι και τα GANs, ωστόσο πριν φτάσουμε στην ανάλυση της δομής και λειτουργίας τους, θα γίνει αναφορά και σε άλλους τύπους τεχνικών και μοντέλων Στατιστικής Παραγωγικής Μοντελοποίησης με Δίκτυα Βαθιάς Μάθησης.

Στατιστική Παραγωγική Μοντελοποίηση

Τα μοντέλα **Στατιστικής Παραγωγικής Μοντελοποίησης** ή Στατιστικά Παραγωγικά Μοντέλα (Statistical Generative Models), είναι εκπαιδεύσιμα μοντέλα τα οποία βασίζονται στην ύπαρξη μεγάλων συνόλων δεδομένων προκειμένου να «αιχμαλωτίσουν» ρητά ή έμμεσα μια πιθανοτική κατανομή ($P(X)$ εάν πρόκειται για παραγωγή χωρίς συνθήκη, $P(X|Y)$ για υπο-συνθήκη παραγωγή/κατανομή - κάτι που αναλύεται εκτενώς παρακάτω). Τα εκπαιδευμένα στατιστικά αυτά μοντέλα καλούνται εν συνεχεία να παράγουν δείγματα που ακολουθούν την ίδια πιθανοτική κατανομή, όπου ως X νοούνται τα παρατηρήσιμα δεδομένα (π.χ. μια εικόνα του συνόλου δεδομένων εκπαίδευσης), ενώ ως Y νοείται η κλάση ή τάξη στην οποία ανήκει το κάθε ένα από τα δεδομένα εκπαίδευσης και η οποία χρησιμοποιείται από το εκπαιδευμένο μοντέλο για την παραγωγή δειγμάτων της αντίστοιχης κλάσης ή τάξης (εάν πρόκειται για μη-επιβλεπόμενη εκπαίδευση το Y συνήθως παραλείπεται οπότε και μιλάμε για παραγωγή/κατανομή χωρίς συνθήκη).

Στη γενική περίπτωση, μας δίνεται ένα σύνολο δεδομένων $\vec{x_i}$ που ακολουθεί κάποια κατανομή $p_{data}(\vec{x})$ και στόχος της Στατιστικής Παραγωγικής Μοντελοποίησης είναι η εύρεση ενός μοντέλου (δηλ. αρχικά μιας οικογένειας μοντέλων και στη συνέχεια βελτιστοποίηση για εύρεση των βέλτιστων παραμέτρων) του οποίου η προσέγγιση της $p_{data}(\vec{x})$, $p_{model;\vec{\theta}}(\vec{x})$ με παραμέτρους τις $\vec{\theta}$ θα έχει τη μικρότερη δυνατή «απόσταση» από αυτή. Εάν

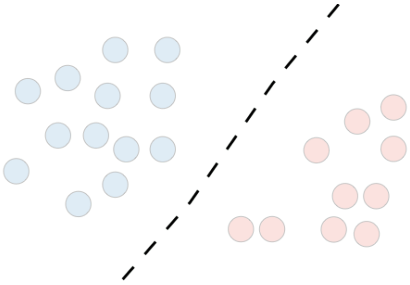
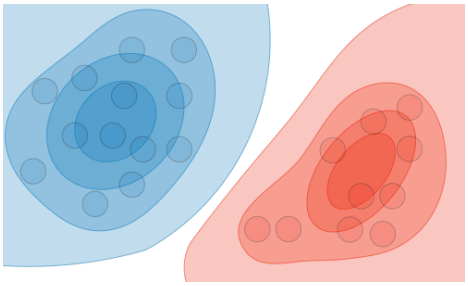
σαν μέτρο της απόστασης κατανομών χρησιμοποιηθεί η KL Divergence [4], τότε προκύπτει ο διαδεδομένος στόχος εκπαίδευσης Στατιστικών Παραγωγικών Μοντέλων: *εύρεση του μοντέλου που μεγιστοποιεί τη λογαριθμική πιθανοφάνεια (log-likelihood) που ανατίθεται στα δεδομένα του συνόλου εκπαίδευσης*.

Επομένως, μπορούμε να θεωρήσουμε ένα Στατιστικό Παραγωγικό Μοντέλο ως μια εκπαιδεύσιμη συνάρτηση κατανομής πιθανότητας για την εκπαίδευση της οποίας απαιτούνται αφενός **δεδομένα** (data) και αφετέρου **πρότερη γνώση** (prior knowledge) η οποία υπεισέρχεται στο μοντέλο είτε ως η μορφή της συνάρτησης κατανομής (π.χ. Gaussian), είτε ως η συνάρτηση κόστους για την εύρεση της βέλτιστης λύσης (π.χ. maximum likelihood - cross entropy), είτε ως ο αλγόριθμος βελτιστοποίησης που θα χρησιμοποιηθεί (π.χ. stochastic gradient descent) ή ως κάτι άλλο. Η ανάγκη ύπαρξης πρότερης γνώσης έρχεται ως άμεση συνέπεια του γνωστού θεωρήματος *No Free Lunch Theorem* στα πλαίσια της βελτιστοποίησης και της βαθιάς μάθησης [21][75].

Σύγκριση Διακριτικών και Παραγωγικών Μοντέλων

Στα πλαίσια της Μηχανικής και Βαθιάς Μάθησης, ως συμμετρικά των Παραγωγικών Μοντέλων νοούνται τα Διακριτικά Μοντέλα (Discriminative Models), τα οποία είναι αυτά που παραδοσιακά χρησιμοποιούνται σε εφαρμογές όπως η Ταξινόμηση (Classification) ή η Ομαδοποίηση (Clustering). Τα Στατιστικά Διακριτικά Μοντέλα (Statistical Discriminative Models) εκπαιδεύονται σε σύνολα δεδομένων με σκοπό να μπορούν να διαχωρίζουν τα εκάστοτε δεδομένα εισόδου σε διάφορες τάξεις (classes) ή ομάδες (clusters). Επομένως, λαμβάνοντας δεδομένα εισόδου, X, αλλά και εξόδου (εφόσον υπάρχουν), Y, καλούνται να «μάθουν» τη συνάρτηση κατανομής πιθανότητας $P(Y|X)$. Αυτό δικαιολογεί τη χρησιμοποίηση του όρου «συμμετρικά» ως προς τα Παραγωγικά Μοντέλα, τα οποία «μαθαίνουν» την $P(X|Y)$, κάτι που φαίνεται και στον πίνακα που ακολουθεί.

Πίνακας 1: Ομοιότητες και διαφορές ανάμεσα στα Παραγωγικά και στα Διακριτικά Μοντέλα Μηχανικής Μάθησης στο πλαίσιο της Αναγνώρισης Προτύπων.
Πηγή: Supervised Learning Cheatsheet - Stanford CS 229 (Machine Learning) [124]

	Διακριτικά Μοντέλα	**Παραγωγικά Μοντέλα**
Στόχος Εκπαίδευσης	Άμεση (ρητή) εκτίμηση της $P(Y\|X)$	Άμεση ή έμμεση εκτίμηση της $P(X\|Y)$
Αποτέλεσμα Εκπαίδευσης	Το Όριο Απόφασης (decision boundary)	Η (υπό-συνθήκη) συνάρτηση πυκνότητας πιθανότητας των δεδομένων εκπαίδευσης
Οπτικοποίηση		
Παραδείγματα Αρχιτεκτονικών	Support Vector Machines [9], Conditional Random Fields [12]	Gaussian Discriminant Analysis (GDA) [2], Naive Bayes Classifier

Στα πλαίσια της Μηχανικής Όρασης με Βαθιά Μάθηση, έχουν αναπτυχθεί αρκετά καινοτόμες και αποδοτικές αρχιτεκτονικές Διακριτικών Μοντέλων. Συγκεκριμένα, έχουν χρησιμοποιηθεί εκτενώς μοντέλα με Βαθιά Συνελικτικά Νευρωνικά Δίκτυα (Deep Convolutional Neural Networks - Deep CNNs) σε εφαρμογές όπως:

- Ταξινόμηση Εικόνας (Image Classification)
- Αναγνώριση Αντικειμένων σε Εικόνα (Object Detection)
- Απόδοση Ετικέτας σε κάθε εικονοστοιχείο μιας Εικόνας (Instance Segmentation)
- Ανάλυση - Ταξινόμηση Βίντεο (Video Analysis - Classification)

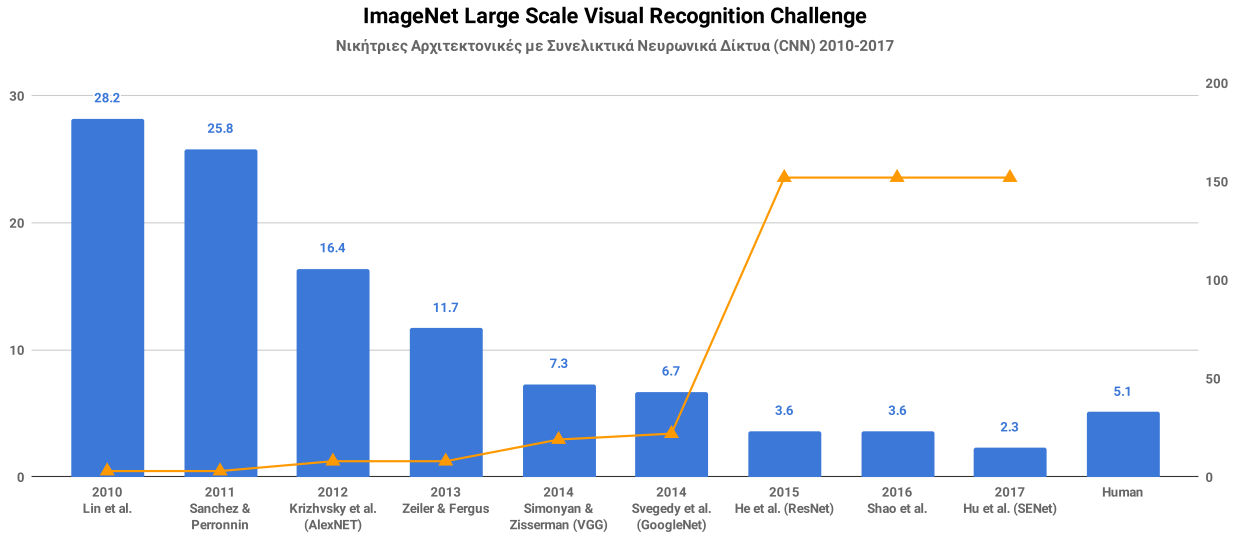
Πολλές από τις καινοτομίες και αρχιτεκτονικές δικτύων που επινοήθηκαν στα Διακριτικά Μοντέλα κατά την εφαρμογή τους στις παραπάνω κατηγορίες με κύρια την ταξινόμηση εικόνων, υιοθετήθηκαν και καθιερώθηκαν και στα Βαθιά Παραγωγικά Μοντέλα, όπως αναλύεται σε επόμενη υποενότητα. Παρακάτω, αναφέρουμε παραδειγματικά δύο ευρέως γνωστές και επιτυχημένες αρχιτεκτονικές Διακριτικών Μοντέλων με Βαθιά Συνελικτι-

κά Νευρωνικά Δίκτυα, αφενός για να τονίσουμε χαρακτηριστικά τους τα οποία έχουν καθιερωθεί και σε αρχιτεκτονικές Βαθιών Παραγωγικών Μοντέλων και αφετέρου γιατί κάνουμε χρήση αμφοτέρων αυτών των μοντέλων για την αξιολόγηση των εικόνων που έχουν παραχθεί από GANs.

Διακριτικά Μοντέλα με Συνελικτικά Νευρωνικά Δίκτυα

Ξεκινώντας από το LeNet-5 [11], τα Συνελικτικά Νευρωνικά Δίκτυα (CNN) έχουν λίγο πολύ μια καθιερωμένη δομή: στοιβαγμένες συνελικτικές στρώσεις ακολουθούμενες από στρώσεις κανονικοποίησης και ομαδοποίησης και στο τέλος ακολουθούμενες από μία ή περισσότερες πλήρως συνδεδεμένες στρώσεις. Παραλλαγές αυτού του βασικού σχεδιασμού είναι αρκετά διαδεδομένες στη βιβλιογραφία για ταξινόμηση εικόνων και έχουν σημειώσει τα καλύτερα αποτελέσματα μέχρι σήμερα στα ευρέως γνωστά σύνολα δεδομένων εικόνων, όπως στο MNIST [10], στο CIFAR [23] και κυρίως στο ImageNET [16] και στην πρόκληση ταξινόμησης εικόνων ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [33], [78]. Σκοπός της τελευταίας ήταν η εκπαίδευση ενός μοντέλου σε ∼1 εκατομμύριο εικόνες το οποίο θα μπορεί σωστά να ταξινομήσει 100.000 άλλες εικόνες σε 1.000 τάξεις. Για τα μεγάλα σύνολα δεδομένων, όπως το ImageNET, η πρόσφατη τάση ήταν η αύξηση του αριθμού και του μεγέθους των στρώσεων (ενν. του αριθμού συνελικτικών φίλτρων ανά στρώση), ενώ γίνονταν χρήση της τεχνικής Dropout [35] για την αντιμετώπιση του overfitting που προκαλούνταν από την αύξηση της χωρητικότητας (capacity) του μοντέλου. Παρακάτω φαίνεται ένα διάγραμμα με την εξέλιξη των αρχιτεκτονικών Συνελικτικών Νευρωνικών Δικτύων στο ImageNET, το οποίο επιβεβαιώνει το παραπάνω. Όλες αυτές είναι αρχιτεκτονικές με Συνελικτικά Νευρωνικά Δίκτυα. Στο σχήμα φαίνεται η τάση για αύξηση του αριθμού των στρώσεων, ειδικά μετά τη χρήση των residual connections το 2015 [41]. Επίσης φαίνεται η απότομη πτώση του error rate (αριστερός κάθετος άξονας) το 2014 με την παρουσίαση των μοντέλων VGGNet και Inception που αναλύονται παρακάτω.

Δύο αξιοσημείωτες αρχιτεκτονικές ή μοντέλα για ταξινόμηση εικόνων με Συνελικτικά Νευρωνικά Δίκτυα, τα οποία εφαρμόζουμε και στη παρούσα εργασία για εξαγωγή χαρακτηριστικών (feature extraction) των εικόνων με σκοπό τη σύγκρισή τους, είναι τα VGGNet και Inception. Ακολουθεί μία σύντομη περιγραφή των μοντέλων αυτών εστιάζοντας σε στοιχεία της αρχιτεκτονικής τους που τα έχουν καθιερώσει ως τα πιο δημοφιλή μοντέλα για ταξινόμηση εικόνων και εξαγωγή χαρακτηριστικών με Συνελικτικά Νευρωνικά



Σχήμα 3: Νικήτριες αρχιτεκτονικές με Συνελικτικά Νευρωνικά Δίκτυα της πρόκλησης ταξινόμησης εικόνων από το σύνολο δεδομένων ImageNET.
Πηγή: CS231n: Convolutional Neural Networks for Visual Recognition, Fei-Fei Li, Standford University, 2018 [94]

Δίκτυα.

Μελέτη περίπτωσης: VGGNet

Το VGGNet [34] είναι ένα μοντέλο για ταξινόμηση εικόνων χρησιμοποιώντας Συνελικτικά Νευρωνικά Δίκτυα, το οποίο επινοήθηκε το 2014 από τους Simonyan και Zisserman στο Visual Geometry Group του Πανεπιστημίου της Οξφόρδης. Μαζί με το GoogleNet που αναλύεται στη συνέχεια αποτέλεσαν τους νικητές του ImageNet Large-Scale Visual Recognition Challenge το 2014, πετυχαίνοντας error rate 7.3% και 6.7% αντίστοιχα στην ταξινόμηση εικόνων, σημειώνοντας σημαντική μείωση σε σχέση με τον προηγούμενο νικητή.

Βασικά στοιχεία που διαφοροποιούν την αρχιτεκτονική του VGGNet από τις προηγούμενες είναι τα εξής:

- Χρησιμοποίηση μικρών συνελικτικών φίλτρων, διάστασης 3×3. Αυτό επέτρεψε την αύξηση του «βάθους» του μοντέλου, αλλά και της σημάντικής βελτίωσης της ποιότητας των εξαγόμενων χαρακτηριστικών στις ενδιάμεσες στρώσεις. Αυτό οφείλεται στο ότι τρία (3) φίλτρα διάστασης 3×3 (το ένα μετά το άλλο) έχουν το ίδιο δεκτικό πεδίο (receptive field) με ένα φίλτρο 7×7, σαν αυτά που χρησιμοποιούνταν στο

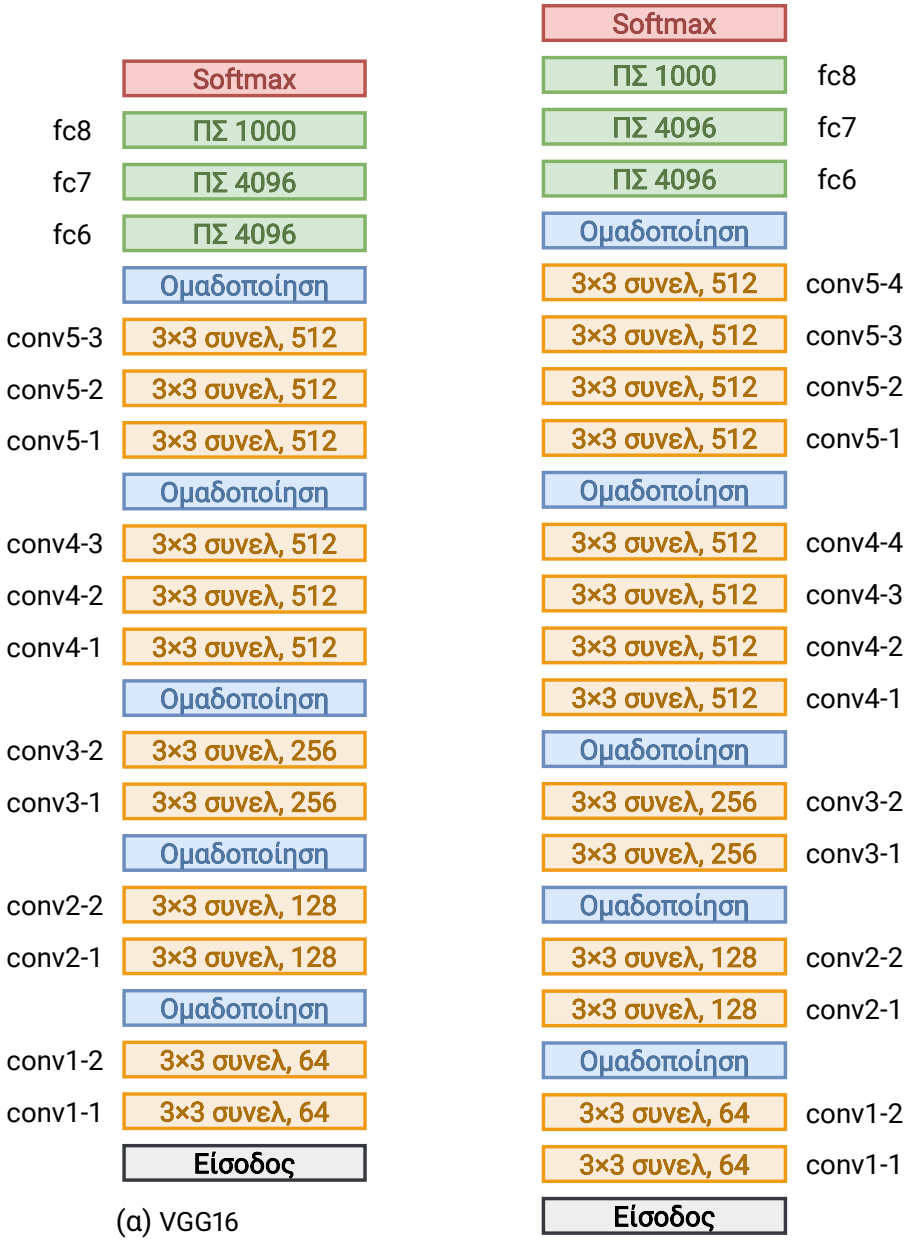
AlexNet [24] για παράδειγμα, περιέχοντας όμως σχεδόν τις μισές εκπαιδεύσιμες παραμέτρους.

- Σημαντικά μεγαλύτερος αριθμός συνελικτικών στρώσεων. Στις δύο παραλλαγές του VGGNet που χρησιμοποιούνται ευρέως ο αριθμός αυτός είναι δεκαέξι (16) για το VGG16 και δεκαεννέα (19) για το VGG19. Για σύγκριση στο μοντέλο AlexNet του 2012 ο αριθμός αυτός ήταν μόλις οχτώ (8).

Στο σχήμα 4, παρακάτω, παρατίθενται σχηματικά οι αρχιτεκτονικές των μοντέλων VGG16 και VGG19 για οπτικοποίηση της δομής τους. Το VGG16 (αριστερά) έχει 16 συνελικτικές στρώσεις και στρώσεις ομαδοποίησης (pooling) πριν τις fully-connected (FC) στρώσεις, όλες με φίλτρα διαστάσεων 3×3. Αντίστοιχα, το VGG19 (δεξιά) έχει 19. Επίσης, όσο κοντινότερα προς την έξοδο βρισκόμαστε, τόσο αυξάνεται ο αριθμός των φίλτρων κάθε συνελικτικής στρώσης, ενώ μειώνεται των μήκος και πλάτος της εισόδου της στρώσης. Οι στρώσεις με την ετικέτα «ΠΣ» είναι πλήρως-συνδεδεμένες (fully-connected) στρώσεις, αυτές με την ένδειξη «συνελ» είναι συνελικτικά φίλτρα των αναγραφόμενων διαστάσεων (αριστερά της ετικέτας) και του αναγραφόμενου αριθμού (δεξιά), ενώ οι στρώσεις με την ετικέτα «Ομαδοποίηση» εφαρμόζουν μείωση στο ήμισυ πλάτους και μήκους κρατώντας το μέγιστο από κάθε διακριτή τετράδα εικονοστοιχείων (max-pooling).

Μελέτη περίπτωσης: Inception

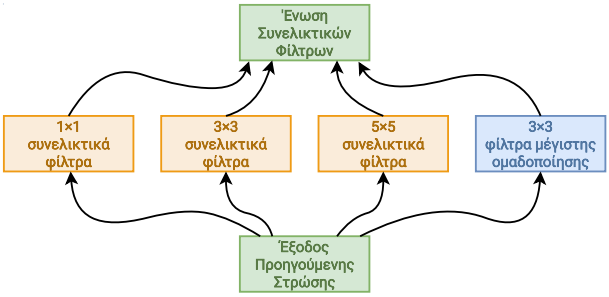
Το μοντέλο Inception ή InceptionNet παρουσιάστηκε την ίδια χρονιά με το VGGNet, δηλαδή στο ImageNet Large-Scale Visual Recognition Challenge του 2014. Αρχικά, είχε ονομαστεί GoogleNet [36] καθώς οι επτά (7) από τους εννέα (9) δημιουργοί του ήταν από την Google. Η λογική των δημιουργών ήταν να ξεφύγουν από την τάση συνεχούς αύξησης του βάθους των συνλεκτικών μοντέλων και να εστιάσουν σε αποδοτικές τοπικές τοπολογίες δικτύων. Έτσι, βασικό δομικό στοιχειών των μοντέλων τύπου Inception είναι ένα υποδίκτυο αποτελούμενο από παράλληλες συνελικτικές στρώσεις (με διαφορετικά μεγέθη φίλτρων) και στρώσεις ομαδοποίησης, το Inception Module, το οποίο φαίνεται στο σχήμα 5 που ακολουθεί. Πρόκειται για μία network-in-network τοπολογία που επιτρέπει σε κάθε module να εξάγει ακόμα καλύτερα και πιο ακριβή χαρακτηριστικά των εικόνων εισόδου. Στην απλή (naive) εκδοχή τους, τα Inception Modules κάνουν παράλληλα συνελικτικά φιλτραρίσματα με φίλτρα διαστάσεων 1×1, 3×3 και 5×5, ενώ στην κανονική εκδοχή τους (που χρησιμοποιήθηκε και στο GoogleNet) γίνεται μείωση των καναλιών (channels) της εισόδου με εκπαιδεύσιμα συνελικτικά φίλτρα 1×1 πριν αυτή «εισέλθει»



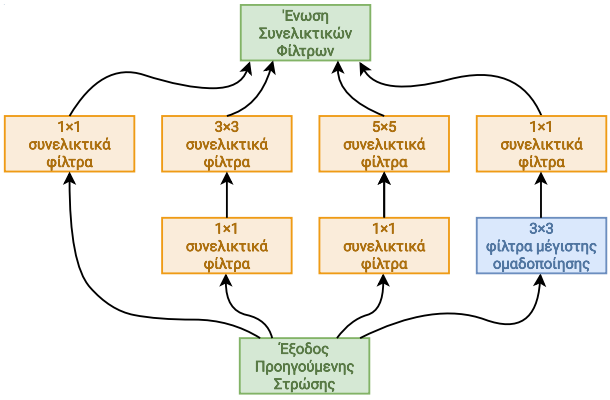
Σχήμα 4: Αρχιτεκτονικές των μοντέλων VGG16 και VGG19 που αναδείχθηκαν νικήτριες στη πρόκληση εντοπισμού και ταξινόμησης, αντίστοιχα, εικόνων από το σύνολο δεδομένων ImageNET το 2014.

Πηγή: Ανακατασκευή από «Very Deep Convolutional Networks for Large-Scale Image Recognition», Karen Simonyan and Andrew Zisserman, 2014 [34]

στα συνελικτικά φίλτρα διαστάσεων 3×3 και 5×5. Αυτό είχε ως αποτέλεσμα σημαντική μείωση του υπολογιστικού κόστους του κάθε εμπρόσθιου περάσματος (forward pass) του μοντέλου και έκανε δυνατή την εκπαίδευσή του.



(α) Απλό (Naive) Inception Module



(β) Inception Module με Μείωση Καναλιών

Σχήμα 5: Inception Modules: Βασικά δομικά στοιχεία του Inception.
Πηγή: Ανακατασκευή από «Going Deeper with Convolutions», Szegedy et al., 2014 [36]

Αυτό που επιτυγχάνεται με τη χρησιμοποίηση των Inception Modules είναι ότι το κάθε ένα από αυτά «βλέπει» την έξοδο του προηγούμενου με διαφορετικά δεκτικά πεδία (receptive fields), με αποτέλεσμα οι πράξεις εσωτερικά του καθενός να γίνονται με μεγαλύτερη αποδοτικότητα.

Μια άλλη σημαντική καινοτομία του GoogleNet και των μοντέλων τύπου Inception είναι η μη-χρησιμοποίηση πλήρως-συνδεδεμένων στρώσεων πριν την έξοδο, πλην μίας μεγέθους ίδιου με του αριθμού των τάξεων εξόδου προκειμένου να μπορεί να εφαρμοστεί η στρώση εξαγωγής πιθανοτήτων, Softmax [17]. Βασιζόμενοι στην παρατήρηση ότι η συντριπτική πλειοψηφία των παραμέτρων εντοπίζονται στις πλήρως-συνδεδεμένες στρώσεις πριν την έξοδο αντίστοιχων μοντέλων, οι δημιουργοί του GoogleNet αφαίρεσαν αυτές τις στρώσεις, κάτι που τους επέτρεψε να φτιάξουν αρκετά πιο σύνθετες δομές εξαγωγής χαρακτηριστικών, δηλαδή τα Inception Modules.

Τέλος, οι δημιουργοί του GoogleNet έχουν προσθέσει δύο ακόμα «βοηθητικούς» ταξινομητές (τους «ταξινομητής 0» και «ταξινομητής 1» που φαίνονται στο σχήμα 6). Οι βοηθητικοί ταξινομητές έχουν ίδια μορφή εξόδου με τον βασικό ταξινομητή, χρησιμοποιήθηκαν προκειμένου να σταθεροποιήσουν και να κάνουν πιο αποδοτική την εκπαίδευση του δικτύου, ενώ δεν χρησιμοποιούνται κατά τη φάση δοκιμής (evaluation). Στο σχήμα 6 παρακάτω, παρατίθεται σχηματικά η βασική δομή του GoogleNet. Εκεί, φαίνονται τα στοιβαγμένα (stacked) Inception Modules καθώς και οι βοηθητικοί ταξινομητές. Αυτή είναι μια περιγραφή σε υψηλό επίπεδο. Για πιο λεπτομερή περιγραφή ο αναγνώστης καλείται να συμβουλευτεί το άρθρο που παρατίθεται στην πηγή του σχήματος.

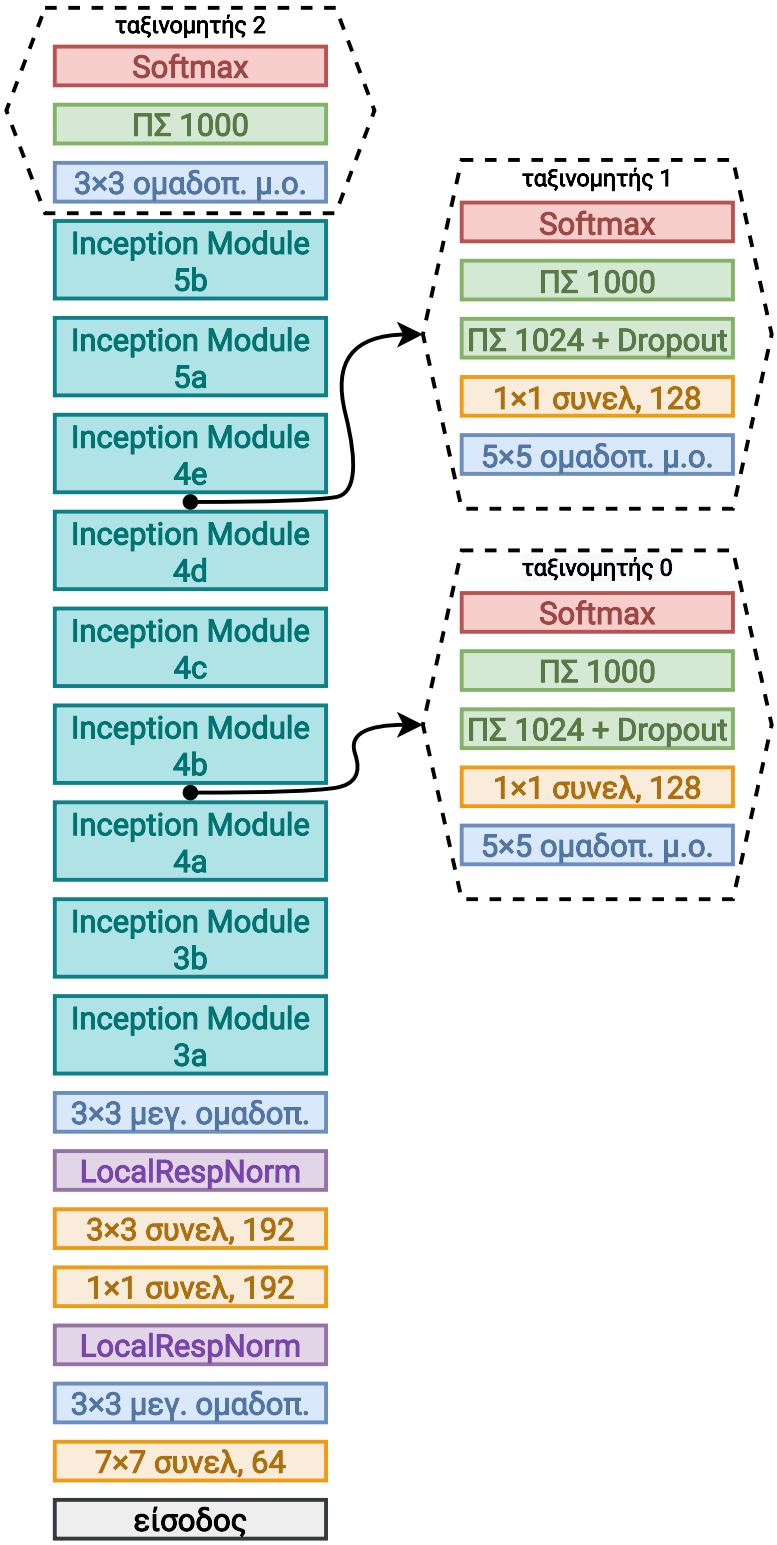
Η αρχιτεκτονική του GoogleNet (ή Inception v1) πέρασε από μια σειρά επαναληπτικών βελτιώσεων από τους δημιουργούς του. Έτσι, σταδιακά παρουσιάστηκαν τα μοντέλα Inception v2 και Inception v3 [46], με έμφαση στην αύξηση της απόδοσης στο εσωτερικό του κάθε Inception Module και της αξιοπιστίας των βοηθητικών ταξινομητών. Επίσης, το 2016, παρουσιάστηκαν τα Inception v4 και Inception-ResNet [60], με περαιτέρω βελτιώσεις της αποδοτικότητας των Inception Modules καθώς, εισαγωγή residual connections (Inception-ResNet v1) και hyperparameter tuning (Inception-ResNet v2).

Βαθιά Παραγωγικά Μοντέλα

Επανερχόμενοι στην Παραγωγική Μοντελοποίηση, μετά τη σύντομη αντιπαραβολή των Διακριτικών Μοντέλων και την παράθεση δύο (2) ευρέως διαδεδομένων αρχιτεκτονικών αυτών με Δίκτυα Βαθιάς Μάθησης, στην παρούσα υποενότητα θα γίνει μια εισαγωγή στα Παραγωγικά Μοντέλα με Δίκτυα Βαθιάς Μάθησης ή Βαθιά Παραγωγικά Μοντέλα (ΒΠΜ) (Deep Generative Models). Τα ΒΠΜ ανήκουν στα μοντέλα Στατιστικής Παραγωγικής Μοντελοποίησης και ουσιαστικά είναι Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) με μία ή περισσότερες (συνήθως πολλές) κρυφές στρώσεις, τα οποία εκπαιδεύονται χρησιμοποιώντας μεγάλα σύνολα δεδομένων προκειμένου να εκτιμήσουν ή να προσεγγίσουν σύνθετες κατανομές πιθανότητας με υψηλή διαστασιμότητα. Ένα επιτυχώς εκπαιδευμένο ΒΠΜ χρησιμοποιείται στη συνέχεια είτε για παραγωγή νέων σημείων (δηλ. δεδομένων) από το χώρο εισόδου, τα οποία ακολουθούν την ίδια κατανομή πιθανότητας με αυτή των δεδομένων εκπαίδευσης, ή για την εκτίμηση της πιθανοφάνειας (likelihood) της κάθε παρατήρησης.

Οι εφαρμογές των ΒΠΜ είναι πολλές και ποικίλες, με κάποιες όπως τα λεγόμενα «Deep Fakes» [105], [110] να έχουν προσελκύσει δημοσιότητα λόγω του κινδύνου που ελλοχεύει ο ρεαλισμός των παραγόμενων τεχνητών εικόνων και βίντεο. Σειρά μεθόδων εμφανίζονται στη βιβλιογραφία, ωστόσο, και σε επιστημονικούς τομείς εφαρμογών, όπου η μηχανική μάθηση θεωρούνταν παραδοσιακά ανίκανη να εισέλθει, όπως η αύξηση της ανάλυσης εικόνων (Image Super-Resolution) με GANs [54], ή η δειγματοληψία καταστάσεων του Σημείου Ισορροπίας σε συστήματα πολλών σωμάτων [106] στη φυσική και η δημιουργία παραμετρικών συνθετικών προσομοιώσεων [92] στη μηχανική ρευστών.

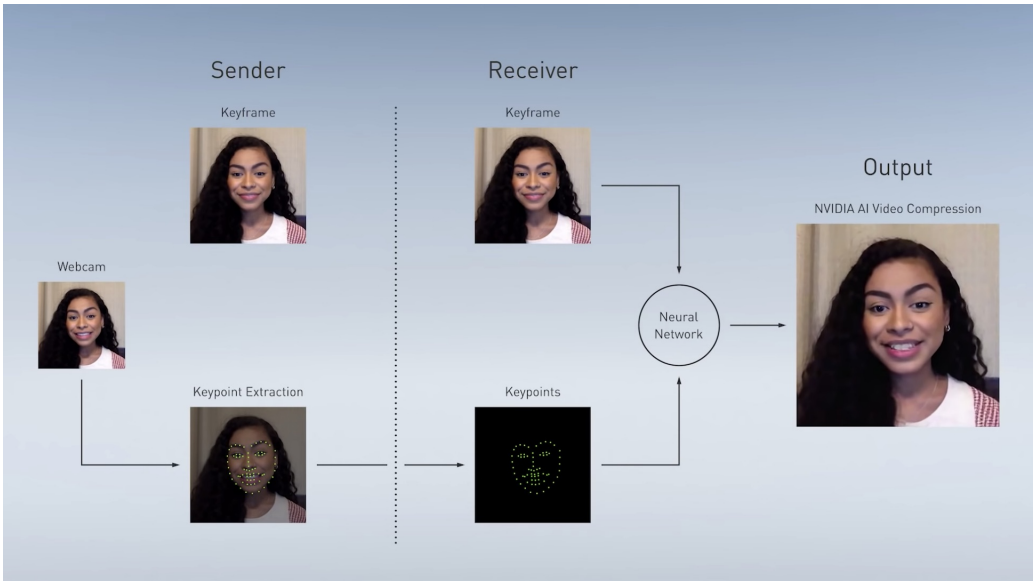
Ίσως την πιο επαναστατική εφαρμογή ΒΠΜ έως σήμερα αποτελεί η πρόσφατη παρουσίαση από την NVIDIA ενός συστήματος που χρησιμοποιεί GANs για συμπίεση βίντεο σε περιβάλλον ζωντανής μετάδοσης, το οποίο παρατίθεται στο σχήμα 7 παρακάτω. Πρόκειται



Σχήμα 6: Μοντέλο GoogleNet ή Inception v1 όπως αλλιώς ονομάστηκε.

Πηγή: Ανακατασκευή από «Going deeper with convolutions», Szegedy et al., 2014 [36]

για τον AI Video Codec του συστήματος NVIDIA Maxine®, όπου ο πομπός στέλνει το αρχικό frame ως εικόνα και για κάθε επόμενο στέλνει τις θέσεις 126 σημείων (keypoints) στα όρια των χαρακτηριστικών του προσώπου του ομιλούντα. Ο δέκτης λαμβάνει την αρχική εικόνα και για κάθε επόμενη λαμβάνει τα keypoints τα οποία τα μετασχηματίζει σε δισδιάστατες εικόνες. Κατόπιν, στη μεριά του δέκτη χρησιμοποιείται ένα προ-εκπαιδευμένο GAN για μετασχηματισμό της εικόνας των keypoints, με συνθήκη την αρχική εικόνα, στο τρέχον frame του βίντεο. Όπως αναλύεται στο άρθρο στην ιστοσελίδα της εταιρείας, αυτό οδηγεί σε δραστική μείωση του απαιτούμενου εύρους ζώνης μετάδοσης δεδομένων για ίδια τελική ποιότητα βίντεο, στην αναβάθμιση της ποιότητας του παραγόμενου βίντεο για ίδιο ρυθμό μετάδοσης και στη δυνατότητα δημιουργίας πλειάδας τεχνητών εφέ βίντεο απλώς αλλάζοντας την αρχική εικόνα - συνθήκη.



Σχήμα 7: AI Video Codec του συστήματος Maxine® της NVIDIA.
Πηγή: «NVIDIA MAXINE: Accelerated SDK with state-of-the-art AI features for building virtual collaboration and content creation applications», NVIDIA (`https://developer.nvidia.com/maxine`)

Στις ενότητες που καταλαμβάνουν το υπόλοιπο αυτού του κεφαλαίου, θα γίνει μια παρουσίαση διαφόρων τύπων ΒΠΜ που έχουν κατά καιρούς παρουσιαστεί στη βιβλιογραφία και έχουν εφαρμοστεί στη πράξη. Έτσι, θα αναφέρουμε τους τρεις πλέον ευρέως χρησιμοποιούμενους τύπους ΒΠΜ: τα Αυτοπαλινδρονούμενα Παραγωγικά Μοντέλα, τους Αυτόματους Κωδικοποιητές και φυσικά των Generative Adversarial Networks που αποτελούν και τον πυρήνα της παρούσας εργασίας. Για κάθε έναν από αυτούς, θα γίνει παράθεση σχετικών μοντέλων και τεχνικών που αναπτυχθεί καθώς και αποτελεσμάτων από τη χρήση τους.

2.2 Αυτοπαλινδρονούμενα Βαθιά Παραγωγικά Μοντέλα

Τα Αυτοπαλινδρονούμενα (autoregressive) Παραγωγικά Μοντέλα αποτελούν μία από τις πρώτες τεχνικές για Παραγωγική Μοντελοποίηση. Η βασική ιδέα που κινητροδότησε τη δημιουργία αυτών των μεθόδων είναι ο κανόνας της αλυσίδας στη Θεωρία Πιθανοτήτων:

$$p(\vec{x}) = p(x_1) * p(x_2|x_1) * ... * p(x_n|x_1, ..., x_{n-1})$$

όπου για τη προσέγγιση των δεσμευμένων πιθανοτήτων έχουν προταθεί ποίκιλλες τεχνικές, με κύριες τη χρήση Παλινδρόμησης που είχε αρχικά δοκιμαστεί [20], [39]) και τη χρήση Βαθιών Νευρωνικών Δίκτυων που δοκιμάστηκαν αργότερα με μεγάλη επιτυχία, όπως αναλύεται και σε επόμενη υποενότητα.

Στη περίπτωση Παραγωγικής Μοντελοποίησης εικόνων, η εικόνα θεωρείται ως μια διατεταγμένη ακολουθία, $\vec{x}$, τυχαίων μεταβλητών (ΤΜ), όπου η κάθε ΤΜ αντιπροσωπεύει τη πιθανότητα το αντίστοιχο εικονοστοιχείο να πάρει τιμή 1/0 (ασπρόμαυρες εικόνες - δυαδικές ΤΜ) ή μία από τις 256 τιμές (0=μαύρο έως 255=άσπρο για εικόνες σε διαβαθμίσεις του γκρι). Η διάταξη των ΤΜ αρχικά γίνονταν όπως και σε ένα σαρωτή εικόνων (raster scan order), δηλαδή γραμμή-γραμμή και από αριστερά προς δεξιά σε κάθε γραμμή. Έτσι, και σύμφωνα με την παραπάνω εξίσωση, x_1 θα είναι το επάνω-αριστερά εικονοστοιχείο μιας εικόνας, x_n το κάτω δεξιά και το n θα ισούται με $W * H$ (W: πλάτος εικόνας σε εικονοστοιχεία, Η: ύψος εικόνας - αρχικά θεωρούνταν μόνο ασπρόμαυρες εικόνες και εικόνες σε διαβαθμίσεις του γκρι, άρα ο αριθμός των καναλιών είναι ένα).

Οι τεχνικές Παραγωγικής Μοντελοποίησης εικόνων με Αυτοπαλινδρονούμενα Παραγωγικά Μοντέλα μπορούν να χωρισθούν σε δυο βασικές οικογένειες, αυτές στις οποίες για την προσέγγιση κατανομών χρησιμοποιείται η Λογιστική Παλινδρόμηση (Logistic Regression) και σε αυτές στις οποίες χρησιμοποιούνται Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks) για τον ίδιο σκοπό. Ακολουθεί μια σύντομη επισκόπηση αμφοτέρων αυτών των οικογενειών με παράθεση αντιπροσωπευτικών παραδειγμάτων από τη βιβλιογραφία.

Μοντελοποίηση βασισμένη στη Λογιστική Παλινδρόμηση

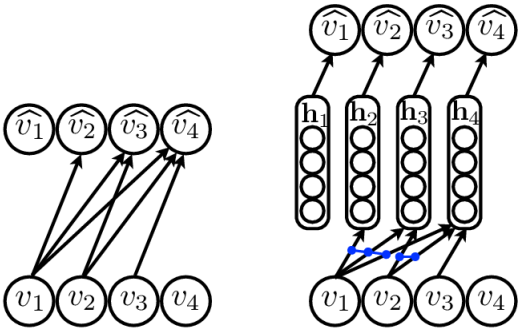
Το μοντέλο Fully-Visible Sigmoid Belief Network (FVSBN) [39], για παράδειγμα, χρησιμοποιούσε Logistic Regression για υπολογισμό των υπο-συνθήκη πιθανοτήτων σε ασπρόμαυρες εικόνες με τη σειρά διάταξης των εικονοστοιχείων όπως του σκάνερ. Έτσι, ο κάθε

όρος του γινομένου του κανόνα αλυσίδας (πλην των πρώτων δύο που είναι τετριμμένες περιπτώσεις) θα είναι:

$$p(X_i = 1|x_1, ..., x_{i-1}; \vec{a}^i) = \sigma(a_0^i + \sum_{j=1}^{i-1} a_j^i x_j)$$

όπου το διάνυσμα $\vec{a}^i = [a_0, ..., a_{i-1}]$ είναι οι (i) παράμετροι της Logistic Regression και $\{x_\tau\}$ είναι οι τιμές που έλαβαν οι προηγούμενες $(i-1)$ ΤΜ. Επειδή πρόκειται για ασπρόμαυρες εικόνες, στην ουσία η έξοδος του αθροίσματος της Λογιστικής Παλινδρόμησης αποτελεί εκτίμηση της παραμέτρου p της Bernoulli κατανομής που ακολουθεί η εκάστοτε ΤΜ. Αντίστοιχα, το μοντέλο Neural Autoregressive Distribution Estimator (NADE) [20] χρησιμοποιεί ένα ΤΝΔ και εν συνεχεία μια Logistic Regression αντί για μόνο την τελευταία για την εκτίμηση της κάθε δεσμευμένης πιθανότητας. Το κέρδος από τη χρήση του ΤΝΔ είναι η εκμάθηση μιας μη γραμμικής σχέσης των μεταβλητών εισόδου και το πέρασμα του αποτελέσματος αυτής στη παλινδρόμηση.

Όπως φαίνεται και στο σχήμα 8, ο υπολογισμός της πιθανοφάνειας νέων εικόνων με παραγωγικά μοντέλα τύπου FVSBN ή NADE ξεκινάει από το πρώτο αριστερά εικονοστοιχείο και συνχίζει προς το κάτω δεξιά, με κάθε επόμενο εικονοστοιχείο να χρησιμοποιεί σαν είσοδο στο αντίστοιχο ΤΝΔ τις τιμές όλων των προηγούμενων εικονοστοιχείων της εικόνας. Οι συνδέσεις με μπλε στο μοντέλο NADE του σχήματος δηλώνουν διαμοιρασμό των βαρών που αντιστοιχίζονται σε κάθε παρατήρηση πριν την είσοδό της στα ΤΝΔ. Για τον τελικό υπολογισμό της $p(\vec{v})$, αν π.χ. $\vec{v} = [1, 0, 0, 1]$, τότε $p(\vec{v}) = \hat{v_1} * (1 - \hat{v_2}) * (1 - \hat{v_3}) * \hat{v_4}$, αφού το $\hat{v_i} = p(V_i = 1|v_1, ..., v_{i-1}; \vec{a}^i)$ (όπου στη περίπτωση του NADE οι παράμετροι $\vec{a}^i$ περιλαμβάνουν και τις παραμέτρους του ΤΝΔ εκτός από αυτές της Logistic Regression). Στο σχήμα 9 φαίνονται παραγωγές εικόνων από το NADE εκπαιδευμένο στο σύνολο δεδομένων MNIST digits. Αυτό που εικονίζεται στα δεξιά είναι δείγματα από τα $\hat{v_i}$'s για $i = 1, ..., 28^2$ (στην ουσία είναι οι παράμετροι της κατανομής Bernoulli). Τέλος, είναι σκόπιμο να αναφερθεί ότι μία βελτίωση του τελευταίου, αποτελεί το μοντέλο Masked Autoregressive Density Estimator (MADE) [40], το οποίο χρησιμοποιεί μια παραλλαγή ενός αυτόματου κωδικοποιητή για το μετασχηματισμό των μεταβλητών εισόδου και μάσκες βαρών για να εξασφαλιστεί ότι κάθε έξοδος επηρεάζεται μόνο από τις προηγούμενες εισόδους, και άρα ότι *ο αυτόματος κωδικοποιητής λειτουργεί ως ένα αυτοπαλινδρονούμενο παραγωγικό μοντέλο*, αρκετά όμως πιο αποδοτικά (παράλληλος υπολογισμός των εξόδων).



Σχήμα 8: Απεικόνιση της διαδικασίας υπολογισμού πιθανοφάνειας (αντίστοιχη χρησιμοποιείται και για παραγωγή) Bernoulli ακολουθίας με τέσσερα (4) στοιχεία από μοντέλα τύπου FVSBN (αριστερά) και NADE (δεξιά).

Πηγή: «CS236: Deep Generative Models», Standford (`https://deepgenerativemodels.github.io`)

Σχήμα 9: Δείγματα εικόνων (δεξιά) που έχουν παραχθεί από το αυτοπαλινδρονούμενο μοντέλο NADE το οποίο εκπαιδεύτηκε σε χειρόγραφα ψηφία από το σύνολο δεδομένων του MNIST (αριστερά).

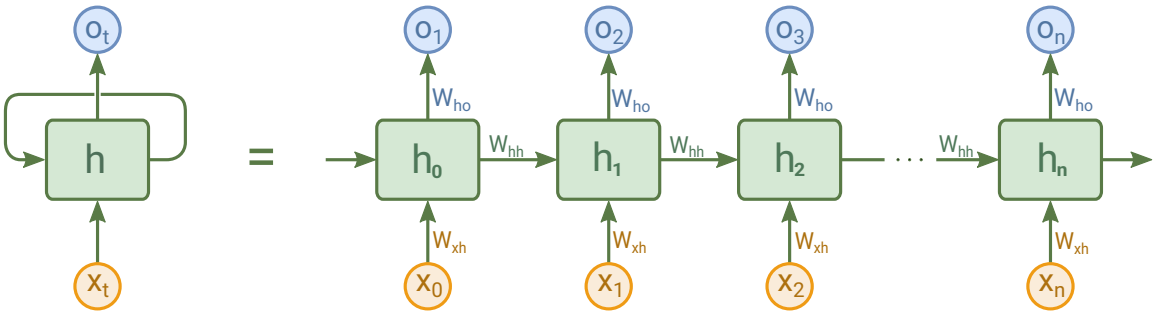
Πηγή: «The Neural Autoregressive Distribution Estimator», Larochelle et. al, 2015 [20]

Μοντελοποίηση με Επαναλαμβανόμενα Νευρωνικά Δίκτυα

Ορμώμενοι από την ιδέα ότι τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (ENN) έχουν σχεδιαστεί για και χρησιμοποιηθεί ευρέως σε ακολουθιακές εισόδους με «μακρές» εξαρτήσεις μεταξύ των δειγμάτων τους, πολλοί ερευνητές στράφηκαν σε αυτά προκειμένου να μοντελοποιήσουν της δεσμευμένες κατανομές σε εφαρμογές Παραγωγικής Μοντελοποίησης μεγάλων ακολουθιών, όπως είναι μια εικόνα ή ένα ηχητικό σήμα. Για μία μεγάλη ακολουθία εισόδου με μακρινές εξαρτήσεις, δεν είναι υπολογιστικά εφικτό (tractable) να σχεδιαστεί μοντέλο που «θυμάται» όλες τις προηγούμενες εισόδους. Για το λόγο αυτό

τα ΕΝΝ διατηρούν μια εσωτερική κατάσταση (στην οποία θεωρητικά έχουν συγκρατηθεί βασικά χαρακτηριστικά των δειγμάτων) την οποία και ανανεώνουν καθώς δέχονται όλο και νεότερα δείγματα στην είσοδο.

Μια απλή μορφή ενός ΕΝΝ φαίνεται στο σχήμα 10 παρακάτω. Η εσωτερική κατάσταση, h_t, διατηρεί μία «περίληψη» των εισόδων έως τη χρονική στιγμή t, ενώ η έξοδος για τη περίπτωση των ΑΠΜ χρησιμοποιείται για να παραμετροποιήσει κάποια υπο-συνθήκη κατανομή, $p(x_t|x_{1:t-1})$. Το δίκτυο εκπαιδεύεται για να βελτιστοποιήσει τις τιμές των παραμέτρων του, W_{xh}, W_{hh} και W_{ho}, ώστε μέσω αυτών το μοντέλο να αναθέτει τη μέγιστη πιθανοφάνεια στα δεδομένα εκπαίδευσης. Έτσι, με την πάροδο του χρόνου το δίκτυο μεταβάλλει την εσωτερική του κατάσταση καθώς και τα βάρη με τα οποία υπολογίζονται οι έξοδοι από την κατάσταση αυτή. Οι έξοδοι με τη σειρά τους, για την περίπτωση Παραγωγικών Μοντέλων, μπορούν να θεωρηθούν ότι προσεγγίζουν τις δεσμευμένες κατανομές πιθανότητας, $p(x_t|x_{1:t-1})$.



Σχήμα 10: Απλό ΕΝΝ μίας εισόδου και μίας εξόδου (φαίνεται και «ξεδιπλωμένο» για να φανούν τα χρονικά βήματα που απαιτούνται για τον υπολογισμό των εξόδων).
Πηγή: Ανακατασκευή από Wikimedia Commons: «A diagram for a one-unit recurrent neural network», fdeloche, 2017

Στα πλαίσια Παραγωγικής Μοντελοποίησης εικόνων με Αυτοπαλινδρονούμενα ΕΝΝ, ίσως η πιο αντιπροσωπευτική δουλειά που έχει παρουσιαστεί στη βιβλιογραφία είναι το μοντέλο PixelRNN του van den Oord το 2016 [63]. Εκεί, για την παραγωγή εικόνων αλλά και την ανάθεση πιθανοφάνειας σε αυτές χρησιμοποιούνταν η σειρά του σαρωτή, ενώ το μοντέλο σχεδιάστηκε για έγχρωμες εικόνες (RGB). Αυτό σημαίνει πως η δεσμευμένη κατανομή για κάθε εικονοστοιχείο απαιτεί τον ορισμό τριών χρωμάτων, ο οποίος επιλέχθηκε να γίνει επίσης ακολουθιακά (πρώτα το κόκκινο, μετά πράσινο και τέλος το

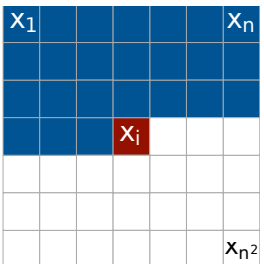
μπλε):

$$p(x_t|x_{1:t-1}) = p(x_t^{red}|x_{1:t-1}) * p(x_t^{green}|x_{1:t-1}, x_t^{red}) * p(x_t^{blue}|x_{1:t-1}, x_t^{red}, x_t^{green})$$

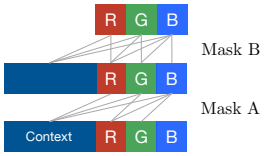
ενώ η κάθε επιμέρους δεσμευμένη κατανομή είναι κατηγορική (multinoulli) με 256 πιθανές τιμές.

Μια βελτίωση του παραπάνω βασιζόμενη στην αντίληψη ότι η τιμή ενός εικονοστοιχείου επηρεάζεται πρωτίστως από τις τιμές των γειτονικών εικονοστοιχείων που έχουν ήδη παραχθεί, ήταν το Αυτοπαλινδρονούμενο Παραγωγικό Μοντέλο PixelCNN [64] στο οποίο χρησιμοποιήθηκαν Συνελικτικές Στρώσεις αντί για ΒΝΔ. Αποτέλεσμα αυτού, ήταν η πολύ ταχύτερη και πιο αποτελεσματική εκπαίδευση του μοντέλου για ίδιας «ποιότητας» παραγόμενα αποτελέσματα σε σχέση με τον προκάτοχό του. Στο σχήμα 11 παρουσιάζεται γραφικά η διαδικασία παραγωγής εικόνων από αυτά τα μοντέλα.

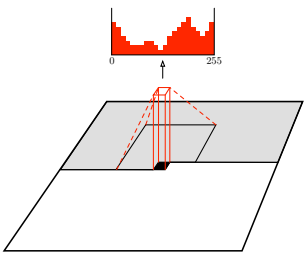
Τέλος, για λόγους πληρότητας, σκόπιμο είναι να αναφερθεί ότι παρόμοιας λογικής Αυτοπαλινδρονούμενα Παραγωγικά Μοντέλα με ΕΝΝ έχουν εφαρμοστεί με επιτυχία και σε εφαρμογής Παραγωγικής Μοντελοποίησης ήχου. Το μοντέλο WaveNet [62] αποτελεί έως σήμερα ίσως το πιο αποτελεσματικό μοντέλο για μετατροπή κειμένου σε ομιλία (Text-to-Speech) ή παραγωγή μουσικής (Music Generation), ενώ υπάρχει από το 2018 στις περισσότερες κινητές συσκευές με λειτουργικό Android.



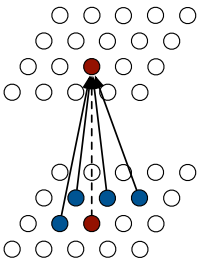
(α) Παραγωγή εικόνας pixel-by-pixel (PixelRNN)



(β) Χρήση μασκών σε δίκτυα τύπου LSTM



(γ) Εφαρμογή συνελικτικών δικτύων για ταχύτητα (PixelCNN)



(δ) Χρήση μασκών στα συνελικτικά φίλτρα

Σχήμα 11: Απεικόνιση της διαδικασίας παραγωγής εικόνων από παραγωγικά μοντέλα με ΕΝΝ, PixelRNN (αριστερά) και PixelCNN (δεξιά).

Πηγή: «Pixel Recurrent Neural Networks» και «Conditional Image Generation with PixelCNN Decoders», van den Oord et al., 2016 [63], [64]

2.3 Παραγωγική Μοντελοποίηση με Αυτόματους Κωδικοποιητές

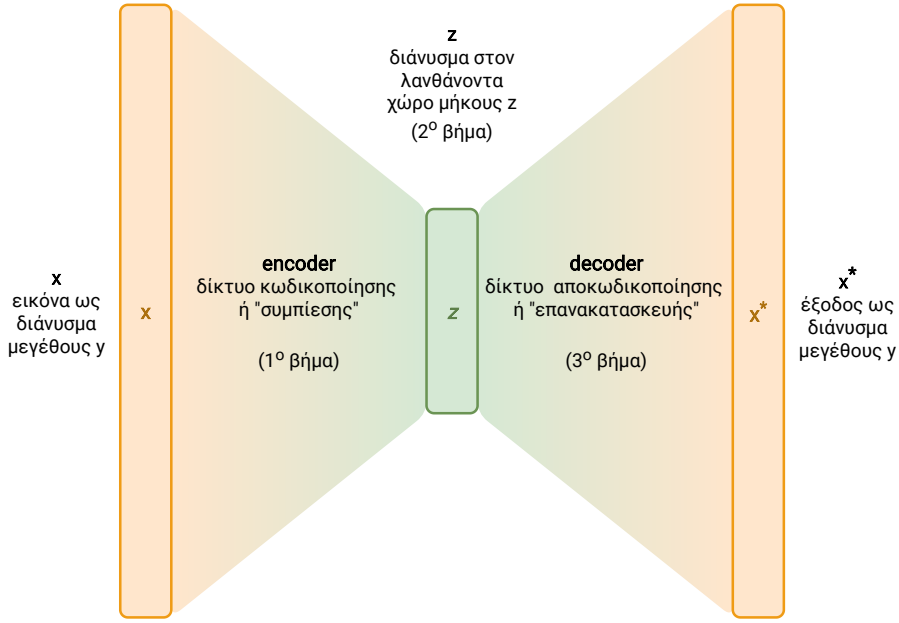
Στη παρούσα υποενότητα θα παρουσιάσουμε τους Αυτόματους Κωδικοποιητές (ΑΚ) και κυρίως μια ειδική εκδοχή τους, τους Μεταβλητούς Αυτόματους Κωδικοποιητές (ΜΑΚ) (Variational Autoencoders - VAEs), οι οποίοι χρησιμοποιούνται συχνά για Παραγωγική Μοντελοποίηση. Αυτή η παρουσίαση, όπως αναφέρθηκε και προηγούμενα, γίνεται διότι πρώτον χρησιμοποιούνται συχνά παραλλαγές των ΜΑΚ ως δομικά στοιχεία Παραγωγικών Μοντέλων με ΠΑΔ και δεύτερον γιατί οι ΜΑΚ αποτελέσαν τις πρώτες προσπάθειες Παραγωγικής Μοντελοποίησης υψηλής-ανάλυσης εικόνων με αξιόλογα αποτελέσματα. Επιπρόσθετα, οι ΑΚ είναι μία από τις τεχνικές και ιδέες που αποτελούν αντικείμενο έρευνας εδώ και δεκατίες στο τομέα της Μηχανικής Μάθησης με ΤΝΔ [5], [6], [8].

Αυτόματοι Κωδικοποιητές

Όπως περιγράφεται και στο όνομά τους, οι Αυτόματοι Κωδικοποιητές είναι ένας τύπος ΤΝΔ που μας βοηθάει να κωδικοποιήσουμε δεδομένα «αυτόματα», δηλ. να βρούμε αποδοτικούς κώδικες για ένα σύνολο δεδομένων χωρίς επίβλεψη (unsupervised learning). Όπως φαίνεται και στο σχήμα 12 παρακάτω, αποτελούνται από δύο μέρη: τον κωδικοποιητή (encoder) που «μαθαίνει» μία συνάρτηση $\vec{z} = f(\vec{x})$ για να μετασχηματίζει την έισοδο $\vec{x}$ σε ένα διάνυσμα $\vec{z}$, που ονομάζεται και *ενδιάμεση αναπαράσταση* και τον αποκωδικοποιητή (decoder) που «μαθαίνει» την αντίστροφη συνάρτηση $\vec{x*} = g(\vec{z}) = g(f(\vec{x}))$ για να ανακατασκευάσει την αρχική είσοδο. Έτσι, η εικόνα εισόδου αρχικά περνάει από τον Κωδικοποιητή (encoder) και μετασχηματίζεται σε ένα διάνυσμα στον λανθάνοντα χώρο σημαντικά μικρότερης διάστασης. Ακολούθως, ο Αποκωδικοποιητής (decoder) λαμβάνει το διάνυσμα αυτό και το μετασχηματίζει σε μία εικόνα ίδιας διάστασης με την αρχική. Τέλος, οι δύο εικόνες συγκρίνονται, προκύπτει το κόστος και οι παράμετροι των δικτύων ανανεώνονται με σκοπό τη μείωση αυτού σύμφωνα με κάποιον αλγόριθμο βελτιστοποίησης.

Ο σκοπός και η κύρια χρήση των ΑΚ είναι η εκμάθηση αυτής της κωδικοποίησης προκειμένου να «συμπιέσουμε» αποδοτικά τα δεδομένα μειώνοντας τη διαστασιμότητά τους (dimensionality reduction) ή να μάθουμε βασικά τους χαρακτηριστικά (feature learning). Οι ΑΚ εκπαιδεύονται μειώνοντας το σφάλμα ανακατασκευής (reconstruction loss), ενώ για τη βελτίωση των παραμέτρων τους συνήθως χρησιμοποιείται ο αλγρόριθμος back-propagation για εύρεση της κατεύθυνσης βελτιώσης σε συνδυασμό με κάποιο αλγρόριθμο

βελτιστοποίησης (όπως οι Stochastic Gradient Descent, RMSProp κ.α.).



Σχήμα 12: Σχηματική αναπαράσταση της λειτουργίας ενός Αυτόματου Κωδικοποιητή (ΑΚ).
Πηγή: Ανακατασκευή από «GANs in action: deep learning with generative adversarial networks», Langr et al., 2019 [104]

Πολλές παραλλαγές των απλών ΑΚ που παραουσιάστηκαν προηγουμένως έχουν εμφανιστεί στη βιβλιογραφεία. Σημαντικότερες όμως από αυτές και πιο διαδεδομένες είναι οι Κανονισμένοι Αυτόματοι Κωδικοποιητές (ΚΑΚ) (Regularized Autoencoders - RAEs) και οι Μεταβλητοί Αυτόματοι Κωδικοποιητές (ΜΑΚ). Ακολούθως, αναλύεται η κάθε μία από αυτές τις παραλλαγές με αντίστοιχα παραδείγματα μοντέλων και επιτυχημένων αρχιτεκτονικών, ενώ στο τέλος αυτής της υποενότητας γίνεται παράθεση παραδειγμάτων χρήσης ΜΑΚ σε Παραγωγική Μοντελοποίηση εικόνων.

Κανονισμένοι Αυτόματοι Κωδικοποιητές (ΚΑΚ)

Για την ευσταθή εκπαίδευση μοντέλων με ΑΚ (ιδιαίτερα στις περιπτώσεις όπου η διάσταση του λανθάνοντος διανύσματος $\vec{z}$ επιτρέπεται να είναι μεγαλύτερη από αυτή της εισόδου) θα πρέπει η χωρητικότητες των δύο επιμέρους δικτύων (του κωδικοποιητή και αποκωδικοποιητή αντίστοιχα) καθώς και η διάσταση του διανύσματος $\vec{z}$ να επιλέγονται με βάση τη πολυπλοκότητα της κατανομής των δεδομένων εκπαίδευσης. Μοντέλα με μεγάλη χωρητικότητα (δηλ. δίκτυα με πολλές στρώσεις και μεγάλο αριθμό εκπαιδεύσιμων παραμέτρων) όταν καλούνται να μάθουν μία συνάρτηση ανακτασκευής για σχετικά απλό

ή μικρό σύνολο δεδομένων, μπορεί εύκολα να οδηγηθούν στην προσέγγιση της $g(f(\vec{x}))$ με τη μοναδιαία συνάρτηση.

Με αφορμή αυτά τα ευρήματα, πολλές ερευνητικές εργασίες οδηγήθηκαν στη χρήση Κανονισμένων (Regularized) ΑΚ προκειμένου να ξεφύγουν από απλά μοντέλα ΤΝΔ με λίγες στρώσεις και να μπορέσουν να εκπαιδεύσουν επιτυχώς βαθιά ΤΝΔ και μοντέλα με μεγάλης διαστσης λανθάνοντες χώρους, χωρίς αυτά απλώς να μαθαίνουν μοναδιαίες συναρτήσεις. Αυτό που ουσιαστικά διαφοροποιεί τους ΚΑΚ από τους απλούς ΑΚ είναι ότι πλέον στη συνατηση κόστους εκτός από τον όρο που μετρά το σφάλμα ανακατασκευής προστίθενται ένας ή περισσότεροι όροι που ενθαρρύνουν το μοντέλο να έχει κι άλλες χρήσιμες ιδιότητες εκτός της μεταφοράς της εισόδου στην εξόδο. Τέτοιες ιδιότητες είναι για παράδειγμα η ενδιάμεση αναπαράσταση να έχει αραιή (sparse) μορφή, το μοντέλο να μπορεί να μειώνει το θόρυβο (denoising) της εισόδου ή να ανέχεται μεταβολές αυτής «συστέλλοντας» (contracting) τις παραμέτρους του. Οι πιο διαδεδομένοι τύποι ΚΑΚ αναλύονται στις παραγράφους που ακολουθούν.

Αραιοί Αυτόματοι Κωδικοποιητές

Οι Αραιοί Αυτόματοι Κωδικοποιητές (Sparse Autoencoders) [14], [15] είναι ίδιοι με τους απλούς ΑΚ με τη προσθήκη ενός όρου στη συνάρτηση κόστους. Έτσι, η συνάρτηση κόσους δεν περιέχει πλέον μόνο το σφάλμα ανακατασκευής, $L(\vec{x}, g(f(\vec{x}))$, αλλά και μία ποινή «αραιότητας» της ενδιάμεσης αναπαράστασης $\vec{z}$, $\Omega(\vec{z})$. Επομένως, το κόστος για κάθε είσοδο στους Αραιούς Αυτόματους Κωδικοποιητές θα είναι:

$$L(\vec{x}; \{\vec{\partial}_f, \vec{\partial}_g\}) = ReconLoss(\vec{x}, g(f(\vec{x}; \vec{\partial}_f); \vec{\partial}_g)) + \lambda_\omega * \Omega(\vec{z} = f(\vec{x}; \vec{\partial}_f))$$

όπου $\vec{\partial}_f$, $\vec{\partial}_g$ είναι οι εκπαιδεύσιμες παράμετροι του δικτύου κωδικοποιήσης και αποκωδικοποίησης αντίστοιχα, $ReconLoss$ είναι μια συνάρτηση απόστασης μεταξύ της εισόδου και της ανακατασκευής της (π.χ. L1/Manhattan, L2/Ευκλείδια κ.α.) και λ_ω ο συντελεστής βαρύτητας του δεύτερου όρου. Στη περίπτωση που η συνάρτηση ποινής της αραιότητας είναι η L1 (ή Manhattan), τότε ο δεύτερος όρος της παραπάνω εξίσωσης γίνεται:

$$\Omega(\vec{z}) = \lambda_\omega * \sum_i |z_i|$$

Η παραπάνω συνάρτηση απόστασης ενθαρύνει το μοντέλο να αναθέτει μικρές τιμές στα στοιχεία του ενδιάμεσου διανύσματος. Ο Glorot et al. χρησιμοποίησαν Rectified Linear Units (ReLUs) ως συναρτήσεις ενεργοποίησης σε μοντέλα Αραιών Αυτόματων Κωδικοποιητών

[19] με αποτελέσμα να ανατίθονται ακριβώς μηδενικές τιμές σε στοιχεία του $\bar{z}$ για διάφορες εισόδους.

Βασικές εφαρμογές των Αραιών ΑΚ είναι η αραιή κωδικοποίηση (sparse coding) [82] καθώς και η εξαγωγή χαρακτηριστικών από το σύνολο δεδομένων [28] τα οποία μπορούν να χρησιμοποιηθούν εν συνεχεία από κάποιο αλγρόριθμο ταξινόμηση ή αναγνώρισης ανωμαλιών (anomaly detection). Περαιτέρω μαθηματική ανάλυση της εκπαίδευσης Αραιών ΑΚ έχει δείξει πως αυτή είναι ισοδύναμη με την εκπαίδευση μέγιστης-πιθανοφάνειας (maximum-likelihood) ενός παραγωγικού μοντέλου που έχει λανθάνουσες τυχαίες μεταβλητές προκειμένου να μάθει την aposteriori συνάρτηση κατανομής πιθανότητας, $p_{model}(\vec{x}|\vec{z})$ [70].

Denoising Αυτόματοι Κωδικοποιητές

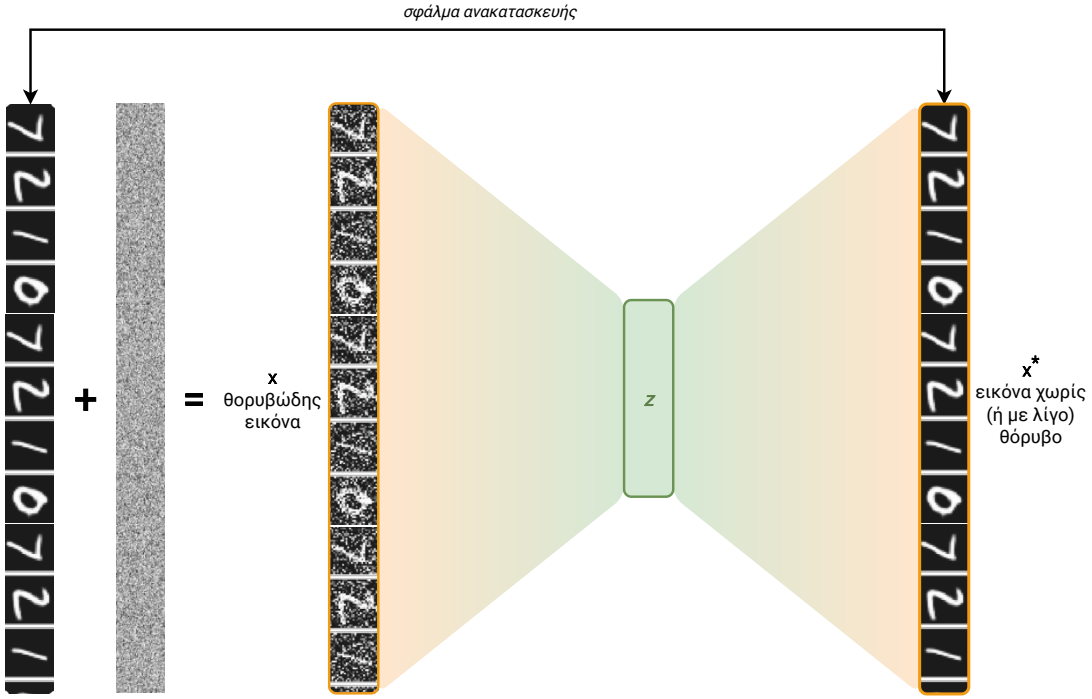
Σε αντίθεση με τους Αραιούς ΑΚ που προσθέτουν ένα δεύτερο όρο στη συνάρτηση κόστους, οι Denoising απλώς αλλάζουν τον τρόπο υπολογισμού του σφάλματος ανακατασκευής προκειμένου να «μάθουν» καλές αναπαραστάσεις. Σε αυτή την παραλλαγή των απλών ΑΚ, λοιπόν, ο κωδικοποιητής δεν δέχεται την εικόνα ως προς την οποία υπολογίζεται το σφάλμα ανακατασκευής, αλλά μια ενθόρυβη εδκοχή της (εικόνα + θόρυβος). Έτσι, όπως απεικονίζεται και στο σχήμα 13 παρακάτω, το δίκτυο καλείται να αποθορυβοποιήσει την εικόνα εισόδου προκειμένου να μειωθεί το σφάλμα ανακατασκευής ως προς την αρχική (χωρίς θόρυβο εικόνα). Επομένως, η νέα συνάρτηση κόστους την οποία εκπαιδεύεται για να ελαχιστοποιήσει ένας Denoising ΑΚ είναι:

$$L(\vec{x}; \{\vec{\partial_f}, \vec{\partial_g}\}) = ReconLoss(\vec{x}, g(f(\vec{x} + noise; \vec{\partial_f}); \vec{\partial_g}))$$

Σύμφωνα με τα παραπάνω, οι Denoising ΑΚ πρέπει να αναιρέσουν τη διάβρωση της εικόνας από το θόρυβο και όχι απλώς να μεταφέρουν την είσοδο στην έξοδο. Η εκπαίδευση ΑΚ με αυτόν τον τρόπο πιέζει τις f και g έμμεσα να μάθουν τη δομή της κατανομής των δεδομένων του συνόλου εκπαίδευσης και άρα μπορούν να λειτουργήσουν επιτυχώς σαν Παραγωγικά Μοντέλα [27], κάτι που είχε αρχικά επεικονιστεί στο σχήμα 1 παραπάνω.

Συστολικοί (Contractive) Αυτόματοι Κωδικοποιητές

Μία άλλος μέθοδος για κανονικοποίηση της συνάρτησης κόστους που θα αναφέρουμε για λόγους πληρότητας είναι παρόμοια με αυτή των Αραιών ΑΚ, δηλ. η πρόσθεση ενός



Σχήμα 13: Σχηματική απεικόνιση της λειτουργίας των Denoising Αυτόματων Κωδικοποιητών.
Πηγή: Ανακατασκευή από «Reconstruct corrupted data using Denoising Autoencoder» (Medium Post), Garima Nishad, 2020, (`https://medium.com/@analytics-vidhya/aeaff4b0958e`)

όρου στη συνάρτηση αυτή. Στην περίπτωση των Συστολικών ΑΚ, εστιάζουμε στο πόσο μεταβάλλεται η ενδιάμεση αναπαράσταση όταν μεταβληθεί η είσοδος, με τη λογική ότι όταν το δίκτυο του κωδικοποιητή (μέσω της εκπαίδευσης συνολικά του Συστολικού ΑΚ) «μάθει» τα βασικά χαρακτηριστικά και τη δομή της κατανοομής των δεδομένων, δεν θα πρέπει η κωδικοποίηση της εισόδου να μεταβάλλεται σημαντικά με την είσοδο. Έτσι, ο πρόσθετος όρος της συνάρτησης κόστους σύμφωνα με τον Rifai et al. [22], θα είναι:

$$\Omega(\vec{z}, \vec{x}) = \lambda_\omega * \sum_i \left\| \nabla_{\vec{x}} z_i^2 \right\|_F^2$$

Επομένως η συνάρτηση κόστους για τη βελτιστοποίηση των Συστολικών ΑΚ θα είναι:

$$L(\vec{x}; \{\vec{\theta_f}, \vec{\theta_g}\}) = ReconLoss(\vec{x}, g(f(\vec{x}; \vec{\theta_f}); \vec{\theta_g})) + \lambda_\omega * \sum_i \left\| \nabla_{\vec{x}} z_i^2 \right\|_F^2$$

κάτι που ενθαρύνει το μοντέλο να μάθει μια συνάρτηση που δεν αλλάζει πολύ για για μικρές αλλαγές της εισόδου (ισοδύναμο με την ενθάρρυνση του μοντέλου να μάθει τη δομή της πιθανοτικής κατανομής εισόδου) αλλά επίσης να μπορεί να ανακτασκευάσει την είσοδο. Στη θεωρία γραμμικών τελεστών, ένας γραμμικός τελεστής λέγεται «συστολικός» εάν όταν εφαρμόζεται σε είσοδο μοναδιαίας νόρμας, δίνει έξοδο με νόρμα

μικρότερης ή ίσης της μονάδας [70]. Έτσι, προσθέντοντας τον όρο ποινής της Frobenius νόρμας του Jacobian πίνακα της της συνάρτης του κωδικοποιητή, $z = f(x)|_{x=\bar{x}}$, τιμωρούμε το μοντέλο όταν έχει μεγάλες τιμές στον Jacobian για κάθε μία από τις εισόδους, δηλ. όταν η τοπική προσέγγιση κατά Taylor της $f(x)$ παίρνει μεγάλες τιμές για κάποια είσοδο. Αυτό με τη σειρά του πιέζει τους επιμέρους τοπικούς γραμμικούς τελεστές του δικτύου να γίνουν συστολικοί, εξού και η ονομασία αυτής της οικογένειας ΑΚ.

Μεταβλητοί Αυτόματοι Κωδικοποιητές (ΜΑΚ)

Ίσως η πιο αξιοσήμειωτη ιδεά στους ΑΚ, η οποία έχει χρησιμοποιηθεί εκτενώς και για Παραγωγική Μοντελοποίηση είναι οι Μεταβλητοί Αυτόματοι Κωδικοποιητές (ΜΑΚ) (Variational Autoencoders - VAEs), η δομή και οι εφαρμογές των οποίων αναλύονται σε αυτήν και την επόμενη υποενότητα. Συνοπτικά, οι ΜΑΚ είναι ΑΚ στους οποίους η κατανομή πιθανότητας των ενδιάμεσων αναπαραστάσεων χρησιμοποιείται στη κανονικοποίηση (regularization) της συνάρτησης κόστους με στόχο οι αναπαράστασεις αυτές (στον λανθάνοντα πιθανοχώρο του μοντέλου) να εκπαιδευθούν αποκτώντας χρήσιμες ιδιότητες που μπορούν κατ΄ επέκταση να χρησιμοποιηθούν για την παραγωγή νέων δεδομένων.

Οι ΜΑΚ ανήκουν σε μια ευρύτερη οικογένεια πιθανοτικών μοντέλων, αυτή των Μοντέλων Λανθάνοντων Μεταβλητών (Latent Variable Models). Στα μοντέλα αυτά, μια σύνθετη πιθανοτική κατανομή μοντελοποιείται μέσω ενός συνόλου λανθάνουσων μεταβλητών (latent variables), με τις οποίες τα δεδομένα εισόδου μετασχήματιζονται ουσιαστικά σε ένα συνεχή διανυσματικό χώρο μικρότερης διάστασης από τον αρχικό. Συγκεκριμένα, δεδομένα, $\mathbf{X}$ τα οποία ακολουθούν μια κατανομή $p_{data}(\vec{x})$, αντιστοιχίζονται σε λανθάνουσες μεταβλητές Z οι οποίες με τη σειρά τους ακολουθούν μια κατανομή, $p(\vec{z})$. Η μορφή της τελευταίας ορίζεται από πριν και γι΄ αυτό η $p(\vec{z})$ ονομάζεται πρότερη (prior) κατανομή, σε αντιδιαστολή με την posterior κατανομή $p_{model}(\vec{z}|\vec{x})$ την οποία προσεγγίζουμε στους ΜΑΚ εκπαιδεύοντας το ΤΝΔ του κωδικοποιητή. Τέλος, το δίκτυο του αποκωδικοποιητή εκπαιδεύεται για να αντιστοιχίζει τις λανθάνουσες μεταβλητές πίσω στον αρχικό χώρο, προσεγγίζοντας δηλαδή την κατανομή $p_{model}(\vec{x}|\vec{z})$.

Στους ΜΑΚ, τα στοιχεία του διανύσματος της ενδιάμεσης αναπαράστασης, $\bar{z}$, αποτελούν τις λανθάνοντες μεταβλητές. Η ελπίδα είναι ότι εφόσον ο κωδικοποιητής έχει καταφέρει να μάθει έναν μετασχηματισμό σε λανθάνοντα χώρο στον οποίο οι λανθάνουσες μεταβλητές αντισοιχούν σε συγκεκρίμενα χαρακτηριστικά ή παράγοντες διαφοροποίησης (factors of variation) των παρατηρίσιμων μεταβλητών (π.χ. των τιμών των εικονοστοιχείων για

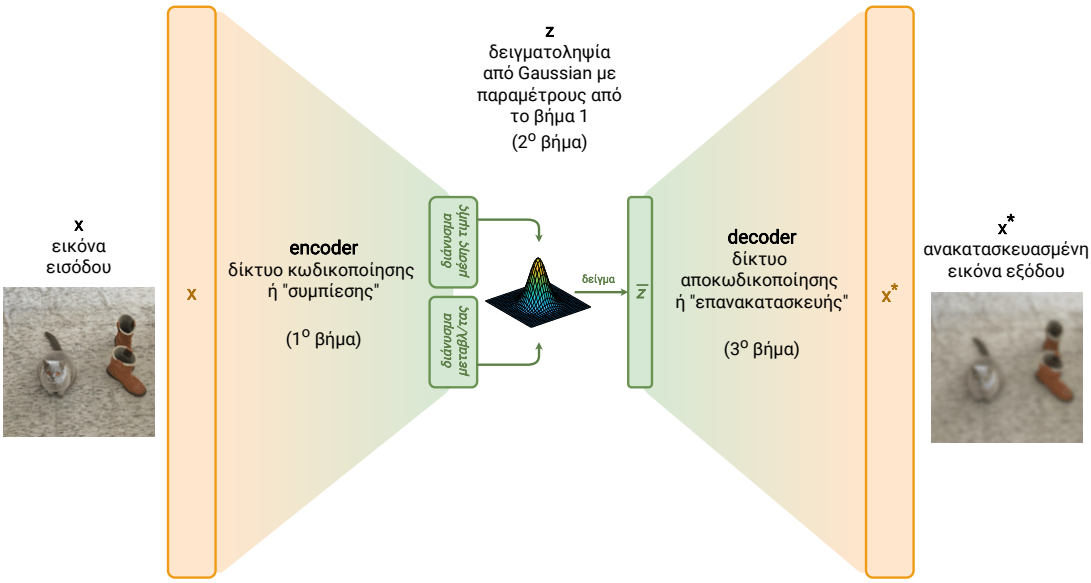
είσοδο εικόνα), τότε θεωρητικά «κινούμενοι» στον χώρο αυτόν (δηλ. κάνοντας τυχαίες δειγματοληψείες από τη prior κατανομή $p(\vec{z})$) θα παράγουμε δείγματα στην έξοδο του αποκωδικοποιητή τα οποία ανήκουν στην κατανομή των δεδομένων εκπαίδευσης. Προκειμένου να επιτευχθεί αυτός ο σκοπός, πρέπει κατά την εκπαίδευση των δύο δικτύων να γίνει η προαναφερθείσα κανονικοποίηση ως προς το λανθάνοντα χώρο. Στους ΜΑΚ για να γίνει αυτό αρχικά αντιστοιχίζουμε κάθε είσοδο σε μία κατανομή στο λανθάνοντα χώρο και όχι απλώς σε ένα σημείο. Έτσι, το ενδιάμεσο διάνυσμα $\vec{z}$ σε ένα ΜΑΚ περιέχει τις παραμέτρους αυτής της κατανομής για την εκάστοτε είσοδο, η οποία συνήθως ορίζεται να είναι Κανονική Ισοτροπική (δηλ. με διαγώνιο πίνακα συμμεταβλητότητας). Κατόπιν, δειγματοληπτούμε από αυτήν την πιθανοτική κατανομή, λαμβάνουμε ένα διάνυσμα $\vec{z} = \overline{z}$ και το ζητάμε από τον αποκωδικοποιητή να ανακατασκευάσει την αρχική είσοδο από αυτό το δείγμα στο λανθάνοντα χώρο, κάτι που απεικονίζεται γραφικά στο σχήμα 14 που ακολουθεί.

Όπως φαίνεται στο σχήμα, η αρχική εικόνα περνάει μέσα από το δίκτυο του κωδικοποιητή η έξοδος του οποίου είναι οι παράμετροι μιας Πολυδιάστατης Ισοτροπικής Κανονικής κατανομής. Ακολούθως αρχικοποιείται μια τέτοια κατανομή και γίνεται δειγματοληψία από αυτή για τη παραγωγή της εισόδου του αποκωδικοποιητή. Ο αποκωδικοποιητής καλείται από αυτό το δείγμα να ανακατασκευάσει με το ελάχιστο σφάλμα την αρχική εικόνα εισόδου. Ο κωδικοποιητής από την άλλη καλείται να μάθει τους βασικούς παράγοντες διαφοροποίησης της κατανομής των δεδομένων με σκοπό να η έξοδός του να έχει τα βασικά στοιχεία της εισόδου. Παραγωγικά μοντέλα που χρησιμοποιούν τους (απλούς) ΜΑΚ τείνουν να παράγουν θολές εξόδους λόγω της θεώρησης Κανονικής κατανομής ως πρότερης.

Η συνάρτηση κόστους επομένως των ΜΑΚ θα περιέχει εκτός από τον όρο του σφάλματος ανακατασκευής και εναν πρόσθετο όρο, την απόσταση Kullback–Leibler μεταξύ της κατανομής που αρχικοποιείται από την έξοδο του κωδικοποιητή και προσεγγίζει την posterior και της ίδιας της posterior κανονικής κατανομής (που εδώ για λόγους απλότητας δίνεται να έχει μέση τιμή 0 και μοναδιαίο πίνακα συμμεταβλητότητας):

$$L\left(\vec{x};\vec{\partial}_f,\vec{\partial}_g\right) = ReconLoss\left(\vec{x}, g\left(Sample\left[f(\vec{x};\vec{\partial}_f)\right];\vec{\partial}_g\right)\right) + \lambda * KL\left[N\left(\vec{\mu}_x,\vec{\sigma}_x\right)||N\left(\vec{0},\vec{1}\right)\right]$$

όπου $KL[]$ είναι η απόσταση Kullback–Leibler η οποία για τη περίπτωση απόστασης μεταξύ



Σχήμα 14: Απεικόνιση του τρόπου λειτουργίας και συγκερκιμένα του εμπρόσθιου περάσματος ενός Μεταβλητού Αυτόματου Κωδικοποιητή.

Πηγή: Ανακατασκευή από «Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational Autoencoders for Unbalancing Data», Teresa et al., 2020 [111]

κανονικών κατανομών έχει κλειστή μορφή [25]:

$$
\begin{aligned}
KL &= \int \left[\frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2) \right] \times p(x)dx \\
&= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} \text{tr} \left\{ E[(x-\mu_1)(x-\mu_1)^T] \, \Sigma_1^{-1} \right\} + \frac{1}{2} E[(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2)] \\
&= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} \text{tr} \, \{I_d\} + \frac{1}{2}(\mu_1-\mu_2)^T \Sigma_2^{-1}(\mu_1-\mu_2) + \frac{1}{2} \text{tr}\{\Sigma_2^{-1}\Sigma_1\} \\
&= \frac{1}{2} \left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - d + \text{tr}\{\Sigma_2^{-1}\Sigma_1\} + (\mu_2-\mu_1)^T \Sigma_2^{-1}(\mu_2-\mu_1) \right].
\end{aligned}
$$

όπου d η διάσταση του διανύσματος ενδιάμεση αναπαράστασης.

Αυτό που ουσιαστικά επιτυγχάνεται με τη επιβολή μιας συγκεκριμένης δομής για τη κατανομή των ενδιάμεσων αναπαράστεσων (ενν. κανονική πρότερη κατανομή) και με τη χρήση του πρόσθετου όρου κανονικοποίησης αυτής της δομής είναι αφενός το ότι «κοντινές» σημασιολογικά είσοδοι «πιέζονται» να αντιστοιχηθούν σε κοντινά σημεία στον λανθάνοντα χώρο. Έτσι το μοντέλο πιέζεται να μάθει βασικά χαρακτηριστικά των εισόδων και όχι απλώς να αντιγράφει την είσοδο στην έξοδο, ενώ ταυτόχρονα μπορούν να γίνουν παραμβολές μεταξύ σημείων του λανθάνοντα χώρου και αυτές να απεικονιστούν στην έξοδο ως ομαλές παραμβολές μεταξύ των εισόδων. Αφετέρου, ο λόγος για τον οποίο χρησιμοποιούνται κανονικές κατανομές σαν priors είναι για να είναι υπολογίσιμη

(tractable) η posterior, $p(\vec{z}|\vec{x})$, κάτι που αλλιώς δεν θα ίσχυε μιας και έχουμε συνεχείς μεταβλητές για την $p(\vec{x})$ και $p(\vec{z}|\vec{x}) = p(\vec{x}|\vec{z})\, p(\vec{z})\, / p(\vec{x})$.

Παραγωγική Μοντελοποίηση με ΜΑΚ

Πριν ολοκληρώσουμε την παρουσιάση των ΑΚ, θεωρούμε σκόπιμο για λόγους πληρότητας αλλά και για καλύτερη σύνδεση με τα Παραγωγικά Μοντέλα που αποτελούν το αντικείμενο της παρούσας εργασίας, να παραθέσουμε ίσως το πιο αντιπροσωπευτικό παραγωγικό μοντέλο το οποίο χρησιμοποιεί ΜΑΚ. Πρόκειται για το μοντέλο Vector-Quantized Variational Autoencoder (VQ-VAE) που παρουσιάστηκε από τον van den Oord et al. το 2017 [83] και βασίζεται στη χρήση ΜΑΚ των οποίων όμως η προτερη κατανομή δεν είναι σταθερή αλλά παραμετρική και εκπαιδεύσιμη. Επίσης, ο κωδικοποιητής δεν βγάζει συνεχείς τιμές αλλά διακριτές (δηλ. ο λανθάνοντας χώρος δεν είναι συνεχής αλλά αποτελείται από μεμονομένα σημεία), μία έμπνευση που προήλθε από τη θεωρία της διανυσματικής κβάντισης (vector quantization) [7].

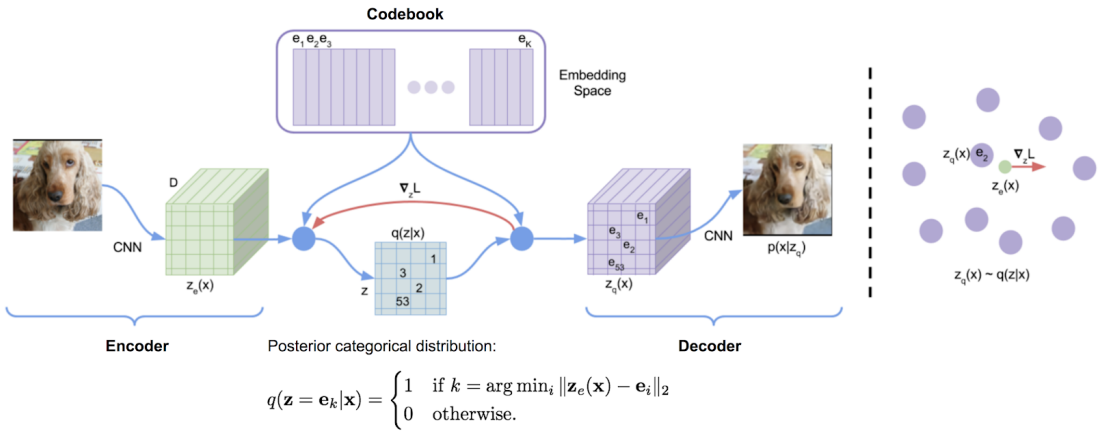
Το μοντέλο VQ-VAE επεκτείνει τον τυπικό ΑΚ προσθέτωντας ένα διακριτό λεξικό κωδικών codebook στο δίκτυο. Το λεξικό αυτό είνα μια λίστα διανυσμάτων με αντιστοιχισμένους δείκτες και χρησιμοποιείται για κβάντιση του διανύσματος ενδιάμεσης αναπαράστασης στην έξοδο του δικτύου κωδικοποίησης. Έτσι, η έξοδος αυτή συγκρίνεται με όλα τα διανύσματα του λεξικού και σαν είσοδο στο δίκτυο αποκωδικοποίησηη επιλέγεται εκείνο με τη μικρότερη ευκλείδια απόσταση:

$$z_q(\vec{x}) = \underset{i}{\operatorname{argmin}} \|z_e(\vec{x}) - \vec{e}_i\|_2$$

όπου $z_e(\vec{x})$ είναι η έξοδος του κωδικοποιητή για είσοδο $\vec{x}$, $\vec{e}_i$ είναι το i-οστό διάνυσμα του λεξικού και $z_q(\vec{x})$ είναι το κβαντισμένο διάνυσμα το οποίο περνάει στην είσοδο του αποκωδικοποιητή (βλ. σχήμα 15).

Ο κωδικοποιητής δεν βγάζει ένα μόνο διάνυσμα στην έξοδό του αλλά ένα σύνολο τέτοιων διανυσμάτων στοιχισμένα π.χ. για εφαρμογή σε είσοδο εικόνων σε ένα πλαίσιο 32×32, καθένα από τα οποία κβαντίζεται ανεξάρτητα. Έτσι δεν τίθεται θέμα ποικολότητας (ή κατάρρευσης της κατανομής - mode collapse - όπως αλλιώς ονομάζεται) καθώς ο αποκωδικοποιητής έχει $|L|^{B*B}$ πιθανές εισόδους, όπου L το μέγεθος του λεξικού και B το μέγεθος του πλαισίου.

Τα διανύσματα του λεξικού είναι και τα ίδια εκπαιδεύσιμες παράμετροι του μοντέλου.



Σχήμα 15: Σχηματική απεικόνιση της λειτουργίας του μοντέλου VQ-VAE.
Πηγή: Ανακατασκευή από «Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational Autoencoders for Unbalancing Data», Teresa et al., 2020 [111]

Επίσης η πρότερη κατανομή αυτών των διανυσμάτων είναι και ίδια εκπαιδεύσιμη. Για την εκπαίδευση του μοντέλου, η συνάρτηση κόστους περιέχει έναν όρο με το σφάλμα ανακτασκευσής και δύο ακόμη όρους για σωστή αντιστοίχιση των εξόδων του κωδικοποιητή στο λεξικό και την κατά το δυνατό καλύτερη συνεισφορά του κωδικοποιητή στην ανανέωση του λεξικού. Ακόμη, για την εκπαίδευση της πρότερης κατανομής οι δημιουργοί δοκίμασαν Αυτοπαλινδρονούμενα μοντέλα και συγκεκριμένα το PixelCNN με αρκετά εντυπωσιακά αποτελέσματα.

(α) Εικόνες από το ImageNET 128×128 (β) Ανακατασκευή από VQ-VAE με L=512 και B=32

Σχήμα 16: Εφαρμογή μοντέλου VQ-VAE σε εικόνες από το σύνολο δεδομένων ImageNET. Ο ρεαλισμός και η ποικιλία των παραγόμενων εικόνων είναι πραγματικά αξιοσημείωτα.
Πηγή: «Neural Discrete Representation Learning», van den Oord et al., 2017 [83]

Τέλος, η παρουσιάση δεν θα ήταν πλήρης εάν δεν αναφέραμε ότι μια εξέλιξη του VQ-VAE η οπόια χρησιμοποιεί ENN για εκπαίδευση της prior (ενν. αφού πρώτα έχει χρησιμοποιηθεί

μια σταθερή για την εκπαίδευση του μοντέλου), είναι το μοντέλο DALL-E που παρουσιάστηκε το 2021 από Ramesh et al. της OpenAI [121] και αποτελεί το πιο εξελιγμένο μοντέλο μετατροπής κειμένου σε εικόνα έως σήμερα (Ιούνιος 2021). Στο σχήμα 17 που ακολουθεί φαίνονται παραγωγές του μοντέλου που επιδεικνύουν την απόλυτη υπεροχή του.

Σχήμα 17: Παραγωγές του μοντέλου DALL-E της OpenAI για είσοδο τη φράση «*an armchair in the shape of an avocado*». Οι εικόνες δεν χρειάζονται περαιτέρω σχολιασμό.
Πηγή: «DALL-E: Creating Images from Text» (OpenAI Blog), Ramesh et al., 2021 [120]

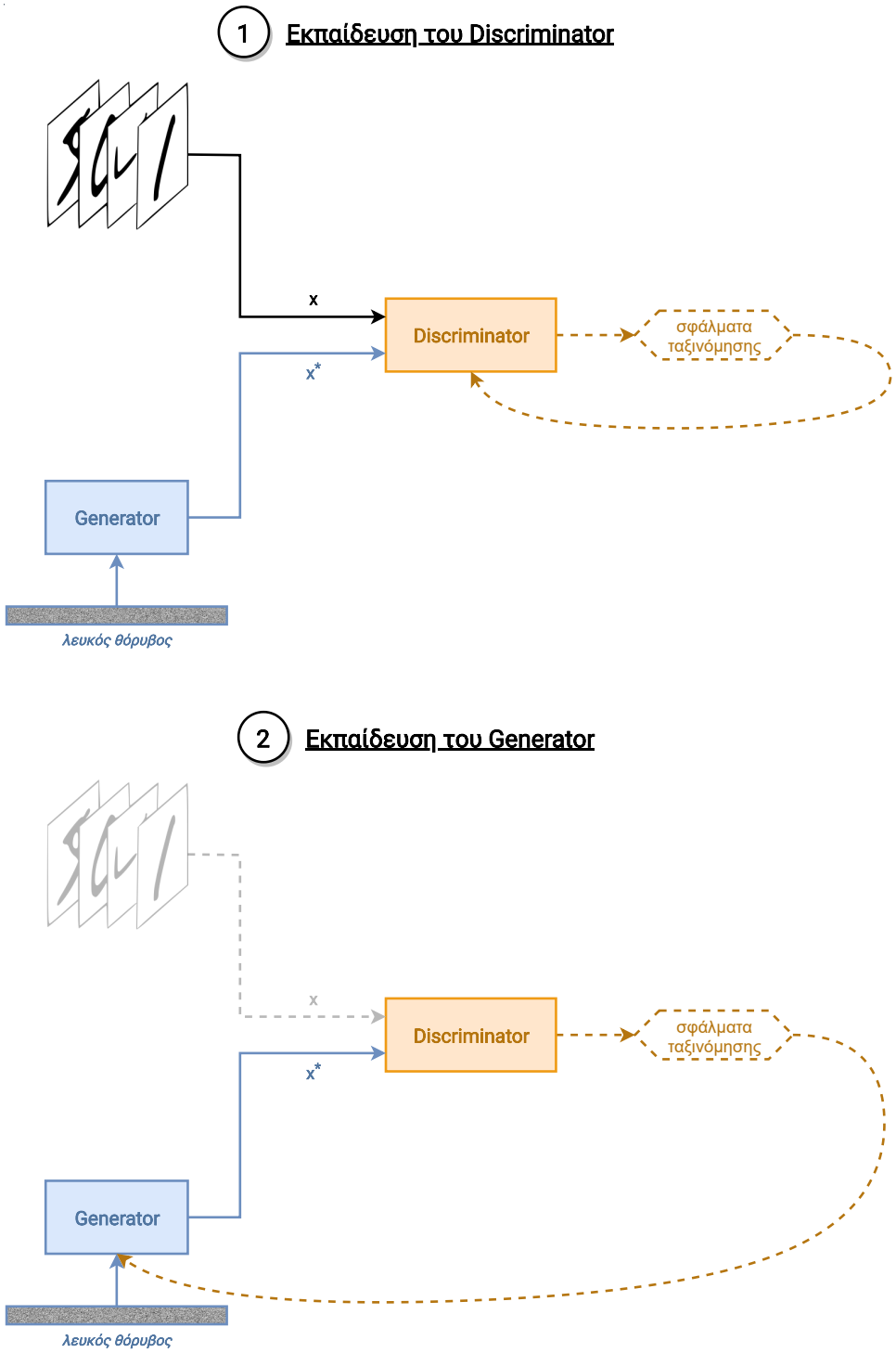
2.4 Παραγωγική Μοντελοποίηση με Generative Adversarial Networks

Μετά από την εισαγωγή σε άλλες μορφές μοντέλων και τεχνικών για Παραγωγική Μοντελοποίηση, επανερχόμαστε στο τέλος του 1ου κεφαλαίου σε μια σύντομη εισαγωγή της δομής και λειτουργίας των Παραγωγικών Αντιπαραθετικών Δικτύων Generative Adversarial Networks (GANs) - θα τα ονομάζουμε GANs για συντομία - στα οποία επικεντρωνόμστε έως το τέλος της παρούσας εργασίας. Τα GANs όπως προαναφέρθηκε, παρουσιάστηκαν από τον Ian Goodfellow et al. το 2014 [29] μια περίοδο που η Παραγωγική Μοντελοποίηση εικόνων (με την οποία ασχολήθηκαν οι δημιουργοί τους) γινόνταν κυρίως με Αυτοπαλινδρονούμενα μοντέλα ή μοντέλα ΑΚ. Αφορμή της αναζήτησης νέων τεχνικών αποτέλεσε το γεγονός ότι η παραγωγική δυνατότητα εξαρτούνταν έντονα στα Αυτοπαλινδρονούμενα μοντέλα από τη σειρά παραγωγής των εικονοστοιχείων, ενώ στους ΑΚ από τη υπόθεση πρότερης κατανομής και τη δομή αυτής. Πηγή έμπνευσης, επίσης, αποτέλεσαν οι μελέτες και προσπάθειες για αντιπαραθετική εκπαίδευση (adversarial training) ΤΝΔ, καθώς είχε βρεθεί πως η υπερβολική χρήση γραμμικών συνάρτησεων οδηγεί ένα μοντέλο με ΤΝΔ να έχει ευαισθησία σε μεταβολές της εισόδου (ενώ σε αντίθεση θα ήταν επιθυμητό να υπάρχει σχετικά σταθερή συμπεριφορά του μοντέλου γύρω από τα δείγματα του συνόλου εκπαίδευσης) [30].

Βασικό στοιχείο της δόμης των GANs, το οποίο τα διαφοροποιεί από τα προηγούμενα

ΒΠΜ, είναι η ύπαρξη και ταυτόχρονη εκπαίδευση δύο δικτύων: του Generator που δημιουργεί δείγματα κατά το δυνατό κοντινότερα σε αυτά του συνόλου εκπαίδευσης και του Discriminator που εκπαιδεύεται να ξεχωρίζει ποια δείγματα προέρχονται από το σύνολο εκπαίδευσης (δηλ. είναι τα πραγματικά - real) και ποια από τον Generator (δηλ. είναι τα τεχνητά ή ψεύτικα - fake). Συγκεκριμένα, σε κάθε βήμα εκπαίδευσης (δηλ. στο εσωτερικό του βρόγχου εκπαίδευσης - σχήμα 18) ο Discriminator, που ανήκει στη κατηγορία των Διακριτικών Μοντέλων (βλ. ενότητα 2.1), δέχεται δείγματα από το σύνολο δεδεμένων εκπαίδευσης και δείγματα που έχουν παραχθεί από τον Generator και εκπαιδεύεται να βγάζει πιθανότητα κοντά στο 1 για τα πρώτα και κοντά στα 0 για τα δεύτερα, ενώ ο Generator που ανήκει στη κατηγορία των Παραγωγικών Μοντέλων εκπαιδεύεται ώστε από είσοδο θόρυβο να βγάζει εικόνες στην έξοδό αρκετά ρεαλιστικές ώστε να μπορέσουν να «ξεγελάσουν» τον Discriminator.

Εμβαθύνοντας λίγο στην ανάλυση του τρόπου εκπαίδευσης των GANs, μπορούμε να πούμε ότι τόσο ο Generator όσο και ο Discriminator αντιπροσωπεύονται από (συνεχώς) διαφορίσιμες συναρτήσεις με εκπαιδεύσιμες παραμέτρους, όπως είναι τα ΤΝΔ, κάθε μία με τη δική της συνάρτηση κόστους. Τα δύο δίκτυα εκπαιδεύονται μέσω back-propagation χρησιμοποιώντας αμφότερα τη συνάρτηση κόστους του Discriminator, αλλά με διαφορετικό στόχο. Ο Discriminator προσπαθεί να μειώσει τη συνάρτηση κόστους τόσο για τα πραγματικά όσο και για τα τεχνητά δείγματα, ενώ ο Generator προσπαθεί να αυξήσει τη συνάρτηση κόστους του Discriminator για τα τεχνητά δείγματα που παράγει. Αξιοσημειώτο είναι, επιπρόσθετα, ότι το σύνολο δεδομένων εκπαίδευσης και μόνο αυτό καθορίζει το είδος των δειγμάτων που ο Generator μαθαίνει να παράγει. Έτσι, εάν για παράδειγμα επιθυμούμε ένα παραγωγικό μοντέλο τύπου GAN να παράγει ρεαλιστικές εικόνες από τοπία (π.χ. για εφαρμογές αναβάθμισης ρεαλισμού των frames ενός παιχνιδιού - βλ. NVIDIA DLSS® [109] και AMD FidelityFX® [118]), θα πρέπει το σύνολο δεδομένων εκπαίδευσης να αποτελείται από ζεύγη χαμηλής-υψηλής ανάλυσης εικόνων τέτοιων τοπίων. Αυτό ελοχεύει ένα σημαντικό κίνδυνο: μη-λεπτομερώς κατασκευασμένα σύνολα δεδομένων εκπαίδευσης μπορεί να οδηγήσουν ΒΠΜ τύπου GAN να παρουσιάζουν έντονη μεροληψία, κάτι που τελικά μειώνει την αξιοπιστία και χρησιμότητά τους [80], [101].



Σχήμα 18: Γραφική αναπαράσταση του εσωτερικού βρόγχου εκπαίδευσης ενός GAN. Ο Discriminator λαμβάνει είτε πραγματικές εικόνες από το σύνολο εκπαίδευσης ή εικόνες που έχουν παραχθεί από τον Generator και βγάζει ένα σκορ «ρεαλιστικότητας» για κάθε μία. Τα σκόρ χρησιμοποιούνται ακολουθιακά από κάθε δίχτυο για ανανέωση των εκάστοτε παραμέτρων.

Πηγή: Ανακατασκευή από «GANs in action: deep learning with generative adversarial networks», Langr et al., 2019 [104]

Πριν την ολοκλήρωση του παρόντος κεφαλαίου, θεωρούμε σκόπιμο να αναφέρουμε επιγραμματικά μερικές περιπτώσεις όπου τα GANs εφαρμόστηκαν ή θεωρούμε πως θα εφαρμοστούν με επιτυχία. Αρχικά, η μοντελοποίηση χαρακτηριστικών ανθρώπινων προσώπων εκτοξεύθηκε με τη χρήση GANs κάτι που απεικονίζεται στο σχήμα 2 στην αρχή του κεφαλαίου. Για τη μοντελοποίηση αυτή ως επί το πλείστο χρησημοποείται το σύνολο δεδομένων Flickr-Faces-HQ Dataset (FFHQ) το οποίο παρουσιάστηκε μαζί με το state-of-the-art μοντέλο *StyleGAN* [99], [101]. ΄Αλλο εντυπωσιακό παράδειγμα είναι η μετατροπή ενός σκίτσου σε μία φωτορεαλιστική απεικόνιση και μάλιστα σε πραγματικό χρόνο, με το μοντέλο GauGAN που αναλύεται στην ενότητα 4.2.2 παρακάτω. Αρκετές προσπάθειες, ακόμη, γίνονται και από την εταιρεία Adobe, με το τμήτα έρευνας αυτής να σχεδιάζει την επόμενη γενιά του Photoshop® με ενσωμάτωση GANs για διόρθωση [93] ή σύνθεση εικόνων [95], ενώ, στα πλαίσια της μοντελοποίησης εικόνων μόδας, από το ίδιο τμήμα έχει παρουσιαστεί ίσως το πιο αποδοτικό μοντέλο για εικονικό δοκιμαστήριο (virtual try-on), το *SieveNet* [112], παράδειγμα εφαρμογής του οποίου φαίνεται στο σχήμα 19 που ακολουθεί.

(α) Αρχική εικόνα (β) Ρούχο για δοκιμή (γ) Έξοδος μοντέλου δοκιμής

Σχήμα 19: Παραγωγή του μοντέλου SieveNet της Adobe για είσοδο την αρχική εικόνα του «δοκιμαστή» (αριστερά), καθώς και του ρούχου προς δοκιμή (κέντρο) και έξοδο την εικόνα στα με το δοκιμαστή να φοράει το ρούχο-στόχο (δεξιά). Το SieveNet θεωρείται το πιο καινοτόμο μοντέλο για virtual try-on, ενώ χρησιμοποιεί και GANs.

Πηγή: «SieveNet: A Unified Framework for Robust Image-Based Virtual Try-On», Jandial et al., 2020 [112]

Κεφάλαιο 3

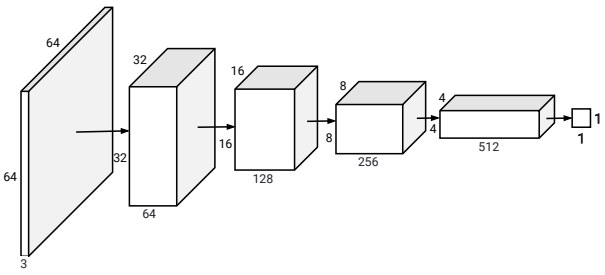
Εκπαιδευση των GANs

Στο κεφάλαιο αυτό θα εμβαθύνουμε την ανάλυσή μας σχετικά με τον τρόπο δόμησης και εκπαίδευσης ΒΠΜ τύπου GAN, θα αναφερθούμε στα διάφορα στησίματα (setups) που έχουν παρουσιαστεί για την αποτελεσματικότερη εκπαίδευση τέτοιων μοντέλων, ενώ στο τέλος θα εστιάσουμε στην αξιολόγηση των παραγωγών από GANs - όλα στα πλαίσια Παραγωγικής Μοντελοποίησης εικόνων. Όπως αναφέρθηκε και στην ανάλυση που προηγήθηκε, τα GANs αποτελούνται τυπικά από δύο (διακριτά) νευρωνικά δίκτυα, καθένα με τη δική του συνάρτηση κόστους για βελτιστοποίηση των δικών του εκπαιδεύσιμων παραμέτρων (βαρών, πολώσεων, τρεχόντων στατιστικών κλπ). Αυτό, όπως θα αναλυθεί στις παρακάτω ενότητες, δυσκολεύει σημαντικά την ευσταθή και επιτυχημένη εκπαίδευση ενός GAN, ωστόσο αποτελεί ίσως το κυριότερο χαρακτηριστικό για την επιτυχία και δημοφιλία τους.

Στις ακόλουθες ενότητες θα αναλύσουμε τον τρόπο εκπαίδευσης των GANs, ξεκινώντας από μία σύγκριση αυτού με τον τρόπο εκπαίδευσης των άμεσων προγόνων τους, των ΑΚ. Στη συνέχεια θα παρουσιάσουμε τις διάφορες μορφές σχηματισμού της συνάρτησης κόστους και εκπαίδευσης των GANs, ενώ στο τέλος του κεφαλαίου θα αναφερθούμε σε τεχνικές αξιολόγησης των παραγόμενων εικόνων από αυτά. Πρωτού συνεχίσουμε, ωστόσο, θεωρούμε σκόπιμο για λόγους πληρότητας να περιγράψουμε και εδώ τα βασικά δομικά στοιχεία ενός GAN: το δίκτυο διάκρισης ή Discriminator και το δίκτυο παραγωγής ή Generator. Σημειώνουμε στο σημείο αυτό πως για το υπόλοιπο της παρούσας εργασίας θα αναφερόμαστε στα δίκτυα αυτά με του αγγλικούς όρους για λόγους απλότητας και ευκολότερης ανάγνωσης.

Δίκτυο Διάκρισης, Discriminator

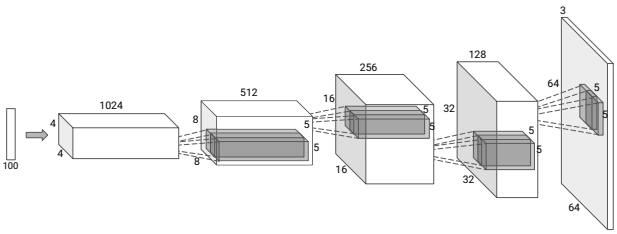
Το πρώτο δομικό μέρος ενός GAN είναι το δίκτυο διάκρισης ή Discriminator δίκτυο όπως ονομάζεται. Αυτό είναι ένα τυπικό δίκτυο ταξινομητή όπως αυτά που παρουσιάστηκαν στα διακριτικά μοντέλα (βλ. ενότητα 2.1). Στα πλαίσια της Παραγωγικής Μοντελοποίησης εικόνων οι Discriminators αποτελούνται από Συνελικτικά Νευρωνικά Δίκτυα, όπως π.χ. ο Discriminator του μοντέλου DCGAN που φαίνεται δεξιά και οποίος αποτελείται από συνελικτικές στρώσεις η μια μετά την άλλη και μια πλήρως συνδεδεμένη στρώση πριν την έξοδο για εξαγωγή της πιθανότητας ρεαλισμού της εισόδου, $p(real|\vec{x})$. Πρόκειται, επομένως, για δίκτυα *δυαδικών ταξινομητών εικόνας*. Γενικά, επειδή το έργο των Discriminators στα GANs είναι αρκετά πιο απλό σε σχέση με αυτό των ταξινομητών εικόνων (για παράδειγμα του ImageNET) που αναφέρθηκαν στο προηγούμενο κεφάλαιο, οι πρώτοι τείνουν να είναι πιο απλά δίκτυα με σημαντικά λιγότερες εκπαιδεύσιμες παραμέτρους από τα τελευταία.



Σχήμα 20: Discriminator του DCGAN (βλ. 4.1.1)
Πηγή: «Semantic Image Inpainting with Perceptual and Contextual Losses», Yeh et al., 2016 [65]

Δίκτυο Παραγωγής, Generator

Το δεύτερο δομικό μέρος ενός GAN είναι το δίκτυο παραγωγής, ή Generator δίκτυο όπως ονομάζεται. Στην περίπτωση αυτή, το δίκτυο εκτελεί έργο αντίστροφο από αυτό του Discriminator, καθώς ο στόχος εδώ είναι η έξοδος να είναι εικόνα και μάλιστα μία που παρουσιάζει ρεαλισμό και ποικιλομορφία μεταξύ των υπόλοιπων εικόνων που παράγονται. Στην απλή του μορφή, ως είσοδο στον Generator δίνεται ένα διάνυσμα λευκού θορύβου (συνήθως γκαουσιανού), το οποίο περνάει μέσα από μια ακολουθία ανεστραμμένων συνελικτικών στρώσεων (transposed convolutions). Παράδειγμα τέτοιου Generator φαίνεται επάνω δεξιά από το μοντέλο DCGAN, το οποίο ήταν και



Σχήμα 21: Generator του DCGAN (βλ. 4.1.1)
Πηγή: «Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks», Radford et al., 2016 [44]

το πρώτο που χρησιμοποιούσε τέτοιες στρώσεις για τον Generator - δομή που πλέον θεωρείται επικρατούσα. Στόχος, επομένως της εκπαίδευσης του Generator είναι να μάθει έμμεσα την κατανομή πιθανότητας των εικόνων (ενν. την πολυδιάστατη κατανομή με διαστάσεις τα εικονοστοιχεία της εικόνας εξόδου) του συνόλου δεδομένων, $p(\vec{x})$, ή τη δεσμευμένη έκδοση αυτής στη περίπτωση που στην είσοδο του Generator υπάρχει και κάποια συνθήκη, $p(\vec{x}|y)$ (βλ. 3.2).

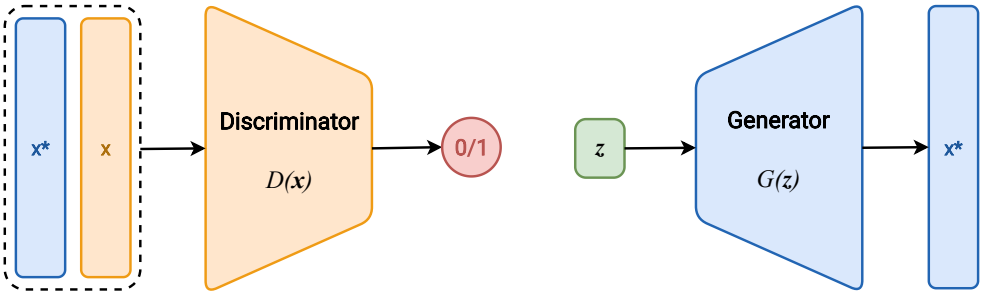
3.1 Συναρτήσεις Κόστους και Παίγνια

Σύγκριση GANs με Αυτόματους Κωδικοποιητές

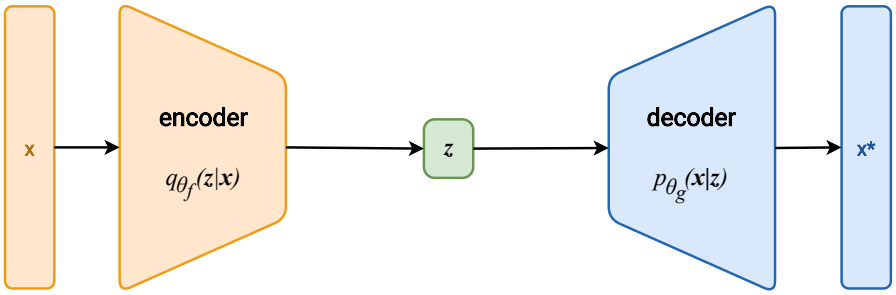
Η αντιπαραβολή των GANs με τους ΑΚ έρχεται φυσικά. Κι αυτό διότι, όπως φαίνεται και στο σχήμα 22 παρακάτω, ο Generator κάνει παρόμοια δουλειά με αυτήν του δικτύου αποκωδικοποίησης στους ΑΚ: λαμβάνοντας στην είσοδό του ένα διάνυσμα, προσπαθεί να παράξει δεδομένα τα οποία δεν ξεχωρίζουν από αυτά του συνόλου εκπαίδευσης. Επίσης, ο Discriminator λαμβάνει εικόνα στην είσοδό του, όπως αντίστοιχα γίνεται και στο δίκτυο κωδικοποίησης ενός ΑΚ.

Ωστόσο, παρουσιάζουν σημαντικές διαφορές, με τη σημαντικότερη να εντοπίζεται στον τρόπο υπολογισμού της συνάρτησης κόστους για βελτιστοποίηση των παραμέτρων του εκάστοτε μοντέλου. Έτσι, ενώ στους ΑΚ αυτή συνήθως «μετράει» το σφάλμα ανακατασκευής μιας αρχικής εικόνας από την ενδιάμεση αναπαράσταση $\vec{z}$, στα GANs χρησιμοποιείται ένα πρόσθετο νευρωνικό δίκτυο που εκπαιδεύεται να διαχωρίζει τα αληθινά από τα τεχνητά δεδομένα, σχηματίζοντας έτσι και τη συνάρτηση κόστους του Generator που μετράει το βαθμό ρεαλισμού των δειγμάτων που παράγει. Έτσι, όπως φαίνεται και στο παρακάτω σχήμα, στους ΜΑΚ, έργο του αποκωδικοποιητή (decoder) είναι να ανακατασκευάσει την είσοδο του κωδικοποιητή (encoder) από δείγμα μια εκπαιδεύσιμης κατανομής, κάτι άμεσα μετρήσιμο από το αντίστοιχο σφάλμα. Στα GANs, ο Generator πρέπει να μάθει να κατασκευάζει εικόνες μη διακρίσιμες από αυτές του συνόλου εκπαίδευσης κάνοντας χρήση όμως ενός άλλου νευρωνικού δικτύου, του Discriminator, για μέτρηση της αποτελεσματικότητάς του, δηλαδή μαθαίνοντας έμμεσα πόσο καλά τα πηγαίνει. Τέλος, στα GANs τα δύο δίκτυα αντιμάχονται, το ένα για να μειώσει τα σφάλματα ταξινόμησης ως προς το ρεαλισμό (Discriminator) και το άλλο για να τα αυξήσει στις εικόνες που παράγει (Generator).

Μια ακόμη διαφοροποίηση των δύο τεχνικών, έγκειται στο γεγονός ότι οι ΑΚ είναι αντιστρέψιμα δίκτυα, με την έννοια ότι για δεδομένη εικόνα εξόδου του αποκωδικοποιητή, ο κωδικοποιητής θα μας δώσει την ενδιάμεση αναπαράσταση από την οποία προέκυψε (ακόμα κι αν πρόκειται για παραμέτρους κατανομής, όπως στους ΜΑΚ). Στα GANs ωστόσο κάτι τέτοιο δεν είναι άμεσα εφικτό, καθώς συνήθως γίνεται μια τυχαία δειγματοληψία (noise-to-image) ή κάποιος μη αντιστρέψιμος μετασχηματισμός (image-to-image) στην είσοδο. Τέλος, μια λιγότερο καίρια διαφορά είναι ότι ο Discriminator σε ένα GAN παίρνει εικόνα και βγάζει (τουλάχιστον στη μορφή που προτάθηκαν αρχικά) έναν αριθμό που σχετίζεται με το βαθμό ρεαλισμού της εισόδου, ενώ το δίκτυο του Κωδικοποιητή (decoder) ενός ΑΚ για είσοδο εικόνα επιστρέφει ένα διάνυσμα στον λανθάνοντα χώρο του μοντέλου.



GANs: Minimax του κόστους από τα σφάλματα ταξινόμησης (ρεαλισμού)



VAEs: Μεγιστοποίηση του κόστους ανακατασκευής από δείγμα της πρότερης κατανομής

Σχήμα 22: Σύγκριση της δομής και του τρόπου εκπαίδευσης ενός GAN με ένα μοντέλο ΜΑΚ (VAE). **Πηγή:** Ανακατασκευή από «Flow-based Deep Generative Models», Lilian Weng, 2018 [98]

Σχηματισμός της συνάρτησης κόστους ενός GAN

Η συνάρτηση κόστους του καθενός από τα επιμέρους δίκτυα ενός GAN, σε αντίθεση με άλλα ΒΠΝ και Διακριτικά Μοντέλα με ΤΝΔ, δεν έχει ως παραμέτρους μόνο αυτές του

αυτού δικτύου, αλλά και του άλλου. Συγκεκριμένα, ακολουθώντας την τυπική ονοματοδοσία, έστω $\vec{\theta_G}$ οι εκπαιδεύσιμες παράμετροι του Generator δικτύου και $\vec{\theta_D}$ του Discriminator. Αντίστοιχα, η συνάρτηση κόστους θα είναι J_G για τον Generator και J_D για τον Discriminator, με κάθε δίκτυο να προσπαθεί να βελτιώσει τη δική του (πιθανόν και με διαφορετικούς αλγορίθμους βελτιστοποίησης). Επομένως, περιφραστικά οι συναρτήσεις κόστους των δικτύων ενός GAN θα είναι [50]:

$J_D\left(\vec{\theta_G}, \vec{\theta_D}\right)$ = ελαχιστοποίηση του σφάλματος ταξινόμησης σε πραγματικά και τεχνητά δεδομένα

$J_G\left(\vec{\theta_D}, \vec{\theta_G}\right)$ = μεγιστοποίηση του σφάλματος ταξινόμησης του Discriminator στα παραγόμενα δεδομένα

Παρότι, όμως, η κάθε συνάρτηση κόστους εμπλέκει και παραμέτρους του άλλου δικτύου, χρησιμοποιείται για να ανανεώσει μόνο αυτές του δικού της. Έτσι, ο αλγόριθμος βελτιστοποίησης του Discriminator, για παράδειγμα, χρησιμοποιεί την $J_D\left(\vec{\theta_D}, \vec{\theta_G}\right)$ και της μερικές παραγώγους αυτής ως προς τις παραμέτρους του Discriminator για την εκπαίδευση των παραμέτρων του Discriminator, $\vec{\theta_D}$, **χωρίς να μπορεί να επηρεάσει τις παραμέτρους του Generator** (ή τις μερικές παραγώγους αυτών). Αντίστοιχα για το δίκτυο του Generator και τον αλγόριθμο βελτιστοποίησης αυτού. Δηλαδή σε κάθε ένα από τα δύο στάδια του εσωτερικού βρόγχου εκπαίδευσης ενός GAN που απεικονίζεται στο σχήμα 18, το ένα από τα δύο δίκτυα παραμένει «παγωμένο» (frozen).

Όπως γίνεται εμφανές από τα παραπάνω τα δύο δίκτυα «αντιμάχονται» το ένα το άλλο, κάτι που ανήκει στην ευρύτερη οικογένεια τεχνικών αντιπαραθετικής εκπαίδευσης (adversarial training). Στο τέλος, τα τεχνητά δείγματα που παράγονται από τον Generator φαίνονται τόσο αληθινά στον Discriminator που το καλύτερο που έχει να κάνει είναι να μαντέψει στη τύχη σε ποια ομάδα δειγμάτων ανήκουν. Τότε, λέμε ότι η εκπαίδευση έχει ολοκληρωθεί επιτυχώς και πλέον μπορούμε να χρησιμοποιήσουμε τον εκπαιδευμένο Generator για να παράγουμε ρεαλιστικά δεδομένα παρόμοια με αυτά του συνόλου εκπαίδευσης. Ωστόσο, για να φτάσουμε στο σημείο αυτό, πρέπει να ρυθμίσουμε τον τρόπο υπολογισμού της συνάρτησης κόστους για κάθε δίκτυο και πως αυτές υπολογίζονται από τα αποτελέσματα της ταξινόμησης στην έξοδο του Discriminator, κάτι που αναλύεται στις παραγράφους που ακολουθούν.

Παίγνιο Μηδενικού Αθροίσματος - Binary Cross-Entropy

Επειδή ο Generator και ο Discriminator μπορούν να βελτιώσουν μόνο τις δικές του παραμέτρους όχι ο ένας του άλλου, η εκπαίδευση ενός GAN μπορεί καλύτερα να περιγραφεί ως ένα παίγνιο. Οι παίκτες αυτού του παιγνίου είναι τα δύο νευρωνικά δίκτυα και όσο το ένα γίνεται καλύτερο, τόσο το άλλο χειροτερεύει και προσπαθεί εκ΄ νέου να βελτιωθεί ώστε να προσπεράσει το πρώτο. Στα πλαίσια της Θεωρίας Παιγνίων, ένα τέτοιο στήσιμο (setup) είναι γνωστό ως **παίγνιο μηδενικού αθροίσματος δύο παικτών** (two-player zero-sum game), όπου τα κέρδη του ενός παίκτη ισούνται με τις απώλειες του άλλου ή, αντίστοιχα, η βελτίωση ενός παίκτη κατά ένα ποσοστό ισοδυναμεί με χειροτέρευση του άλλου κατά το ίδιο ποσοστό.

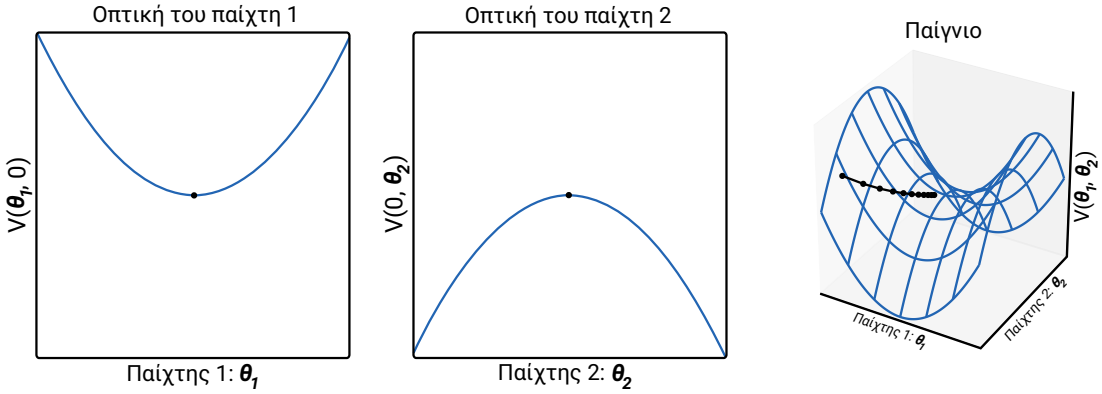
Σε όλα τα παίγνια μηδενικού αθροίσματος υπάρχει ένα σημείο ισορροπίας (ή μια minimax[1] λύση όπως λέγεται), γνωστό ως *Nash equilibrium*, ενός σημείου στο οποίο κανένας παίκτης δεν μπορεί να βελτιώσει την κατάστασή ή τις αποδόσεις του αλλάζοντας τις ενέργειές του. Αυτό συμβαίνει στα GANs, όταν όπως αναφέρθηκε και προηγούμενα, ο Generator παράγει ρεαλιστικές εικόνες μη-διακρίσιμες από αυτές του συνόλου εκπαίδευσης και ο Discriminator στην καλύτερη μπορεί να μαντέψει τυχαία εάν ένα παράδειγμα δεδομένων στην είσοδό του προέρχεται από το πραγματικό σύνολο δεδομένων ή έχει παραχθεί από τον Generator [50]. Τότε λέγεται ότι το *GAN έχει συγκλίνει* και η εκπαίδευσή του ολοκληρώνεται, κάτι που στην πράξη είναι πολύ δύσκολο να επιτευχθεί, χωρίς όμως αυτό να εμποδίζει την επιτυχή εφαρμογή τέτοιων μοντέλων στην πράξη [104].

Από μία πιο μαθηματική σκοπιά, το Nash equilibrium στα GANs επιτυγχάνεται όταν η συνάρτηση κόστους του Generator, $J_G\left(\vec{\theta_G}, \vec{\theta_D}\right)$ έχει ελαχιστοποιηθεί ως προς τις δικές του εκπαιδεύσιμες παραμέτρους, $\vec{\theta_G}$ και ταυτόχρονα η συνάρτηση κόστους του Discriminator, $J_D\left(\vec{\theta_D}, \vec{\theta_G}\right)$ έχει ελαχιστοποιηθεί ως προς τις δικές του εκπαιδεύσιμες παραμέτρους, $\vec{\theta_D}$, κάτι που απεικονίζεται γραφικά στο σχήμα 23 παρακάτω: ο παίκτης 1 (Discriminator) προσπαθεί να μειώσει τη συνάρτηση κόστους, V, βελτιστοποιώντας τις παραμέτρους του, θ_1, ενώ ο παίκτης 2 προσπαθεί να μειώσει τη δική του συνάρτηση κόστους, $-V$ (δηλ. να μεγιστοποιήσει τη V) βελτιστοποιώντας τις δικές του παραμέτρους, θ_2. Η συνολική συνάρτηση κόστους φαίνεται στο πλεγματοειδές σχήμα δεξιά του χώρου των συνολικών παραμέτρων των δύο δικτύων. Επίσης σε αυτό το σχήμα, φαίνεται (μαύρη γραμμή με

[1]Στη Θεωρία Παιγνίων, minimax τιμή ενός παίκτη είναι η μικρότερη τιμή που οι άλλοι παίκτες μπορούν να αναγκάσουν τον παίκτη να λάβει, χωρίς να γνωρίζουν τις ενέργειές του, ή ισοδύναμα, είναι η μεγαλύτερη εξασφαλισμένη αξία που μπορεί να πάρει ο παίκτης όταν ξέρει τις ενέργειες των άλλων παικτών.

τελείες) το μονοπάτι σύγκλισης στο Nash equilibrium περίπου στο κέντρο αυτού, στο διάσελο (saddle-point)

Στην αρχική έκδοση των GANs [29], ως συνάρτηση υπολογισμού του κόστους για κάθε δίκτυο είχε προταθεί η Δυαδική Διασταυρούμενη Εντροπία (Binary Cross-Entropy - BCE), η οποία και εφαρμόζεται σε εργασίες ταξινόμησης όπου υπάρχουν δύο κατηγορίες, όπως η εργασία που κάνει ο Discriminator. Όπως αναλύεται στη συνέχεια, αυτός δεν είναι ο μόνος τρόπος υπολογισμού του κόστους του κάθε δικτύου και μάλιστα ο ίσως λιγότερο χρησιμοποιούμενος λόγω του κορεσμού (saturation) και κατ΄ επέκταση η εξαφάνιση των παραγώγων (vanishing gradients) που προκαλεί η χρήση λογαρίθμων.



Σχήμα 23: Σχηματική απεικόνιση της εκπαίδευσης ενός GAN ως ένα παίγνιο μηδενικού αθροίσματος.

Πηγή: Ανακατασκευή από Goodfellow, 2019, `https://www.iangoodfellow.com/slides/2019-05-07.pdf`

Η συνάρτηση κόστους Binary Cross-Entropy

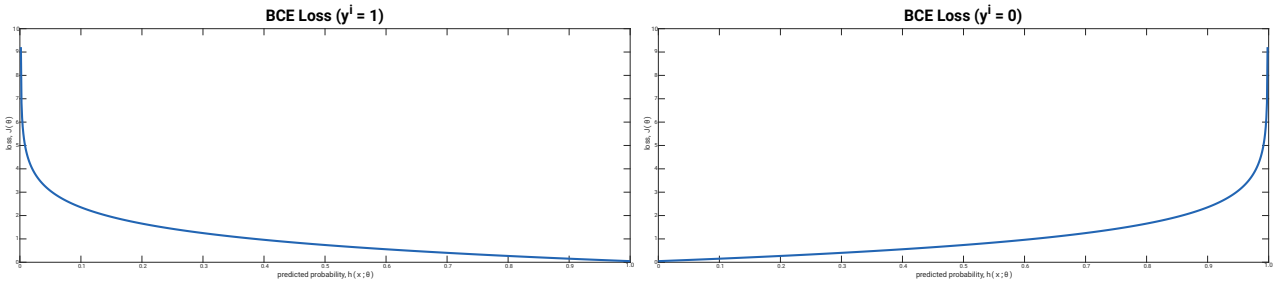
Η συνάρτηση κόστους Binary Cross-Entropy για ένα σύνολο m δειγμάτων ανά ομάδα (batch) έχει ως εξής:

$$J_m\left(\vec{\theta}\right) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}log\left(h\left(x^{(i)};\vec{\theta}\right)\right) + \left(1-y^{(i)}\right)log\left(1-h\left(x^{(i)};\vec{\theta}\right)\right)\right] \tag{3.1}$$

όπου το αρχικό άθροισμα και η διαίρεση με τον αριθμό των δειγμάτων προσεγγίζει τον τελεστή μέσης τιμής, $x^{(i)}$ είναι το i-οστό δείγμα, $y^{(i)}$ είναι η ετικέτα του i-οστού δείγματος και $\vec{\theta}$ το διάνυσμα των εκπαιδεύσιμων παραμέτρων του μοντέλου. Κατά την εκπαίδευση του Discriminator ενός GAN οι ετικέτες θα είναι 1 για τα πραγματικά δείγματα και 0 για τα τεχνητά, ενώ για την εκπαίδευση του Generator ισχύει το αντίστροφο, δηλαδή μαζί με τα

τεχνητά δείγματα θα δίνεται η ετικέτα 1 προκειμένου να υπολογίσει το κατά πόσο μπορεί να «ξεγελάσει» τον Discriminator.

Εστιάζοντας στο σχηματισμό της συνάρτησης κόστους και στις τιμές που αυτή λαμβάνει για τις ετικέτες $0/1$ που δίνονται κατά την εκπαίδευση ενός GAN, βλέπουμε ότι όταν η ετικέτα είναι 1 δρα μόνο ο πρώτος όρος του αθροίσματος, οποίος παίρνει τιμές από το 0 (όταν η $h\left(x^{(i)};\vec{\partial}\right)$ βγάζει τιμές κοντά στο 1) έως το $-\infty$ (όταν η $h\left(x^{(i)};\vec{\partial}\right)$ βγάζει τιμές κοντά στο 0). Αντίστοιχα, για ετικέτα 0, δρα μόνο ο δεύτερος όρος του αθροίσματος, οποίος παίρνει τιμές από το 0 (όταν η $h\left(x^{(i)};\vec{\partial}\right)$ βγάζει τιμές κοντά στο 0) έως το $-\infty$ (όταν η $h\left(x^{(i)};\vec{\partial}\right)$ βγάζει τιμές κοντά στο 1). Λαμβάνοντας, τέλος, υπόψη και το αρνητικό πρόσημο στην αρχή της 3.1, βλέπουμε ότι η παραπάνω προσέγγιση της Binary Cross-Entropy για ένα batch παίρνει τιμές από 0 έως $+\infty$ όταν η συνάρτηση ταξινόμησης, $h(x)$ με παραμέτρους τις ∂ παίρνει τιμές από το 0 έως το 1. Επιλογικά, η συνάρτηση κόστους Binary Cross-Entropy, έχει δύο μέρη (ένα για κάθε τάξη) και λαμβάνει τιμές κοντά στο 0 για σωστή ταξινόμηση (διαγώνιος confusion matrix) ενώ προσεγγίζει το θετικό άπειρο για λάθος (αντιδιαγώνιος confusion matrix) - συμπεριφορά που απεικονίζεται γραφικά στο σχήμα 24 παρακάτω.



(α) Για ετικέτα $y^{(i)} = 1$ (β) Για ετικέτα $y^{(i)} = 0$

Σχήμα 24: Γραφική αναπαράσταση της συνάρτησης κόστους Binary Cross-Entropy για ετικέτες 1 (πραγματικών εικόνων) αριστερά και 0 (τεχνητών) δεξιά.

Πηγή: Ανακατασκευή από Generative Adversarial Networks Specialization, Zhou et al., DeepLearning.AI, 2021 [122]

Σχηματισμός Συναρτήσεων Κόστους σε ένα GAN

Με βάση την ανάλυση που προηγήθηκε, είμαστε πλέον σε θέση να γράψουμε σε κλειστή μορφή τη συνάρτηση κόστους που καλείται να βελτιστοποιήσει το κάθε δίκτυο. Έτσι, για τον Discriminator, η συνάρτηση κόστους εάν χρησιμοποιηθεί η Binary Cross-Entropy και με δεδομένο ότι κατά την εκπαίδευσή ενός GAN τα πραγματικά δεδομένα είναι κατά σύμβαση

αντιστοιχισμένα με την ετικέτα 1 και τα τεχνητά με την 0, θα είναι:

$$J_D\left(\vec{\theta_D}, \vec{\theta_G}\right) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} log\left(h\left(x^{(i)}; \vec{\theta}\right)\right) + \left(1 - y^{(i)}\right) log\left(1 - h\left(x^{(i)}; \vec{\theta}\right)\right)\right] \tag{3.2}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[log\left(D\left(x^{(i)}; \vec{\theta_D}\right)\right) + log\left(1 - D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right)\right)\right] \tag{3.3}$$

$$= -\frac{1}{m}\sum_{i=1}^{m} log\left(D\left(x^{(i)}; \vec{\theta_D}\right)\right) - \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right)\right) \tag{3.4}$$

$$\approx -\mathbb{E}_{x \sim p_{data}} log\left[D\left(x\right)\right] - \mathbb{E}_{z \sim p_{prior}} log\left[1 - D\left(G\left(z\right)\right)\right] \tag{3.5}$$

όπου $D(x)$ είναι η έξοδος Discriminator (δηλ. η πιθανότητα ρεαλισμού της εισόδου x), $G(z)$ είναι η έξοδος του Generator δικτύου για είσοδο τυχαίο διάνυσμα z (δηλ. μία τεχνητή εικόνα), p_{data} η κατανομή που ακολουθούν τα δεδομένα εισόδου (στις εικόνες αυτή θα είναι μια πολύ υψηλής διαστασιμότητας κατανομή που μόνο έμμεσα και προσεγγιστικά μπορεί να μοντελοποιήσει ένα παραγωγικό μοντέλο τύπου GAN) και p_{prior} η prior κατανομή από την οποία δειγματολειπτούμε για να πάρουμε το τυχαίο διάνυσμα στην είσοδο του Generator. Εφόσον, όπως αναφέρθηκε, ο Discriminator βγάζει πιθανότητα και άρα $D(x) \in [0, 1]$, προκύπτει ότι για την ελαχιστοποίηση της συνάρτησης κόστους του, ο Discriminator πρέπει να μάθει να αναθέτει υψηλή πιθανότητα στα δείγματα με την ετικέτα 1 (τα οποία προέρχονται από το σύνολο δεδομένων εκπαίδευσης) και χαμηλή σε αυτά που παράγονται από τον Generator.

Το δίκτυο του Generator, με τη σειρά του, προσπαθεί να «ξεγελάσει» αυτό του Discriminator ώστε οι πιθανότητες που αναθέτει στα τεχνητά δείγματα στην έξοδό του να είναι υψηλές. Στοχεύει, δηλαδή, να μεγιστοποιήσει τον δεύτερο όρο της συνάρτησης κόστους του Discriminator - εξάλλου μόνο αυτόν τον όρο μπορεί να επηρεάσει προκειμένου η συνάρτηση κόστους του Discriminator προκειμένου αυτή να αυξηθεί. Επομένως, για τον Generator θα ισχύει:

$$J_G\left(\vec{\theta_G}, \vec{\theta_D}\right) = \frac{1}{m}\sum_{i=1}^{m}\left[\left(1 - y^{(i)}\right) log\left(1 - h\left(x^{(i)}; \vec{\theta}\right)\right)\right] \tag{3.6}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[log\left(1 - D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right)\right)\right] \tag{3.7}$$

$$= \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right)\right) \tag{3.8}$$

$$\approx \mathbb{E}_{z \sim p_{prior}} log\left[1 - D\left(G\left(z\right)\right)\right] \tag{3.9}$$

όπου το αρνητικό πρόσημο στην αρχή έχει πλέον αφαιρεθεί καθώς ο Generator προσπαθεί ελαχιστοποιώντας τη δική του συνάρτηση κόστους να αυξήσει αυτή του Discriminator,

ενώ όλα τα υπόλοιπα μεγέθη είναι όπως πριν. Επειδή, ο πρώτος όρος της εξίσωσης 3.5 εξαρτάται μόνο από το σύνολο δεδομένων εκπαίδευσης, συνηθίζεται στη βιβλιογραφία η παραπάνω συνάρτηση κόστους του Generator να δηλώνεται ως το αρνητικό της συνάρτησης κόστους του Discriminator:

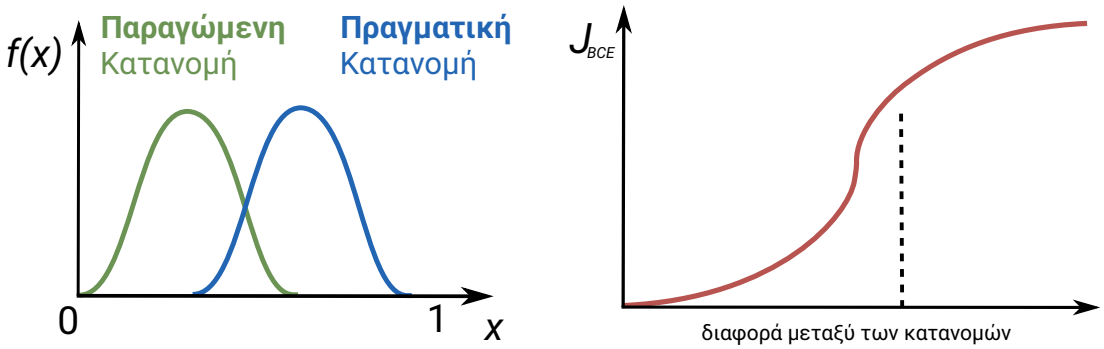
$$J_G\left(\vec{\theta_G}, \vec{\theta_C}\right) = -J_D\left(\vec{\theta_C}, \vec{\theta_G}\right) \tag{3.10}$$

κάτι που εξηγεί γιατί η εκπαίδευση ενός GAN χρησιμοποιώντας τη συνάρτηση κόστους Binary Cross-Entropy είναι ισοδύναμη με ένα παίγνιο μηδενικού αθροίσματος δύο παικτών. Τέλος, όπως αποδεικνύεται στο [29] αυτό το στήσιμο (setup) για εκπαίδευση των GANs έχει την πρόσθετη ιδιότητα ότι η συνάρτηση κόστους είναι ασυμπτωτικά συνεπής με τη απόσταση Jensen-Shannon (Jensen-Shannon divergence - JSD) μεταξύ της κατανομής των δεδομένων και αυτής που έχει μάθει ο Generator και από την οποία παράγει τα (τεχνητά) δείγματα στην έξοδό του.

Ένα βασικό μειονέκτημα που παρουσιάζει η χρήση της Binary Cross-Entropy και συγκεκριμένα η ύπαρξη των λογαρίθμων στις συναρτήσεις κόστους είναι ο κορεσμός. Εμβαθύνοντας, όταν δύο κατανομές είναι «μακριά» η Binary Cross-Entropy παίρνει υψηλές τιμές ωστόσο αυτές φτάνουν σε έναν κορεσμό και έτσι περαιτέρω μετακίνηση των κατανομών δεν διαφοροποιεί σημαντικά τις τιμές αυτές. Κατ΄ επέκταση, ιδιαίτερα στην αρχική φάση της εκπαίδευσης όπου η κατανομή που έχει μάθει ο Generator απέχει αρκετά από την πραγματική πολυδιάστατη κατανομή των δεδομένων εκπαίδευσης, η Binary Cross-Entropy στις συναρτήσεις κόστους δεν δίνει σαφή «πληροφορία» για το πως πρέπει να μεταβάλλει τις παραμέτρους του. Το ίδιο όμως συμβαίνει και όταν αυτές οι κατανομές είναι σχετικά κοντά, δηλαδή από κάποιο σημείο εκπαίδευσης και ύστερα. Έτσι, όπως φαίνεται και στο σχήμα 25, όταν οι κατανομές είναι αρκετά μακριά ή κοντά το κόστος παρουσιάζει κορεσμό. Αυτό προκαλεί την εξαφάνιση των παραγώγων (vanishing gradients), κάτι που έχει οδηγήσει στη χρήση εναλλακτικών συναρτήσεων κόστους για την εκπαίδευση GANs.

Αυτό το πρόβλημα είναι γνωστό ως vanishing gradients και οδηγεί συχνά την εκπαίδευση ενός GAN σε Συρρίκνωση Ρυθμών[2] (Mode Collapse). Η Συρρίκνωση Ρυθμών στα πλα-

[2]Ρυθμός μιας πιθανοτικής κατανομής ονομάζεται μία «περιοχή» αυτής με υψηλή συγκέντρωση παρατηρήσεων. Για παράδειγμα, σε μία Κανονική κατανομή, η περιοχή γύρω από τη μέση τιμή είναι ο μοναδικός ρυθμός της κατανομής, ενώ υπάρχουν άλλες κατανομές με περισσότερους από έναν ρυθμούς χωρίς αναγκαστικά αυτοί να αντιστοιχίζονται σε κάποια ροπή. Στο σύνολο δεδομένων χειρόγραφων ψηφίων του MNIST, για παράδειγμα, εντοπίζουμε 10 τέτοιους ρυθμούς, έναν για κάθε ψηφίο και η Συρρίκνωση Ρυθμών θα οδηγούσε έναν Generator στην παραγωγή εικόνων από ένα μόνο ψηφίο.



Σχήμα 25: Γραφική απεικόνιση του προβλήματος κορεσμού της συνάρτησης κόστους Binary Cross-Entropy.

Πηγή: Ανακατασκευή από Generative Adversarial Networks Specialization, Zhou et al., DeepLearning.AI, 2021 [122]

ίσια της Παραγωγικής Μοντελοποίησης έχει να κάνει με την ποικιλία (diversity) των δειγμάτων που μπορεί να παράξει ένας εκπαιδευμένος Generator και συγκεκριμένα με τη δραστική συρρίκνωση αυτής σε δείγματα μίας μόνο κατηγορίας ή κλάσης. Όπως αναλύεται από τον Mao et al. [56], η χρήση της συνάρτησης κόστους Binary Cross-Entropy οδηγεί τον Generator στο «ασφαλές» μονοπάτι να παράγει καλά δείγματα μίας μόνο κλάσης, δηλαδή ενθαρρύνει τη Συρρίκνωση Ρυθμών. Αυτή η παρατήρηση, έχει αφενός οδηγήσει στη χρήση έντονης κανονικοποίησης των συναρτήσεων κόστους εκπαίδευσης GANs, αλλά αφετέρου έχει αποτελέσει τον λόγο που πλέον δεν χρησιμοποιούνται σχεδόν πουθενά συναρτήσεις κόστους Binary Cross-Entropy στην εκπαίδευση των GANs, με τη θέση τους να έχουν πάρει άλλες οι οποίες επιδεικνύουν λιγότερο ή καθόλου κορεσμό. Οι βασικότερες από αυτές, η συνάρτηση κόστους Μέσου Τετραγωνικού Σφάλματος και η Απόσταση Wasserstein, αναλύονται στις παραγράφους που ακολουθούν.

Συνάρτηση Κόστους Ελαχίστων Τετραγώνων (LSGAN)

Μία άλλη συνάρτηση κόστους που δοκιμάστηκε στην εκπαίδευση GANs με σκοπό τη σταθεροποίηση αυτής και τη μείωση των περιπτώσεων Κατάρρευσης Ρυθμών και εξαφάνισης των παραγώγων (vanishing gradients), είναι η συνάρτηση κόστους Ελαχίστων Τετραγώνων (Least Squares Loss). Για τον υπολογισμό της συνάρτησης κόστους Ελαχίστων Τετραγώνων χρησιμοποιείται το μέσο τετραγωνικό σφάλμα (mean square error - MSE) μεταξύ των προβλέψεων του Discriminator και των «ιδανικών» προβλέψεων αντίστοιχα για κάθε δίκτυο και για πραγματικές/τεχνητές εικόνες (Discriminator: 1 για τις πραγματικές εικόνες και 0 για τις τεχνητές, Generator: 1 για τις τεχνητές εικόνες που παράγει και με

τις οποίες θέλει να «ξεγελάσει» τον Discriminator ότι είναι πραγματικές).

Πρωτοεμφανιζόμενη στα πλαίσια εκπαίδευσης GANs στο μοντέλο LSGAN [56], η συνάρτηση κόστους Ελαχίστων Τετραγώνων βασίζεται στη μέθοδο ελαχίστων τετραγώνων από τη Στατιστική, όπου προσπαθούμε μεταβάλλοντας ένα σύνολο παραμέτρων να ελαχιστοποιήσουμε το άθροισμα των τετραγώνων των αποστάσεων ή σφαλμάτων μιας ποσότητας ενδιαφέροντος από μία ποσότητα αναφοράς[3]. Στα GANs, όπως αναφέρθηκε, η αναφορά είναι οι εκάστοτε ετικέτες και η ποσότητα ενδιαφέροντος είναι οι οι εκάστοτε προβλέψεις του Discriminator. Σε αυτή την περίπτωση, ωστόσο, ο Discriminator δεν καλείται να βγάλει στην έξοδό του μια πιθανότητα (δηλ. μια τιμή από 0 έως 1, π.χ. χρησιμοποιώντας Sigmoid συνάρτηση εξόδου), αλλά μια οποιαδήποτε τιμή, παρόλο που με το πέρας μιας επιτυχημένης εκπαίδευσης αυτή θα είναι κοντά στις τιμές των ετικετών.

Αντιπαραβάλλοντας με τις εξισώσεις 3.5 και 3.9 της προηγούμενης παραγράφου, οι συναρτήσεις κόστους τις οποίες προσπαθούν να ελαχιστοποιήσουν τα δίκτυα ενός GAN που εκπαιδεύεται με τη συνάρτηση κόστους Ελαχίστων Τετραγώνων, θα είναι:

$$J_D\left(\vec{\theta_D}, \vec{\theta_G}\right) = \frac{1}{m}\sum_{i=1}^{m}\left[\left(D\left(x^{(i)}; \vec{\theta_D}\right) - 1\right)^2 + \left(D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right) - 0\right)^2\right] \tag{3.11}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[\left(D\left(x^{(i)}; \vec{\theta_D}\right) - 1\right)^2 + \left(D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right)\right)^2\right] \tag{3.12}$$

$$\approx \mathbb{E}_{x\sim p_{data}}\left[(D(x) - 1)^2\right] + \mathbb{E}_{z\sim p_{prior}}\left[(D(G(z)))^2\right] \tag{3.13}$$

για τον Discriminator, ενώ για τον Generator αντίστοιχα και μετά από πράξεις, θα είναι

$$J_G\left(\vec{\theta_G}, \vec{\theta_D}\right) = \frac{1}{m}\sum_{i=1}^{m}\left[\left(D\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_D}\right) - 1\right)^2\right] \tag{3.14}$$

$$\approx \mathbb{E}_{z\sim p_{prior}}\left[(D(G(z)) - 1)^2\right] \tag{3.15}$$

αφού ο Generator θέλει να «δει» πόσο μακριά από τα δείγματα με ετικέτα 1 κατατάσσει ο Discriminator τα δείγματα που παρήγαγε.

Σε αντίθεση με τη συνάρτηση κόστους Binary Cross-Entropy, στο σχηματισμό της συνάρτησης κόστους Ελαχίστων Τετραγώνων δεν εμφανίζονται οι λογάριθμοι. Αυτό έχει ως άμεση συνέπεια το άθροισμα των δύο όρων της συνάρτησης κόστους του Discriminator

[3]Συνήθης εφαρμογή της μεθόδου Ελαχίστων Τετραγώνων στη Στατιστική αποτελεί η Γραμμική Παλινδρόμηση. Εκεί, δοθέντων κάποιων σημείων ή δεδομένων, μας ζητείται να βρούμε μια ευθεία, οι παράμετροι της οποίας ελαχιστοποιούν το άθροισμα των τετραγώνων των (κάθετων) αποστάσεων από τα σημεία.

να μην παρουσιάζει αυτή τη σιγμοειδή μορφή και άρα το φαινόμενο κορεσμού εδώ είναι σημαντικά μειωμένο. Συγκεκριμένα, μονάχα στην περίπτωση όπου ή έξοδος του Discriminator είναι 1 για τα πραγματικά δίκτυα και 0 για τα τεχνητά (δηλ. τέλεια πρόβλεψη), η συνάρτηση κόστους της 3.13 θα δώσει μηδενικά gradients[4]. Όπως αναλύεται και στο άρθρο του LSGAN, αυτό έχει σαν αποτέλεσμα αντίστοιχη μείωση σε εμφάνιση του φαινομένου Συρρίκνωσης Ρυθμών σε διάφορες αρχιτεκτονικές GANs που δοκιμάστηκαν και σημαντικά σταθερότερη εκπαίδευση (δηλ. πιο ομαλές τιμές των συναρτήσεων κόστους) [56]. Στη πλειοψηφία των μοντέλων που αναπτύχθηκαν στα πλαίσια της παρούσας εργασίας γίνεται χρήση της συνάρτησης κόστους Ελαχίστων Τετραγώνων τόσο λόγω αυτών που ελέχθησαν παραπάνω που έχουν να κάνουν με την ευστάθεια της εκπαίδευσης, όσο και λόγω της ταχύτητας υπολογισμού αυτής.

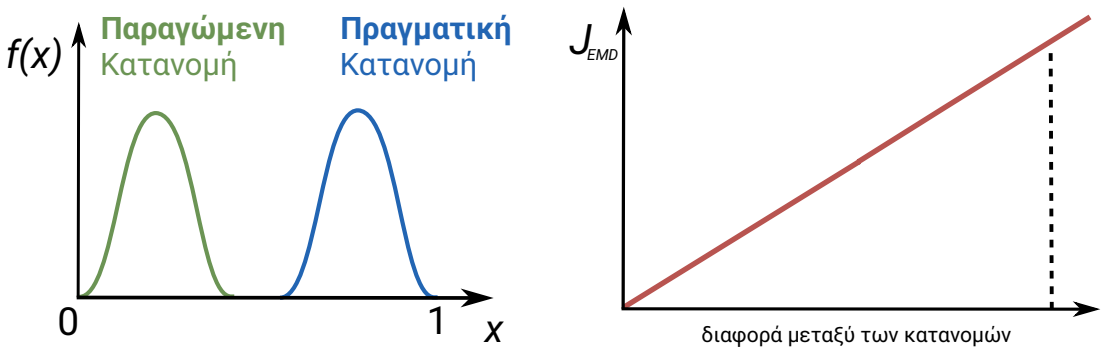
Συνάρτηση Κόστους Wasserstein (WGAN)

Ταυτόχρονα σχεδόν με τη συνάρτηση κόστους Ελαχίστων Τετραγώνων, παρουσιάστηκε και μία εναλλακτική συνάρτηση κόστους, η οποία βασίζεται στην απόσταση Μετακίνησης Εδάφους (Earth Mover's Distance - EMD) μεταξύ δύο κατανομών και η οποία επίσης είχε σαν στόχο την εξάλειψη του προβλήματος της Κατάρρευσης Ρυθμών που συχνά συνόδευε την εκπαίδευση GANs με συνάρτηση κόστους Binary Cross-Entropy. Πρόκειται για τη συνάρτηση κόστους Wasserstein, η οποία είναι μια συνάρτηση που μετράει την απόσταση μεταξύ δύο κατανομών και επειδή βασίζεται στην EMD το κάνει πιο εύρωστα και αποδοτικά από την Binary Cross-Entropy. Πιο συγκεκριμένα και με βάση τις δύο κατανομές που απεικονίζονται στο σχήμα 25 (μία αυτή που έχει μάθει ο Generator και μία η κατανομή των δεδομένων εκπαίδευσης), η EMD μετράει την απόσταση μεταξύ των δύο κατανομών προσεγγίζοντας το έργο που απαιτείται για να γίνει η παραγόμενη κατανομή (δηλ. αυτή που έχει μάθει ο Generator) ίση με με την πραγματική. Διαισθητικά, έαν η παραγόμενη κατανομή ήταν σωρός από χώμα, η EMD θα μας έδινε μια προσέγγιση της

[4]Όταν αναφέρουμε την έκφραση «*η συνάρτηση κόστους θα δώσει παραγώγους ή gradients*» εννοούμε το πρώτο βήμα του αλγορίθμου back-propagation για υπολογισμό των μερικών παραγώγων της συνάρτησης κόστους ως προς κάθε εκπαιδεύσιμη παράμετρο. Η έκφραση αυτή έγκειται στον υπολογισμό των παραγώγων της συνάρτησης κόστους ως προς τις εξόδους του Discriminator, οι οποίες διοχετεύονται αρχικά μέχρι την είσοδο του Discriminator και κατόπιν αναδρομικά φτάνουν στις στρώσεις έως και την είσοδο του Generator. Κορεσμός της συνάρτησης κόστους, επομένως, είναι όταν το feedback που λαμβάνει ο Discriminator στην έξοδό του δεν μεταβάλλεται καθόλου ή μεταβάλλεται λίγο για μικρές αλλαγές της εξόδου του - και τότε δεν μπορεί να γνωρίζει προς ποια κατεύθυνση πρέπει να «κινηθεί» στον χώρο των εκπαιδεύσιμων παραμέτρων προκειμένου να μειώσει τη συνάρτηση κόστους.

«δυσκολίας» μετακίνησης του χώματος ώστε ο τελικός σωρός (μετά τη μετακίνηση) να έχει τη μορφή και να βρίσκεται στην ίδια θέση με αυτόν της πραγματικής κατανομής.

Το πρόβλημα με τα vanishing gradients και την Binary Cross-Entropy είναι ότι όσο πιο μακριά (αντ. κοντά) είναι οι δύο κατανομές τόσο οι τιμές τείνουν όλο και πιο κοντά στο 1 (αντ. στο 0). Αντίθετα, στην EMD και άρα στη συνάρτηση κόστους Wasserstein δεν υπάρχει κάποια μέγιστη ή ελάχιστη τιμή - οι τιμές που μπορεί να πάρει η συνάρτηση είναι πρακτικά όλο το $\mathbb{R}$. Έτσι, ακόμα και όταν οι κατανομές είναι αρκετά μακριά, όπως στην αρχή της εκπαίδευσης, οι τιμές που δίνει η EMD δεν παρουσιάζουν κορεσμό. Όπως φαίνεται και στο σχήμα 26 που ακολουθεί, όταν οι κατανομές είναι αρκετά μακριά ή κοντά το κόστος δεν παρουσιάζει κορεσμό, κάτι που διαφοροποιεί την EMD από την Binary Cross-Entropy (βλ. σχήμα 25). Ως αποτέλεσμα, η χρήση της EMD στις συναρτήσεις κόστους εκπαίδευσης GANs εξαλείφει το φαινόμενο εξαφάνισης των παραγώγων (vanishing gradients), κάτι που έχει οδηγήσει στη υιοθέτησή της (μέσω της προσεγγιστικής της μορφής, της συνάρτησης κόστους Wasserstein), για την εκπαίδευση GANs. Ως αποτέλεσμα αυτού, η εκπαίδευση GANs με τη συνάρτηση κόστους Wasserstein μειώνει τη πιθανότητα κατάρρευσης ρυθμών στον Generator, οδηγώντας τον σε πιο ευσταθή εκπαίδευση πιθανόν με καλύτερα αποτελέσματα.



Σχήμα 26: Γραφική απεικόνιση της εξόδου της Απόστασης Μετακίνησης Εδάφους (EMD) μεταξύ δύο κατανομών, αυτής που έχει μάθει ο Generator και της πραγματικής.
Πηγή: Ανακατασκευή από Generative Adversarial Networks Specialization, Zhou et al., DeepLearning.AI, 2021 [122]

Επανερχόμενοι στη συνάρτηση κόστους Wasserstein (Wasserstein Loss - W-Loss), αυτή όπως αναφέρθηκε προσεγγίζει την απόσταση Μετακίνησης Εδάφους (EMD). Η προσέγγιση στα πλαίσια της εκπαίδευσης GANs, σύμφωνα με τον Arjovsky et al. στο άρθρο τους

Wasserstein GAN [69], είναι η ακόλουθη:

$$W\left(\mathbb{P}_{data}, \mathbb{P}_{gen}\right) = \mathbb{E}_{x\sim p_{data}}\left[f\left(x\right)\right] - \mathbb{E}_{\vec{z}\sim p_{prior}}\left[f\left(\hat{x} = G\left(\vec{z}\right)\right)\right] \tag{3.16}$$

όπου $\mathbb{P}_{data}, \mathbb{P}_{gen}$ είναι οι κατανομές των δεδομένων και του Generator αντίστοιχα, p_{prior} είναι η κατανομή από την οποία δειγματοληπτείται το τυχαίο διάνυσμα στην είσοδο του Generator και f είναι συνάρτηση συνεχής κατά Lipschitz[5] (με σταθερά $K = 1$). Ο Discriminator (που όταν χρησιμοποιείται η W-Loss ονομάζεται Critic επειδή πλέον δεν καλείται να κάνει δυαδική ταξινόμηση - για λόγους απλότητας θα κρατήσουμε την αρχική ονομασία) θέλει να μεγιστοποιήσει αυτή τη συνάρτηση κόστους απομακρύνοντας κατά το δυνατό περισσότερα τις δύο κατανομές ώστε να είναι πιο εύκολο το έργο του, ενώ ο Generator θέλει να την ελαχιστοποιήσει, καθώς έτσι τα δείγματα που παράγει θα είναι όλο και πιο κοντά σε αυτά του συνόλου εκπαίδευσης. Επομένως, οι συναρτήσεις κόστους Wasserstein των δύο δικτύων θα είναι:

$$J_C\left(\vec{\theta_C}, \vec{\theta_G}\right) = -\frac{1}{m}\sum_{i=1}^{m}\left[C\left(x^{(i)}; \vec{\theta_C}\right)\right] + \frac{1}{m}\sum_{i=1}^{m}\left[C\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_C}\right)\right] \tag{3.17}$$

$$\approx -\mathbb{E}_{x\sim p_{data}}\left[C\left(x\right)\right] + \mathbb{E}_{z\sim p_{prior}}\left[C\left(G\left(z\right)\right)\right] \tag{3.18}$$

για τον Discriminator (συμβ. με «c» από το Critic που όπως αναφέρθηκε χρησιμοποιήθηκε στο WGAN), ενώ για τον Generator αντίστοιχα θα είναι:

$$J_G\left(\vec{\theta_G}, \vec{\theta_c}\right) = -\frac{1}{m}\sum_{i=1}^{m}\left[C\left(G\left(\vec{z}^{(i)}; \vec{\theta_G}\right); \vec{\theta_C}\right)\right] \tag{3.19}$$

$$\approx -\mathbb{E}_{z\sim p_{prior}}\left[C\left(G\left(z\right)\right)\right] \tag{3.20}$$

όποτε και φαίνεται ότι ελαχιστοποιώντας τη συνάρτηση κόστους του ο Generator μεγιστοποιεί τον δεύτερο όρο και άρα τη συνάρτηση κόστους του Discriminator - εξού και η λέξη «adversary» (= αντιπαράθεση).

Εστιάζοντας στη συνεχή κατά 1-Lipschitz συνάρτηση f, αυτή στα GANs θα είναι το ίδιο το δίκτυο του Discriminator, το οποίο λαμβάνοντας μια εικόνα, x, καλείται να δώσει έναν

[5] Μία μονοδιάστατη συνάρτηση είναι συνεχής κατά Lipschitz με σταθερά K όταν υπάρχει διπλός κώνος (σε σχήμα X) που σχηματίζεται από τις ευθείες $y = \pm Kx$ και του οποίου το κέντρο μπορεί να κινείται επάνω στη συνάρτηση έτσι ώστε ολόκληρη η καμπύλη της συνάρτησης να παραμένει πάντα έξω από τον διπλό κώνο (δηλ. μέσα στην αριστερή και δεξιά μεριά του X). Αυτό εξασφαλίζει, ότι σε κανένα της σημείο, η συνάρτηση δεν αυξάνει πιο γρήγορα από K. Για $K = 1$, για παράδειγμα, ο κώνος αποτελείται από τις ευθείες $y = \pm x$ και η συνάρτηση δεν μπορεί σε κανένα σημείο του πεδίου ορισμού της να αυξάνει πιο γρήγορα από γραμμικά. Έτσι, συναρτήσεις όπως οι εκθετικές δεν είναι συνεχής κατά Lipschitz με σταθερά $K = 1$.

πραγματικό αριθμό. Επομένως, η υπό συζήτηση συνάρτηση, η οποία θα συμβολίζεται με $c(x)$ (αντί για $D(x)$ όπως στις προηγούμενες παραγράφους), θα είναι:

$$c : \mathcal{X} \to \mathbb{R}, \|c\|_L \leq 1 \tag{3.21}$$

όπου $\mathcal{X}$ είναι το πεδίο ορισμού των εικόνων του συνόλου εκπαίδευσης (π.χ. για έγχρωμες εικόνες 64×64 με 8 bits/pixel θα είναι ο χώρος $255^{3*64*64} = 255^{12288}$), ενώ η συνθήκη στα δεξιά δηλώνει ότι η συνάρτηση πρέπει να ικανοποιεί τη συνθήκη συνέχειας 1-Lipschitz. Για να προσεγγίζει επιτυχώς ένα νευρωνικό δίκτυο με εκπαιδεύσιμες παραμέτρους $\vec{\theta}$ μία συνάρτηση συνεχή κατά 1-Lipschitz, θα πρέπει η το μέτρο των μερικών του παραγώγων της εξόδου του δικτύου ως προς τις εκπαιδεύσιμες παραμέτρους να είναι το πολύ 1 *σε κάθε σημείο του πεδίου ορισμού* [69]. Έτσι, όταν ένα GAN εκπαιδεύεται με συνάρτηση κόστους Wasserstein, το νευρωνικό δίκτυο του Discriminator θα πρέπει να ικανοποιεί την ακόλουθη συνθήκη συνέχειας ώστε να είναι μια συνεχής συνάρτηση κατά 1-Lipschitz:

$$\left\|\nabla_{\vec{\theta_C}} C\left(x; \vec{\theta_C}\right)\right\|_2 \leq 1 \quad \forall x \in \mathcal{X} \iff \|c\|_L \leq 1 \tag{3.22}$$

Η επιβολή αυτή της συνθήκης εξασφαλίζει ότι η συνάρτηση κόστους είναι *έγκυρη* κατά τη μέτρηση της απόστασης κατανομών με την EMD (στην οποία βασίζεται η Wasserstein), με την έννοια ότι δεν είναι μόνο συνεχής και διαφορίσιμη, αλλά και *δεν αυξάνει υπερβολικά γρήγορα*.

Πρακτικά, ωστόσο, η παραπάνω συνθήκη είναι αδύνατο να επιβληθεί, διότι απαιτεί αξιολόγηση των παραγώγων του Discriminator σε κάθε σημείο του πεδίου ορισμού του, δηλαδή σε κάθε πιθανή εικόνα εισόδου. Επίσης η αξιολόγηση θα πρέπει να επαναλαμβάνεται κάθε φορά που μεταβάλλονται οι παράμετροι του δικτύου. Έχουν προταθεί διάφοροι τρόποι για επιβολή της συνθήκης της 3.22 στο νευρωνικό δίκτυο του Discriminator, με κυρίαρχες τον Ψαλιδισμό Βαρών (Weight Clipping), την Ποινή Παραγώγων (Gradient Penalty) και την Κανονικοποίηση Φάσματος (Spectral Normalization), οι οποίες και αναλύονται ακολούθως. Στο σημείο αυτό είναι σκόπιμο αναφερθεί ότι ο λόγος που προχωράμε σε μια λεπτομερή ανάλυση της συνάρτησης κόστους Wasserstein είναι ότι βρέθηκε και στα δικά μας πειράματα (βλ. 6) ότι αυτή οδηγεί σε πιο σταθερή εκπαίδευση και πιο ρεαλιστικές και υψηλής ποιότητας εικόνες στα μοντέλα που εκπαιδεύτηκαν.

Επιβολή συνθήκης συνέχειας κατά Lipschitz με Ψαλιδισμό Βαρών

Η μέθοδος που προτάθηκε στο WGAN [69] προκειμένου να επιβληθεί η συνθήκη για συνέχεια κατά 1-Lipschitz είναι ο Ψαλιδισμός των Βαρών (Weight Clipping) του Discriminator

σε ένα προκαθορισμένο πεδίο τιμών. Αυτό που επιτυγχάνεται με αυτόν τον τρόπο είναι τα βάρη δεν μεταβάλλονται πολύ σε κάθε βήμα και άρα έμμεσα εξασφαλίζεται ότι η παράγωγος της συνάρτησης του Discriminator παραμένει ελεγχόμενη. Έτσι, αυτό που πρακτικά δοκίμασαν οι συγγραφείς του WGAN είναι σε κάθε βήμα του αλγορίθμου βελτιστοποίησης του Discriminator, μετά την ανανέωση των βαρών του (δηλ. σχεδόν όλων των εκπαιδεύσιμων παραμέτρων), γίνονταν ψαλιδισμός αυτών και έτσι όσα ήταν μεγαλύτερα από τη προκαθορισμένη μέγιστη ή μικρότερα από την ελάχιστη μεταβάλλονταν στην αντίστοιχη μέγιστη ή ελάχιστη τιμή (clipping). Κατά την εκπαίδευση του WGAN προτάθηκε οι τιμές αυτές να είναι ± 0.01.

Μονολότι η μέθοδος αυτή βοήθησε στη σταθεροποίηση και την επιτυχία της εκπαίδευσης του WGAN σε διάφορα σύνολα δεδομένων, όπως παρατηρήθηκε λίγο αργότερα, ο ψαλιδισμός εκπαιδεύσιμων παραμέτρων περιορίζει την ικανότητα μάθησης του Discriminator καθώς τον εμποδίζει να «εντοπίσει» καλά τοπικά ελάχιστα της συνάρτησης κόστους. Κατ΄ επέκταση περιορίζονται και οι αναδράσεις προς τον Generator άρα γενικότερα η χρήση Ψαλιδισμού Βαρών στη συνάρτηση κόστους Wasserstein μειώνει την απόδοση του GAN. Η παρατήρηση αυτή οδήγησε στην υιοθέτηση μιας πιο ομαλής μεθόδου επιβολής της συνθήκης 3.22, της Ποινής Παραγώγων που αναλύεται ακολούθως.

Ενθάρρυνση συνθήκης συνέχειας κατά Lipschitz με Ποινή Παραγώγων

Λίγο αργότερα από την αρχική πρόταση εκπαίδευσης GANs με τη συνάρτηση κόστους Wasserstein, ο Gulrajani et al. παρουσιάσαν στο άρθρο τους «Improved Training of Wasserstein GANs» [71] μια βελτίωση της μεθόδου Ψαλιδισμού Βαρών προκειμένου να επιβληθεί η συνθήκη της 3.22 στον Discriminator. Η μέθοδός τους, που ονομάζεται Ποινή Παραγώγων (Gradient Penalty), είναι αρκετά πιο απλή και ομαλή καθώς έγκειται μονάχα στην προσθήκη ενός όρου κανονικοποίησης (regularizing term) στη συνάρτηση κόστους του Discriminator. Πιο συγκεκριμένα, αυτό που κάνει ο όρος κανονικοποίησης είναι να αναθέτει μία ποινή όταν η νόρμα των μερικών παραγώγων της εξόδου του Discriminator ως προς την είσοδό του είναι μεγαλύτερη από 1.

Ωστόσο, σύμφωνα με την 3.22 θα πρέπει να ελέγξουμε τις παραγώγους αυτές για κάθε είσοδο του Discriminator, κάτι πρακτικά αδύνατο. Αντί αυτού, οι συγγραφείς του [71] πρότειναν να παρθούν κάποια δείγματα εικόνων ως γραμμικές παρεμβολές μεταξύ πραγματικών και τεχνητών εικόνων και με βάση αυτά να υπολογιστούν οι παράγωγοι και η ποινή. Για παράδειγμα, αντί η νόρμα των παραγώγων να υπολογιστεί ξεχωριστά για

μια εικόνα από το σύνολο εκπαίδευσης, x, και μία από την έξοδο του Generator, $\hat{x}$, θα υπολογιστεί σε μία μίξη των δύο, $\tilde{x} = \epsilon * x + (1 - \epsilon) * \hat{x}$. Επομένως, ο όρος κανονικοποίησης για την ανάθεση ποινής σε παραγώγους νόρμας μεγαλύτερης της μονάδας, θα είναι:

$$reg_{GP} = \left(\left\|\nabla_{\vec{\partial_C}} C(\tilde{x}; \tilde{x})\right\|_2 - 1\right)^2 \tag{3.23}$$

και άρα οι συναρτήσεις κόστους τις οποίες προσπαθούν να ελαχιστοποιήσουν τα δύο νευρωνικά δίκτυα ενός GAN το οποίο εκπαιδεύεται με συνάρτηση κόστους Wasserstein και Gradient Penalty (WGAN+GP), θα είναι:

$$J_C\left(\vec{\partial_C}, \vec{\partial_G}\right) = -\frac{1}{m}\sum_{i=1}^{m}\left[C\left(x^{(i)}; \vec{\partial_C}\right)\right] + \frac{1}{m}\sum_{i=1}^{m}\left[C\left(G\left(\vec{z}^{(i)}; \vec{\partial_G}\right); \vec{\partial_C}\right)\right] + \lambda_{GP} * reg_{GP} \tag{3.24}$$

$$\begin{aligned} &= -\frac{1}{m}\sum_{i=1}^{m}\left[C\left(x^{(i)}; \vec{\partial_C}\right)\right] + \frac{1}{m}\sum_{i=1}^{m}\left[C\left(G\left(\vec{z}^{(i)}; \vec{\partial_G}\right); \vec{\partial_C}\right)\right] \\ &\quad + \lambda_{GP} * \frac{1}{m}\sum_{i=1}^{m}\left[\left(\left\|\nabla_{\vec{\partial_C}} C\left(\epsilon * x + (1 - \epsilon) * \hat{x}; \vec{\partial_C}\right)\right\|_2 - 1\right)^2\right] \end{aligned} \tag{3.25}$$

$$\approx -\mathbb{E}_{x \sim p_{data}}[C(x)] + \mathbb{E}_{z \sim p_{prior}}[C(G(z))] + \lambda_{GP} * \mathbb{E}_{\tilde{x} \sim p_{\tilde{x}}}\left[(\|\nabla_{\tilde{x}} C(\tilde{x})\|_2 - 1)^2\right] \tag{3.26}$$

για τον Discriminator, με λ_{GP} να είναι συντελεστής βαρύτητας του κανονικοποιητή, ο οποίος στο [71] είχε τεθεί στη σταθερή τιμή $\lambda_{GP} = 10.0$. Για τον Generator δεν υπάρχει κάποια αλλαγή και άρα και εδώ θα είναι:

$$J_G\left(\vec{\partial_G}, \vec{\partial_c}\right) = -\frac{1}{m}\sum_{i=1}^{m}\left[C\left(G\left(\vec{z}^{(i)}; \vec{\partial_G}\right); \vec{\partial_C}\right)\right] \tag{3.27}$$

$$\approx -\mathbb{E}_{z \sim p_{prior}}[C(G(z))] \tag{3.28}$$

Όπως φαίνεται, η μέθοδος αυτή δεν επιβάλλει τη συνθήκη συνέχειας κατά 1-Lipschitz, απλώς ενθαρρύνει τις παραγώγους να μην απομακρύνονται από τη μονάδα, ή σωστότερα να μην μεταβάλλονται πολύ σε σχέση με τη μεταβολή της εικόνας στην είσοδο. Ωστόσο, αυτή η μέθοδος έχει αποδειχτεί στο παραπάνω άρθρο ότι δουλεύει καλά και κυρίως πολύ καλύτερα από τη μέθοδο Ψαλιδισμού Βαρών. Βασικό μειονέκτημα της μεθόδους είναι η καθυστέρηση που εισάγει ο πρόσθετος υπολογισμός παραγώγων και παρεμβολών. Τέλος, και για λόγους πληρότητας, παρουσιάζεται στην επόμενη παράγραφο μια ακόμη μέθοδος για επιβολή ποινής σε απότομες μεταβολές των παραμέτρων του Discriminator, η οποία επίσης έχει δοκιμαστεί κατά την εκπαίδευση των μοντέλων GANs της παρούσας εργασίας.

Ενθάρρυνση συνθήκης συνέχειας κατά Lipschitz με Κανονικοποίηση Φάσματος

Η κανονικοποίηση φάσματος ενός πίνακα βαρών είναι κάτι που είχε αρχικά χρησιμοποιηθεί από τον Miyato et al. το 2017 για αύξηση της ικανότητας γενίκευσης βαθιών νευρωνικών δικτύων [85]. Προχωρώντας ένα βήμα περαιτέρω, οι συγγραφείς εφάρμοσαν ένα χρόνο αργότερα Κανονικοποίηση Φάσματος στα βάρη συγκεκριμένων συνλεκτικών στρώσεων του Discriminator ενός GAN, προκειμένου να ενθαρρύνουν κατά αυτόν τον τρόπο τον Discriminator να ικανοποιεί τη συνθήκη συνέχειας κατά Lipschitz κάποιας σταθεράς $K \geq 1$ και έτσι να σταθεροποιήσουν την εκπαίδευσή του [96]. Οι συγγραφείς μελέτησαν τη συμπεριφορά του μοντέλου που ανέπτυξαν, Spectral-Normalized GAN (SN-GAN), όχι μόνο κατά την εκπαίδευση με συνάρτηση κόστους Wasserstein αλλά και με συνάρτηση κόστους Ελαχίστων Τετραγώνων και διαπίστωσαν πως η Κανονικοποίηση Φάσματος στον Discriminator οδηγεί σε πιο σταθερή εκπαίδευση με καλύτερα αποτελέσματα σε αμφότερες τις περιπτώσεις - κάτι που *μπορούμε να επιβεβαιώσουμε και εμείς* στα μοντέλα που εκπαιδεύσαμε και στα οποία έγινε χρήση της Κανονικοποίησης Φάσματος.

Από μαθηματικής σκοπιάς, το φάσμα ενός πίνακα, W, $\sigma(W)$, είναι το σύνολο όλων των ιδιοτιμών του (εάν πρόκειται για τετραγωνικό πίνακα) ή των ιδιάζουσων τιμών[6] του (για μη-τετραγωνικό πίνακα). Επιπρόσθετα, η φασματική (ή τελεστική) νόρμα του πίνακα W, $\|W\|_2$, είναι η τετραγωνική ρίζα της μέγιστης ιδιάζουσας τιμής αυτού, δηλαδή για τη φασματική νόρμα θα ισχύει [117]:

$$\|W\|_2 = max\left\{ \sqrt{\lambda} : \ \lambda \in \sigma\left(W^T W\right)\right\} \tag{3.29}$$

Στα νευρωνικά δίκτυα, ο πίνακας W αντιπροσωπεύει έναν πίνακα βαρών κάποιας στρώσης σε ένα ΤΝΔ. Για την εφαρμογή της Φασματικής Κανονικοποίησης πρέπει κάθε τιμή του πίνακα να διαιρεθεί με τη φασματική του νόρμα. Ως αποτέλεσμα, ο φασματικά κανονικοποιημένος πίνακας, $\overline{W_{SN}}$, μπορεί να εκφραστεί ως:

$$\overline{W_{SN}} = \frac{W}{\|W\|_2} \tag{3.30}$$

[6]Οι ιδιάζουσες τιμές ενός πίνακα προκύπτουν από την Παραγοντοποίηση Ιδιάζουσων Τιμών (Singular Value Decomposition - SVD). Η SVD είναι μια γενίκευση της Παραγοντοποίησης Ιδιοτιμών, δηλαδή της γραφής ενός πίνακα ως γινόμενο πινάκων που περιέχουν τα ιδιοδιανύσματα και τις ιδιοτιμές του (κανονική μορφή). Η γενίκευση έγκειται στο ότι η SVD πρώτα «τετραγωνοποιεί» έναν πίνακα με πολλαπλασιασμό με τον ανάστροφό του και κατόπιν βρίσκει τα ιδιοδιανύσματα και τις ιδιοτιμές του γινομένου - αυτά λέγονται ιδιάζοντα διανύσματα και ιδιάζουσες τιμές αντίστοιχα. Έτσι η SVD για έναν πίνακα W, θα είναι: $W = U\Sigma V^T$, όπου U, V ορθοκανονικοί πίνακες και Σ διαγώνιος πίνακας με τις ιδιάζουσες τιμές.

Στην πράξη, ο υπολογισμός της SVD του πίνακα W, προκειμένου να υπολογιστεί η φασματική του νόρμα, είναι αρκετά υπολογιστικά απαιτητική διαδικασία, και γι΄ αυτό οι συγγραφείς του SN-GAN [96] δοκίμασαν μια προσεγγιστική λύση: προσέγγισαν τη μέγιστη ιδιάζουσα τιμή και τα αριστερά και δεξιά ιδιάζοντα διανύσματα της SVD, $\tilde{u}$ και $\tilde{v}$ αντίστοιχα, με μία παραλλαγή της επαναληπτικής μεθόδου δύναμης[7]. Πιο συγκεκριμένα, πρότειναν την επαναληπτική προσέγγιση της φασματική νόρμας ενός πίνακα βαρών, W, ως εξής:

$$\tilde{u} := \frac{W^T \tilde{u}}{\|W^T \tilde{u}\|_2} \tag{3.31}$$

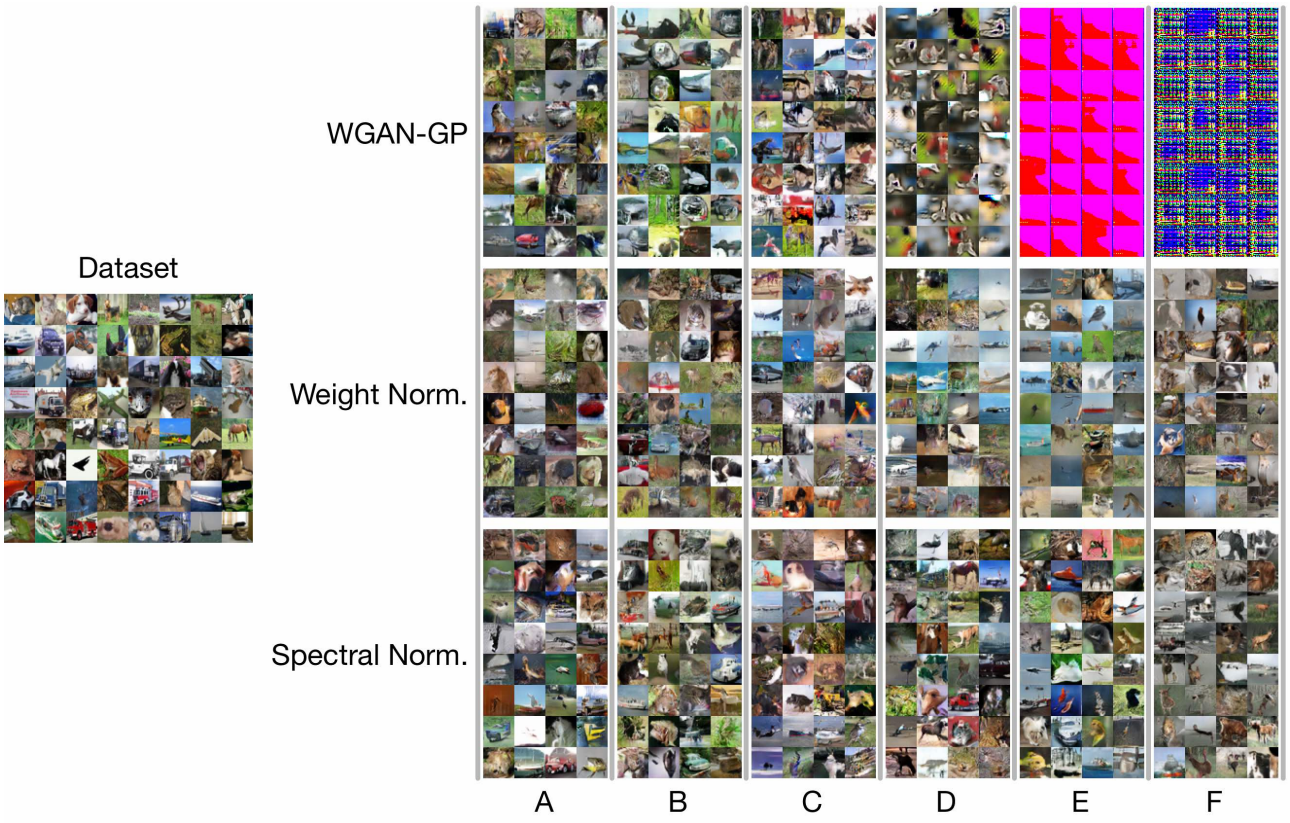
$$\tilde{v} := \frac{W^T \tilde{v}}{\|W^T \tilde{v}\|_2} \tag{3.32}$$

$$\|W\|_2 \approx \tilde{u}^T W \tilde{v} \tag{3.33}$$

όπου $\|\cdot\|_2$ είναι η L2 (Ευκλείδεια) νόρμα ενός διανύσματος, ενώ οι συγγραφείς βρήκαν «*πως μία επανάληψη αρκεί για ικανοποιητική προσέγγιση*» της φασματικής νόρμας.

Αυτό που αποδεικνύεται στην ανάλυση των συγγραφέων στο [96] είναι ότι η εφαρμογή της Φασματικής Κανονικοποίησης στις τελευταίες συνελικτικές στρώσεις του Discriminator ισοδυναμεί με την επιβολή ποινής στα πρώτα ιδιάζουσα στοιχεία (singular components) του πίνακα βαρών με προσαρμοζόμενο συντελεστή, ο οποίος αποτρέπει το στηλοχώρο του πίνακα W από το να επικεντρώνεται σε μία συγκεκριμένη κατεύθυνση μεταβολής (μέσω των αλγορίθμων βελτιστοποίησης που στα GANs συνήθως «κοιτούν» μόνο την πρώτη παράγωγο) κατά τη διάρκεια της εκπαίδευσης. Αυτό έχει οδηγήσει σε πιο σταθερή και εύρωστη εκπαίδευση ειδικά των πιο πολύπλοκων μοντέλων, ενώ στις περισσότερες περιπτώσεις οδήγησε και σε καλύτερα παραγόμενα αποτελέσματα. Όπως γίνεται αντιληπτό από τις παραγόμενες εικόνες στο σχήμα 27, η Φασματική Κανονικοποίηση υπερέχει σε ότι αφορά την ευστάθεια της εκπαίδευσης για ποικίλες ρυθμίσεις και παραλλαγές των μοντέλων. Τέλος, για λόγους πληρότητας είναι σκόπιμο να αναφερθεί ότι παρόλο που κάποιες συναρτήσεις κόστους ή κάποιες τεχνικές, όπως αυτή, έχουν καλύτερη θεωρητική υποστήριξη και ιδιότητες από κάποιες άλλες, στην πράξη έχει αποδειχτεί ότι διαφορετικές συναρτήσεις κόστους (και κανονικοποιήσεις αυτών) αποδίδουν καλύτερα

[7]Στην αριθμητική ανάλυση, η μέθοδος δύναμης (power method) χρησιμοποιείται για την εύρεση της μεγαλύτερης ιδιοτιμής ενός πίνακα, έστω A. Η μέθοδος δύναμης μπορεί να αναπαρασταθεί από τις σχέσεις [26]: $\vec{v_p} = A\vec{u_p}$ και $\vec{u_{p+1}} = \vec{v_p}/max(\vec{v_p})$ για $p = 0, 1, 2, ..., A = W^T W$, $\vec{v_0}$ αρχική προσέγγιση του ιδιοδιανύσματος του A και $\vec{u_0} \sim N(\vec{0}, \mathbb{I})$. Η προσέγγιση της μέγιστης ιδιοτιμής είναι το μέγιστο στοιχείο του $\vec{v}$ και η επαναληπτική διαδικασία τερματίζει όταν η διαφορά μεταξύ διαδοχικών μέγιστων τιμών είναι μικρή.



Σχήμα 27: Σύγκριση των παραγόμενων εικόνων έξι (6) παραλλαγών (κυρίως ως προς τις παραμέτρους του αλγόριθμου βελτιστοποίησης) ενός μοντέλου GAN για διαφορετικές τεχνικές κανονικοποίησης: στην πρώτη γραμμή χρησιμοποιείται συνάρτηση κόστους Wasserstein και κανονικοποίησης Ποινής Παραγώγων, στη δεύτερη χρησιμοποιείται συνάρτηση κόστους Ελαχίστων Τετραγώνων με κανονικοποίηση Απόσβεσης Βαρών (Weight Decay) και στην τρίτη χρησιμοποιείται επίσης συνάρτηση κόστους Ελαχίστων Τετραγώνων με Φασματική Κανονικοποίηση στις τελευταίες στρώσεις του Discriminator (η συνάρτηση κόστους δεν αλλάζει).
Πηγή: «Spectral Normalization for Generative Adversarial Networks», Miyato et al., 2018 [96]

ή χειρότερα μεταξύ διαφορετικών μοντέλων, αλλά ακόμη και στο ίδιο μοντέλο μεταξύ διαφορετικών συνόλων δεδομένων εκπαίδευσης.

3.2 Υπο-συνθήκη Παραγωγή και Ελεγξιμότητα

Τα μοντέλα τα οποία συζητήθηκαν έως τώρα ήταν στην πλειονότητά τους μοντέλα GANs τα οποία προσπαθούσαν απλώς να μιμηθούν εικόνες από το σύνολο εκπαίδευσης, δηλαδή εκπαιδεύονταν χωρίς επίβλεψη (unsupervised training). Στις δύο υποενότητες που ακολουθούν, θα παρουσιάσουμε μοντέλα τα οποία εκπαιδεύονται με επίβλεψη (supervised training) προκειμένου να μάθουν δεσμευμένες πιθανοτικές κατανομές, με τη συνθήκη να είναι άλλοτε η τάξη που θέλουμε να ανήκει μια παραγόμενη εικόνα και άλλοτε συγκεκριμένα (οπτικά) χαρακτηριστικά που θέλουμε αυτή να έχει. Σκοπός αυτών των υποενοτήτων είναι να φέρουν τον αναγνώστη ένα βήμα πιο κοντά στα GANs του πραγματικού κόσμου, όπου στις περισσότερες περιπτώσεις δεν αρκεί η δυνατότητα παραγωγής ρεαλιστικών εικόνων που μιμούνται ένα σύνολο εκπαίδευσης, αλλά απαιτείται αυτό να γίνεται ελεγχόμενα.

Στην παραγωγή χωρίς συνθήκη, το GAN εκπαιδεύεται λαμβάνοντας τυχαίο θόρυβο στην είσοδο του Generator και μια ομάδα είτε τεχνητών ή πραγματικών εικόνων στην είσοδο του Discriminator. Έτσι, εάν επιθυμούμε την παραγωγή ρεαλιστικών εικόνων από μια συγκεκριμένη τάξη ή με συγκεκριμένα χαρακτηριστικά, πρέπει να δοκιμάζουμε (evaluate) τον Generator με διαφορετικές τυχαίες εισόδους έως ότου (από τύχη) λάβουμε μια εικόνα στην έξοδό του με τα επιθυμητά χαρακτηριστικά. Προς τον σκοπό αυτό, αφιερώνουμε αυτήν την ενότητα, όπου θα αναλύσουμε τεχνικές τόσο για παραγωγή εικόνων από συγκεκριμένη τάξη όσο και για την ελεγξιμότητα στην παραγωγή εικόνων με στόχο την ύπαρξη συγκεκριμένων χαρακτηριστικών.

Υπο-συνθήκη Παραγωγή Εικόνων

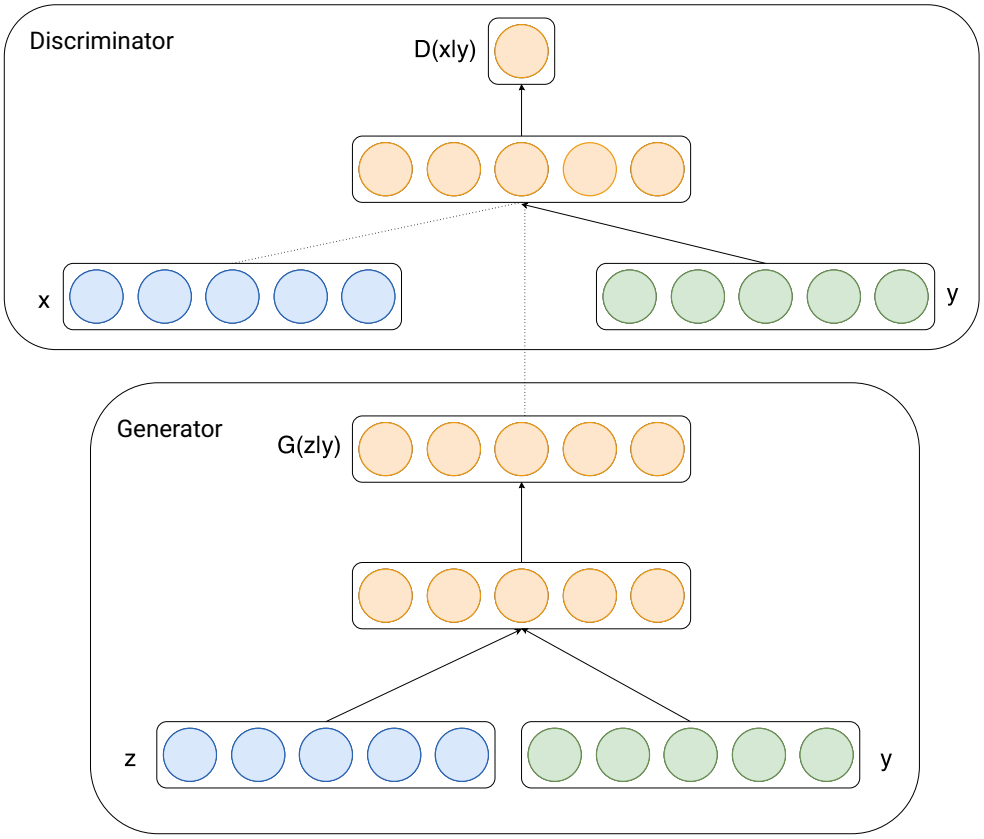
Ξεκινώντας με την Υπο-συνθήκη Παραγωγή (Conditional Generation), τόσο κατά τη φάση εκπαίδευσης όσο και κατά τη φάση της δοκιμής ο Generator λαμβάνει εκτός από το διάνυσμα τυχαίου θορύβου και πληροφορία για την τάξη της εικόνας που καλείται να παράγει. Αντίστοιχα, ο Discriminator εκτός από μία ομάδα εικόνων και τις ετικέτες με την τάξη που αντιστοιχούν στη κάθε μία. Επομένως, σε αυτήν την περίπτωση απαιτείται *η ύπαρξη επισημασμένου (annotated) συνόλου δεδομένων* προκειμένου να γίνει η εκπαίδευση του GAN.

Conditional GAN (CGAN)

Η υπο-συνθήκη παραγωγή στα πλαίσια των GANs, ξεκινώντας από τον Mirza et al. και το μοντέλο Conditional GAN (CGAN) [32], επέτρεψε την παραγωγή δειγμάτων από οποιαδήποτε τάξη ή ετικέτα των δεδομένων - εφόσον αυτή έχει «εκπροσωπηθεί» ικανοποιητικά στο σύνολο δεδομένων εκπαίδευσης. Ο τρόπος που προτάθηκε στο [32] προκειμένου να δίνεται η πληροφορία της τάξης/ετικέτας τόσο στον Generator όσο και στον Discriminator είναι χρησιμοποιώντας τα λεγόμενα διανύσματα one-hot.

Έτσι, στην περίπτωση του Generator, μαζί με το τυχαίο διάνυσμα στην είσοδό του δίνεται (συνενωμένο) και ένα διάνυσμα μήκους ίσου με τον αριθμό των διαφορετικών τάξεων/ετικετών του συνόλου εκπαίδευσης με όλα τα στοιχεία μηδενικά πλην ενός, αυτού που αντιστοιχεί στην τάξη που επιθυμούμε να ανήκει η παραγόμενη εικόνα, που είναι μονάδα. Ο λόγος που παραμένει και το διάνυσμα τυχαίου (γκαουσιανού συνήθως) θορύβου στην είσοδο, είναι διότι κατ΄ αυτόν τον τρόπο υπάρχει κάποια τυχαιότητα και άρα κάποια ποικιλομορφία στα δείγματα που παράγονται από τη συγκεκριμένη τάξη. Αυτό όμως που εξασφαλίζει ότι πράγματι ο Generator εκπαιδεύεται για να παράγει δείγματα από τη συγκεκριμένη τάξη που περιγράφεται στο one-hot διάνυσμα στην είσοδό του, είναι το ότι αντίστοιχη πληροφορία για την τάξη δίνεται και στον Discriminator, όπως απεικονίζεται στο σχήμα 28 παρακάτω. Στα πλαίσια της Παραγωγικής Μοντελοποίησης εικόνων, η συνθήκη της τάξης δίνεται στον Discriminator ως ένας one-hot πίνακας, δηλαδή ένας τρισδιάστατος πίνακας ίδιου πλάτους και μήκους με τις πραγματικές εικόνες αλλά με βάθος (ή αριθμό καναλιών) όσο και οι διαφορετικές τάξεις/ετικέτες του συνόλου εκπαίδευσης. Όλα αυτά τα κανάλια περιέχουν μηδενικές τιμές πλην ενός, αυτού που αντιστοιχεί στην ετικέτα της εικόνας εισόδου το οποίο έχει μονάδες. Έτσι, όπως απεικονίζεται στο σχήμα, στον Generator δίνεται το τυχαίο διάνυσμα, $\vec{z}$ καθώς και το διάνυσμα one-hot της τάξης. Αντίστοιχα στον Discriminator τα κανάλια (3 για έγχρωμες εικόνες, 1 για ασπρόμαυρες) της εικόνας εισόδους συνενώνονται με τα one-hot κανάλια της τάξης. Αυτό οδηγεί τον Generator να μάθει να παράγει εικόνες που ανήκουν στην επιθυμητή τάξη των δεδομένων εκπαίδευσης.

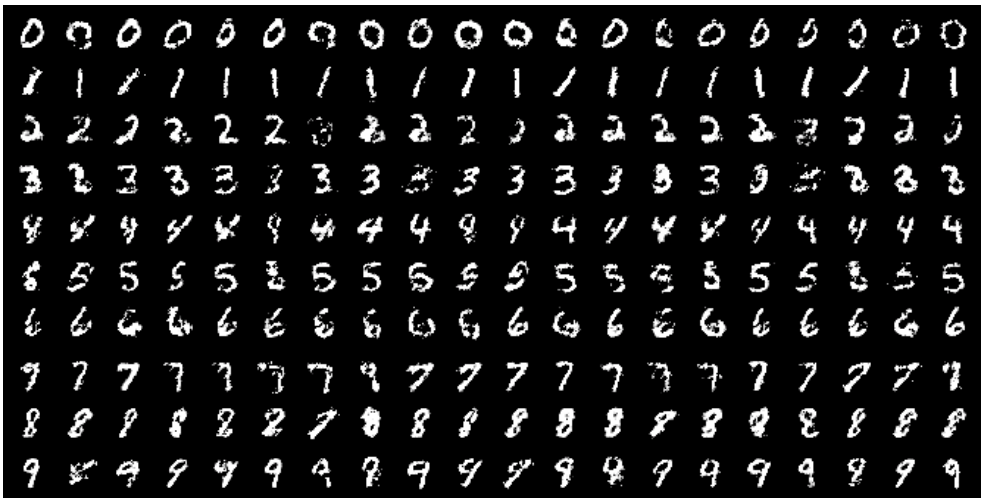
Τέλος, στο σχήμα 29, που ακολουθεί φαίνονται παραγωγές του μοντέλου CGAN για κάθε μία από τις 10 διαφορετικές ετικέτες του συνόλου εκπαίδευσης χειρόγραφων ψηφίων του MNIST. Στις παραχθείσες εικόνες της κάθε γραμμής του σχήματος, ο Generator λάμβανε στην είσοδό του το ίδιο διάνυσμα one-hot συνενωμένο με διάνυσμα γκαουσιανού τυχαίου θορύβου. Έτσι οι εικόνες της κάθε γραμμής έχουν παραχθεί υπο-συνθήκη την



Σχήμα 28: Τρόπος εμβύθισης της πληροφορίας της τάξης ή της ετικέτας στον Generator και Discriminator του Conditional GAN.

Πηγή: Ανακατασκευή από «Conditional Generative Adversarial Nets», Mirza et al., 2014 [32]

εμβύθιση της ετικέτας του αντίστοιχου ψηφίου.



Σχήμα 29: Παραγωγές του μοντέλου Conditional GAN (CGAN) το οποίο έχει εκπαιδευτεί στο σύνολο δεδομένων χειρόγραφων ψηφίων του MNIST.

Πηγή: «Conditional Generative Adversarial Nets», Mirza et al., 2014 [32]

Ελέγξιμη Παραγωγή

Μία εναλλακτική μέθοδος ελέγχου των παραγόμενων δειγμάτων ενός GAN είναι αυτός να γίνει λιγότερο κατά τη διάρκεια και κυρίαρχα μετά το πέρας της εκπαίδευσης των Παραγωγικών Μοντέλων, κάτι που γενικά ονομάζεται Ελέγξιμη Παραγωγή (Controllable Generation). Η παραγωγή υπο-συνθήκη κάνει χρήση των ετικετών του συνόλου δεδομένων εκπαίδευσης προκειμένου τα παραχθέντα δείγματα να ανήκουν σε κάποια επιθυμητή τάξη. Η ελέγξιμη παραγωγή από την άλλη, εστιάζει στον έλεγχο των χαρακτηριστικών που είναι επιθυμητό να βρίσκονται στα παραχθέντα δείγματα, κάτι που γίνεται ακόμη και μετά το πέρας της εκπαίδευσης ενός GAN. Σε ένα ήδη εκπαιδευμένο GAN, για παράδειγμα, η ελέγξιμη παραγωγή ισοδυναμεί με την εύρεση εκείνης της εισόδου από τον λανθάνοντα χώρο του Generator - ο οποίος ονομάζεται και χώρος-z (z-space) - που οδηγεί στην παραγωγή εικόνων με συγκεκριμένα χαρακτηριστικά.

Η ελέγξιμη παραγωγή στα GANs εστιάζεται στην τροποποίηση συγκεκριμένων οπτικών χαρακτηριστικών της εξόδου ενός Generator. Στο σχήμα 30 παρακάτω, για παράδειγμα, οι συγγραφείς του άρθρου «Interpreting the Latent Space of GANs for Semantic Face Editing» [108], αναζήτησαν μέσω ενός πρόσθετου νευρωνικού δικτύου τις εισόδους εκείνες ενός προ-εκπαιδευμένου Generator που αλλάζουν σε μία αρχική παραγωγή ορισμένα επιθυμητά οπτικά χαρακτηριστικά, όπως η ηλικία, το φύλο, η πόζα κ.α. Σημαντικό σημείο υπενθύμισης είναι ότι η εκπαίδευση του μοντέλου που χρησιμοποιήθηκε (PGGAN - βλ. υποενότητα 4.1.2 του επόμενου κεφαλαίου) είχε ολοκληρωθεί και ύστερα γίνονταν αυτή η «αναζήτηση» στον z-χώρο της εισόδου του Generator. Συγκεκριμένα, οι συγγραφείς άλλαζαν τους λανθάνοντες «κωδικούς», όπως τους ονόμασαν, στην είσοδο ενός καλά εκπαιδευμένου μοντέλου GAN. Η πρώτη στήλη του παρακάτω σχήματος φαίνεται η αρχική σύνθεση από το μοντέλο PGGAN, ενώ κάθε μία από τις άλλες στήλες φαίνονται τα αποτελέσματα του ελέγχου ενός συγκεκριμένου χαρακτηριστικού.

Επομένως, σε αντίθεση με την Υπο-συνθήκη Παραγωγή, η Ελέγξιμη δεν απαιτεί επισημασμένα σύνολα δεδομένων εκπαίδευσης. Έτσι, ενώ στην πρώτη μπορούμε απλώς να παράγουμε από ένα εκπαιδευμένο GAN εικόνες από μία τάξη, στη δεύτερη μπορούμε να παράγουμε που έχουν περισσότερο ή λιγότερο από κάποια επιθυμητά χαρακτηριστικά. Μία πρόσθετη διαφοροποίηση, είναι ότι για την υπο-συνθήκη παραγωγή απαιτείται υπο-συνθήκη εκπαίδευση, δηλαδή συνένωση της εισόδου των δικτύων ενός GAN με διανύσματα ή πίνακες που αντιπροσωπεύουν τη ζητούμενη τάξη, ενώ η ελεγξιμότητα της παραγωγής έγκειται απλώς στον χειρισμό των διανυσμάτων στον z-χώρο.

Σχήμα 30: Τροποποίηση ποικίλων χαρακτηριστικών του προσώπου μέσω της μεταβολής των στοιχείων της εισόδου του Generator.
Πηγή: «Interpreting the Latent Space of GANs for Semantic Face Editing», Shen et al., 2019 [108]

Όσο απλή και να φαίνεται ως ιδέα και ως τεχνική, η Ελεγξιμότητα στην παραγωγή δεν είναι εύκολη ούτε δεδομένη. Αυτό, σύμφωνα με τους συγγραφείς των [48] και [108], οφείλεται σε δύο βασικούς λόγους: στη συσχέτιση (correlation) μεταξύ των οπτικών χαρακτηριστικών στον χώρο των εικόνων εξόδου και στο ότι ο z-χώρος εισόδου είναι μπερδεμένος entangled. Επεξηγώντας, όταν διαφορετικά χαρακτηριστικά στον χώρο εξόδου έχουν υψηλή συσχέτιση μεταξύ τους, γίνεται αρκετά δύσκολο να ελέγξει κανείς το ένα χωρίς να επηρεάσει το άλλο (όπως π.χ. στην περίπτωση του φύλου και της ύπαρξης γενειάδας).

Επιπρόσθετα, πολλές φορές ο z-χώρος που μαθαίνει ένας noise-to-image Generator[8] είναι μπερδεμένος entangled, με την έννοια ότι διαφορετικές κατευθύνσεις (άξονες) στον χώρο αυτό δεν αντιστοιχούν σε διαφορετικά χαρακτηριστικά στον χώρο εξόδου. Πρακτικά, αλλάζοντας τα ένα στοιχείο του τυχαίου διανύσματος ενός μπερδεμένου z-χώρου, ενώ θα θέλαμε στην έξοδο του εκπαιδευμένου Generator να αλλάζει ένα χαρακτηριστικό, αλλάζουν περισσότερα από ένα και μάλιστα όχι μόνον αυτά που έχουν συσχέτιση

[8]Στα αρχικά μοντέλα GANs, αλλά και σε κάποια από τα πιο σύγχρονα και εξελιγμένα, ο Generator λαμβάνει ένα τυχαίο διάνυσμα στην είσοδό του και καλείται να παράξει μια ρεαλιστική εικόνα. Αυτού του τύπου οι Generators αλλά και γενικότερα η Παραγωγική Μοντελοποίηση ονομάζεται noise-to-image, σε αντιδιαστολή με άλλους τύπους όπως π.χ. η παραγωγή ή μετασχηματισμός εικόνας-σε-εικόνα (image-to-image transform) στην οποία η είσοδος ή συνθήκη στον Generator είναι μια εικόνα στην οποία καλείται να κάνει ένα σύνολο ρεαλιστικών μετασχηματισμών.

μεταξύ τους. Όταν συμβαίνει αυτό η υψηλής-ποιότητας (βλ. σχήμα 30) ελέγξιμη παραγωγή καθίσταται αδύνατη, ενώ για να μη συμβεί αυτό έχουν προταθεί διάφορες τεχνικές, από αύξηση της διάστασης του διανύσματος και προσθήκη κανονικοποιητών στη συνάρτηση κόστους, έως και τη προσθήκη ολόκληρων ΤΝΔ (όπως στο StyleGAN - βλ. υποενότητα 4.1.3 του επόμενου κεφαλαίου) για μετασχηματισμό του αρχικού μπερδεμένου z-χώρου σε ένα μη-μπερδεμένο (disentangled) ο οποίος χρησιμοποιείται ως χώρος εισόδου του Generator.

Σχετική τεχνική με την ελεγξιμότητα στην παραγωγή από έναν εκπαιδευμένο Generator είναι και αυτή της παρεμβολής (interpolation) μεταξύ παραγωγών. Η παρεμβολή χρησιμοποιείται για την παραγωγή ενδιάμεσων δειγμάτων μεταξύ δύο παραγωγών. Πιο συγκεκριμένα, εφόσον έχουμε το διάνυσμα στον λανθάνοντα χώρο για κάθε μία από τις δύο παραγωγές, έστω $\vec{z_1}$ και $\vec{z_2}$, μπορούμε εφαρμόζοντας γραμμική παρεμβολή να πάρουμε ένα ή περισσότερα σημεία του διανύσματος $\vec{z_1} - \vec{z_2}$, να τροφοδοτήσουμε τον Generator με αυτά για να πάρουμε ενδιάμεσες παραγωγές, στις οποίες (τουλάχιστον θεωρητικά) θα υπάρχει ομαλή μετάβαση των χαρακτηριστικών από την αρχική έως την τελική εικόνα. Η διαφορά μεταξύ των δύο διανυσμάτων είναι η κατεύθυνση στο z-χώρο που πρέπει να κινηθούμε για να ελέγξουμε την παραγωγή, κάτι που απεικονίζεται για τον Generator του μοντέλου InfoGAN [48] στο σχήμα 31 παρακάτω. Επέκταση αυτής της ιδέας, είναι η χρήση ενός προ-εκπαιδευμένου ταξινομητή αναγνώρισης των επιθυμητών χαρακτηριστικών για εύρεση της κατεύθυνσης αναζήτησης: κρατώντας «παγωμένα» τόσο τον Generator όσο και τον εξωτερικό ταξινομητή, μεταβάλλουμε το διάνυσμα εισόδου, $\vec{z}$, στην κατεύθυνση των παραγώγων (gradient ascent) της πιθανότητας ύπαρξης ενός χαρακτηριστικού ως προς την είσοδο του Generator (αυτή η τεχνική χρησιμοποιήθηκε από του συγγραφείς του [108] στο σχήμα 30).

(α) Περιστροφή (β) Αλλαγή πλάτους

Σχήμα 31: Παρεμβολές μεταξύ αρχικών (αριστερά) και τελικών (δεξιά) εικόνων ενός συνόλου δεδομένων 3D καρεκλών. Παρατηρούμε, ότι επειδή το GAN έχει εκπαιδευτεί να ξεμπερδεύει (disentangle) τον λανθάνοντα χώρο του, οι παρεμβολές αλλάζουν ένα χαρακτηριστικό της εικόνας εξόδου (εδώ είναι η περιστροφή στις αριστερά εικόνες και το πλάτος στις δεξιά).
Πηγή: «InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets», Chen et al., 2016 [48]

3.3 Προκλήσεις για Ευσταθή Εκπαίδευση

Η παρούσα ενότητα αφιερώνεται σε μία σύντομή παράθεση δυσκολιών και προκλήσεων για επιτυχημένη εκπαίδευση GANs, με έμφαση να δίνεται στο πως αυτή σταθεροποιείται για τη μεγάλη χρονική διάρκεια που συνήθως απαιτείται για καλά αποτελέσματα. Η πιο σημαντική πρόκληση που καλείται να αντιμετωπίσει ο σχεδιαστής ενός GAN για Παραγωγική Μοντελοποίηση εικόνων είναι η αξιολόγηση των παραχθέντων δειγμάτων με τρόπο συστηματικό και ερμηνεύσιμο. Ωστόσο, αυτό είναι τόσο κομβικό που αφιερώνουμε την επόμενη ενότητα σε τεχνικές αξιολόγησης εικόνων που παράγονται από GANs. Σε αυτήν την ενότητα, θα αναλύσουμε τρεις βασικές προκλήσεις και προτάσεις για την αντιμετώπισή τους: το πρόβλημα της Συρρίκνωσης Ρυθμών ή το αντίθετο της, την υπεργενίκευση ή υπερ-ρεαλισμό, την ύπαρξη πόλωσης καθώς και τρίτον, την (πολύ) αργή σύγκλιση και εκπαίδευση μοντέλων GANs.

Συρρίκνωση Ρυθμών και Υπερ-ρεαλισμός

Τι προκαλεί τη Συρρίκνωση Ρυθμών

Όπως αναφέρθηκε και σε προηγούμενες υποενότητες, η εκπαίδευση του Generator με στόχο να αυξήσει τη συνάρτηση κόστους του Discriminator πολλές φορές τον οδηγεί στην εκμάθηση των χαρακτηριστικών μιας τάξης των δεδομένων, που πιθανόν είναι

πιο εύκολο να απεικονιστούν. Οι σχεδιαστές και χρήστες όμως ενός τέτοιου μοντέλου αναμένουν ότι ένας Generator πού έχει ικανά αιχμαλωτίσει τη δομή της δομής της πιθανοτικής κατανομής των δεδομένων εκπαίδευσης, θα μπορεί να παράγει δείγματα όχι μόνο ρεαλιστικά αλλά και με ποικιλομορφία. Όπως φαίνεται και στο σχήμα 32 παρακάτω, ένα GAN το οποίο υποφέρει από Συρρίκνωση Ρυθμών πρακτικά δεν είναι χρήσιμο, καθώς οι έξοδοι του Generator δεν μπορούν να χρησιμοποιηθούν για καμία πρακτικά χρήσιμη εφαρμογή.

(α) Χωρίς Συρρίκνωση Ρυθμών (β) Με Συρρίκνωση Ρυθμών

Σχήμα 32: Παραγωγές από δύο διαφορετικά μοντέλα GAN, αμφότερα τα οποία έχουν εκπαιδευτεί στο σύνολο δεδομένων χειρόγραφων ψηφίων του MNIST. Στο αριστερά (Unrolled GAN) η εκπαίδευση έχει ολοκληρωθεί εύρωστα και με επιτυχία. Στο δεξιά (DCGAN), υπάρχει έντονο το φαινόμενο Συρρίκνωσης Ρυθμών με αποτέλεσμα ο Generator να έχει φτάσει στο λεγόμενο σημείο «*τερματισμού εκπαίδευσης*» (end of training).
Πηγή: «Unrolled Generative Adversarial Networks», Metz et al., 2016 [57]

Πως όμως προκαλείται η Συρρίκνωση Ρυθμών κατά την εκπαίδευση ενός GAN; Η απάντηση περιγράφεται με ωραίο τρόπο στο [125] και έχει ως εξής:
Εάν ο Generator αρχίσει να παράγει την ίδια έξοδο (ή ένα μικρό σύνολο εξόδων) ξανά και ξανά, η καλύτερη στρατηγική του Discriminator είναι να μάθει να απορρίπτει πάντα αυτήν την έξοδο. Αλλά εάν η επόμενη επανάληψη (update) του Discriminator «κολλήσει» σε τοπικό ελάχιστο της συνάρτησης κόστους και δεν βρει την καλύτερη στρατηγική, τότε είναι πολύ εύκολο για την επόμενη επανάληψη του Generator να βρει την πιο πιθανή έξοδο για τον τρέχοντα Discriminator. Κάθε επανάληψη του Generator υπερ-βελτιστοποιείται (over-fitted) για τον συγκεκριμένο Discriminator και ο Discriminator δεν καταφέρνει ποτέ

να μάθει την έξοδο από αυτό το τοπικό ελάχιστο. Ως αποτέλεσμα, οι παραγωγές του Generator περιστρέφονται μέσα σε ένα μικρό σύνολο διακριτών δειγμάτων, κάτι που οδηγεί σε Συρρίκνωση Ρυθμών και τελικά στον τερματισμό της εκπαίδευσης.

Αντιμετώπιση της Συρρίκνωσης Ρυθμών

Για την αντιμετώπιση αυτού του φαινομένου κατά την εκπαίδευση των GANs έχουν προταθεί διάφορες μέθοδοι, κάποιες από τις οποίες αναφέρονται ακολούθως. Αρχικά, όπως αναφέρθηκε η χρήση συνάρτηση κόστους χωρίς περιοχές κορεσμού, δηλαδή συναρτήσεις κόστους όπως Ελαχίστων Τετραγώνων ή Wasserstein, επιλύουν το πρόβλημα της εξαφάνισης παραγώγων (vanishing gradients) της συνάρτησης κόστους και βοηθούν σημαντικά με το πρόβλημα της Συρρίκνωσης Ρυθμών. Συγκεκριμένα, άρθρα όπως το [116] συνιστούν πως η χρήση συνάρτησης κόστους Ελαχίστων Τετραγώνων σε συνδυασμό με Φασματική Κανονικοποίηση οδηγεί σε εύρωστη εκπαίδευση με ικανοποιητική ποιότητα και ποικιλομορφία στα παραχθέντα δείγματα.

΄Αλλες προσπάθειες, όπως τα Unrolled GANs [57], χρησιμοποιούν μια συνάρτηση κόστους του Generator που ενσωματώνει όχι μόνο τις τρέχουσες εξόδους του Discriminator, αλλά και αυτές των επόμενων εκδόσεων ή επαναλήψεων του Discriminator. Έτσι, ο Generator δεν μπορεί να υπερ-βελτιστοποιηθεί για μία έκδοση του Discriminator με αποτέλεσμα να μειώνεται έως και να εξαλείφεται η Συρρίκνωση Ρυθμών. Τέλος, όπως θα αναφερθεί πιο εκτενώς κατά την παρουσίαση του StyleGAN (βλ. υποενότητα 4.1.3) έχουν προταθεί διάφορες ενδιάμεσες στρώσεις στο δίκτυο του Discriminator οι οποίες υπολογίζουν την τυπική απόκλιση των εικόνων της κάθε ομάδας (batch) και τη συνενώνουν σαν ένα πρόσθετο κανάλι στην έξοδο συνελικτικών στρώσεων. Κατ΄ αυτόν τον τρόπο, ο Discriminator μαθαίνει να συνυπολογίζει και αυτόν τον παράγοντα προκειμένου να ταξινομήσει σωστότερα τις εικόνες, ενώ ο Generator μαθαίνει να παράγει ομάδες εικόνων με ποικιλομορφία, βοηθώντας έτσι και στη Συρρίκνωση Ρυθμών.

Υπερ-ρεαλισμός

Για λόγους πληρότητας, αναφέρουμε στην παρούσα παράγραφο το θέμα του υπερ-ρεαλισμού που πρόσφατα αναδείχθηκε κατά την εκπαίδευση GANs. Πρόκειται για την περίπτωση που ο Generator έχει ικανή χωρητικότητα (capacity) για να αιχμαλωτίσει πλήρως την πιθανοτική κατανομή των δεδομένων εκπαίδευσης και εκτός αυτού να είναι σε θέση να παράγει ρεαλιστικές εικόνες που δεν θα μπορούσαν όμως να ανήκουν στο

σύνολο δεδομένων εκπαίδευσης. Όπως φαίνεται για παράδειγμα στο σχήμα 33 παρακάτω, οι εικόνες πλην της τελευταίας επιδεικνύουν έντονο ρεαλισμό. Ωστόσο, λόγω της πολύ υψηλής χωρητικότητας του Generator, το μοντέλο παρήγαγε και εικόνες που αν και περιέχουν ρεαλιστικά χαρακτηριστικά δεν θα μπορούσαν ποτέ να βρίσκονται στο σύνολο εκπαίδευσης, όπως η εικόνα στα δεξιά. Αυτός ο υπερ-ρεαλισμός συνοδεύει αρκετές φορές μοντέλα GANs τα οποία εκπαιδεύονται επιτυχώς και ταυτόχρονα έχουν ικανά μεγάλη χωρητικότητα.

(α) Ρεαλιστικό Δείγμα (β) Ρεαλιστικό Δείγμα (γ) Ρεαλιστικό Δείγμα (δ) Υπερ-Ρεαλισμός

Σχήμα 33: Υπο-συνθήκη παραγωγές από το μοντέλο BigGAN.
Πηγή: «Large Scale GAN Training for High Fidelity Natural Image Synthesis», Brock et al., 2018 [88]

Όπως αναλύεται στην επόμενη ενότητα, υπάρχουν μετρικές αξιολόγησης των παραγόμενων εικόνων και συγκεκριμένα οι μετρικές Inception Score, Precision και Recall οι οποίες έμμεσα υπολογίζουν την ύπαρξη υπερ-ρεαλισμού κατά την εκπαίδευση των GANs. Ωστόσο, αν και πρόκληση ή μη επιθυμητό χαρακτηριστικό της εκπαίδευσης σύγχρονων και μεγάλων GANs, ο υπερ-ρεαλισμός μάλλον αποτελεί περισσότερο σημάδι ευσταθούς εκπαίδευσης παρά κάτι αρνητικό.

Για την αντιμετώπιση του υπερ-ρεαλισμού κατά τη φάση δοκιμής (evaluation) του Generator, αυτό που πρακτικά γίνεται είναι το τυχαίο διάνυσμα στην είσοδό του να δειγματοληπτείται από περικομμένη κανονική truncated κατανομή. Αυτό, που έχει ονομαστεί truncation trick, οδηγεί στην τροφοδότηση του Generator με διανύσματα που είναι πιο κοντά στη μέση τιμή της πρότερης κανονική κατανομής, κάτι που οδηγεί τον Generator να παράγει όλο και λιγότερα από τα «μη-κοινά» δείγματα στην έξοδό του.

Πόλωση και Μεροληψία στα GANs

Η παρούσα υποενότητα αφιερώνεται σε ένα θέμα που απασχολεί την ευρύτερη ερευνητική κοινότητα της Μηχανικής Μάθησης και φυσικά αυτής της Παραγωγικής Μοντελο-

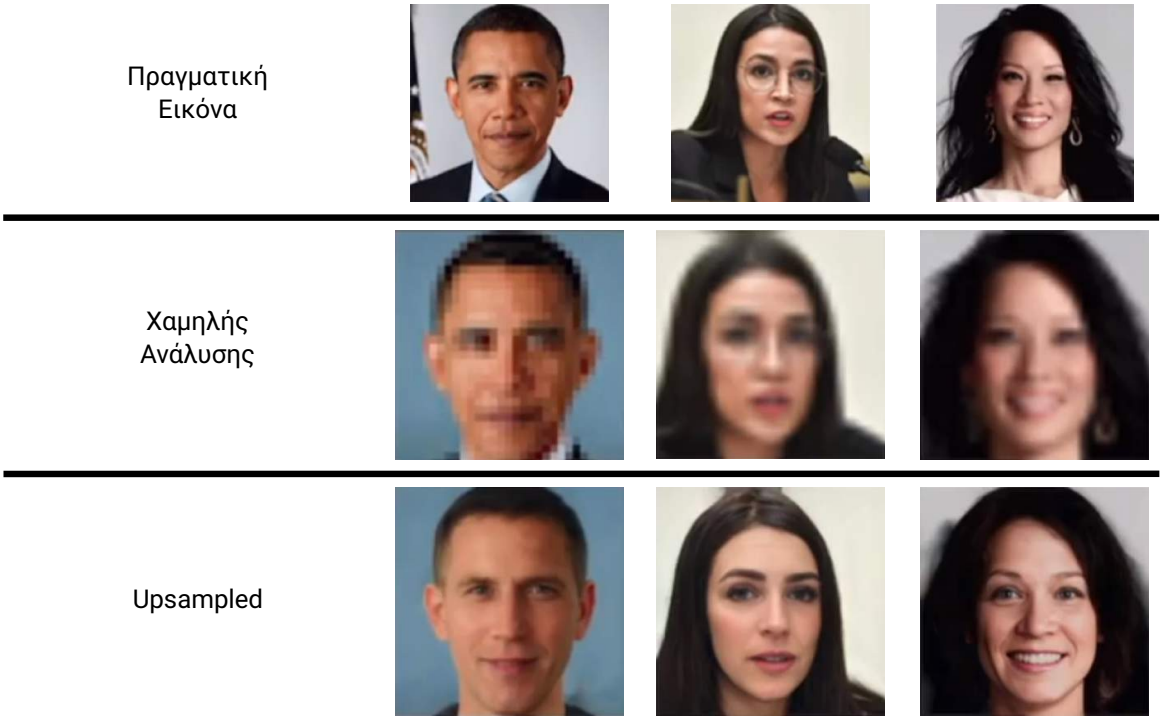
ποίησης εικόνων: την ύπαρξη Πόλωσης και Μεροληψίας τόσο στους αλγορίθμους και τα μοντέλα που χρησιμοποιούνται, όσο και στα ίδια τα σύνολα δεδομένων εκπαίδευσης. Πιο συγκεκριμένα, θα αναφερθούμε στον τρόπο με τον οποίο εισάγεται πόλωση στην εκπαίδευση των μοντέλων, όπως σε ένα GAN, και θα αναφέρουμε παραδείγματα επιτυχώς εκπαιδευμένων GANs που παρουσιάζουν πόλωση στα παραγόμενα δείγματά τους.

Ξεκινώντας, ίσως το πιο διαδεδομένο παράδειγμα ύπαρξης πόλωσης σε αλγορίθμους μηχανικής μάθησης είναι αυτό του συστήματος εκτίμησης ρίσκου (risk assessment) COMPAS που χρησιμοποιούνταν συμβουλευτικά σε πολλά αμερικανικά δικαστήρια. Στόχος ήταν το σύστημα αυτό να αποτελεί βοηθό ενός δικαστή στην εκτίμηση της πιθανότητας ένας κατηγορούμενος να υποτροπιάσει και να προβεί εκ νέου σε παράνομες πράξεις εάν δεν προφυλακιστεί. Το 2016, σε μια έρευνα που μετρούσε την ακρίβεια του συστήματος COMPAS [53], διαπιστώθηκε ότι «*οι έγχρωμοι κατηγορούμενοι έχουν διπλάσιες πιθανότητες από τους λευκούς να χαρακτηριστούν υψηλότερου ρίσκου, αλλά στην πραγματικότητα δεν προβαίνουν ξανά στην ίδια πράξη*», ενώ το COMPAS προέβλεπε ότι «*οι λευκοί είναι πολύ πιο πιθανό από τους μαύρους να χαρακτηριστούν χαμηλού κινδύνου αλλά προχωρούσαν και πάλι σε παράνομες πράξεις*». Διαπίστωσαν επίσης ότι μόνο το 20 τοις εκατό των ανθρώπων που προέβλεπε ότι θα διαπράττουν βίαια εγκλήματα, όντως συνέχισαν να το κάνουν, κάτι που φανερώνει την έντονη φυλετική μεροληψία του αλγορίθμου και την αναξιοπιστία του.

Στα πλαίσια των GANs, μεροληψία και πόλωση εισάγετε τόσο λόγω της σχεδίασης του μοντέλου και της συνάρτησης κόστους, όσο και από την κατασκευή του συνόλου δεδομένων εκπαίδευσης. Από τη μία η Συρρίκνωση Ρυθμών και από την άλλη η απουσία κανονικοποιητών της συνάρτησης κόστους των GANs οδηγεί πολλές φορές το μοντέλο να μεγεθύνει την πόλωση που πιθανό να υπάρχει στο σύνολο δεδομένων ή, τουλάχιστον, να μην την περιορίζει. Εξίσου σημαντική πηγή μεροληψίας αποτελεί και η μη-ύπαρξη ικανής ποικιλομορφίας στο σύνολο δεδομένων εκπαίδευσης. Έχοντας ως δεδομένο ότι ένα επιτυχώς εκπαιδευμένο GAN είναι σε θέση να παράγει δείγματα που θα μπορούσαν να ανήκουν στο σύνολο δεδομένων εκπαίδευσης, η ύπαρξη πόλωσης στο τελευταίο σχεδόν αναπόφευκτα θα «περάσει» και στα παραγόμενα δείγματα του μοντέλου.

Για λόγους πληρότητας, παραθέτουμε στο σχήμα 34, παρακάτω, ένα παράδειγμα ύπαρξης φυλετικής πόλωσης σε ένα επιτυχώς εκπαιδευμένο GAN. Το μοντέλο, που ονομάζεται PULSE [113], χρησιμοποιεί το StyleGAN (βλ. υποενότητα 4.1.3) για να αυξήσει την ανάλυση (upsampling) σε χαμηλής ανάλυσης εικόνες με ανθρώπινα πρόσωπα. Η ύπαρξη πόλωσης

είναι εμφανής, ωστόσο οι συγγραφείς δεν μπορούν με βεβαιότητα να την αποδώσουν μόνο στο σύνολο δεδομένων ή μόνο στην αρχιτεκτονική κάποιου από τα νευρωνικά δίκτυα που αναπτύχθηκαν - κάτι που δηλώνει τη δυσκολία εύρεσης της πηγής ύπαρξης μεροληψίας και της δυσκολίας εξάλειψής της. Έτσι, και στις τρεις περιπτώσεις του σχήματος, παρατηρείται η ύπαρξη φυλετικής πόλωσης ή μεροληψίας. Συγκεκριμένα, ειδικά στην πρώτη και τελευταία στήλη, φαίνεται ότι μοντέλο προσπαθεί να βγάλει πρόσωπα λευκών αντί για έγχρωμων ανθρώπων, παρόλο που οι χαμηλής ανάλυσης είσοδοι (δεύτερη γραμμή) προτάσσουν διαφορετικά. Η επιλογή αυτή από το μοντέλο γίνεται αφενός γιατί εκπαιδεύτηκε με περισσότερες εικόνες λευκών ανθρώπων και άρα προσπαθεί να βγάλει ένα «μέσο» χρώμα δέρματος, αλλά αφετέρου σύμφωνα με τους συγγραφείς οφείλεται και στην αρχιτεκτονική του μοντέλου StyleGAN που χρησιμοποιούν.



Σχήμα 34: Τρεις διαφορετικές παραγωγές του μοντέλου PULSE (μία για κάθε στήλη) όπου παρατηρείται φυλετική πόλωση ή μεροληψία του μοντέλου.
Πηγή: Ανακατασκευή από «PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models», Menon et al., 2020 [113]

Θα θέλαμε να σημειώσουμε στο σημείο αυτό, πως πόλωση και μεροληψία υπέρ των δεδομένων του συνόλου εκπαίδευσης είναι κάτι που παρατηρήθηκε έντονα και στα μοντέλα που αναπτύχθηκαν στα πλαίσια της παρούσας εργασίας.

Σύγκλιση και Χρόνος Εκπαίδευσης

Καθώς ο Generator βελτιώνεται με την εκπαίδευση, η απόδοση του Discriminator επιδεινώνεται επειδή ο Discriminator δεν μπορεί εύκολα διακρίνει μεταξύ των πραγματικών και τεχνητών εικόνων. Εάν ο Generator, για παράδειγμα, επιτύχει «τέλεια» εκπαίδευση, τότε ο Discriminator το καλύτερο που έχει να κάνει είναι να μαντεύει τυχαία και αμερόληπτα (ακρίβεια 50%). Στην πραγματικότητα, ο Discriminator αναστρέφει ένα νόμισμα για να κάνει την πρόβλεψή του [125].

Αυτή η εξέλιξη θέτει ένα πρόβλημα για τη σύγκλιση του GAN στο σύνολό της: η ανατροφοδότηση των ταξινομήσεων στην έξοδο του Discriminator γίνεται όλο και λιγότερο σημαντική με την πάροδο του χρόνου. Εάν το ΓΑΝ συνεχίσει να εκπαιδεύεται πέρα από το σημείο που ο Discriminator δίνει εντελώς τυχαία ανατροφοδότηση, τότε ο Generator αρχίζει να εκπαιδεύεται για ανεπιθύμητη ανατροφοδότηση και η δική του ποιότητα μπορεί ακόμα και να καταρρεύσει. Γενικότερα, η σύγκλιση της εκπαίδευσης στα GANs είναι συνήθως μια ασταθής κατάσταση [77], κάτι που κάνει τη σχεδίαση της αρχιτεκτονικής των δικτύων αλλά και των παραμέτρων εκπαίδευσης περισσότερο τέχνη και έμπνευση παρά μεθοδολογία. Αξιοσημείωτο στο σημείο αυτό, είναι ότι η αστάθεια στην εκπαίδευση των GANs (καθώς και η μη-αντιστρεψιμότητά του) έχουν αποτρέψει την εγκατάλειψη μεθόδων όπως οι MAK για Παραγωγική Μοντελοποίηση εικόνων, μιας και που οι τελευταίοι συνήθως έχουν αρκετά εύρωστη εκπαίδευση.

Για τη βελτίωση και σταθεροποίηση της σύγκλισης στα GANs έχουν προταθεί αρκετές τεχνικές, οι οποίες εστιάζουν κυρίως στη χρήση κανονικοποιητών (regularizers) της συνάρτησης κόστους. ο Arjovsky et al., για παράδειγμα, πρότειναν στο μοντέλο τους Regularized Jensen-Shannon GAN (Regularized JSGAN) . Σε μια διαφορετική προσέγγιση, ο Arjovsky et al., πρότειναν τη προσθήκη θορύβου στις εισόδους του Discriminator [68], κάτι όπως αναφέρουν ισοδυναμεί με κανονικοποίηση της συνάρτησης κόστους του Discriminator παρόμοια με την προηγούμενη.

3.4 Αξιολόγηση Παραγόμενων Δειγμάτων από GANs

Η αξιολόγηση των παραγόμενων δειγμάτων από GANs ήταν και παραμένει μείζον ερευνητικό πρόβλημα. Αυτό συμβαίνει διότι, στην περίπτωση παραγωγής εικόνων για παράδειγμα, είναι πολύ δύσκολο να βρεθεί μία εύρωστη μετρική για την αξιολόγηση

του ρεαλισμού των εικόνων που παράγει ένας Generator ή της απόστασης μεταξύ των πιθανοτικών κατανομών του Generator και της πραγματικής. Σε αντιδιαστολή, για την αξιολόγηση ενός ταξινομητή εικόνων, πρέπει απλώς να συγκριθεί η εκάστοτε έξοδός του με την αντίστοιχη ετικέτα, κάτι αδύνατο στα πλαίσια της παραγωγικής μοντελοποίησης και ειδικά στην παραγωγή εικόνων με GANs. Παρ΄ όλη τη δυσκολία που παρουσιάζει η μέτρηση της παραγωγικής δυνατότητας ενός GAN, έχουν προταθεί αρκετές τεχνικές αξιολόγησης ανεξάρτητες του μοντέλου (model-independent metrics) - με ορισμένες από αυτές να έχουν προταθεί αρκετά πρόσφατα.

Κατά μια έννοια, η μη-ύπαρξη ενός γενικά αποδεκτού Discriminator που θα μπορούσαμε να χρησιμοποιήσουμε για την αξιολόγηση και σύγκριση διαφορετικών Generators [122], μας αναγκάζει να καταφύγουμε σε άλλες τεχνικές, ακόμα και στην ανάθεση σε ανθρώπους της εργασίας διάκρισης μεταξύ πραγματικών/τεχνητών εικόνων (για έμμεση αξιολόγηση της ποιότητας των παραγόμενων δειγμάτων) [123]. Στις τεχνικές αυτές (πλην της ανάθεσης σε ανθρώπους) αυτό που πρακτικά αξιολογείται είναι τα εξής:

- **Ποιότητα (Fidelity)**: η πρώτη ιδιότητα που θα θέλαμε να έχουν τα παραγόμενα δείγματα από έναν Generator ενός μοντέλου GAN είναι αυτά να είναι υψηλής ποιότητας, δηλαδή να επιδεικνύουν ρεαλισμό και να έχουν περιορισμένα έως καθόλου artifacts.

- **Ποικιλομορφία (Diversity)**: η δεύτερη και ίσως σημαντικότερη ιδιότητα που θα θέλαμε να έχουν τα παραγόμενα δείγματα από έναν Generator είναι αυτά να παρουσιάζουν ποικιλομορφία, δηλαδή ο Generator να είναι σε θέση να παράγει ρεαλιστικά δείγματα αλλά από διάφορες και ποικίλες τάξεις των δεδομένων εκπαίδευσης. Υπό αυτήν την έννοια, η αντιμετώπιση του προβλήματος της Συρρίκνωσης Ρυθμών γίνεται με σκοπό την αύξηση αυτής της ιδιότητας των παραγόμενων δειγμάτων.

Όπως είναι εύκολα αντιληπτό, τόσο η ποιότητα όσο και η ποικιλομορφία των παραγόμενων δειγμάτων ενός GAN, αν και εύκολα αξιολογείται από το σύστημα όρασης ενός ανθρώπου, είναι δύσκολο να περιγραφεί στον υπολογιστή και άρα να μετρηθεί εύρωστα και αξιόπιστα. Πρακτικά, αυτό που γίνεται με τις μετρικές αξιολόγησης εικόνων από ΒΠΜ σε ότι αφορά την ποιότητα αυτών (πρώτη ιδιότητα), είναι η σύγκριση[9] ομάδων τεχνητών

[9]Για τη σύγκριση εικόνων και εξαγωγή της «απόστασης» μεταξύ τους δεν εργαζόμαστε στον χώρο των εικονοστοιχείων (pixel space), αλλά αντ΄ αυτού περνάμε τις εικόνες από κάποιο ΣΝΔ εξαγωγής χαρακτηριστικών και κατόπιν συγκρίνουμε τις εξόδους κάποιας από τις τελευταίες συνελικτικές στρώσεις του δικτύου αυτού. Έτσι, η σύγκριση των εικόνων γίνεται στο χώρο των χαρακτηριστικών (feature space) και άρα χρησιμοποιεί μια πιο υψηλού επιπέδου αναπαράσταση, για πιο πιο εύρωστη και αξιόπιστη σύγκριση.

εικόνων με ομάδες πραγματικών και ο υπολογισμός κάποιας «απόστασης» μεταξύ των ομάδων αυτών. Αντίστοιχα, για τη δεύτερη ιδιότητα, αυτή της ποικιλομορφίας, αυτό που γίνεται είναι η προσπάθεια ταξινόμησης των παραγόμενων δειγμάτων με σκοπό τον υπολογισμό της ποικιλίας των τάξεων στις οποίες ένας εξωτερικός ταξινομητής κατατάσσει τα παραγόμενα και τα πραγματικά δείγματα, κάτι που αναλύεται πιο λεπτομερώς σε επόμενη υποενότητα.

Οι κυρίαρχες μετρικές αξιολόγησης των παραγόμενων εικόνων από GANs είναι η εξής τέσσερις (4): Inception Score (IS), Fréchet Inception Distance (FID), Precision-Recall-F_1 και Structural Similarity (SSIM). Υπάρχουν κι άλλες όπως το Perceptual Path Length (PPL) αλλά οι παραπάνω τέσσερις (4) αφενός είναι οι πιο διαδεδομένες στη βιβλιογραφία των GANs, ενώ αφετέρου κάνουμε χρήση και των τεσσάρων για την αξιολόγηση των μοντέλων τα οποία αναπτύχθηκαν και εκπαιδεύτηκαν στην παρούσα εργασία. Παρακάτω, αφιερώνουμε μία υποενότητα για την ανάλυση της κάθε μίας από τις μετρικές αξιολόγησης που χρησιμοποιήθηκαν.

Inception Score (IS)

Ίσως η πρώτη μέθοδος για αξιολόγηση των παραγόμενων εικόνων από ΒΠΜ που αναπτύχθηκε είναι το Inception Score (IS). Όπως περιγράφεται και στο όνομά της, αυτή η μετρική υπολογίζεται με βάση την έξοδο (δηλ. τις πιθανότητες ταξινόμησης) του διακριτικού μοντέλου Inception (βλ. υποενότητα 2.1) και συγκεκριμένα της τρίτης έκδοσης αυτού (Inception v3 [46]). Πρωτοεμφανιζόμενη από τον Salimans et al. στο άρθρο τους «*Improved Techniques for Training GANs*» [59], η μετρική χρησιμοποιεί την multinoulli κατανομή στην έξοδο του Inception v3, το οποίο έχει προ-εκπαιδευτεί στο ImageNET.

Αναλυτικά, ο υπολογισμός του Inception Score έχει ως εξής:
Σε κάθε μία από τις παραχθείσες (τεχνητές) εικόνες, εφαρμόζεται το μοντέλο Inception το οποίο δίνει την (υπο-συνθήκη) multinoulli κατανομή, $p(Y = y_i|X = x)$ όπου Y είναι η τ.μ. με 1000 πιθανές τιμές $y_i, i = 1...1000$ (όσες και οι διακριτές ετικέτες του ImageNET) και x μια εικόνα στην είσοδο του μοντέλου. Στις εικόνες στις οποίες περιέχονται ερμηνεύσιμα αντικείμενα αναμένουμε η υπό-συνθήκη κατανομή ετικετών $p(y|x)$ να έχει χαμηλή εντροπία (ιδανικά θα θέλαμε μία ετικέτα να έχει πιθανότητα κοντά στη μονάδα και όλες οι άλλες κοντά στο 0), ένα δείγμα της καλής ποιότητας (fidelity) των εικόνων που παράγει ο Generator. Επιπλέον, επιδιώκουμε το μοντέλο να παράγει εικόνες με ποικιλομορφία (diversity), οπότε θα θέλαμε η περιθώρια κατανομή $\int_z p(y|x = G(z))\, dz$ να έχει υψηλή

εντροπία (ιδανικά να είναι κοντά στην ομοιόμορφη κατανομή). Συνδυάζοντας αυτές τις δύο απαιτήσεις, η μετρική που προτάθηκε στο [59] είναι:

$$IS = exp\left(\mathbb{E}_{x\sim p_{model}}\left[KL\left(p\left(y|x\right)||p\left(y\right)\right)\right]\right) \tag{3.34}$$

$$= exp\left(\mathbb{E}_{x\sim p_{model}}\left[p\left(y|x\right) * log\left(\frac{p(y|x)}{p(y)}\right)\right]\right) \tag{3.35}$$

όπου η ύπαρξη του εκθετικού γίνεται έτσι ώστε οι τιμές του Inception Score να είναι πιο εύκολα συγκρίσιμες, KL είναι η απόσταση Kullback-Leibler, ενώ όπως αναφέρθηκε η $p(y|x)$ υπολογίζεται από το Inception v3.

Από μαθηματικής σκοπιάς, λόγω του εκθετικού και του γεγονότος ότι η KL παίρνει μη-αρνητικές τιμές, το πιθανό εύρος τιμών της μετρικής Inception Score είναι $[0, \infty)$. Πρακτικά ωστόσο, λόγω του ότι έχουμε κατηγορική υπό-συνθήκη κατανομή, οι τιμές που λαμβάνει το Inception Score είναι από 1.0 (υψηλής εντροπίας υπο-συνθήκη κατανομή - χαμηλής ποιότητας δείγματα) έως N_{class}, που είναι ο αριθμός των διακριτών τάξεων του συνόλου δεδομένων εκπαίδευσης (τότε θα έχουμε και τις δύο επιθυμητές ιδιότητες στα παραγόμενα δείγματα - υψηλή ποιότητα και μεγάλη ποικιλομορφία), κάτι που φαίνεται ακολούθως:

$$\left.\begin{aligned} p\left(y|x\right)_{perfect} &= \delta\left(y-\hat{y_x}\right), \forall x \sim p_{model} \\ p\left(y\right)_{perfect} &= \frac{1}{N_{class}} \end{aligned}\right\} \iff IS_{perfect} = exp\left(\mathbb{E}_x\left[\delta\left(y-\hat{y_x}\right) * log\left(\frac{\delta\left(y-\hat{y_x}\right)}{1/N_{class}}\right)\right]\right)$$

$$= exp\left(1 * log\left(N_{class}\right)\right)$$

$$= N_{class} \tag{3.36}$$

$$\left.\begin{aligned} p\left(y|x\right)_{worst} &= \frac{1}{N_{class}}, \forall x \sim p_{model}(random) \\ p\left(y\right) &= \frac{1}{N_{class}} \end{aligned}\right\} \iff IS_{worst} = exp\left(\mathbb{E}_x\left[\frac{1}{N_{class}} * log\left(\frac{\frac{1}{N_{class}}}{\frac{1}{N_{class}}}\right)\right]\right)$$

$$= exp\left(\frac{1}{N_{class}} * log\left(1\right)\right)$$

$$= exp(0) = 1 \tag{3.37}$$

Επομένως, στο εύρος $\left[1, \frac{1}{N_{class}}\right]$ *όσο μεγαλύτερο είναι το Inception Score τόσο το καλύτερο.*

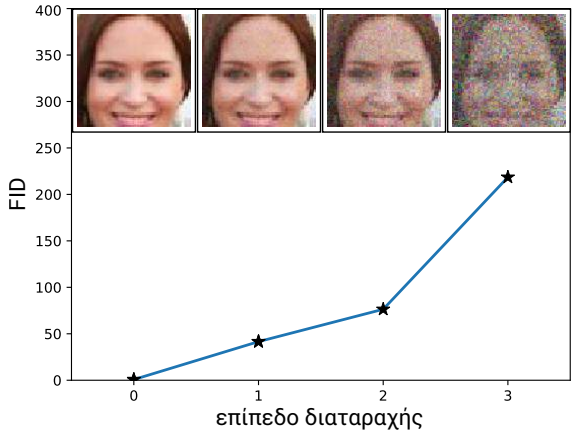
Βασικό μειονέκτημα τις μετρικής Inception Score είναι ότι «βλέπει» μονάχα τις τεχνητές εικόνες στην έξοδο του Generator και δεν τις συγκρίνει με τις πραγματικές εικόνες στις

οποίες αυτός εκπαιδεύτηκε. Επίσης καίριο μειονέκτημα, είναι η έντονη εξάρτηση της μετρικής από τις εξόδους ενός δικτύου που είχε αρχικά εκπαιδευτεί για ταξινόμηση εικόνων του ImageNET. Έτσι, εάν το σύνολο δεδομένων αποτελείται από αρκετά διαφορετικές κατηγορίες εικόνων, είναι πολύ πιθανό η μετρική να γίνει εντελώς αναξιόπιστη (με την έννοια ότι για καλύτερα οπτικά αποτελέσματα αυτή μπορεί να χειροτερεύει). Τέλος, όπως φαίνεται και στην εξίσωση 3.36, το μοντέλο θα μπορούσε να βγάζει την ίδια τάξη ($\hat{y_x}$) κάθε φορά και εφόσον τα δείγματα της τάξης αυτής είναι αρκετά ρεαλιστικά θα λάμβανε το τέλειο Inception Score, κάτι που «ενθαρρύνει» τη Συρρίκνωση Ρυθμών. Είναι πλέον ευρέως αποδεκτό πως άλλες μετρικές, όπως η FID που αναλύεται στη συνέχεια, είναι πιο εύρωστες και περισσότερο συσχετισμένες με την ανθρώπινη κρίση.

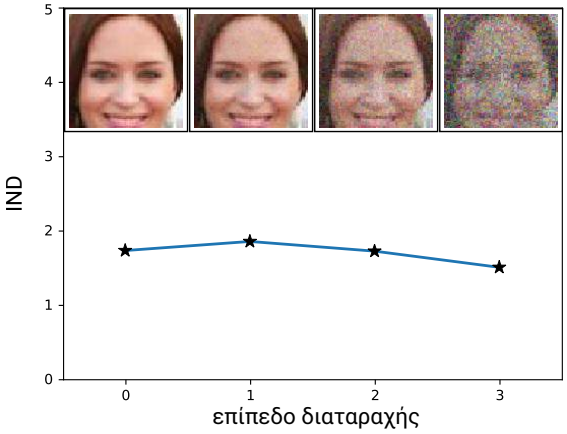
Fréchet Inception Distance (FID)

Μία άλλη μετρική αξιολόγησης των παραγόμενων εικόνων από GANs είναι η απόσταση Fréchet μεταξύ διανυσμάτων χαρακτηριστικών του Inception v3, *Fréchet Inception Distance* (FID). Η μετρική αυτή, η οποία παρουσιάστηκε από τον Heusel et al. στο άρθρο τους «*GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*» [73], προσπαθεί να λύσει τα προβλήματα αξιοπιστίας του Inception Score και να συμβαδίζει περισσότερο με την ανθρώπινη κρίση. Για να το πετύχει αυτό, διαφοροποιείται από το Inception Score σε δύο βασικά σημεία: κατά πρώτον χρησιμοποιεί και τις πραγματικές εικόνες για τη μέτρηση της ποιότητας των τεχνητών και κατά δεύτερον δεν χρησιμοποιεί την τελική έξοδο του ταξινομητή Inception (δηλ. τις πιθανότητες ταξινόμησης) αλλά τα διανύσματα χαρακτηριστικών στην έξοδο της τελευταίας στρώσης πριν την πρώτη πλήρως-συνδεδεμένη στρώση αυτού. Αυτό έχει οδηγήσει σε μια μετρική σημαντικά πιο εύρωστη και περισσότερο συμβατή με την ανθρώπινη κρίση, κάτι που φαίνεται και στο σχήμα 35 που έπεται.

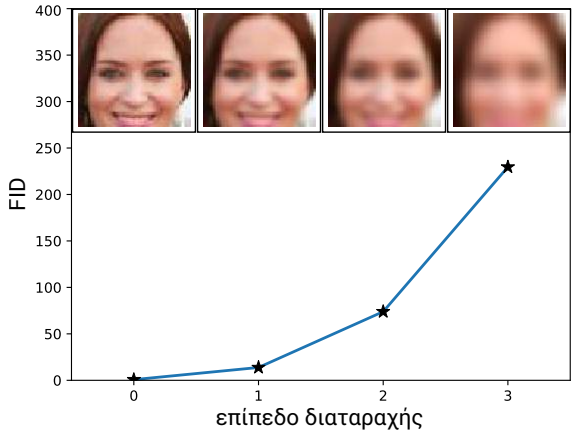
Όπως φαίνεται στο σχήμα αυτό, για αυξανόμενη διαταραχή μιας εικόνας αναφοράς, φαίνεται πως η FID αυξάνει μονότονα κάτι που αποτελεί σημάδι ευρωστίας και αξιοπιστίας. Αντίθετα, το Inception Score φαίνεται σχεδόν επίπεδο, ή χειρότερα αυξάνει (η IND μειώνεται), όπως στην περίπτωση γκαουσιανού θολώματος της εικόνας (δεύτερη σειρά).



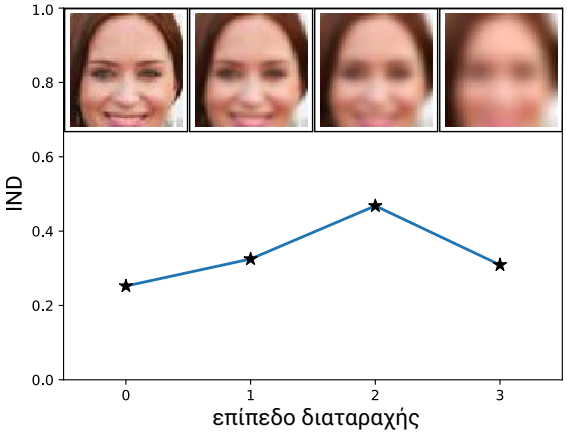
(α) Μεταβολή FID στον θόρυβο



(β) Μεταβολή IND στον θόρυβο



(γ) Μεταβολή FID στο θόλωμα



(δ) Μεταβολή IND στο θόλωμα

Σχήμα 35: Σύγκριση μετρικής FID με τη μετρική Inception Score (για την ακρίβεια μια παραλλαγή του αντίστροφου Inception Score, που οι συγγραφείς ονόμασαν Inception Distance - IND).
Πηγή: Ανακατασκευή από «GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium», Heusel et al., 2017 [73]

Εξαγωγή Διανυσμάτων Χαρακτηριστικών

Για τον υπολογισμό των διανυσμάτων χαρακτηριστικών ή των εμβυθίσεων (embeddings) όπως αλλιώς αποκαλούνται τα διανύσματα χαρακτηριστικών όταν λαμβάνονται από τις τελευταίες συνελικτικές ή pooling στρώσεις ενός ταξινομητή εικόνων, ακολουθούμε την εξής διαδικασία (γραμμένη σε τρίτο πρόσωπο για ευκολότερη ανάγνωση):
Πάρε έναν προ-εκπαιδευμένο ταξινομητή εικόνων και αφαίρεσε τις πλήρως-συνδεμένες στρώσεις στην έξοδό του καθώς και τη σιγμοειδή συνάρτηση εξόδου (αφού δεν ενδιαφερόμαστε για χρήση του ως ταξινομητή). Τροφοδότησέ τον με μία εικόνα στην είσοδό του και πραγματοποίησε ένα εμπρόσθιο πέρασμα. Πάρε την έξοδο της τελευταίας στρώσης του περικομμένου δικτύου (δηλ. της στρώσης πριν την πρώτη πλήρως-συνδεδμένη στρώση

του αρχικού), μετασχημάτισέ τη σε ένα μακρύ διάνυσμα (flatten) και επέστρεψε αυτό το διάνυσμα ως εμβύθιση της εικόνας εισόδου.

Όταν για την εξαγωγή των εμβυθίσεων χρησιμοποιούμε έναν ταξινομητή που έχει προ-εκπαιδευθεί στο σύνολο δεδομένων ImageNET (κάτι πολύ σύνηθες), τότε οι εμβυθίσεις λέγονται *εμβυθίσεις του ImageNET (ImageNET embeddings)*. Η μετρική FID χρησιμοποιεί το Inception v3 προ-εκπαιδευμένο στο ImageNET για την εξαγωγή αυτών των εμβυθίσεων, μία τάση που έχει ακολουθηθεί γενικότερα στις μετρικές αξιολογήσεις των GANs. Ό λόγος που χρησιμοποιούνται δίκτυα ταξινόμησης εικόνας προ-εκπαιδευμένα στο ImageNET, είναι διότι αυτό θεωρείται ένα πολύ γενικό σύνολο δεδομένων αποτελούμενο από περίπου 1,3 εκατομμύρια εικόνες οργανωμένες σε 1000 τάξεις (όπως δόθηκε στην πρόκληση ILSVRC το 2012) και άρα τα διανύσματα χαρακτηριστικών θα είναι αρκετά αντιπροσωπευτικά (δηλ. δύο κοντινές οπτικά εικόνες θα έχουν και κοντινές σε νόρμα εμβυθίσεις του ImageNET). Ωστόσο μπορούν να χρησιμοποιηθούν ταξινομητές προ-εκπαιδευμένοι και σε άλλα σύνολα δεδομένων, μιας και που πλέον υπάρχει πληθώρα τέτοιων ταξινομητών εκπαιδευμένων σε ποικίλα σύνολα δεδομένων.

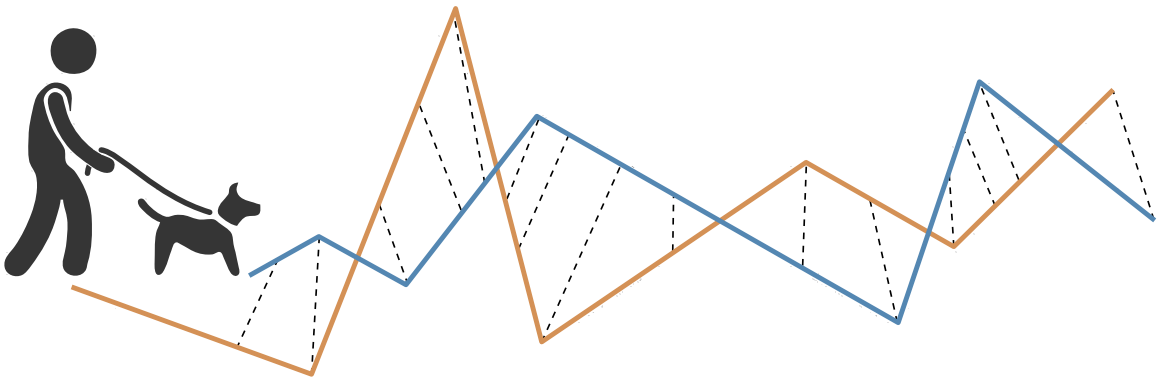
Στο σημείο αυτό αξίζει να τονίσουμε πως θεωρητικά μπορούμε να λάβουμε τις εμβυθίσεις μιας εικόνας από την έξοδο οποιαδήποτε στρώσης ενός ταξινομητή εικόνων και όχι απαραίτητα από την τελευταία. Ωστόσο, ο λόγος για τον οποίο η τελευταία συνελικτική (ή pooling εφόσον υπάρχει) στρώση σε τέτοια δίκτυα ονομάζεται στρώση χαρακητριστκών (feature layer) είναι διότι ενώ οι αρχικές συνελικτικές στρώσεις τέτοιων δικτύων μαθαίνουν να αναγνωρίζουν απλά χαρακτηριστικά (όπως αναγνώριση ακμών), οι τελευταίες συνελικτικές στρώσεις μαθαίνουν να αναγνωρίζουν και να ενεργοποιούνται σε πολύ πιο σύνθετες δομές και χαρακτηριστικά (όπως η ύπαρξη ενός προσώπου ή ενός πολύπλοκου αντικειμένου). Έτσι, η χρήση αυτών ως στρώσεων εξαγωγής εμβυθίσεων δίνει διανύσματα που αιχμαλωτίζουν επαρκώς τα οπτικά χαρακτηριστικά της εικόνας εισόδου.

Όταν ως ταξινομητής χρησιμοποιείται το Inception v3 και ως στρώση λήψης της εμβύθισης η τελευταία pooling στρώση (max-pooling 8×8) του δικτύου, τότε τα διανύσματα χαρακτηριστικών ή εμβυθίσεις έχουν *2048 στοιχεία*. Συγκεκριμένα, η έξοδος της στρώσης είναι 1×1×2048 και μετά το flattening θα έχουμε ένα μονοδιάστατο διάνυσμα μήκους 2048. Επομένως, μέσω της παραπάνω διαδικασίας εξαγωγής των εμβυθίσεων, κάθε εικόνα *αντιστοιχίζεται από ένα σημείο του αρχικού υψηλής-διαστασιμότητας χώρου εισόδου (χώρος των εικονοστοιχείων - στο Inception v3 αυτός θα είναι* $\mathbb{R}^{299*299*3}$*), στον πολύ*

μικρότερης διάστασης χώρο $\mathbb{R}^{2048}$. Ως αποτέλεσμα αυτής της αντιστοίχισης, οι συγκρίσεις εικόνων γίνονται πολύ πιο αξιόπιστα σε αυτόν τον νέο χώρο των εμβυθίσεων (embedding space) σε σχέση με τον αρχικό και άρα ο υπολογισμός της απόστασης τους μέσω της απόστασης των διανυσμάτων-εμβυθίσεών τους θα είναι και αυτός πιο αξιόπιστος.

Απόσταση Fréchet

Πριν προχωρήσουμε στον τύπο υπολογισμού της μετρικής FID παραθέτουμε σε αυτήν την παράγραφο τον ορισμό της απόστασης Fréchet, καθώς αυτή αποτελεί τη βάση της μετρικής. Η απόσταση Fréchet προτάθηκε το 1906 από τον γάλλο μαθηματικό Maurice Fréchet και δημοσιεύθηκε στο πρώτο κεφάλαιο του συνεδρίου «*The Rendiconti del Circolo Matematico di Palermo*» [1] εκείνης της χρονιάς, για τη μέτρηση της απόστασης μεταξύ καμπύλων. Δεν θα εμβαθύνουμε στα μαθηματικά του ορισμού της απόστασης Fréchet, θα παραθέσουμε ωστόσο έναν διαισθητικό ορισμό.



Σχήμα 36: Διαισθητικός ορισμός της απόστασης Fréchet μεταξύ δύο καμπύλων: η απόσταση Fréchet ισούται με το ελάχιστο μήκος του λουριού που απαιτείται ώστε ακολουθώντας ο καθένας διαφορετική καμπύλη, να φτάσουν στο τέλος (χωρίς δυνατότητα οπισθοδρόμησης).
Πηγή: Ανακατασκευή από «Build Better Generative Adversarial Networks», Zhou et al., [Online Course] [122]

Ο διαισθητικός ορισμός της απόστασης Fréchet μεταξύ δύο καμπύλων, σύμφωνα και με το σχήμα 36 που προηγείται, έχει ως εξής [119]:
Έστω ένας άνθρωπος ο οποίος περπατά πάνω σε μια πεπερασμένη καμπύλη κρατώντας τον σκύλο του με λουρί, με τον σκύλο να περπατά πάνω σε μια ξεχωριστή πεπερασμένη καμπύλη. Ο καθένας μπορεί να αλλάξει την ταχύτητά του για να διατηρήσει χαλαρό το λουρί, αλλά κανένας δεν μπορεί να κινηθεί προς τα πίσω. Τότε, ως απόσταση Fréchet μεταξύ των δύο καμπυλών ορίζεται το ελάχιστο μήκος του λουριού που απαιτείται για

να διασχίσουν (ο άνθρωπος και ο σκύλος) τις ξεχωριστές διαδρομές τους από την αρχή έως το τέλος. Εάν, για παράδειγμα οι δύο καμπύλες ήταν δύο ομόκεντροι κύκλοι, τότε η απόσταση Fréchet μεταξύ τους θα ισούνταν με τη διαφορά των ακτίνων τους.

Υπολογισμός της FID

Η απόσταση Fréchet μπορεί εκτός από καμπύλες να υπολογιστεί και μεταξύ κατανομών. Συγκεκριμένα, υπάρχουν αρκετές (συνεχείς κυρίως) κατανομές για τις οποίες υπάρχει αναλυτική έκφραση υπολογισμού της απόστασης Fréchet. Η μετρική FID υποθέτει κανονικές κατανομές, για τις οποίες η απόσταση υπολογίζεται ως εξής:

$$\left.\begin{array}{l} X \sim \mathcal{N}(\vec{\mu_X}, \Sigma_X) \\ Y \sim \mathcal{N}(\vec{\mu_Y}, \Sigma_Y) \end{array}\right\} \Longrightarrow d(X,Y) = \|\vec{\mu_X} - \vec{\mu_Y}\|^2 + Tr\left(\Sigma_X + \Sigma_Y - 2\sqrt{\Sigma_X \Sigma_Y}\right) \tag{3.38}$$

όπου $\vec{\mu}$ είναι τα διανύσματα μέσης τιμής, Σ οι πίνακες συμμεταβλητότητας των πολυδιάστατων κανονικών κατανομών που ακολουθούν οι τυχαίες μεταβλητές X και Y και $d(X,Y)$ είναι η απόσταση Fréchet μεταξύ των δύο κανονικών κατανομών.

Βασική ιδέα πίσω από τη μετρική αξιολόγησης Fréchet Inception Distance (FID) είναι ότι οι πολυδιάστατες κανονικές κατανομές μπορούν να προσεγγίσουν ικανοποιητικά τις κατανομές των εμβυθίσεων των εικόνων. Έτσι, για να υπολογίσουν την απόσταση μεταξύ δύο συνόλων εικόνων (στα GANs αυτά θα είναι το σύνολο μιας ομάδας πραγματικών και μίας ομάδας τεχνητών εικόνων), αυτό που πρότειναν οι συγγραφείς του [73] είναι μετά την εξαγωγή των εμβυθίσεων των εικόνων της κάθε ομάδας, να προσαρμοστεί μία κανονική κατανομή ανά ομάδα. Κατόπιν, υπολογίζεται αναλυτικά η απόσταση Fréchet μεταξύ των δύο κανονικών κατανομών σύμφωνα με τη σχέση 3.38. Άρα, *η μετρική Fréchet Inception Distance είναι η απόσταση Fréchet μεταξύ των κανονικών κατανομών που προσαρμόζονται στις εμβυθίσεις των εικόνων της κάθε ομάδας εικόνων εισόδου*.

Συμπερασματικά, η μέθοδος για υπολογισμό της απόστασης μεταξύ δύο ομάδων (batches) από εικόνες μέσω της μετρικής FID είναι η ακόλουθη:

1. **Υπολογισμός Εμβυθίσεων**: για κάθε ομάδα και για κάθε εικόνα της ομάδας, η εικόνα τροφοδοτείται στο περικομμένο Inception v3, η έξοδος του οποίου (μετά το flatten) είναι η εμβύθιση της εικόνας. Έτσι, αντί για ομάδες εικόνων θα έχουμε ομάδες διανυσμάτων εμβύθισης.

2. **Προσαρμογή Κανονικών Κατανομών**: για κάθε ομάδα εμβυθίσεων υπολογίζεται το

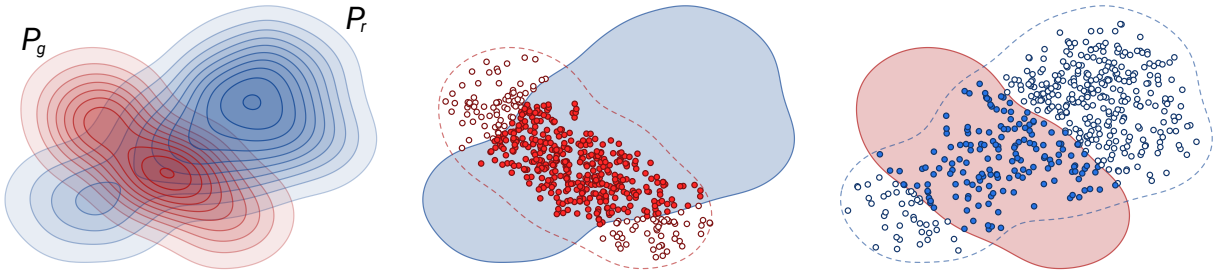
η (δειγματική) μέση εμβύθιση και ο (δειγματικός) πίνακας συμμεταβλητότητας των εμβυθίσεων.

3. **Υπολογισμός Απόστασης Fréchet**: ως τελευταίο βήμα υπολογίζεται η απόσταση Fréchet μεταξύ των δύο κανονικών κατανομών σύμφωνα με τη σχέση 3.38. Αυτή είναι η τιμή της μετρικής FID.

Όπως γίνεται φανερό, πρόκειται για μία μετρική απόστασης και άρα όσο μικρότερη είναι η τιμή της τόσο πιο κοντά θα είναι οι παραγόμενες εικόνες στις πραγματικές. Άρα, το εύρος τιμών της μετρικής FID είναι $[0, +\infty)$, ενώ στην πράξη είναι γενικά αποδεκτό πως αυτή είναι από τις πλέον αξιόπιστες μετρικές για σύγκριση των παραγωγών μεταξύ διαφορετικών GANs. Στα μειονεκτήματα της μετρικής αξίζει να αναφέρουμε την υψηλές υπολογιστικές απαιτήσεις λόγω του υπολογισμού του πίνακα συμμεταβλητότητας και της τετραγωνικής ρίζας, καθώς και ότι για να είναι αξιόπιστη χρειάζεται οι δειγματικές ροπές να υπολογιστούν από πολλά δείγματα (οι συγγραφείς αναφέρουν περισσότερα από 50.000 - ωστόσο σε αρκετά άρθρα αυτό το κατώφλι γίνεται 10.000). Ένα άλλο, ίσως λιγότερο σημαντικό, μειονέκτημα της μετρικής FID προέρχεται από τη χρήση του Inception προ-εκπαιδευμένου στο ImageNET για εξαγωγή των εμβυθίσεων. Εάν, ωστόσο, το σύνολο δεδομένων εκπαίδευσης ενός GAN περιέχει αρκετά διαφορετικές κατηγορίες εικόνων, τότε ίσως ο ταξινομητής να μην μπορέσει να αιχμαλωτίσει όλα τα οπτικά χαρακτηριστικά αυτού του συνόλου δεδομένων και άρα να επιστρέφει αραιές και λιγότερο χρήσιμες εμβυθίσεις.

Precision, Recall και F_1-Score στα GANs

Μία άλλη μετρική αξιολόγησης των παραγόμενων εικόνων από GANs, η οποία προτάθηκε αρκετά πρόσφατα και έχει γνωρίσει μεγάλη δημοφιλία είναι μια προσαρμοσμένη εκδοχή των κλασικών μετρικών αξιολόγησης ταξινομητών, Precision, Recall και F_1-Score. Η μετρική αυτή, ή καλύτερα η ομάδα μετρικών, οι οποίες παρουσιάστηκαν από τον Kynkäänniemi et al. στο άρθρο τους «*Improved Precision and Recall Metric for Assessing Generative Models*» [103], αποτελούν μία αναβαθμισμένη εκδοχή των αντίστοιχων μετρικών αξιολόγησης ταξινομητών, προσαρμοσμένες όμως στην αξιολόγηση των GANs. Συγκεκριμένα, οι συγγραφείς επιχειρούν να αξιολογήσουν με μεγαλύτερη σαφήνεια την απόδοση ενός GAN διαχωρίζοντας και μετρώντας ξεχωριστά την ποιότητα των παραγόμενων εικόνων από την ποικιλομορφία τους.



(α) Παράδειγμα κατανομών (β) Precision (γ) Recall

Σχήμα 37: Απεικόνιση του ορισμού των μετρικών Precision και Recall για κατανομές. Αριστερά (α) φαίνονται οι κατανομές των πραγματικών (μπλε), P_r, και παραγόμενων (κόκκινο) εικόνων, P_g. Η Precision (β) μετρά την πιθανότητα ένα τυχαίο δείγμα (ενν. εικόνα) από την P_g να πέσει στην υποστήριξη της P_r, ενώ η Recall μετρά την πιθανότητα ένα τυχαίο δείγμα (ενν. εικόνα) από την P_r να πέσει στην υποστήριξη της P_g.

Πηγή: «Improved Precision and Recall Metric for Assessing Generative Models», Kynkäänniemi et al., 2019 [103]

Η έμπνευση πίσω από τις μετρικές αυτές είναι ότι ιδανικά θα θέλαμε η πιθανοτική κατανομή που μαθαίνει ο Generator, P_g, να έχει ίδια δομή και όμοιες και σε κοντινές θέσεις ρυθμούς (modes) όπως η πραγματική κατανομή των δεδομένων εκπαίδευσης, P_r. Κατ΄ επέκταση, οι συγγραφείς του [103] πρότειναν τη μέτρηση της υποστήριξης[10] αυτών των κατανομών και ορισμό των μετρικών ως προς την επικάλυψη των υποστηρίξεων - κατ΄ αναλογία με τους παραδοσιακούς ορισμούς των μετρικών. Έτσι, όπως περιγράφεται και στο σχήμα 37 παραπάνω, οι μετρικές ορίζονται ως εξής:

- **Precision**: μετρά την πιθανότητα ένα τυχαίο δείγμα (ενν. εικόνα) από την P_g να πέσει στην υποστήριξη της P_r και άρα (θεωρητικά) υπολογίζει το λόγο του εμβαδού της περιοχής επικάλυψης προς το εμβαδό της υποστήριξης της κατανομής του Generator:

$$\text{Precision} = \frac{\text{περιοχή επικάλυψης}}{\text{υποστήριξη της κατανομής των τεχνητών}} \tag{3.39}$$

  Όπως φαίνεται, επομένως, η μετρική Precision κοιτάει πόσα παραγόμενα δείγματα από τον Generator θα μπορούσαν να ανήκουν στην πραγματική κατανομή των δεδομένων εκπαίδευσης και, υπό αυτήν την έννοια, **η Precision μετράει την ποιότητα των παραγόμενων δειγμάτων**.

[10] Ως υποστήριξη (support) μιας πιθανοτικής κατανομής νοείται το μέγιστο υποσύνολο του πεδίου ορισμού της για το οποίο η κατανομή έχει μη-μηδενικές τιμές. Έτσι, στην περίπτωση της μονοδιάστατης κανονικής κατανομή, η υποστήριξη είναι περίπου τέσσερις (4) τυπικές αποκλίσεις αριστερά και δέξια της μέσης τιμής.

- **Recall**: μετρά την πιθανότητα ένα τυχαίο δείγμα (ενν. εικόνα) από την P_r να πέσει στην υποστήριξη της P_g και άρα (θεωρητικά) υπολογίζει το λόγο του εμβαδού της περιοχής επικάλυψης προς το εμβαδό της υποστήριξης της πραγματικής κατανομής των δεδομένων εκπαίδευσης:

$$\text{Recall} = \frac{\text{περιοχή επικάλυψης}}{\text{υποστήριξη της κατανομής των πραγματικών}} \tag{3.40}$$

  Όπως φαίνεται, επομένως, η μετρική Recall κοιτάει πόσα από τα πραγματικά δείγματα του συνόλου εκπαίδευσης θα μπορούσαν (θεωρητικά) να παραχθούν από τον Generator και, υπό αυτήν την έννοια, **η Recall μετράει την ποικιλομορφία των παραγόμενων δειγμάτων**, καθώς μικρό Recall σημαίνει ότι ο Generator δεν μπορεί να μοντελοποιήσει αρκετά από τα χαρακτηριστικά των πραγματικών εικόνων.

- **F_1-Score**: όπως και στο κλασικό τύπο υπολογισμού του F_1-Score, έτσι και εδώ, αυτή η μετρική λαμβάνει υπόψη της τόσο το Precision όσο και το Recall, ως εξής:

$$F_1\text{-Score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.41}$$

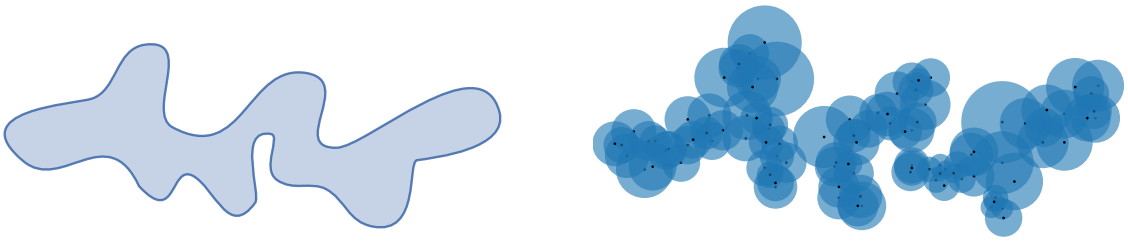
  όπου όπως φαίνεται από τον ορισμό του, το F_1-Score μετράει την ισορροπία μεταξύ της ποιότητας (από την Precision) και της ποικιλομορφίας (από την Recall) των δειγμάτων που μπορεί να παράξει ο Generator.

Υπολογισμός από τις Εμβυθίσεις

Επειδή ο υπολογισμός των παραπάνω μετρικών δεν μπορεί να γίνει άμεσα από τους αντίστοιχους ορισμούς τους, αυτό που πρότειναν οι συγγραφείς του [103] είναι μια προσέγγιση των ορισμών που βασίζεται σε μια προσεγγιστική μορφή των υποστηρίξεων των κατανομών. Συγκεκριμένα, προκειμένου να απαντήσουν το ερώτημα «*ανήκει το δείγμα x στην υποστήριξη της κατανομής P;*» και με δεδομένο ότι αυτό που δίνεται στην είσοδο τις διαδικασίας είναι απλώς δύο ομάδες εικόνων, ακολουθούν την εξής διαδικασία:

1. **Υπολογισμός Εμβυθίσεων**: για κάθε ομάδα και για κάθε εικόνα της ομάδας, η εικόνα τροφοδοτείται στο περικομμένο Inception v3, η έξοδος του οποίου (μετά το flatten) είναι η εμβύθιση της εικόνας. Έτσι, αντί για ομάδες εικόνων θα έχουμε ομάδες διανυσμάτων εμβύθισης, όπως συνέβη και κατά τον υπολογισμό της μετρικής FID.

2. **Προσέγγιση της υποστήριξης από τις Εμβυθίσεις**: κατόπιν, για κάθε ομάδα διανυσμάτων εμβύθισης, υπολογίζεται μια προσέγγιση της πραγματικής υποστήριξης ή



(α) Πραγματικό manifold (β) Προσεγγιστικό manifold

Σχήμα 38: (α) Παράδειγμα ενός πραγματικού manifold (αντίστοιχο της υποστήριξης κατανομών) στον χώρο των εμβυθίσεων κάποιας ομάδας εικόνων. (β) Εκτίμηση του manifold από δείγματα της ομάδας και σχεδίαση υπερσφαιρών με κέντρο το κάθε δείγμα και ακτίνα ίση με την απόσταση του k-οστού κοντινότερου γείτονα της εκάστοτε εμβύθισης.

Πηγή: «Improved Precision and Recall Metric for Assessing Generative Models», Kynkäänniemi et al., 2019 [103]

manifold της κάθε κατανομής, αρχικά υπολογίζοντας όλες τις αποστάσεις ανά ζεύγη των διανυσμάτων και κατόπιν παίρνοντας υπερσφαίρες στον χώρο των εμβυθίσεων, με κέντο το εκάστοτε διάνυσμα και ακτίνα ίση με την απόστασή του από το k-οστό κοντινότερο διάνυσμα, κάτι που απεικονίζεται στο σχήμα 38 παραπάνω. Όπως φαίνεται, η μετρική είναι *παραμετρική* με παράμετρο k, με τους συγγραφείς να προτείνουν μία τιμή κοντά στο $k = 3$ ως ιδανική για την πλειονότητα των περιπτώσεων.

3. **Συνάρτηση ελέγχου αν Δείγμα ανήκει σε Manifold**: για την απάντηση επομένως του ερωτήματος εάν μία εμβύθιση, ϕ, ανήκει στον όγκο (ή την προσέγγιση) του manifold μιας ομάδας εμβυθίσεων, $\mathbf{\Phi}$, οι συγγραφείς προτείνουν το εξής:

$$f(\phi, \mathbf{\Phi}) = \begin{cases} 1, \text{ εάν } \|\phi - \phi'\|_2 \leq \|\phi' - \text{NN}_k(\phi', \mathbf{\Phi})\|_2 \text{ για τουλάχιστον ένα } \phi' \in \mathbf{\Phi} \\ 0, \text{ αλλιώς,} \end{cases} \tag{3.42}$$

ελέγχοντας δηλαδή εάν υπάρχει απόσταση μεταξύ της εμβύθισης ϕ και οποιαδήποτε άλλης εμβύθισης που ανήκει στο $\mathbf{\Phi}$, η οποία να είναι μικρότερη ή ίση από την απόσταση της «άλλης» από τον k-οστό κοντινότερο γείτονά της.

4. **Υπολογισμός Precision και Recall**: για κάθε ομάδα εικόνων προκύπτει όπως αναφέρθηκε στο 1 μία ομάδα εμβυθίσεων, έστω $\mathbf{\Phi}_r$ για τις πραγματικές εικόνες και

$\mathbf{\Phi}_g$ για τις τεχνητές. Η έξοδος επομένως της $f(\phi, \mathbf{\Phi}_r)$ θα είναι 1 εάν η εικόνα από την οποία προέκυψε η εμβύθιση ανήκει στην υποστήριξη των πραγματικών εικόνων (αντίστοιχα για τις τεχνητές). Έχοντας, επομένως, τον τρόπο απάντησης στα ερωτήματα ύπαρξης μέλους, παρακάτω παραθέτουμε τον τρόπο υπολογισμού των μετρικών, όπως δόθηκε στο [103]:

$$\text{Precision}(\mathbf{\Phi}_r, \mathbf{\Phi}_g) = \frac{1}{|\mathbf{\Phi}_g|} \sum_{\phi_g \in \mathbf{\Phi}_g} f(\phi_g, \mathbf{\Phi}_r) \tag{3.43}$$

$$\text{Recall}(\mathbf{\Phi}_r, \mathbf{\Phi}_g) = \frac{1}{|\mathbf{\Phi}_r|} \sum_{\phi_r \in \mathbf{\Phi}_r} f(\phi_r, \mathbf{\Phi}_g) \tag{3.44}$$

ενώ το F_1-Score υπολογίζεται από τη σχέση 3.41 παραπάνω.

Επιλογικά, σημειώνουμε, πως η μετρικές Precision και Recall όπως προτάθηκαν στο [103] αποτελούν τις τελευταίες και πιο σύγχρονες (state-of-the-art) μετρικές αξιολόγησης της αποδοτικότητας των GANs και επιτρέπουν την πιο εύρωστη σύγκριση μεταξύ τους. Επίσης, τα πολύ καλά εκπαιδευμένα GAN τείνουν να τα πηγαίνουν καλύτερα στη μετρική Recall από ότι στη μετρική Precision λόγω του φαινομένου του υπερ-ρεαλισμού που συναντάται σε αυτά. Μειονέκτημα της μετρικής αποτελεί και εδώ η εξάρτηση των εμβυθίσεων από το σύνολο δεδομένων εκπαίδευσης του δικτύου εξαγωγής τους, που και εδώ συνήθως είναι το Inception v3.

Structural Similarity Index (SSIM)

Η τελευταία μετρική αξιολόγησης των αφορμάται επίσης από της ιδέα σύγκρισης μίας ομάδας πραγματικών εικόνων από μία τεχνητών, με σκοπό την ταυτόχρονο αξιολόγηση της ποιότητας και ποικιλομορφίας των παραγόμενων εικόνων. Πρόκειται για τη μετρική *Structural Similarity Index (SSIM)* η οποία αρχικά παρουσιάστηκε από τον Wang et al. το 2004 στο άρθρο τους «*Image Quality Assessment: From Error Visibility to Structural Similarity*» [13].

Πρόκειται για μια τεχνική διαφορετική από αυτές που προηγήθηκαν, με την έννοια ότι δεν προτάθηκε για αξιολόγηση των παραγωγών από GANs, αλλά για πιο «ρεαλιστική» σύγκριση μεταξύ δύο εικόνων. Για το σκοπό αυτό δεν θα επεκταθούμε ιδιαίτερα στον αναλυτικό τρόπο υπολογισμού της μετρικής, θα μείνουμε ωστόσο στα εξής σημεία της μετρικής για τη σύγκριση δύο εικόνων:

- **Εμπνευσμένη από το Ανθρώπινο Σύστημα Όρασης**: σύμφωνα με τους συγγραφείς το

ανθρώπινο σύστημα όρασης έχει βελτιστοποιηθεί στη γρήγορη αναγνώριση δομών (structures) και χρήση αυτών για σύγκριση μεταξύ των εικόνων. Προκειμένου να πετύχουν το διαχωρισμό της δομής μιας εικόνας από τη μεριά της κάμερας ή τον φωτισμό των αντικειμένων. Έτσι, οι συγγραφείς *συγκρίνουν χωριστά μια εκτίμηση της φωτεινότητας, της αντίθεσης και της δομής των δύο εικόνων* και στο τέλος συνδυάζουν τις διαφορές για το τελικό index.

- **Σύγκριση φωτεινότητας**: ως φωτεινότητα, λαμβάνεται η (δειγματική) μέση τιμή της έντασης των εικονοστοιχείων. Για τη σύγκριση της φωτεινότητας μεταξύ δύο εικόνων, $\mathbf{x}$ και $\mathbf{y}$, οι συγγραφείς προτείνουν τη χρήση της σχέσης:

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + \epsilon}{\mu_x^2\mu_y^2 + \epsilon} \tag{3.45}$$

- **Σύγκριση αντίθεσης**: ως αντίθεση, λαμβάνεται η (δειγματική) τυπική απόκλισης της έντασης των εικονοστοιχείων. Για τη σύγκριση της αντίθεσης μεταξύ δύο εικόνων, $\mathbf{x}$ και $\mathbf{y}$, οι συγγραφείς προτείνουν τη χρήση της σχέσης:

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + \epsilon}{\sigma_x^2\sigma_y^2 + \epsilon} \tag{3.46}$$

- **Σύγκριση δομής**: αφού οι εικόνες κανονικοποιηθούν (αφαιρεθεί η μέση τιμή και διαιρεθούν με την τυπική απόκλιση), υπολογίζεται η (δειγματική) ετεροσυσχέτιση μεταξύ των τιμών των κανονικοποιημένων εικόνων, σ_{xy}. Κατόπιν, για τη σύγκριση της δομής μεταξύ δύο εικόνων, $\mathbf{x}$ και $\mathbf{y}$, οι συγγραφείς προτείνουν τη χρήση της σχέσης:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + \epsilon}{\sigma_x\sigma_y + \epsilon} \tag{3.47}$$

Έτσι, οι συγγραφείς ορίζουν την πλήρη μετρική SSIM για τη σύγκριση δύο εικόνων, $\mathbf{x}$ και $\mathbf{y}$, ως εξής:

$$SSIM(\mathbf{x}, \mathbf{y}) = \frac{\left(2\mu_x\mu_y\right)\left(\sigma_{xy} + \epsilon\right)}{\left(\mu_x^2\mu_y^2 + \epsilon\right)\left(\sigma_x\sigma_y + \epsilon\right)} \tag{3.48}$$

- **Υπολογισμός από Συνελικτική Στρώση**: οι συγγραφείς πρότειναν έναν εναλλακτικό τρόπο (προσεγγιστικού) υπολογισμού της μετρικής, αυτού του ολισθαίνοντος τοπικού παραθύρου. Αυτός υπολογίζει αρκετές μετρικές SSIM για τα εικονοστοιχεία των εικόνων ενός τοπικού παραθύρου (στη δική μας υλοποίηση αυτό είναι

11×11px) και κατόπιν παίρνει το μέσο όρο των μετρικών. Οι επιμέρους μετρικές υπολογίζονται από μία ειδική συνελικτική στρώση με φίλτρα που υλοποιούν τον υπολογισμό της σχέσης 3.48 για κάθε θέση του παραθύρου. Αυτή η παραλλαγή της SSIM, που ονομάζεται Mean SSIM (MSSIM), υπολογίζεται ξεχωριστά για κάθε κανάλι, ενώ αν ακολούθως πάρουμε τον μέσο όρο των SSIM των καναλιών, η μετρική ονομάζεται Channel Mean SSIM (C-MSSIM και είναι αυτή η οποία χρησιμοποιείται για την αξιολόγηση των παραγόμενων εικόνων από GANs (κατά τη σύγκρισή τους με πραγματικές εικόνες)

Κεφάλαιο 4

Εφαρμογές των GANs

Το παρόν κεφάλαιο το αφιερώνουμε σε μια παράθεση διαφόρων εφαρμογών των GANs που έχουν παρουσιαστεί στη βιβλιογραφία και έχουν εφαρμοστεί στην πράξη. Σε ό,τι ακολουθεί, θα αναφερθούμε σε άρθρα και μοντέλα, παραλλαγές των οποίων έχουμε χρησιμοποιήσει στην παρούσα εργασία, καθώς και κάποιες άλλες εφαρμογές των GANs στην Παραγωγική Μοντελοποίηση εικόνων για λόγους πληρότητας.

Έτσι, ξεκινάμε το κεφάλαιο αναφέροντας εφαρμογές των GANs στην παραγωγή εικόνων από θόρυβο (παραγωγή χωρίς συνθήκη), ενώ στη συνέχεια παραθέτουμε αντίστοιχες εφαρμογές GANs στον (συζευγμένο ή μη) μετασχηματισμό εικόνας-σε-εικόνα (υπό-συνθήκη παραγωγή). Θέλουμε να τονίσουμε στο σημείο αυτό, πως σε καμία περίπτωση δεν επιχειρούμε μια εξαντλητική παράθεση έργων που έχουν παρουσιαστεί στη βιβλιογραφία, παρά εστιάζουμε στα μοντέλα και τεχνικές εκπαίδευσης που έχουμε κάνει χρήση ή θεωρούμε απαραίτητο να συμπεριληφθούν για λόγους πληρότητας.

4.1 Παραγωγή Εικόνας από Θόρυβο

Η πρώτη κατηγορία εφαρμογών GANs στην οποία εστιάζουμε είναι αυτή της παραγωγής εικόνων χωρίς συνθήκη. Σε αυτήν την κατηγορία, συνήθως ως είσοδο στον Generator δίνεται ένα τυχαίο διάνυσμα λευκού θορύβου (γκαουσιανού ως επί το πλείστο), ενώ για την εκπαίδευση του μοντέλου δεν απαιτείται επισημασμένο (annotated) σύνολο δεδομένων. Τα μοντέλα που ανήκουν σε αυτήν την κατηγορία λέγεται ότι παράγουν εικόνα από θόρυβο (noise-to-image generation).

Όπως έχει αναφερθεί, από τα τέσσερα (4) μοντέλα που αναπτύχθηκαν το ένα ανήκει στην κατηγορία αυτών που παράγουν εικόνα από θόρυβο. Ωστόσο, θα ξεκινήσουμε αυτήν την ενότητα με την περιγραφή του πρώτου μοντέλου GAN αυτής της κατηγορίας που χρησιμοποίησε αποκλειστικά συνελικτικές στρώσεις στην αρχιτεκτονική του Generator, του Deep Convolutional GAN (DCGAN). Κατόπιν, θα προχωρήσουμε σε πιο εξελιγμένα μοντέλα όπως είναι το Progressive Growing GAN (PGGAN), ενώ στο τέλος της ενότητας θα μιλήσουμε για το πιο εξελιγμένο GAN που έχει δημοσιευθεί έως τη συγγραφή της παρούσας (Ιούλιος 2021), το StyleGAN.

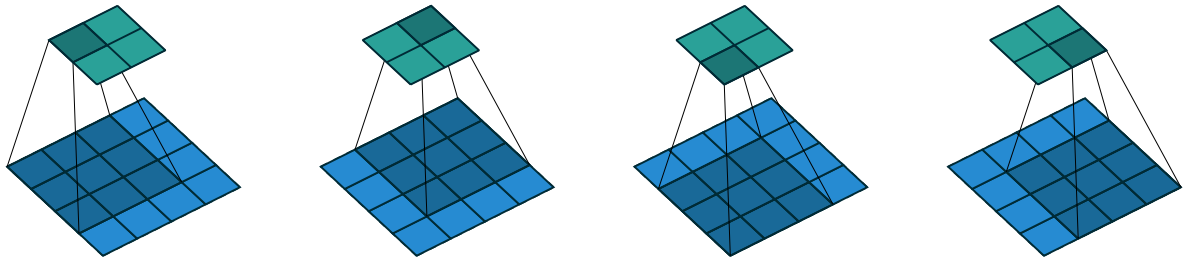
4.1.1 Παραγωγή με Συνελικτικά Δίκτυα: DCGAN

Στα αρχικά μοντέλα GANs, τόσο ο Generator όσο και ο Discriminator υλοποιούνταν με ΤΝΔ δύο ή περισσότερων κρυφών στρώσεων. Το 2015 έγινε η πρώτη αποτελεσματική προσπάθεια σχεδίασης GANs με συνελικτικά νευρωνικά δίκτυα (CNNs) από την Emily Denton et al. στο μοντέλο τους Laplacian Pyramid of Generative Adversarial Networks (LAPGAN) [38]. Ωστόσο το μοντέλο τους ήταν εξαιρετικά πολύπλοκο και υπολογιστικά «βαρύ» και αν και αρκετά αποτελεσματικό (το πιο αποτελεσματικό GAN του 2015, με το 40% των παραχθέντων εικόνων να χαρακτηρίζονται ως αληθινές από ανθρώπους) δεν αποτέλεσε τη βάση των σύγχρονων μοντέλων GANs με CNNs.

Αντίθετα, το 2016 ο Radford et al. παρουσίασαν ένα πολύ αποδοτικό και ταυτόχρονα «απλό» και αποτελεσματικό μοντέλο GAN, αποτελούμενο αποκλειστικά από συνελικτικές στρώσεις και στρώσεις κανονικοποίησης. Το μοντέλο τους, Deep Convolutional GAN (DCGAN) [44], σημάδεψε μια από τις πιο σημαντικές αρχικές καινοτομίες στη σχεδίαση GANs, λόγω της απλής αρχιτεκτονικής του και των εντυπωσιακών αποτελεσμάτων του. Ίσως η πιο βασική τεχνική που χρησιμοποίησαν στη σχεδίαση του μοντέλου τους και η οποία τους επέτρεψε να σταθεροποιήσουν την εκπαίδευση και να λύσουν το πρόβλημα της εξαφάνισης των παραγώγων λόγω των πολλών συνελικτικών στρώσεων, ήταν η χρήση Κανονικοποίησης Ομάδας (Batch Normalization). Η κανονικόποιηση αυτή καθώς και κάποια βασικά στοιχεία για τις (ορθές και ανάστροφες) συνελικτικές στρώσεις που αποτέλεσαν δομικά στοιχεία του DCGAN παρουσιάζονται παρακάτω, ενώ στο τέλος της ενότητας παρατίθεται η αρχιτεκτονική των δικτύων του DCGAN καθώς και μετρικές και αποτελέσματα από την εφαρμογή του.

Συνελικτικές Στρώσεις και Διαμοιρασμός Παραμέτρων

Σε αντίθεση με τα κανονικά feed-forward ΤΝΔ των οποίων οι νευρώνες είναι τοποθετημένοι σε επίπεδες πλήρως-συνδεδεμένες στρώσεις, στα CNNs οι στρώσεις είναι τοποθετημένες σε ένα τρισδιάστατο πλέγμα (πλάτος × ύψος × βάθος). Οι συνελίξεις υλοποιούνται με εκπαιδεύσιμα φίλτρα μικρού δεκτικού πεδίου (receptive field), τα οποία ολισθαίνουν κατά πλάτος και ύψος σε όλη την προηγούμενη στρώση, ενώ το καθένα έχει βάθος όσο όλο το βάθος της προηγούμενης στρώσης. Σε κάθε βήμα, καθώς το συνελικτικό φίλτρο ολισθαίνει στην είσοδό του, βγάζει ως έξοδο το εσωτερικό γινόμενο μεταξύ της εισόδου και των παραμέτρων του. Αυτό έχει ως αποτέλεσμα την παραγωγή ενός δισδιάστατου χάρτη ενεργοποίησης (activation map), κάτι που απεικονίζεται στο σχήμα που ακολουθεί.



Σχήμα 39: Ένα συνελικτικό φίλτρο 3×3 καθώς ολισθαίνει σε είσοδο 4×4. Σε κάθε βήμα το φίλτρο κινείται μία θέση (αριστερά προς δεξιά, επάνω προς κάτω) με αποτέλεσμα να προκύπτει έξοδος 2×2 μετά από 4 βήματα. Οι παράμετροι του φίλτρου είναι σταθεροί σε όλα τα βήματα που απαιτούνται για συνέλιξη με την είσοδο, ενώ το βάθος εισόδου παραλείπεται για λόγους απλότητας.
Πηγή: «A guide to convolution arithmetic for deep learning», Dumoulin et al., 2016 [49]

Πολλοί τέτοιοι χάρτες ενεργοποίησης στοιβάζονται ο ένας πάνω στον άλλο για να δημιουργήσουν την (τρισδιάστατη) έξοδο της τρέχουσας στρώσης (όταν η είσοδος είναι η προηγούμενη στρώση). Το νέο βάθος ισούται με τον αριθμό των χαρτών ενεργοποίησης ή, ισοδύναμα, με τον αριθμό των διαφορετικών συνελικτικών φίλτρων της στρώσης. Σημαντικό στοιχείο της αρχιτεκτονικής των συνελικτικών δικτύων είναι ο *διαμοιρασμός των παραμέτρων* για να προκύψει ο κάθε χάρτης ενεργοποίησης. Αυτός έγκειται στο γεγονός ότι καθώς το φίλτρο ολισθαίνει στην είσοδό του οι παράμετροι του παραμένουν σταθερές. Έτσι κάθε χάρτης ενεργοποίησης προκύπτει από ένα σύνολο παραμέτρων ίσο σε αριθμό με τις διαστάσεις του φίλτρου από το οποίο προέκυψε ($\text{πλάτος}_{\text{φίλτρου}} \times \text{ύψος}_{\text{φίλτρου}} \times \text{βάθος}_{\text{προηγ. στρώσης}}$). Γενικά, ο διαμοιρασμός παραμέτρων μειώνει δραστικά

τον αριθμό των εκπαιδεύσιμων παραμέτρων στα CNNs, κάτι που αφενός βοηθάει με την υπερ-προσαρμογή (over-fitting) και αφετέρου επιταχύνει και σταθεροποιεί την εκπαίδευση σε σχέση με τα παραδοσιακά βαθιά ΤΝΔ.

Κανονικοποίηση Ομάδας Batch Normalization

Τα GANs συνήθως χρειάζονται πολύ χρόνο για να εκπαιδευτούν, ειδικά όταν θέλουμε να παράξουμε ρεαλιστικές και υψηλής ανάλυσης εικόνες. Ωστόσο, η εκπαίδευση των GANs είναι συχνά ασταθής, καθώς αυτά είναι πολύ πιο σύνθετα μοντέλα σε σύγκριση με διακριτικά όπως οι ταξινομητές εικόνων. Έτσι, κάθε κόλπο που επιταχύνει και σταθεροποιεί την εκπαίδευση είναι καίριας σημασίας για αυτά τα μοντέλα. Ένα τέτοιο «κόλπο» είναι και η Κανονικοποίηση Ομάδας (Batch Normalization) που προτάθηκε από τον Ioffe et al. στο άρθρο τους «*Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*» [42]. Η Κανονικοποίηση Ομάδας, η οποία έχει αποδειχθεί ιδιαίτερα αποτελεσματική στη σταθεροποίηση της εκπαίδευσης, αποτέλεσε κυρίαρχο δομικό στοιχείο της αρχιτεκτονικής και αποτελεσματικότητας του DCGAN.

Το πρόβλημα που επιλύει η χρήση της Κανονικοποίσης Ομάδας είναι αυτό που ονομάζεται εσωτερική συμμεταβλητή μετατόπιση (internal covariate shift). Συμμεταβλητή μετατόπιση ονομάζεται το φαινόμενο μετατόπισης της μορφής της συνάρτησης κόστους (στο χώρο των παραμέτρων ενός ΤΝΔ) εξαιτίας των διαφορετικών κατανομών (ως προς τα δείγματα της εκάστοτε ομάδας) των εισόδων ενός μοντέλου και συνήθως προέρχεται από μη-σωστά επεξεργασμένο σύνολο δεδομένων εκπαίδευσης. Αυτό που παρατήρησαν, ωστόσο, οι συγγραφείς του [42] είναι ότι και με επαρκώς κανονικοποιημένο σύνολο δεδομένων, βαθιά ΤΝΔ παρουσίαζαν ασταθή εκπαίδευση και πολύ αργή σύγκλιση. Αυτό οφείλονταν στο ότι καθώς μεταβάλλονταν τα βάρη του δικτύου, μεταβάλλονταν και οι κατανομές των εξόδων της κάθε στρώσης, με αποτέλεσμα να εμφανίζεται παρόμοιο πρόβλημα με το αρχικό αλλά για τις συναρτήσεις κόστους του κάθε νευρώνα, κάτι που ονόμασαν *internal covariate shift*. Για την επίλυσή του, πρότειναν την κανονικοποίηση της εξόδου του κάθε νευρώνα ενός ΤΝΔ ώστε όλες οι έξοδοι της εκάστοτε ομάδας για τον συγκεκριμένου νευρώνα να έχουν αρχικά μηδενική μέση τιμή και μοναδιαία διακύμανση και κατόπιν αυτές να πηγαίνουν στις εκπαιδεύσιμες παραμέτρους γ και β αντίστοιχα.

Με βάση τα παραπάνω, η Κανονικοποίηση Ομάδας στα ΤΝΔ γίνεται ως εξής:
Έστω, $z_i^{[l]}$ η έξοδος του i-οστού νευρώνα της l-οστής στρώσης ενός ΤΝΔ. Επίσης, έστω ότι στην είσοδο του νευρωνικού (και άρα κατ΄ επέκταση και σε κάθε στρώση αυτού) δίνονται

ομάδες B δειγμάτων, τότε η έξοδος του νευρώνα μετά την Κανονικοποίηση Ομάδας θα είναι [42]:

$$\mu_B \longleftarrow \frac{1}{B}\sum_{b=1}^{B}\left(z_i^{[l]}\right)_b \tag{4.1}$$

$$\sigma_B^2 \longleftarrow \frac{1}{B}\sum_{b=1}^{B}\left[\left(z_i^{[l]}\right)_b - \mu_B\right]^2 \tag{4.2}$$

$$\left(\hat{z}_i^{[l]}\right)_b \longleftarrow \frac{\left(z_i^{[l]}\right)_b - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4.3}$$

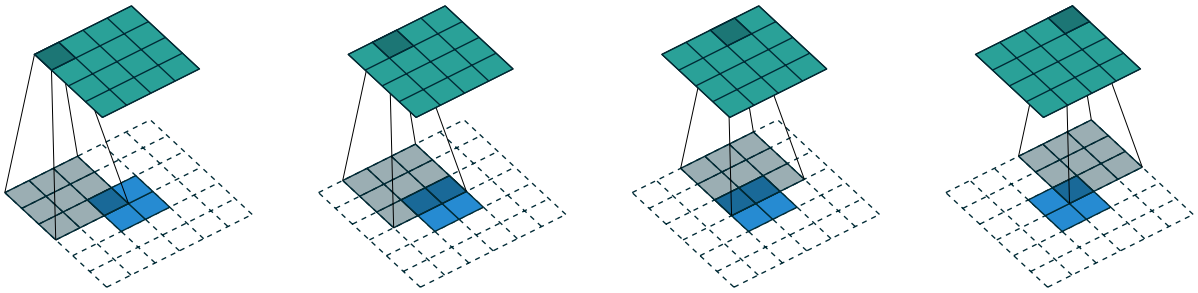
$$\left(y_i^{[l]}\right)_b \longleftarrow \gamma\left(\hat{z}_i^{[l]}\right)_b + \beta, \quad b = 1, ..., B \tag{4.4}$$

Στην περίπτωση των CNNs, η Κανονικοποίηση ομάδας γίνεται ως προς μία ομάδα καναλιών, δηλαδή για ένα συγκεκριμένο βάθος στην έξοδο μιας συνελικτικές στρώσης. Επίσης, κατά τη διάρκεια της φάσης δοκιμής (evaluation phase) του μοντέλου, χρησιμοποιούνται στατιστικά των μέσων τιμών και των διακυμάνσεων που κρατήθηκαν από τη φάση εκπαίδευσης. Τέλος, για λόγους πληρότητας να αναφέρουμε πως η κανονικοποίηση ομάδας είναι λιγότερο αποτελεσματική για μειούμενο αριθμό δειγμάτων στην ομάδα, καθώς και για περιπτώσεις όπου τα δείγματα αυτά παρουσιάζουν αρκετά διαφορετικές στατιστικές (όπως π.χ. στην παραγωγή εικόνων με ποικιλομορφία).

Ανάστροφες Συνελικτικές Στρώσεις (Transposed Convolutions)

Ακολούθως, θα προβούμε σε μία σύντομη περιγραφή των ανάστροφων συνελικτικών στρώσεων (transposed convolutional layers), οι οποίες χρησιμοποιούνται κατά κανόνα στους Generators των GANs. Οι ανάστροφες συνελίξεις είναι ουσιαστικά μία τεχνική για αύξηση του πλάτους και μήκους μιας στρώσης (upsampling) μέσω φίλτρων εκπαιδεύσιμων παραμέτρων. Αναλυτικά, για είσοδο πλάτους και μήκους 2, δηλαδή 2×2, και το συνελικτικό φίλτρο πολλαπλασιάζεται με κάθε στοιχείο της εισόδου (πολλαπλασιασμός αριθμού με διάνυσμα - όχι εσωτερικό γινόμενο) και το αποτέλεσμα αποθηκεύεται στον χάρτη ενεργοποίησης της εξόδου με βάση την τεχνική overlap-and-add (δηλαδή στα σημεία της επικάλυψης γίνεται πρόσθεση της νέας με την υπάρχουσα τιμή). Σύμφωνα με το [49], αυτό είναι ισοδύναμο με εφαρμογή περιθωρίου padding στην είσοδο και πραγματοποίησης της κανονικής συνέλιξης, κάτι που απεικονίζεται στο σχήμα 40 παρακάτω:

Οι ανάστροφες συνελικτικές στρώσεις χρησιμοποιούνται ως βασικά δομικά στοιχεία στον Generator του DCGAN με στόχο τη σταδιακή αύξηση της ανάλυσης των στρώσεων



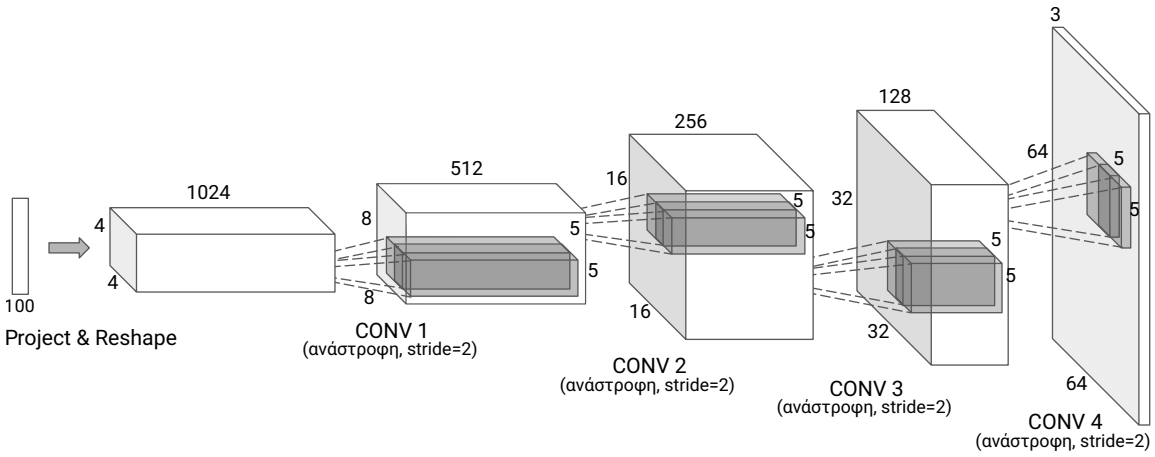
Σχήμα 40: Το ανάστροφο της συνέλιξης 3×3: ένα ανάστροφο συνελικτικό φίλτρο 3×3 καθώς ολισθαίνει σε είσοδο 2×2 (με padding 2). Σε κάθε βήμα το φίλτρο κινείται μία θέση (αριστερά προς δεξιά, επάνω προς κάτω) με αποτέλεσμα να προκύπτει έξοδος 4×4 μετά από 16 βήματα. Οι παράμετροι του φίλτρου είναι σταθεροί σε όλα τα βήματα που απαιτούνται για ανάστροφη συνέλιξη με την είσοδο, ενώ το βάθος εισόδου παραλείπεται για λόγους απλότητας.
Πηγή: «A guide to convolution arithmetic for deep learning», Dumoulin et al., 2016 [49]

έως την τελική ανάλυση που θέλουμε να έχουν οι εικόνες στην έξοδό του. Για λόγους πληρότητας, σκόπιμο είναι να αναφέρουμε πως οι ανάστροφες συνελίξεις λόγω του overlap που υπάρχει σε κάποια σημεία της εξόδου, εμφανίζουν ένα εφέ σκακιέρας (checkerboard pattern) [58]. Αυτό έχει αναγκάσει αρκετούς ερευνητές στην αναζήτηση και εφαρμογή διαφορετικών τεχνικών upsampling, όπως η χρήση ντερμενιστικού upsampling (π.χ. με –bilinear, bicubic, nearest neighboors˝ παρεμβολή) ακολουθούμενο από κανονικές συνελικτικές στρώσεις (βλ. PGGAN, υποενότητα 4.1.2).

Αρχιτεκτονική του μοντέλου DCGAN

Το τέλος της ενότητας το αφιερώνουμε στην παράθεση της αρχιτεκτονικής των δικτύων του DCGAN και ορισμένων από τα αρχικά αποτελέσματα από την εφαρμογή του. Έτσι, αρχικά παραθέτουμε το δίκτυο του Generator ως αντιπροσωπευτική απεικόνιση όσο και πιο τεχνικά σε μορφή πίνακα. Όπως φαίνεται αυτός αποτελείται από ανάστροφες συνελικτικές στρώσεις, στρώσεις κανονικοποίησης ομάδας και ανορθωμένες γραμμικές μονάδες (rectified linear units - ReLU)[1], ενώ δεν συμπεριλαμβάνονται *καθόλου πλήρως-συνδεδεμένες ή pooling στρώσεις* κάτι πρωτοποριακό τη στιγμή της παρουσίασής του DCGAN. Έτσι, σχηματικά ο Generator του DCGAN έχει ως εξής:

[1]Οι ανορθωμένες γραμμικές μονάδες (rectified linear units - ReLU) είναι συναρτήσεις ενεργοποίησης οι οποίες έχουν τη μορφή: $f(x) = max(0, x)$ (ReLU), ή $f_a(x) = max(-ax, x)$ με a μικρή αρνητική σταθερά όπως -0.1 (Leaky ReLU).



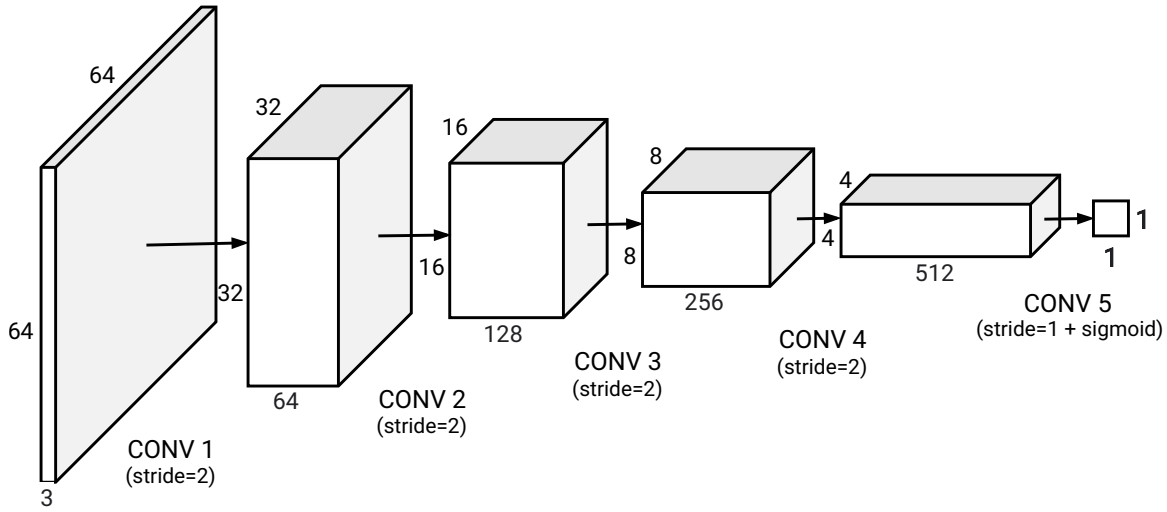
Σχήμα 41: Αρχιτεκτονική του Generator του DCGAN κατά την εφαρμογή του μοντέλου στο σύνολο δεδομένων LSUN [47].

Πηγή: Ανακατασκευή από «Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks», Radford et al., 2015 [44]

Όπως φαίνεται στο σχήμα 41, το δίκτυο του Generator αποτελείται από πέντε (5) ανάστροφες συνελικτικές στρώσεις (τη στρώση προβολής (projection layer) και τις στρώσεις CONV 1-4), όλες με μέγεθος φίλτρου 5×5 και stride 2. Στο τέλος βγαίνει μια εικόνα έγχρωμη (3 κανάλια) και μεγέθους 64×64 εικονοστοιχεία.

Ακολουθώντας συμμετρική σχεδίαση, το δίκτυο του Discriminator του DCGAN αποτελείται από συνελικτικές στρώσεις με stride=2, στρώσεις κανονικοποίησης ομάδας και ανορθωμένες γραμμικές μονάδες (Leaky ReLU, $a = -0.2$). Η αρχιτεκτονική του Discriminator απεικονίζεται στο σχήμα 42 παρακάτω. Όπως φαίνεται στο σχήμα αυτό, το δίκτυο του Discriminator αποτελείται από πέντε (5) συνελικτικές στρώσεις (τις στρώσεις CONV 1-4), όλες με μέγεθος φίλτρου 5×5 και stride 2 και τη στρώση εξόδου με stride 1) και σιγμοειδή συνάρτηση εξόδου. Στο τέλος βγαίνει μια πιθανότητα από 0 έως 1 εκπαιδεύεται να αντιστοιχεί στο βαθμό ρεαλισμού της εισόδου.

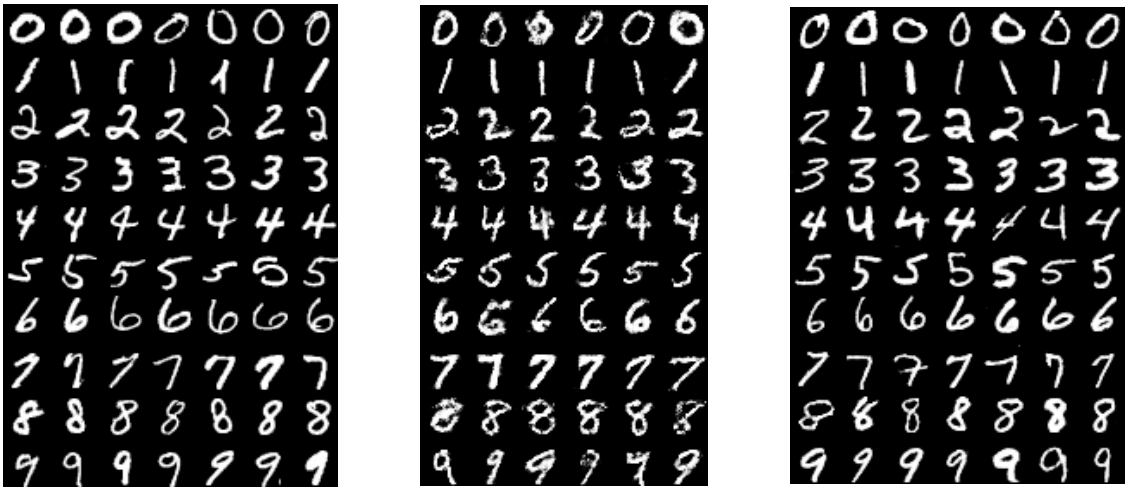
Όπως αναφέρθηκε, το DCGAN είχε μια αρκετά πρωτοπόρα αρχιτεκτονική και εάν και δεν πέτυχε επαναστατικά αποτελέσματα ήταν το πρώτο μοντέλο GAN χωρίς πλήρως-συνδεδεμένες στρώσεις το οποίο κατάφερε να εκπαιδευθεί ευσταθώς για χιλιάδες επαναλήψεις. Παρακάτω, παραθέτουμε από το [44] παραγωγές του DCGAN το οποίο εκπαιδεύτηκε στο σύνολο δεδομένων χειρόγραφων ψηφίων του MNIST, οι οποίες συγκρίνονται με το αρχικό GAN (το οποίο χρησιμοποιούσε πλήρως-συνδεδεμένες στρώσεις) καθώς και με τις πραγματικές εικόνες του συνόλου εκπαίδευσης. Όπως φαίνεται στο σχήμα 43, ο



Σχήμα 42: Αρχιτεκτονική του Discriminator του DCGAN.

Πηγή: Ανακατασκευή από «Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks», Radford et al., 2015 [44]

Generator του DCGAN δίνει αρκετά ρεαλιστικά δείγματα - τα περισσότερα δυσδιάκριτα από αυτά του συνόλου εκπαίδευσης.



(α) Πραγματικές εικόνες (β) Αρχικό GAN (γ) DCGAN

Σχήμα 43: Σύγκριση παραγόμενων εικόνων του DCGAN (δεξιά) και του αρχικού GAN (κέντρο) όταν αμφότερα έχουν εκπαιδευτεί με το σύνολο δεδομένων χειρόγραφων ψηφίων του MNIST (αριστερά).

Πηγή: «Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks», Radford et al., 2015 [44]

4.1.2 Σταδιακή Παραγωγή: PGGAN

Περνάμε τώρα στην παρουσίαση του μοντέλου Progressively Growing GAN (PGGAN) το οποίο παρουσιάστηκε από τον Karras et al. στο άρθρο τους «*Progressive Growing of GANs for Improved Quality, Stability, and Variation*» [74]. Το μοντέλο αυτό ήταν το πρώτο μοντέλο που παρήγαγε πολύ υψηλής ποιότητας και ποικιλομορφίας εικόνες υψηλής ανάλυσης (1024×1024), αποτελώντας το πρώτο μοντέλο της οικογένειας μοντέλων StyleGAN (κάποια μοντέλα από την οποία αναλύουμε στις επόμενες υποενότητες). Σε ότι ακολουθεί θα αναφέρουμε σημαντικές καινοτομίες του μοντέλου, όπως η σταδιακή αύξηση της ανάλυσης και η στρώση τυπικής απόκλισης ομάδας, θα παραθέσουμε στοιχεία της αρχιτεκτονικής των δικτύων και στο τέλος θα δώσουμε κάποια από τα αποτελέσματα εφαρμογής του PGGAN στην πράξη.

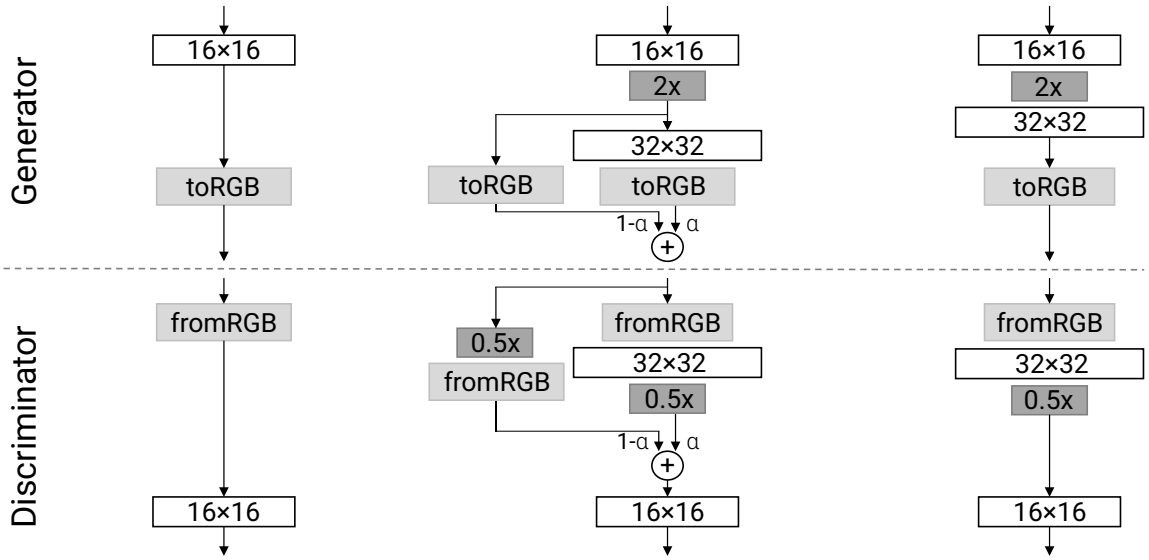
Σταδιακή Αύξηση Ανάλυσης (Progressing Growing)

Το κύριο στοιχείο σχεδίασης και εκπαίδευσης του PGGAN είναι αυτό της σταδιακής αύξησης της ανάλυσης των δικτύων με σκοπό τη σταθερότερη εκπαίδευση. Στην πράξη, οι συγγραφείς του [74] είδαν ότι η εκπαίδευση του GAN σε στάδια οδηγεί σε πολύ πιο γρήγορη και σταθερή εκπαίδευση όταν οι τελικές αναλύσεις των εικόνων που επιθυμούμε να παραχθούν από τον Generator ή να διακριθούν από τον Discriminator είναι μεγάλες (δηλ. μεγαλύτερες από 256×256).

Συνοπτικά, αυτό που γίνεται είναι η αρχικοποίηση των δικτύων για παραγωγή εικόνων πολύ μικρής ανάλυσης, π.χ. 4×4 ή 8×8. Αφότου η εκπαίδευση του μοντέλου σταθεροποιηθεί στην ανάλυση αυτή (κατί σχετικά εύκολο για το μέγεθος των δικτύων) και στα δύο δίκτυα προστίθενται νέες στρώσεις είτε για παραγωγή ή για διάκριση εικόνων διπλάσιας ανάλυσης (αντίστοιχα 8×8 ή 16×16). Κατόπιν, μετά από κάποια epochs και αφού η εκπαίδευση των μοντέλων έχει εκ νέου σταθεροποιηθεί[2] γίνεται η ίδια διαδικασία αύξησης της ανάλυσης των δικτύων, κάτι που συνεχίζεται έως ότου η ανάλυση φτάσει στην επιθυμητή των εικόνων εξόδου του Generator. Ακολουθώντας, επομένως, αυτή τη λογική της εκπαίδευσης του GAN σε στάδια, οι συγγραφείς ήταν σε θέση να εκπαιδεύσουν επιτυχώς μοντέλα που παράγουν υψηλής ανάλυσης και ρεαλισμού εικόνες - με σημαντικά καλύτερα αποτελέσματα σε σύγκριση με προηγούμενα μοντέλα.

[2] Η έννοια της σταθεροποίησης της εκπαίδευσης ενός GAN και ειδικά ενός αρκετά πολύπλοκου όπως το PGGAN δεν είναι σαφώς ορισμένη. Μετά από πληθώρα πειραμάτων, οι συγγραφείς του [74] κατέληξαν σε ένα σύνολο breakpoints με βάση τις συναρτήσεις κόστους των δικτύων, τα οποία και δημοσιοποίησαν.

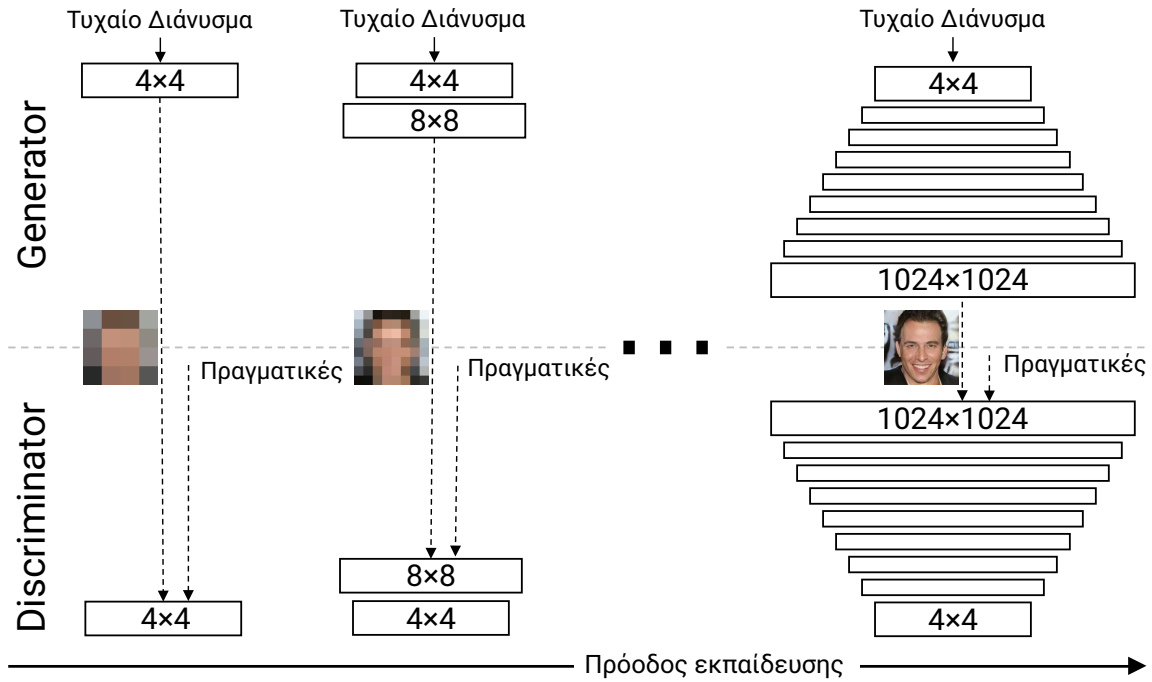
Αν και η αύξηση της ανάλυσης των δικτύων μέσω της προσθήκης νέων στρώσεων ακούγεται απλή, στην πράξη οι συγγραφείς κατέληξαν ότι αυτό θα πρέπει να γίνει με ομαλό τρόπο: όταν εισάγονται νέες στρώσεις οι συγγραφείς εισάγουν ένα παράλληλο μονοπάτι που τις παρακάμπτει. Σε κάθε ένα από τα δύο μονοπάτια ανατίθενται συντελεστές βαρύτητας a (για τις νέες στρώσεις) και $(1-a)$ (για την «παράκαμψη»), με την παράμετρο μίξης a να ξεκινάει από 0 και σταδιακά να πηγαίνει έως το 1, οπότε στην ουσία καταργείται το πρόσθετο παράλληλο μονοπάτι. Σχηματικά αυτό φαίνεται στη γραφική απεικόνιση που ακολουθεί. Στο σχήμα αυτό φαίνεται η ομαλή μετάβαση από την ανάλυση 16×16 στην ανάλυση 32×32 τόσο για τον Generator όσο και για τον Discriminator. Η μετάβαση γίνεται μέσω ενός μονοπατιού παράκαμψης των νέων στρώσεων και την αντίστοιχη παράμετρο μίξης a. Η παράμετρος αυτή ξεκινάει από 0, οπότε απλώς προστίθεται μια upsampling στρώση στον Generator και η αντίστοιχη downsampling στον Discriminator της προηγούμενης ανάλυσης, έως 1, οπότε τα δίκτυα έχουν μεγαλώσει και έχουν ένα μπλοκ νέων συνελικτικών στρώσεων το καθένα.



Σχήμα 44: Απεικόνιση του τρόπου σταδιακής αύξησης των δικτύων του PGGAN.
Πηγή: Ανακατασκευή από «Progressive Growing of GANs for Improved Quality, Stability, and Variation», Karras et al., 2017 [74]

Οι πραγματικές εικόνες δίνονται στον Discriminator, για την εκπαίδευσή του, στην εκάστοτε ανάλυση εικόνων που δέχεται. Συνολικά, στο σχήμα 45, παρακάτω, φαίνεται το πως σταδιακά εκπαιδεύεται το δίκτυο για να παράγει εικόνες υψηλής ανάλυσης. Εκεί, απεικονίζεται η ομαλή μετάβαση από την ανάλυση 16×16 στην ανάλυση 32×32 τόσο για τον Generator όσο και για τον Discriminator. Η μετάβαση γίνεται μέσω ενός μονοπατιού

παράκαμψης των νέων στρώσεων και την αντίστοιχη παράμετρο μίξης a. Η παράμετρος αυτή ξεκινάει από 0, οπότε απλώς προστίθεται μια upsampling στρώση στον Generator και η αντίστοιχη downsampling στον Discriminator της προηγούμενης ανάλυσης, έως 1, οπότε τα δίκτυα έχουν μεγαλώσει και έχουν ένα μπλοκ νέων συνελικτικών στρώσεων το καθένα.

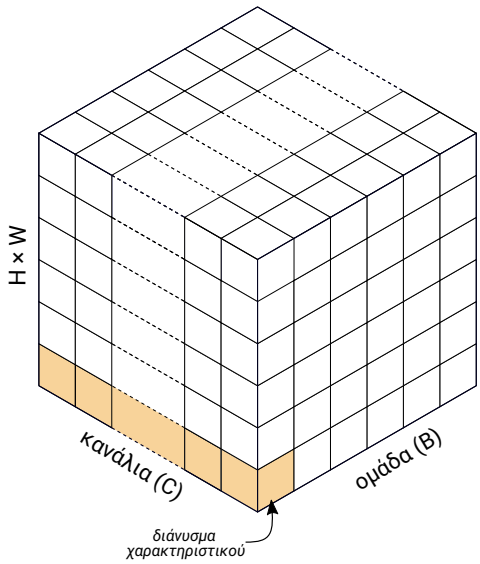


Σχήμα 45: Συνολική απεικόνιση σταδιακής αύξησης των δικτύων του PGGAN.
Πηγή: Ανακατασκευή από «Progressive Growing of GANs for Improved Quality, Stability, and Variation», Karras et al., 2017 [74]

Κανονικοποίηση Εικονοστοιχείων (Pixel Normalization)

Παράλληλα, μαζί με την καινοτομία στη σταδιακή αύξηση της ανάλυσης, οι συγγραφείς του [74] δοκίμασαν και μια νέα μορφή κανονικοποίησης των εξόδων των συνελικτικών στρώσεων του Generator. Αυτή, την οποία ονομάσαν Κανονικοποίηση Διανύσματος Χαρακτηριστικού ανά Εικονοστοιχείο (Pixel-wise Feature Vector Normalization) ή Κανονικοποίηση Εικονοστοιχείων (Pixel Normalization) όπως αλλιώς ονομάζεται, ακολουθεί τις συνελικτικές στρώσεις του Generator του PGGAN προκειμένου «*να αποτραπεί το σενάριο όπου τα μεγέθη στον Generator και στον Discriminator στρέφονται εκτός ελέγχου ως αποτέλεσμα της αντιπαραθετικής εκπαίδευσης.*».

Αναλυτικά, για κάθε δείγμα της ομάδας (batch), και γιά κάθε θέση κατά πλάτος (x) και



Σχήμα 46: Σχηματική απεικόνιση της Κανονικοποίησης Εικονοστοιχείων.
Πηγή: Ανακατασκευή από «PowerNorm: Rethinking Batch Normalization in Transformers», Yeh et al., 2020 [115]

ύψος (y) της εξόδου της προηγούμενης συνελικτικής στρώσης κανονικοποιούνται όλα τα στοιχεία του διανύσματος που αποτελείται από την τιμή όλων των χαρτών ενεργοποίησης στη συγκεκριμένη θέση πλάτους και ύψους ως προς την ευκλείδεια νόρμα αυτού. Δηλαδή, αυτό που γίνεται είναι το εξής:

$$\left\|\vec{a_{x,y}}\right\| \longleftarrow \sqrt{\frac{1}{C}\sum_{c=1}^{C}\left(\vec{a_{x,y}}[c]\right)^2} \tag{4.5}$$

$$\vec{a_{x,y}} \longleftarrow \frac{\vec{a_{x,y}}}{\left\|\vec{a_{x,y}}\right\| + \epsilon}, \quad (x,y) = (1,1),\ldots,(W,H) \tag{4.6}$$

όπου ϵ σταθερά της τάξης 10^{-8} (για αριθμητική σταθερότητα), C ο αριθμός των καναλιών ή χαρτών ενεργοποίησης και (x, y) η θέση του εκάστοτε εικονοστοιχείου ή διανύσματος χαρακτηριστικών όπως τονίζεται στο σχήμα 46 δεξιά. Σύμφωνα με τους συγγραφείς του [74] είναι «*εντυπωσιακό ότι αυτός ο βαρύς περιορισμός δεν φαίνεται να βλάπτει τον Generator με κανέναν τρόπο, και μάλιστα στα περισσότερα σύνολα δεδομένων δεν αλλάζει πολύ τα αποτελέσματα, αλλά αποτρέπει την κλιμάκωση των μεγεθών σήματος πολύ αποτελεσματικά όταν χρειάζεται*».

Τυπική Απόκλιση Ομάδας (Batch Standard Deviation)

Στην προσπάθεια τους για αύξηση της ποικιλομορφίας των παραγόμενων δειγμάτων, οι συγγραφείς του PGGAN πρότειναν τη χρήση μιας νέας στρώσης στον Discriminator,

που μετράει την τυπική απόκλιση των δειγμάτων της ομάδας Batch Standard Deviation. Συγκεκριμένα, προς το τέλος του Discriminator προστίθεται ένας χάρτης ενεργοποίησης στην έξοδο κάποιας συνελικτικής στρώσης στον οποίο αποθηκεύονται στατιστικά μεταξύ των δειγμάτων της ομάδας. Τα στατιστικά εξάγονται συγκρίνοντας τις εξόδους της προηγούμενης συνελικτικής στρώσης για κάθε δείγμα της ομάδας.

Επεξηγηματικά, πρώτα υπολογίζεται η τυπική απόκλιση για κάθε διάνυσμα χαρακτηριστικό σε κάθε θέση (x, y) ως προς την ομάδα, κάτι που αποτελεί εκτίμηση της πραγματικής τυπικής απόκλισης μεταξύ των εικόνων (πραγματικών ή τεχνητών αντίστοιχα). Έπειτα, υπολογίζεται ο μέσος όρος αυτών των εκτιμήσεων για όλες τα διανύσματα χαρακτηριστικών και όλες τις χωρικές θέσεις ώστε να προκύψει μία μόνο τιμή. Κατόπιν, αυτή η τιμή αντιγράφεται σε όλες τις θέσεις ενός νέου χάρτη ενεργοποίησης διαστάσεων ίδιων με αυτών των υπολοίπων χαρτών. Ο χάρτης ενώνεται με τους υπόλοιπους απλώς προσθέτοντας τον σε αυτούς.

Αυτός η στρώση θα μπορούσε να εισαχθεί οπουδήποτε στον Discriminator, αλλά οι συγγραφείς μετά από πειράματα κατέληξαν ότι το καλύτερο είναι αυτή να εισαχθεί προς το τέλος του Discriminator, κάτι που φαίνεται και στην αρχιτεκτονική του PGGAN που παρουσιάζεται στην επόμενη παράγραφο. Τονίζεται στο σημείο αυτό ότι οι τιμές της τυπικής απόκλισης είναι σταθερές (μη-παραγωγίσιμες) και απλώς χρησιμοποιούνται από τονDiscriminator για καλύτερη διάκριση των κατανομών, κάτι που έμμεσα «πιέζει» τον Generator να αυξήσει την ποικιλομορφία των δειγμάτων που παράγει αποφεύγοντας τη Συρρίκνωση Ρυθμών.

Αρχιτεκτονικής του PGGAN - Αποτελέσματα

Πριν προχωρήσουμε στην παράθεση της αρχιτεκτονικής των δικτύων του PGGAN, αξίζει να σημειωθεί ότι οι συγγραφείς δεν χρησιμοποιήσαν ανάστροφες συνελικτικές στρώσεις για να αποφύγουν το πρόβλημα σκακιέρας, όπως ειπώθηκε προηγούμενα. Έτσι, αυτό που κάνουν είναι ένα αρχικό upsampling ακολουθούμενο από συνελικτικές στρώσεις μοναδιαίου βήματος (δηλ. που δεν μειώνουν το πλάτος και μήκος των χαρτών ενεργοποίησης εισόδου), κάτι που φαίνεται και στην αρχιτεκτονική του Generator.

Παρακάτω, παραθέτουμε σε μορφή πίνακα την πλήρη αρχιτεκτονική των δικτύων του PGGAN όπως παρουσιάζονται στο [74]. Στον πίνακα 2, λοιπόν, φαίνονται τα δίκτυα πλήρως ανεπτυγμένα (χωρίς δηλαδή τα μονοπάτια παράκαμψης που χρησιμοποιούνται κατά την

αρχική φάση εκπαίδευσης). Εκεί με «Conv» συμβολίζονται οι συνελικτικές στρώσεις με το μέγεθος των συνελικτικών φίλτρων να αναγράφεται δίπλα, με «{Up,Down}sample» συμβολίζονται οι στρώσεις που κάνουν {up,down}sampling στους χάρτες ενεργοποίησης της εισόδου και με «Fully-connected» συμβολίζεται η πλήρως-συνδεδεμένη στρώση στην έξοδο του Discriminator. Τέλος, για λόγους πληρότητας μετά τον πίνακα παραθέτουμε μερικά παραγόμενα δείγματα του μοντέλου στο σύνολο δεδομένων προσώπων διασήμων υψηλής ανάλυσης, CelebA-HQ [43], στο σχήμα 47.

Σχήμα 47: Μερικές από τις καλύτερες παραγωγές του μοντέλου PGGAN το οποίο εκπαιδεύτηκε στο σύνολο δεδομένων προσώπων διασήμων υψηλής-ανάλυσης, CelebA-HQ. Φαίνεται τόσο η εξαιρετική ποιότητα όσο και η μεγάλη ποικιλομορφία των παραγόμενων δειγμάτων του Generator του PGGAN.
Πηγή: «Progressive Growing of GANs for Improved Quality, Stability, and Variation», Karras et al., 2017 [74]

4.1.3 StyleGAN

Ένα χρόνο μετά την παρουσίαση του PGGAN, δηλαδή το 2018, οι δημιουργοί του προχώρησαν στην αναβάθμισή του και στην παρουσίαση του μοντέλου StyleGAN στο άρθρο τους «*A Style-Based Generator Architecture for Generative Adversarial Networks*» [91]. Το μοντέλο είχε μικρότερο αριθμό αλλά όχι μικρότερης σημασίας καινοτομίες, οι οποίες προκύπτουν από τη θεώρηση ότι μια εικόνα συντίθεται από ένα σύνολο μεταβαλλόμενων χαρακτηριστικών (τα οποία ονομάζουν «στιλ» (styles)), τόσο για τις παραγόμενες εικόνες όσο

Generator	Ενεργ.	Σχήμα Εξόδου	Αρ. Παραμ.
Τυχαίο Διάνυσμα	–	512 × 1 × 1	–
Conv 4×4	LReLU	512 × 4 × 4	4.2M
Conv 3×3	LReLU	512 × 4 × 4	2.4M
Upsample	–	512 × 8 × 8	–
Conv 3×3	LReLU	512 × 8 × 8	2.4M
Conv 3×3	LReLU	512 × 8 × 8	2.4M
Upsample	–	512 × 16 × 16	–
Conv 3×3	LReLU	512 × 16 × 16	2.4M
Conv 3×3	LReLU	512 × 16 × 16	2.4M
Upsample	–	512 × 32 × 32	–
Conv 3×3	LReLU	512 × 32 × 32	2.4M
Conv 3×3	LReLU	512 × 32 × 32	2.4M
Upsample	–	512 × 64 × 64	–
Conv 3×3	LReLU	256 × 64 × 64	1.2M
Conv 3×3	LReLU	256 × 64 × 64	590k
Upsample	–	256 × 128 × 128	–
Conv 3×3	LReLU	128 × 128 × 128	295k
Conv 3×3	LReLU	128 × 128 × 128	148k
Upsample	–	128 × 256 × 256	–
Conv 3×3	LReLU	64 × 256 × 256	74k
Conv 3×3	LReLU	64 × 256 × 256	37k
Upsample	–	64 × 512 × 512	–
Conv 3×3	LReLU	32 × 512 × 512	18k
Conv 3×3	LReLU	32 × 512 × 512	9.2k
Upsample	–	32 × 1024 × 1024	–
Conv 3×3	LReLU	16 × 1024 × 1024	4.6k
Conv 3×3	LReLU	16 × 1024 × 1024	2.3k
Conv 1×1	γραμμική	3 × 1024 × 1024	51
Συνολικός Αριθμός Παραμέτρων			**23.1M**

Discriminator	Ενεργ.	Σχήμα Εξόδου	Αρ. Παραμ.
Εικόνα εισόδου	–	3 × 1024 × 1024	–
Conv 1×1	LReLU	16 × 1024 × 1024	64
Conv 3×3	LReLU	16 × 1024 × 1024	2.3k
Conv 3×3	LReLU	32 × 1024 × 1024	4.6k
Downsample	–	32 × 512 × 512	–
Conv 3×3	LReLU	32 × 512 × 512	9.2k
Conv 3×3	LReLU	64 × 512 × 512	18k
Downsample	–	64 × 256 × 256	–
Conv 3×3	LReLU	64 × 256 × 256	37k
Conv 3×3	LReLU	128 × 256 × 256	74k
Downsample	–	128 × 128 × 128	–
Conv 3×3	LReLU	128 × 128 × 128	148k
Conv 3×3	LReLU	256 × 128 × 128	295k
Downsample	–	256 × 64 × 64	–
Conv 3×3	LReLU	256 × 64 × 64	590k
Conv 3×3	LReLU	512 × 64 × 64	1.2M
Downsample	–	512 × 32 × 32	–
Conv 3×3	LReLU	512 × 32 × 32	2.4M
Conv 3×3	LReLU	512 × 32 × 32	2.4M
Downsample	–	512 × 16 × 16	–
Conv 3×3	LReLU	512 × 16 × 16	2.4M
Conv 3×3	LReLU	512 × 16 × 16	2.4M
Downsample	–	512 × 8 × 8	–
Conv 3×3	LReLU	512 × 8 × 8	2.4M
Conv 3×3	LReLU	512 × 8 × 8	2.4M
Downsample	–	512 × 4 × 4	–
Τυπ. απόκλ. ομάδ.	–	513 × 4 × 4	–
Conv 3×3	LReLU	512 × 4 × 4	2.4M
Conv 4×4	LReLU	512 × 1 × 1	4.2M
Fully-connected	γραμμική	1 × 1 × 1	513
Συνολικός Αριθμός Παραμέτρων			**23.1M**

Πίνακας 2: Αρχιτεκτονική του Generator (αριστερά) και του Discriminator (δεξιά) που χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου PGGAN στο σύνολο δεδομένων CelebA-HQ [43] για παραγωγή εικόνων 1024×1024.

Πηγή: Ανακατασκευή από «Progressive Growing of GANs for Improved Quality, Stability, and Variation», Karras et al., 2017 [74]

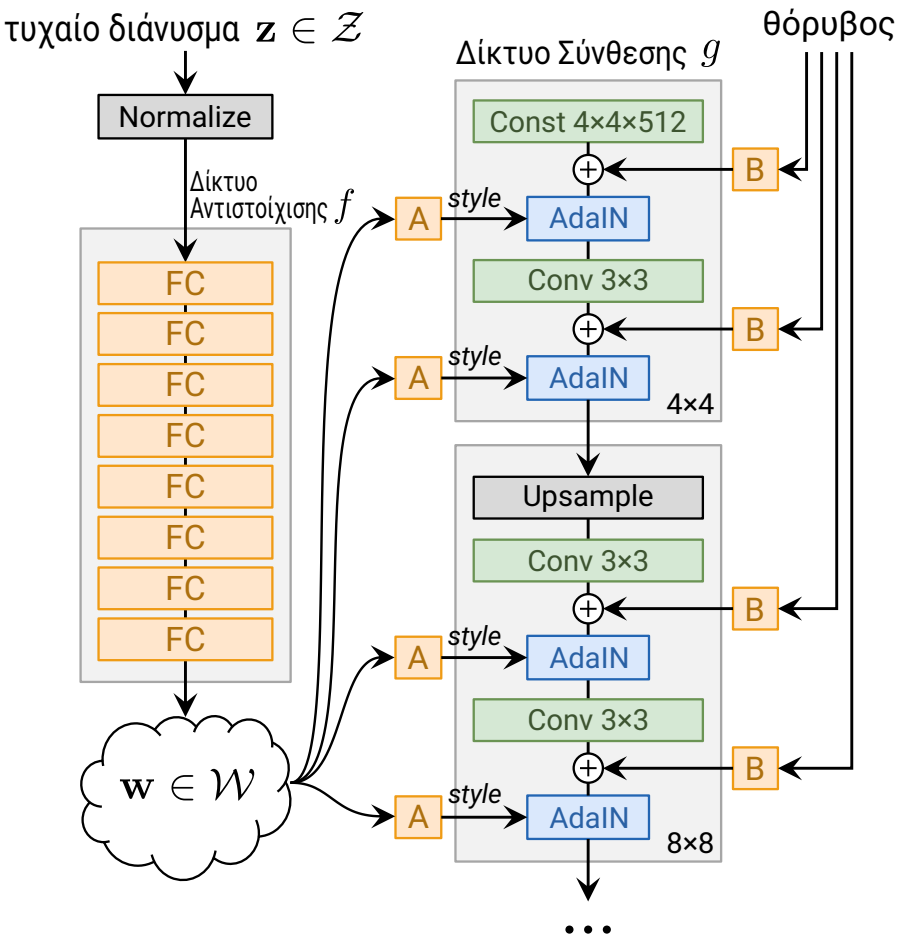
και στις πραγματικές του συνόλου εκπαίδευσης. Το StyleGAN εκτός από το ότι αποτελεί (μαζί με τις επόμενες εκδόσεις του) το καλύτερο (state-of-the-art) μοντέλο, η πρώτη του έκδοση που παρουσιάζεται στην παρούσα υποενότητα, θεωρείται ότι αποτελεί σημείο καμπής στην απόδοση των GANs λόγω της ποιότητας των παραγόμενων εικόνων.

Οι στόχοι σχεδίασης αυτού του δικτύου ξεκινώντας από το PGGAN ήταν αφενός η καλύτερη ποιότητα (fidelity) των υψηλής-ανάλυσης παραγόμενων εικόνων και αφετέρου η αύξηση της ποικιλομορφίας χωρίς να θυσιάζεται η ποιότητα. Επίσης, οι συγγραφείς ήθελαν να αυξήσουν την ελεγξιμότητα του μοντέλου τους (όπως π.χ. η προσθήκη γυαλιών σε ένα πρόσωπο χωρίς να επηρεάζονται τα υπόλοιπα χαρακτηριστικά ή η αλλαγή του χρώματος των μαλλιών), κάτι που το κατάφεραν σε σημαντικό βαθμό. Για το σκοπό αυτό, οι συγγραφείς έδωσαν περισσότερο έμφαση στην επανασχεδίαση του Generator, ενώ αντίθετα εκείνη την περίοδο, όπως οι ίδιοι αναφέρουν, οι περισσότερες προσπάθειες γίνονταν για βελτίωση της σχεδίασης και λειτουργίας του Discriminator.

Εμβαθύνοντας λίγο στις συνεισφορές του [91], αυτές συνοψίζονται σε τρεις (3) βασικές καινοτομίες: το Δίκτυο Αντιστοίχισης Θορύβου, η Προσαρμοστική Κανονικοποίηση Δείγματος και η Έγχυση Θορύβου. Οι καινοτομίες αυτές αναλύονται στις παραγράφους που ακολουθούν, ενώ στο τέλος της υποενότητας παραθέτουμε αποτελέσματα από την εκπαίδευσή του StyleGAN σε σύνολα δεδομένων εικόνων υψηλής ανάλυσης. Για αναφορά, παραθέτουμε παρακάτω το πλήρως δίκτυο του Generator, το οποίο όπως αναφέρθηκε βασίζεται στον Generator του PGGAN, δηλαδή εκπαιδεύεται σταδιακά με αναλύσεις εικόνων που ξεκινούν από 4×4 και με διαδοχικούς διπλασιασμούς φτάνουν έως 1024×1024. Όπως φαίνεται στο σχήμα αυτό (σχήμα 48), το τυχαίο διάνυσμα περνάει αρχικά από ένα δίκτυο «ξεμπερδέματος» και κατόπιν εισάγεται με εκπαιδεύσιμα βάρη στις στρώσεις κανονικοποίησης δειγμάτων. Η παραγωγή του Δικτύου Σύνθεσης ξεκινά από μια σταθερή τιμή, ενώ θόρυβος εγχέεται μετά τις συνελικτικές στρώσεις για αύξηση της ποικιλομορφίας. Κατά την εκπαίδευση του μοντέλου υπάρχουν και μονοπάτια παράκαμψης τα οποία εδώ παραλείπονται.

Δίκτυο Αντιστοίχισης Θορύβου (Noise Mapping Network)

Ο Generator του StyleGAN («*Style-based Generator*») ακολουθεί τη γενική ιδέα του αρχικού GAN: για είσοδο διάνυσμα γκαουσιανού θορύβου (από τον λανθάνοντα χώρο του Generator) εκπαιδεύεται να παράγει ρεαλιστικές εικόνες και, στην περίπτωση αυτή, υψηλής ανάλυσης. Αυτό που διακρίνει, ωστόσο, τον Style-based Generator από τον αρχικό - κάτι που



Σχήμα 48: Αρχιτεκτονικη του Generator του μοντέλου StyleGAN.

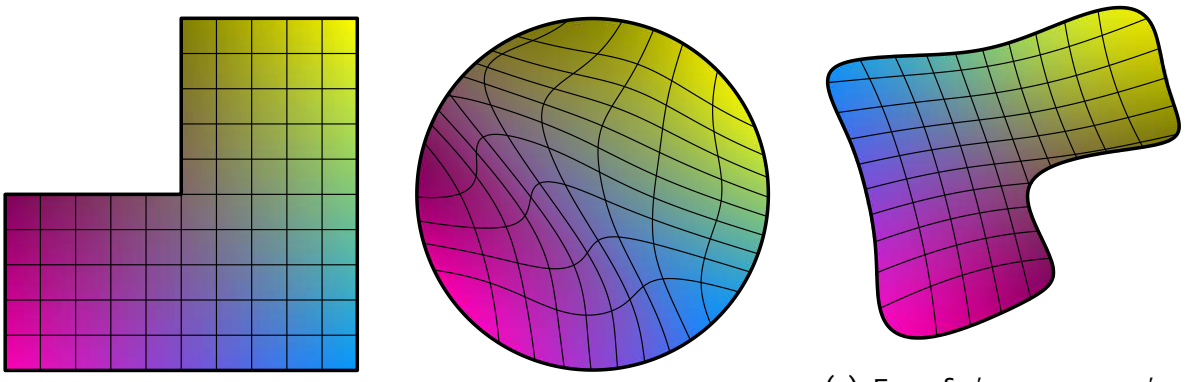
Πηγή: Ανακατασκευή από «A Style-Based Generator Architecture for Generative Adversarial Networks», Karras et al., 2018 [91]

συνιστά και την πρώτη σημαντική καινοτομία του StyleGAN - είναι ότι το διάνυσμα αυτό δεν δίνεται απευθείας στον Generator αλλά υπάρχουν οι εξής δύο διαφοροποιήσεις:

- **Δίκτυο Αντιστοίχισης Θορύβου**: το διάνυσμα θορύβου στην είσοδο του Generator περνάει αρχικά από ένα ΤΝΔ, τύπου multi-layer perceptron (MLP), αποτελούμενο από πλήρως-συνδεδεμένες στρώσεις ακολουθούμενες από συναρτήσεις ενεργοποίησης τύπου ReLU. Ο στόχος εδώ είναι ο μη-γραμμικός μετασχηματισμός της πρότερης (prior) κατανομής από την οποία δειγματοληπτείται το αρχικό διάνυσμα σε μία που οδηγεί σε πιο disentagled χώρο (βλ. σχήμα 49). Στο σχήμα αυτό απεικονίζονται τα εξής: (α) Παράδειγμα συνόλου εκπαίδευσης - εδώ λείπουν δείγματα για ορισμένους συνδυασμούς παραλλαγών (π.χ. μακριά μαλλιά αρσενικών). (β) Αυτό αναγκάζει την αντιστοίχιση από το z-χώρο σε χαρακτηριστικά εικόνας (όπως στο αρχικό GAN)

να καμπυλωθεί έτσι ώστε ο απαγορευμένος συνδυασμός να εξαφανιστεί στο z-χώρο για να αποφευχθεί η δειγματοληψία μη έγκυρων συνδυασμών. (γ) Η εκμάθηση αντιστοίχισης από το z-χώρο, μέσω του Δικτύου Αντιστοίχισης Θορύβου, στο w-χώρο είναι ικανή να «αναιρέσει» μεγάλο μέρος της παραμόρφωσης.

- **Εισαγωγή σε πολλές στρώσεις**: η έξοδος του παραπάνω δικτύου είναι ένα διάνυσμα, $\vec{w}$, (συνήθως) ίδιας διάστασης με το αρχικό, το οποίο όμως δεν δίνεται στο υπόλοιπο μέρος του Generator με τον κλασικό τρόπο (δηλ. στην πρώτη στρώση). Αντ΄ αυτού εισάγεται σε πολλά σημεία στο υπόλοιπο μέρος του Generator και συγκεκριμένα στις στρώσεις Προσαρμοστικής Κανονικοποίησης Δείγματος (βλ. επόμενη παράγραφο).



(α) Κατανομή χαρακτηριστικών εικόνας στο σύνολο εκπαίδευσης

(β) Απευθείας αντιστοίχιση από τον z-χώρο στα χαρακτηριστικά

(γ) Εκπαιδεύσιμη αντιστοίχιση από τον z-χώρο στον w-χώρο και μετά στα χαρακτηριστικά

Σχήμα 49: Ενδεικτικό παράδειγμα επίδρασης του Δικτύου Αντιστοίχισης Θορύβου όταν υπάρχουν δύο παράγοντες παραλλαγής (factors of variation) (δηλ. χαρακτηριστικά εικόνας, π.χ. αρρενωπότητα και μήκος μαλλιών).
Πηγή: «A Style-Based Generator Architecture for Generative Adversarial Networks», Karras et al., 2018 [91]

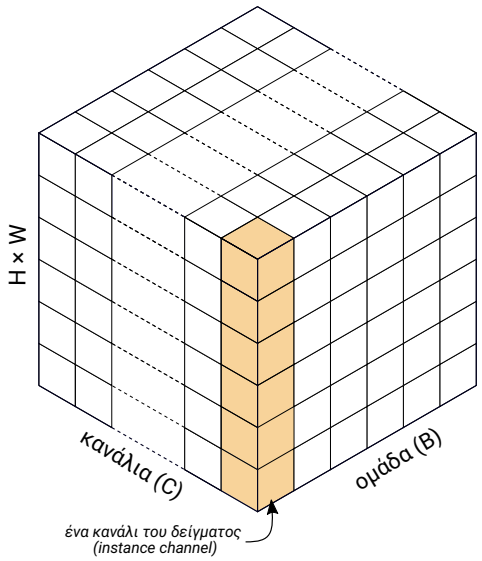
Πιο αναλυτικά, ο λόγος ύπαρξης του δικτύου αντιστοίχισης θορύβου είναι η αύξηση της ελεγξιμότητας του Generator μέσω της μείωσης του entanglement (μπερδέματος - προτιμούμε τη χρήση του αγγλικού όρου για το συγκεκριμένο) του χώρου εισόδου του Generator. Έτσι, οι συγγραφείς περνούν το αρχικό τυχαίο διάνυσμα, $\vec{z}$, από ένα απλό ΤΝΔ (MLP) με οκτώ (8) πλήρως συνδεδεμένες στρώσεις, στην έξοδο του οποίου βγαίνει το «*διάνυσμα ενδιάμεσου θορύβου*», $\vec{w}$. Το διάνυσμα αυτό είναι ίδιου μεγέθους με το αρχικό, ωστόσο σύμφωνα με τους συγγραφείς του [91], βρίσκεται σε έναν μετασχηματισμένο χώρο οι βάσεις του οποίου εκπαιδεύονται για να αντιστοιχούν σε οπτικά χαρακτηριστικά των

εικόνων εξόδου. Θα ήταν ίσως παράλογο, εξάλλου, να αναμένουμε οι διαστάσεις ενός γκαουσιανού χώρου, να αντιστοιχούν χωρίς κάποιο μετασχηματισμό σε χαρακτηριστικά του πολύ-υψηλής διαστασιμότητας χώρου των εικόνων εξόδου.

Ως αποτέλεσμα της εφαρμογής του Δικτύου Αντιστοίχισης Θορύβου, οι παρεμβολές (interpolations) στον w-χώρο είναι πολύ πιο ομαλές, ενώ οι συγγραφείς έδειξαν πως είναι δυνατή η εύρεση γραμμικών «κατευθύνσεων» μεταβολής συγκεκριμένων χαρακτηριστικών (όπως π.χ. η ύπαρξη γενειάδας ή το χρώμα των μαλλιών). Το διάνυσμα αυτό χρησιμοποιείται ως είσοδος στο *Δίκτυο Σύνθεσης* του Generator, όπου εισέρχεται μέσω εκπαιδεύσιμων μετασχηματισμών σε κάθε στρώση Προσαρμοστικής Κανονικοποίησης Δείγματος, όπως αναλύεται ακολούθως.

Προσαρμοστική Κανονικοποίηση Δείγματος (Adaptive Instance Normalization)

Αφού έχει παραχθεί το διάνυσμα ενδιάμεσου θορύβου, $\vec{w}$, πρέπει να εισαχθεί στο Δίκτυο Σύνθεσης του Generator. Όπως αναφέρθηκε, αυτό γίνεται μέσω μιας παραλλαγής της στρώσης Κανονικοποίησης Δείγματος, της στρώσης Προσαρμοστικής Κανονικοποίησης Δείγματος (Adaptive Instance Normalization), όπου για την προσαρμογή χρησιμοποιείται το μετασχηματισμένο από ένα ΤΝΔ διάνυσμα $\vec{w}$. Θα ξεκινήσουμε με μία σύντομη περιγραφή της Κανονικοποίησης Δείγματος και κατόπιν θα αναλύσουμε πως υλοποιείται η Προσαρμοστική εκδοχή αυτής.



Σχήμα 50: Σχηματική απεικόνιση της Κανονικοποίησης Δείγματος.

Πηγή: Ανακατασκευή από «Instance Normalization: The Missing Ingredient for Fast Stylization», Ulyanov et al., 2016 [61]

Ξεκινώντας, η Κανονικοποίηση Δείγματος (Instance Normalization) παρουσιάστηκε αρχικά από τον Ulyanov et al. στο άρθρο τους «*Instance Normalization: The Missing Ingredient for Fast Stylization*» [61]. Το όνομά της προέρχεται από το ότι, σε αντίθεση με την Κανονικοποίηση Ομάδας, η κανονικοποίηση γίνεται ξεχωριστά για κάθε ένα από τα δείγματα (ή τις εξόδους των συνελικτικών στρώσεων) της ομάδας (batch), που εναλλακτικά ονομάζονται instances. Επίσης, εδώ όμοια με την Κανονικοποίηση Ομάδας, η κανονικοποίηση γίνεται ξεχωριστά για κάθε κανάλι της εισόδου (ή αντίστοιχα για κάθε χάρτη ενεργοποίησης της εξόδου της προηγούμενης συνελικτικής στρώσης), όπως φαίνεται και στο σχήμα 50 επάνω.

Από μαθηματική σκοπιά, η Κανονικοποίηση Δείγματος, ορίζεται ως εξής:

$$\mu_{C_i^{[b]}} \longleftarrow \frac{1}{|H \times W|} \sum_{i=1}^{|H \times W|} \left(\vec{c}_i^{[b]}\right) \tag{4.7}$$

$$\sigma^2_{C_i^{[b]}} \longleftarrow \frac{1}{|H \times W|} \sum_{i=1}^{|H \times W|} \left[\left(\vec{c}_i^{[b]}\right) - \mu_{C_i^{[b]}}\right]^2 \tag{4.8}$$

$$\vec{c}_i^{[b]} \longleftarrow \frac{\vec{c}_i^{[b]} - \mu_{C_i^{[b]}}}{\sqrt{\sigma^2_{C_i^{[b]}} + \epsilon}}, \quad b = 1, ..., B \quad i = 1, ..., C \tag{4.9}$$

όπου $\vec{c}_i^{[b]}$ περιέχει όλα τα στοιχεία του i-οστού καναλιού (ή χάρτη ενεργοποίησης) και του b-οστού δείγματος μέσα στην ομάδα, το οποίο και κανονικοποιείται ώστε να έχει μηδενική μέση τιμή και μοναδιαία διακύμανση. Άρα, η Κανονικοποίηση Δείγματος γίνεται ως προς κάθε δείγμα και συγκεκριμένα ως προς κάθε κανάλι του κάθε δείγματος της ομάδας, ενώ η Κανονικοποίηση Ομάδας γίνεται ως προς κάθε κανάλι αλλά *όλων* των δειγμάτων της ομάδας.

Στο σημείο αυτό και πριν προχωρήσουμε αξίζει να επαναορίσουμε την έννοια του «στιλ» σύμφωνα με τους συγγραφείς του StyleGAN: *Εφαρμογή στιλ ισοδυναμεί με την κλιμάκωση (scaling) και τη μετατόπιση shifting των τυπικών (κανονικοποιημένων) εξόδων των Συνελικτικών Στρώσεων*. Με βάση αυτόν τον ορισμό, είμαστε σε θέση να περιγράψουμε την Προσαρμοστική Κανονικοποίηση Δείγματος. Η προσαρμογή, λοιπόν, αφορά ακριβώς αυτήν την εφαρμογή στιλ, το οποίο προέρχεται από το διάνυσμα ενδιάμεσου θορύβου. Η λογική είναι η εξής: *Πάρε ένα διάνυσμα από έναν (σχετικά) disentangled χώρο, και χρησιμοποίησέ το για να μεταβάλλεις με εκπαιδεύσιμο τρόπο τις εξόδους διαφόρων συνελικτικών στρώσεων του Generator, εφαρμόζοντας έτσι ένα σύνολο από στιλ*.

Για τη μετατροπή του διανύσματος ενδιάμεσου θορύβου, $\vec{w}$, σε συντελεστές κλιμάκωσης

και μετατόπισης σε κάθε στρώση Κανονικοποίησης Δείγματος, οι συγγραφείς πρότειναν τη χρήση δύο ΤΝΔ τύπου MLP ανά στρώση. Έτσι, για την προσαρμογή των κανονικοποιημένων τιμών στην έξοδο της Κανονικοποίησης Δείγματος υπολογίζονται οι ακόλουθοι συντελεστές:

$$\vec{w}^{[b]} \longrightarrow [A_{sc}] \longrightarrow (y_{sc})_i^{[b]} \tag{4.10}$$

$$\vec{w}^{[b]} \longrightarrow [A_{sh}] \longrightarrow (y_{sh})_i^{[b]}, \quad b = 1, ..., B \quad i = 1, ..., C \tag{4.11}$$

όπου $(y_{sc})_i^{[b]}$ είναι ο συντελεστής κλιμάκωσης και $(y_{sh})_i^{[b]}$ ο συντελεστής μετατόπισης του i-οστού καναλιού (ή χάρτη ενεργοποίησης), $[A_{sc}]$ το MLP για εξαγωγή των συντελεστών κλιμάκωσης και $[A_{sh}]$ το MLP για εξαγωγή των συντελεστών μετατόπισης για κάθε κανάλι, B είναι ο αριθμός των δειγμάτων στην ομάδα και C ο αριθμός των καναλιών (ή χαρτών ενεργοποίησης). Σημειώνεται, ότι τα δίκτυα $[A_{sc}]$ και $[A_{sh}]$ εμπεριέχονται στα εκάστοτε τετράγωνα με την ένδειξη «A» στο σχήμα 48 παραπάνω.

Επομένως, μετά την εφαρμογή των «στιλ» που εξήχθησαν, η έξοδος της στρώσης Προσαρμοστικής Κανονικοποίησης Δείγματος, θα είναι:

$$\begin{aligned}
\mu_{C_i^{[b]}} &\longleftarrow \frac{1}{|H\times W|}\sum_{i=1}^{|H\times W|}\left(\vec{c}_i^{[b]}\right) \\
\sigma^2_{C_i^{[b]}} &\longleftarrow \frac{1}{|H\times W|}\sum_{i=1}^{|H\times W|}\left[\left(\vec{c}_i^{[b]}\right) - \mu_{C_i^{[b]}}\right]^2 \\
\vec{w}^{[b]} &\longrightarrow [A_{sc}] \longrightarrow (y_{sc})_i^{[b]} \\
\vec{w}^{[b]} &\longrightarrow [A_{sh}] \longrightarrow (y_{sh})_i^{[b]} \\
\vec{c}_i^{[b]} &\longleftarrow (y_{sc})_i^{[b]} * \frac{\vec{c}_i^{[b]} - \mu_{C_i^{[b]}}}{\sqrt{\sigma^2_{C_i^{[b]}} + \epsilon}} + (y_{sh})_i^{[b]}, \quad b = 1, ..., B \quad i = 1, ..., C
\end{aligned} \tag{4.12}$$

με τα μεγέθη να είναι όπως προηγουμένως.

Έγχυση Θορύβου (Noise Injection)

Μία μικρότερη καινοτομία που διαπίστωσαν οι συγγραφείς του StyleGAN ότι έχει εντυπωσιακά αποτελέσματα είναι η έγχυση (injection) μετασχηματισμένου θορύβου στις εξόδους των συνελικτικών στρώσεων και *πριν* τις στρώσεις Προσαρμοστικής Κανονικοποίησης Δείγματος. Συγκεκριμένα, προκειμένου να εισάγουν στοχαστικές διαφοροποιήσεις για αύξηση της ποικιλομορφίας των παραγόμενων εικόνων, οι συγγραφείς του StyleGAN

πρότειναν την έγχυση θορύβου στις εξόδους των συνελικτικών στρώσεων μέσω ενός εκπαιδεύσιμου μετασχηματισμού.

Πιο αναλυτικά, έστω η έξοδος μιας συνελικτικής στρώσης του Δικτύου Σύνθεσης του Style-based Generator, διαστάσεων H×W×C (δηλαδή C στον αριθμό χάρτες ενεργοποίησης, ο καθένας ύψους H και πλάτους W). Τότε ο θόρυβος που θα εγχυθεί θα είναι αρχικά ένας δισδιάστατος πίνακας τυχαίων τιμών (από κανονική κατανομή) διαστάσεων H×W. Ταυτόχρονα, σε κάθε στρώση Έγχυσης Θορύβου, υπάρχει ένα σύνολο C εκπαιδεύσιμων παραμέτρων που είναι τα βάρη με τα οποία πολλαπλασιάζεται ο θόρυβος πριν αυτός προστεθεί στους χάρτες ενεργοποίησης της συνελικτικής στρώσης. Έτσι, ο αρχικός θόρυβος, πολλαπλασιασμένος με το βάρος που αντιστοιχεί στον εκάστοτε χάρτη ενεργοποίησης, προστίθεται σε αυτόν. Λαμβάνοντας υπόψη ότι έχουμε N στον αριθμό δείγματα ανά ομάδα, η πράξη που συντελείται στη στρώση Έγχυσης Θορύβου έχει ως εξής:

$$noise_{N,1,W,H} \longleftarrow randn(N, 1, W, H) \tag{4.13}$$

$$Wn_{N,C} \longleftarrow SGD(Wn) \tag{4.14}$$

$$featuremaps \longleftarrow featuremaps + Wn_{N,C,1,1} * noise_{N,1,W,H} \tag{4.15}$$

όπου $featuremaps$ είναι οι χάρτες ενεργοποίησης της προηγούμενης συνελικτικής στρώσης. Παρακάτω, παραθέτουμε δύο εικόνες, μία από Generator που έχει εκπαιδευτεί με στρώσεις Έγχυσης Θορύβου και μία χωρίς. Τέλος, οι συγγραφείς παρατήρησαν ότι όσο πιο αργά (στις τελευταίες στρώσεις) του Generator εγχέεται ο θόρυβος τόσο αυτός τείνει να επηρεάζει τις παραγόμενες εικόνες σε πιο μικρές λεπτομέρειες όπως τη θέση ή την υφή των μαλλιών ή την ύπαρξη ρυτίδων κλπ., ενώ το αντίθετο συμβαίνει για τις αρχικές στρώσεις.

Παραγωγές του StyleGAN και Μίξη Στιλ

Πριν ολοκληρώσουμε την περιγραφή της πρώτης έκδοσης του StyleGAN, παραθέτουμε σε αυτήν την παράγραφο μερικές παραγωγές του StyleGAN που φανερώνουν την υπεροχή και αποτελεσματικότητά του.

Ενώ ακολούθως παραθέτουμε μία εικόνα μίξης στιλ, δηλαδή δίνοντας διαφορετικά διανύσματα ενδιάμεσου (που αντιστοιχούν σε διαφορές παραγωγές) θορύβου στις στρώσεις Προσαρμοστικής Κανονικοποίησης Δείγματος του Δικτύου Σύνθεσης του Generator. Μάλιστα, οι συγγραφείς παρατήρησαν ότι οι στρώσεις που θα γίνει η μίξη, επηρεάζει το

(α) Με θόρυβο (β) Χωρίς θόρυβο

Σχήμα 51: Επίδραση της έγχυσης θορύβου μετά τις συνελικτικές στρώσεις του Generator του StyleGAN. Απ΄ ότι φαίνεται, η απουσία του θορύβου οδηγεί σε πιο θολές εικόνες με λιγότερη λεπτομέρεια.

Πηγή: «A Style-Based Generator Architecture for Generative Adversarial Networks», Karras et al., 2018 [91]

από πόσο πιο λεπτομερή ή πιο χονδροειδή coarse στιλ μεταφέρονται από την εκάστοτε εικόνα. Στο σχήμα 53, στις αρχικές στρώσεις Προσαρμοστικής Κανονικοποίησης Δείγματος δίνονται τα διανύσματα $\vec{w}$ από την εικόνα Β, ενώ στα υπόλοιπα δίνονται κανονικά από την Α. Ως αποτέλεσμα αυτού, οι παραγωγές της εικόνας Α να μεταβάλλονται με χονδροειδή χαρακτηριστικά από τη Β (όπως το φύλο και η ηλικία).

StyleGAN2

Αξίζει να αναφερθεί, για λόγους πληρότητας, πως οι συγγραφείς των PGGAN και StyleGAN αναβάθμισαν εκ΄ νέου το τελευταίο ένα χρόνο αργότερα, παρουσιάζοντας το 2019 στο άρθρο τους «*Analyzing and Improving the Image Quality of StyleGAN*» [102] το μοντέλο StyleGAN v2. Το νέο μοντέλο περιείχε αρκετές βελτιώσεις και καινοτομίες, με σημαντικότερες τις εξής δύο (2):

- **Κατάργηση της Προσαρμοστικής Κανονικοποίησης Δείγματος**: Πλέον δεν χρησιμο-

Σχήμα 52: Μερικές από τις παραγωγές του μοντέλου StyleGAN (με τυχαία διαλογή) το οποίο εκπαιδεύτηκε στο σύνολο δεδομένων προσώπων υψηλής-ανάλυσης του Flicker, FFHQ. Φαίνεται τόσο η εξαιρετική ποιότητα όσο και η μεγάλη ποικιλομορφία των παραγόμενων δειγμάτων του Generator του StyleGAN.

Πηγή: «A Style-Based Generator Architecture for Generative Adversarial Networks», Karras et al., 2018 [91]

ποιούν τις στρώσεις Προσαρμοστικής Κανονικοποίησης Δείγματος, καθώς παρατήρησαν πως κάποια artifacts που συστηματικά δημιουργούνταν στις παραγόμενες εικόνες οφείλονταν στις στρώσεις αυτές. Αντ΄ αυτών εισήγαγαν μια παραλλαγή των παραδοσιακών συνελικτικών στρώσεων, τις Modulated Convolutional Layers όπου χρησιμοποιούν και πάλι τα διανύσματα $\vec{w}$ για επιβολή των στιλ, απλά πλέον αυτό

Σχήμα 53: Αποτέλεσμα μίξης στιλ, δηλ. των διανυσμάτων ενδιάμεσων θορύβων, από διαφορετικές παραγωγές του Generator του StyleGAN.

Πηγή: «A Style-Based Generator Architecture for Generative Adversarial Networks», Karras et al., 2018 [91]

γίνεται στα βάρη και όχι στις εξόδους των συνελικτικών στρώσεων.

- **Κατάργηση της Σταδιακής Αύξησης των Δικτύων**: Οι συγγραφείς παρατήρησαν πως αφενός λόγω των νέων συνελικτικών στρώσεων η σταδιακή εκπαίδευση δεν προσέφερε σημαντικά πλεονεκτήματα στον χρόνο και στη σταθερότητα της εκπαίδευσης, αλλά αφετέρου δημιουργούσε και προβλήματα λανθασμένης ευθυγράμμισης οπτικών χαρακτηριστικών. Αντ΄ αυτού, χρησιμοποίησαν skip connections στον Generator (βλ. pix2pix 4.2.1) και residual connections στον Διαφοριμινατορ.

Δεν θα επεκταθούμε περαιτέρω, λόγω του ότι δεν υλοποιήσαμε μοντέλο που να βασίζεται στο StyleGAN v2. Θα παραθέσουμε ωστόσο μερικές παραγωγές του Generator του StyleGAN v2 όπου φαίνεται για ποιο λόγο τα μοντέλα τύπου StyleGAN αποτέλεσαν σημείο καμπής στην ποιότητα των παραγόμενων εικόνων από μοντέλα Παραγωγικής Μοντελοποίησης. Αυτές φαίνονται στο σχήμα 54 παρακάτω. Ο ενδιαφερόμενος ανα-

γνώστης καλείται να επισκεφθεί τις εξής ιστοσελίδες για περισσότερες παραγωγές από το μοντέλο StyleGAN v2:

- **https://thispersondoesnotexist.com**: Παραγωγές του μοντέλου StyleGAN v2 το οποίο έχει εκπαιδευθεί στο σύνολο δεδομένων ανθρώπινων προσώπων του Flicker, Flicker Faces High-Quality (FFHQ).
- **https://thiscatdoesnotexist.com**: Παραγωγές του μοντέλου StyleGAN v2 το οποίο έχει εκπαιδευθεί στο σύνολο δεδομένων προσώπων ζώων του Flicker, Animal Faces High-Quality (AFHQ).

Σχήμα 54: Τέσσερις από τις καλύτερες παραγωγές του μοντέλου StyleGAN v2 το οποίο εκπαιδεύτηκε στο σύνολο δεδομένων προσώπων υψηλής-ανάλυσης του Flicker, FFHQ.
Πηγή: «Analyzing and Improving the Image Quality of StyleGAN», Karras et al., 2019 [102]

Στις εικόνες του σχήματος 54 δεν παρατηρούνται τα artifacts αυτών του StyleGAN, ενώ η

ποιότητα και η ποικιλομορφία είναι εντυπωσιακές.

4.2 Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα

Περνάμε ακολούθως σε μία άλλη κατηγορία εφαρμογών των GANs στην Παραγωγική Μοντελοποίηση εικόνων, αυτήν της Συζευγμένης Μετατροπής Εικόνας-σε-Εικόνα Paired Image-to-Image Translation. Σε αυτήν την κατηγορία, τα μοντέλα GANs εκπαιδεύονται και δοκιμάζονται υπο-συνθήκη, με τη συνθήκη να είναι και η ίδια μία εικόνα (σε αντίθεση με άλλα μοντέλα όπως το Conditional GAN - βλ. παράγραφο 3.2 - στα οποία η συνθήκη είναι η ετικέτα της τάξης). Γενικά, με τη φράση «μετατροπή εικόνας-σε-εικόνα», εννοούμε την εφαρμογή οποιουδήποτε απλού ή πιο σύνθετου μετασχηματισμού σε μία εικόνα εισόδου, όπως για παράδειγμα ο (σύνθετος) μετασχηματισμός μιας ασπρόμαυρης σε μία έγχρωμη εικόνα (όπως στο [97]).

Στα πλαίσια της Συζευγμένης Μετατροπής Εικόνα-σε-Εικόνα με GANs ο Generator λαμβάνει ως συνθήκη μία εικόνα και χρησιμοποιώντας το περιεχόμενο (δηλ. κάτι «σταθερό») και τα στιλ αυτής, καλείται στην έξοδό του να παράξει μία άλλη εικόνα από διαφορετικό πεδίο (domain) από την αρχική. Εκτός από τον χρωματισμό εικόνων που αναφέρθηκε, τα GANs έχουν εφαρμοστεί με επιτυχία σε εφαρμογές όπως η αύξηση της ανάλυσης εικόνων (δηλ. της συζευγμένης μετατροπής μίας χαμηλής ανάλυσης εικόνα σε μία υψηλής - μοντέλο SRGAN [54]), ή η αναπαλαίωση ασπρόμαυρων ταινιών (όπως εδώ: *DeOldify Facebook F8 Movie Colorization Demo*), ή η μετατροπή ενός χάρτη κατάτμησης (segmentation map) σε μία ρεαλιστική εικόνα (όπως με το μοντέλο pix2pix που αναλύεται ακολούθως), ή μετατροπή σκίτσου σε ρεαλιστική εικόνα και αρκετές άλλες. Στην παρούσα εργασία, σε ότι αφορά τη Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα, κάνουμε χρήση του μοντέλου pix2pix, γι΄ αυτό και για το υπόλοιπο αυτής της ενότητας θα εστιάσουμε σε αυτό.

4.2.1 pix2pix

Προχωρώντας, θα αναλύσουμε τη αρχιτεκτονική και τον τρόπο λειτουργίας του μοντέλου pix2pix, τύπου GAN, το οποίο παρουσιάστηκε από τον Isola et al. στο άρθρο τους «*Image-to-Image Translation with Conditional Adversarial Networks*» [51]. Όπως αναφέρθηκε, το μοντέλο αυτό χρησιμοποιείται για Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα («pixels-to-pixels mapping») και επομένως για είσοδο μία εικόνα από το σύνολο δεδομένων ενός πεδίου καλείται, εφαρμόζοντας έναν εκπαιδεύσιμο μετασχηματισμό, να δώσει μία εικόνα

στην έξοδό του που αφενός να είναι ρεαλιστική και αφετέρου να είναι «κοντά» στις εικόνες του συνόλου δεδομένων ενός διαφορετικού πεδίου. Πρακτικά, λοιπόν, για την εκπαίδευση αυτών των μοντέλων απαιτείται η ύπαρξη ενός επισημασμένου συνόλου δεδομένων, με την έννοια ότι θα πρέπει οι εικόνες να είναι οργανωμένες σε ζεύγη εισόδων-εξόδων. Πριν συνεχίσουμε, θεωρούμε χρήσιμο να αναφέρουμε πως για τη Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα εκτός από GANs έχουν δοκιμαστεί και άλλες μορφές ΤΝΔ, όπως οι Αυτόματοι Κωδικοποιητές, ωστόσο η χρήση των πρώτων οδηγεί σε πιο ρεαλιστικά και ποιοτικώς αναβαθμισμένα αποτελέσματα.

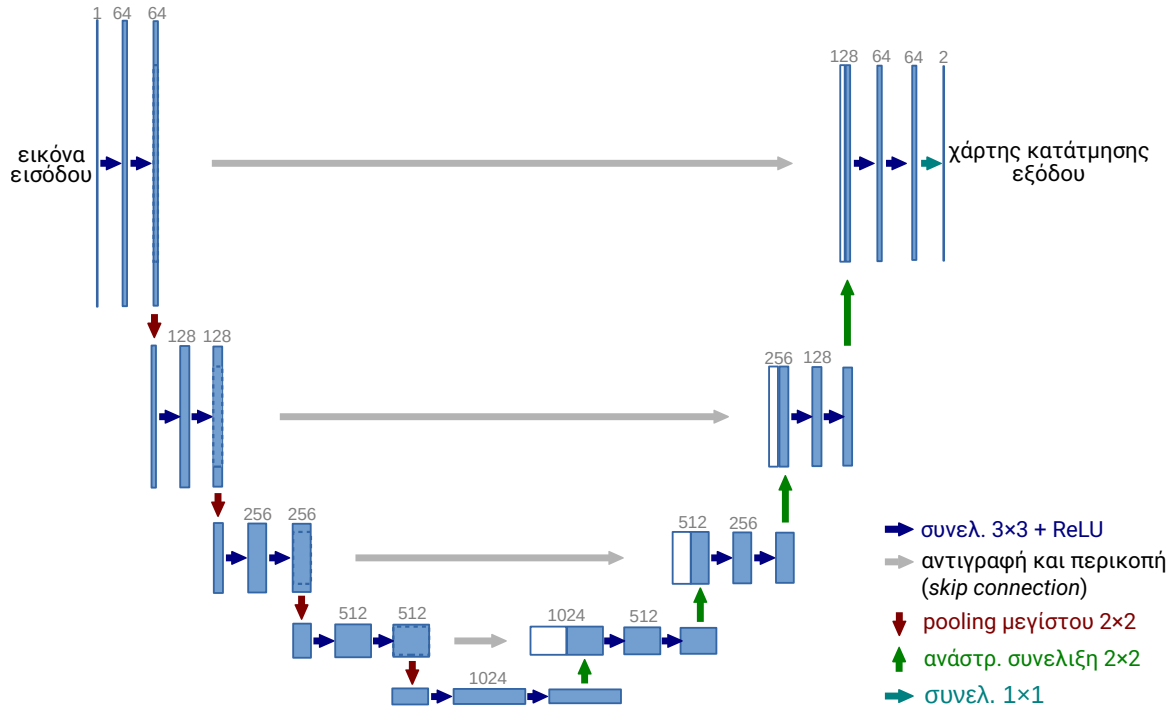
Σε ότι αφορά τα επιμέρους δίκτυα του μοντέλου pix2pix, συναντάμε και εδώ την παραδοσιακή μορφή των GANs, δηλαδή ένα δίκτυο για τον Generator και ένα για τον Discriminator. Ωστόσο, υπάρχουν σημαντικές διαφοροποιήσεις σε σύγκριση με τα δίκτυα των μοντέλων που προηγήθηκαν, οι οποίες έχουν κάνουν με το διαφορετικό της εφαρμογής αλλά και με σειρά καινοτομιών οι οποίες προτάθηκαν από του συγγραφείς του μοντέλου. Σημειώνουμε, πως ο αρχικός στόχος των συγγραφέων του [51] ήταν η παραγωγή ρεαλιστικών εικόνων από τους χάρτες κατάτμησης (segmentation maps)[3] αυτών. Ακολούθως, αναλύουμε αρχικά τον Generator και κατόπιν τον Discriminator, ενώ στο τέλος αυτής της υποενότητας παραθέτουμε παραγόμενα αποτελέσματα και αναβαθμίσεις του μοντέλου που προτάθηκαν αργότερα.

Generator του pix2pix: U-Net

Οι συγγραφείς του pix2pix χρησιμοποίησαν ως Generator μία παραλλαγή του δικτύου U-Net, το οποίο είχε αρχικά προταθεί από τον Ronneberger et al., στο άρθρο τους «*Convolutional networks for biomedical image segmentation*» [45], για παραγωγή χαρτών κατάτμησης από εικόνες οργανικών ιστών. Τονίζουμε στο σημείο αυτό, πως το δίκτυο U-Net στην ουσία κάνει και αυτό συζευγμένη μετατροπή εικόνας-σε-εικόνα, στο απλούστερο όμως έργο παραγωγής χαρτών κατάτμησης από πραγματικές εικόνες (κάτι που πιθανότατα δεν θα ήταν πιο ωφέλιμο εάν γίνονταν με μοντέλο GAN). Ωστόσο, στο αντίστροφο και σημαντικά πολυπλοκότερο έργο παραγωγής ρεαλιστικών εικόνων από χάρτες κατάτμησης, η χρήση ενός μοντέλου τύπου GAN, όπως το pix2pix, οδηγεί σε σημαντικά καλύτερα αποτελέσματα

[3]Οι χάρτες κατάτμησης είναι ουσιαστικά εικόνες σε κάθε εικονοστοιχείο των οποίων ανατίθεται ένα χρώμα από σύνολο διακριτών χρωμάτων. Το σύνολο αυτό περιέχει τόσα στον αριθμό χρώματα όσες είναι και οι διακριτές τάξεις των δεδομένων εκπαίδευσης. Έτσι, κάθε εικονοστοιχείο ταξινομείται, με αποτέλεσμα ο χάρτης να δείχνει τη θέση και μορφή όλων των αντικειμένων που υπάρχουν στην εικόνα (το παρασκήνιο λαμβάνει ένα ουδέτερο χρώμα).

από άποψη ρεαλισμού και συνοχής. Ακολούθως, αρχικά αναλύουμε το U-Net και κατόπιν τις διαφοροποιήσεις αυτού για τη σχεδίαση του Generator του pix2pix.



Σχήμα 55: Αρχιτεκτονική του δικτύου U-Net.

Πηγή: Ανακατασκευή από «U-Net: Convolutional Networks for Biomedical Image Segmentation», Ronneberger et al., 2015 [45]

Το δίκτυο U-Net μοιάζει σημαντικά με τα δίκτυα Αυτόματων Κωδικοποιητών που αναλύθηκαν στην ενότητα 2.3. Αποτελείται, δηλαδή, από μία σειρά συνελικτικών στρώσεων οι οποίες μειώνουν το πλάτος και ύψος έως ένα σημείο στένωσης bottleneck, ακολουθούμενες από ανάστροφες συνελικτικές στρώσεις οι οποίες αυξάνουν το πλάτος και ύψος έως την τελική στρώση εξόδου. Οι πρώτες ανήκουν στο υποδίκτυο του encoder, ενώ οι τελευταίες σε αυτό του decoder. Υπάρχουν, ωστόσο, δύο σημαντικές διαφοροποιήσεις μεταξύ του U-Net και των ΑΚ. Αρχικά, δεν εκπαιδεύεται με στόχο την αυτοκωδικοποίηση της εισόδου (δηλ. παραγωγής της ίδιας της εισόδου στην έξοδο), αλλά με στόχο τον μετασχηματισμό αυτής σε κάποιο άλλο πεδίο. Δεύτερη και σημαντικότερη διαφοροποίηση είναι ότι οι συγγραφείς του U-Net πρότειναν τη χρήση skip connections μεταξύ στρώσεων του ίδιου επιπέδου του encoder και decoder. Όπως φαίνεται και στο σχήμα 55 παραπάνω, αυτές αντιγράφουν αυτούσιους (ίσως περικομμένους κατά πλάτος και ύψος για ταύτιση των διαστάσεων) τους χάρτες ενεργοποίησης στις εξόδους κάποιων συνελικτικών στρώσεων του encoder και τους επικολλούν ενώνοντάς τους με αυτούς στην έξοδο των

αντίστοιχων (ενν. ομοεπίπεδων) ανάστροφων συνελικτικών στρώσεων του decoder. Ο λόγος ύπαρξης των λεγόμενων skip connections είναι διττός: από τη μία βοηθούν στη ροή της πληροφορίας κατά την κατασκευή της εικόνας εξόδου από τη στρώση στένωσης - ειδικά με την πληροφορία που αφορά το περιεχόμενο ή τη σταθερή δομή μιας εικόνας που δεν επηρεάζονται σημαντικά κατά τη μετατροπή - αλλά κυρίως διότι επιλύει το πρόβλημα εξαφάνισης των παραγώγων κατά την εκτέλεση του προς τα πίσω περάσματος και του back-propagation, κάτι που μαστίζει τα βαθιά συνελικτικά νευρωνικά δίκτυα με συναρτήσεις ενεργοποίησης ReLU (ανορθωμένες γραμμικές μονάδες).

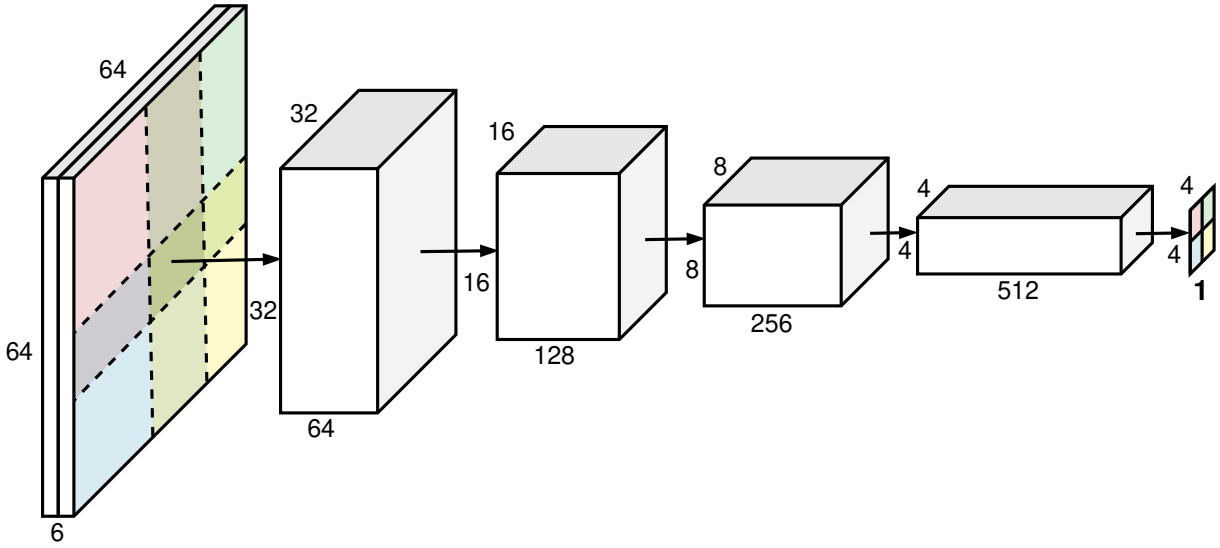
Συνοπτικά, η αρχιτεκτονική του δικτύου U-Net που απεικονίζεται στο σχήμα 55 έχει ως εξής: οκτώ (8) συνελικτικές στρώσεις στο υποδίκτυο του encoder επιδρούν στην εικόνα εισόδου έως το σημείο στένωσης, ενώ οκτώ (8) ανάστροφες συνελικτικές στρώσεις μετατρέπουν την ενδιάμεση αυτή αναπαράσταση στον χάρτη κατάτμησης στην έξοδο. Με γκρι σημειώνονται οι skip connection που μεταφέρουν περικομμένα αντίγραφα των χαρτών ενεργοποίησης των στρώσεων του encoder στις ομοεπίπεδες στρώσεις του decoder για καλύτερη ροή της πληροφορίας.

Περνάμε τώρα στην περιγραφή του Generator του pix2pix. Αυτός, όπως αναφέρθηκε, βασίζεται στο δίκτυο U-Net, έχοντας όμως μερικές σημαντικές διαφορές καθώς η εικόνα εξόδου δεν είναι πλέον χάρτης κατάτμησης αλλά μια έγχρωμη εικόνα. Αρχικά, οι συγγραφείς του pix2pix προσέθεσαν στρώσεις κανονικοποίησης ομάδας (batch normalization) μετά από κάθε συνελικτική στρώση του encoder και ανάστροφη συνελικτική στρώση του decoder και πριν τις αντίστοιχες στρώσεις ενεργοποίησης ReLU. Επιπρόσθετα, οι στρώσεις ενεργοποίησης του encoder είναι Leaky ReLU με συντελεστή αρνητικής κλίσης ίσο με 0.2. Τέλος, τα περιθώρια (paddings) των συνελικτικών στρώσεων και η τελική συνελικτική στρώση εξόδου του decoder έχουν προσαρμοστεί ώστε οι διαστάσεις των εικόνων εξόδου να ταυτίζονται με αυτές των εικόνων εισόδου (256×256×3 για όλα τα σύνολα δεδομένων που εφάρμοσαν το μοντέλο του στο [51]).

Discriminator του pix2pix: PatchGAN

Ο Discriminator του pix2pix αποτελεί μια σημαντική καινοτομία και συνεισφορά των συγγραφέων στο σχεδιασμό GANs. Αυτός, τον οποίο ονόμασαν *PatchGAN Discriminator*, βασίζεται στον Discriminator του DCGAN (βλ. υποενότητα 4.1.1) ωστόσο έχει μια καίρια διαφορά: οι συγγραφείς έχουν αφαιρέσει μία ή περισσότερες συνελικτικές στρώσεις από το τέλος του Discriminator καθώς και την πλήρως-συνδεδεμένη στρώση. Ως αποτέλεσμα

αυτού, η έξοδος του PatchGAN Discriminator προκύπτει απλώς από μία συνελικτική στρώση που εφαρμόζεται στο σημείο που «κόβεται» το δίκτυο και η οποία βγάζει ένα κανάλι (ή χάρτη ενεργοποίησης), ακολουθούμενη από μια σιγμοειδή συνάρτηση ενεργοποίησης. Επίσης, λόγω του ότι έχουμε υπο-συνθήκη παραγωγή, στην είσοδο του Discriminator εκτός από την τεχνητή εικόνα / επιθυμητή εικόνα εξόδου, δίνεται και η εικόνα εισόδου του Generator ή η συνθήκη, κάτι που φαίνεται στο σχήμα 56 παρακάτω. Όπως φαίνεται εκεί, για είσοδο (πραγματική ή τεχνητή) εικόνα 64×64 καθώς και τη συνθήκη της εισόδου του Generator επίσης 64×64, ο PatchGAN Discriminator θα δώσει μετά από πέντε (5) συνελικτικές στρώσεις (ακολουθούμενες από στρώσεις Κανονικοποίησης Ομάδας και ενεργοποίησης Leaky ReLU αρνητικής κλίσης 0.2). Η έξοδος είναι πίνακας πιθανοτήτων ρεαλισμού του κάθε μέρους (patch) της εισόδου. Το δεκτικό πεδίο της κάθε τιμής ως προς την είσοδο χρωματίζεται διαφορετικά.



Σχήμα 56: Αρχιτεκτονικη του PatchGAN Discriminator.

Πηγή: Κατασκευή με βάση την αρχιτεκτονική που δίνεται στο «Image-to-Image Translation with Conditional Adversarial Networks», Isola et al., 2016 [51]

Έτσι, στην έξοδο του PatchGAN Discriminator, θα έχουμε έναν πίνακα πιθανοτήτων, κάθε στοιχείο του οποίου εκπαιδεύεται να δίνει την πιθανότητα το αντίστοιχο μέρος (patch) της εισόδου να είναι αφενός ρεαλιστικό και αφετέρου να «ταιριάζει» με το αντίστοιχο μέρος της εικόνας συνθήκης (βλ. χρωματισμένες περιοχές - δεκτικά πεδία - στο σχήμα 56 παραπάνω). Λόγω αυτής της σχεδιαστικής επιλογής, σε κάθε βήμα ο Discriminator δίνει πολύ περισσότερη πληροφορία ανάδρασης (feedback) στον Generator, κάτι που οδηγεί σε πιο γρήγορη και σταθερή εκπαίδευση. Επίσης, οι συγγραφείς του [51] διαπίστωσαν ότι

η χρήση του PatchGAN Discriminator τους επέτρεψε να εκπαιδεύσουν το μοντέλο τους με αρκετά μικρού μεγέθους σύνολα δεδομένων, όπως αυτό με ζεύγη εικόνων-χαρτών κατάτμησης με λιγότερα από 2000 ζεύγη στο σύνολο εκπαίδευσης. Τέλος, αναφέρουμε ότι όταν ως συνάρτηση κόστους για την εκπαίδευση του εκάστοτε μοντέλου GAN χρησιμοποιείται η Binary Cross-Entropy, τότε όπως αναφέρθηκε εφαρμόζεται σιγμοειδής συνάρτηση για τον σχηματισμό του πίνακα πιθανοτήτων. Θα μπορούσαμε, ωστόσο, να χρησιμοποιήσουμε οποιαδήποτε από τις αναφερθείσες συναρτήσεις κόστους, όπως η συνάρτηση κόστους Ελαχίστων Τετραγώνων (την οποία και χρησιμοποιούμε με τον PatchGAN Discriminator σε κάποια από τα μοντέλα που εκπαιδεύτηκαν).

Εκπαίδευση του pix2pix - Αποτελέσματα

Μια άλλη καινοτομία (στα πλαίσια εκπαίδευσης GANs) που εφάρμοσαν οι συγγραφείς του pix2pix έχει να κάνει με τη συνάρτηση κόστους εκπαίδευσης του μοντέλου τους. Συγκεκριμένα, χρησιμοποιήσαν την Binary Cross-Entropy προσθέτοντας, ωστόσο, και έναν όρο κανονικοποίησης που μετράει την απόσταση στον χώρο των εικονοστοιχείων (pixel distance loss term). Η βασική ιδέα προέρχεται από την εφαρμογή του pix2pix στο σύνολο δεδομένων χαρτών κατάτμησης-εικόνων, όπου για είσοδο ενός χάρτη κατάτμησης οι πιθανές «σωστές» απαντήσεις είναι περισσότερες από μία, ωστόσο εκείνοι ήθελαν να «πιέσουν» έμμεσα τον Generator να καταλήγει σε μία συγκεκριμένη. Έτσι, μαζί με τη συνάρτηση κόστους του GAN (adversarial loss) χρησιμοποιούν και έναν όρο που μετράει την απόσταση L1 (Manhattan) μεταξύ της εικόνας που παράγει ο Generator και της ιδανικής (πραγματικής) εικόνας που καλούνταν να παράξει. Αυτή ήταν μία από τις πρώτες εφαρμογές GANs που ο *Generator έμμεσα βλέπει την εικόνα που καλείται να παράξει* (όπως γίνονταν για παράδειγμα στου AK).

Έτσι η συνάρτηση κόστους εκπαίδευσης του GAN θα είναι [51]:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{y \sim p_{data}(y)} \left[\log D(y; \vec{\theta_D}) \right] + \mathbb{E}_{x \sim p_{data}(x), z \sim p_z(z)} \left[\log(1 - D(G(x, z; \vec{\theta_G}); \vec{\theta_D}) \right] \tag{4.16}$$

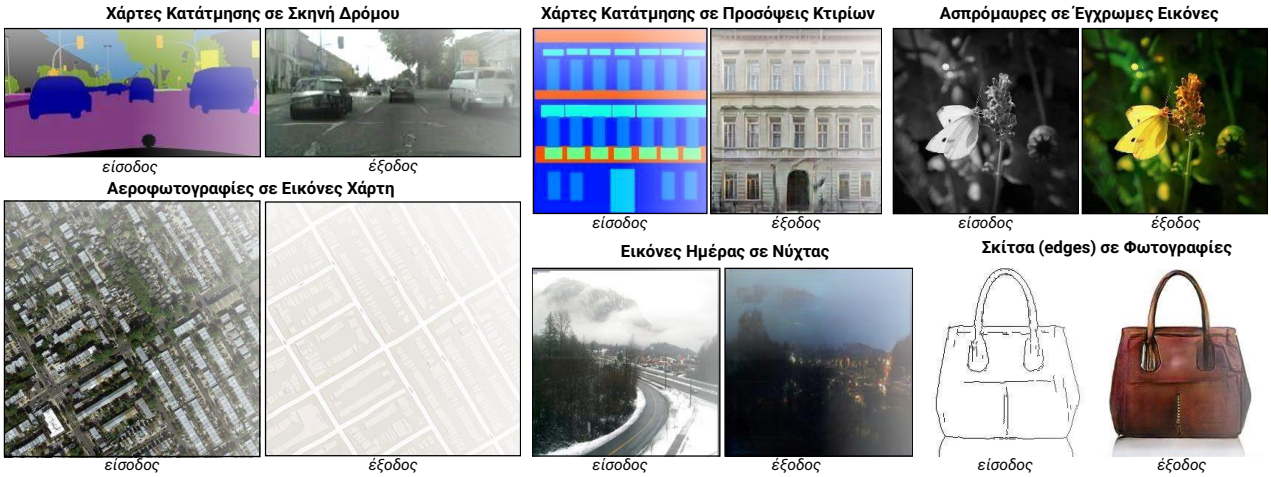
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y \sim p_{data}(x,y), z \sim p_z(z)} \left[\left\| y - G(x, z; \vec{\theta_G}) \right\|_1 \right] \tag{4.17}$$

$$J_D \left(\vec{\theta_D}, \vec{\theta_G} \right) = -\mathcal{L}_{cGAN}(G, D) \tag{4.18}$$

$$J_G \left(\vec{\theta_G}, \vec{\theta_D} \right) = \mathcal{L}_{cGAN}(G, D) + \lambda_{L1} * \mathcal{L}_{L1}(G) \tag{4.19}$$

όπου το «cGAN» προέρχεται από το ότι έχουμε υπο-συνθήκη παραγωγή (Conditional GAN - CGAN). Στο σχήμα 57 παρακάτω, παραθέτουμε παραγωγές του μοντέλου pix2pix σε διάφορα σύνολα δεδομένων που έχει δοκιμαστεί από τους συγγραφείς του, στο

[51]. Αξιοσημείωτο σχετικά με τις παραγωγές στο σχήμα αυτό είναι ότι οι συγγραφείς χρησιμοποίησαν αρκετά μικρού μεγέθους σύνολα δεδομένων, όλα με λιγότερα από 2000 ζεύγη εικόνων.



Σχήμα 57: Εφαρμογή του μοντέλου pix2pix για συζευγμένη μετατροπή εικόνας-σε-εικόνα σε διάφορα σύνολα δεδομένων εκπαίδευσης. Σε όλα τα σύνολα δεδομένων οι συγγραφείς χρησιμοποίησαν ίδια αρχιτεκτονική του μοντέλου και ίδιες παραμέτρους εκπαίδευσης, ενώ όλες οι παραγωγές είναι υπο-συνθήκη της εκάστοτε εικόνας εισόδου.
Πηγή: Ανακατασκευή από «Image-to-Image Translation with Conditional Adversarial Networks», Isola et al., 2016 [51]

4.2.2 pix2pixHD

Ένα χρόνο μετά την παρουσίαση του pix2pix, ο Wang et al. από το ίδιο εργαστήριο του Berkeley/NVIDIA παρουσίασαν στο άρθρο τους «*High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs*» [84], μία αναβάθμιση του μοντέλου pix2pix, το *pix2pixHD*, με σκοπό τη συζευγμένη μετατροπή υψηλής ανάλυσης εικόνων (διαστάσεων μεγαλύτερων από 1024×1024). Η ενδελεχής ανάλυση της αρχιτεκτονικής του μοντέλου pix2pixHD αποφεύγεται μιας και που δεν γίνεται χρήση του στα πλαίσια της παρούσας εργασίας, ωστόσο πριν παραθέσουμε ενδεικτικά αποτελέσματα από τη χρήση του θα αναφέρουμε τις βασικές αλλαγές/καινοτομίες του.

Έτσι, αυτό που αποτέλεσε κλειδί για την επιτυχή εκπαίδευση του pix2pixHD σε πολύ-υψηλής ανάλυσης εικόνες είναι η αύξηση της χωρητικότητας συνολικά του μοντέλου καθώς και άλλες καινοτομίες οι οποίες συνοψίζονται στις εξής τέσσερις (4):

- **Residual Generator**: ο Generator πλέον είναι σημαντικά πιο περίπλοκος και μεγαλύτε-

ρης χωρητικότητας καθώς αποτελείται από πολλές περισσότερες συνελικτικές στρώσεις οι οποίες όμως πλέον εντοπίζονται μεταξύ του encoder και του decoder, δεν αλλάζουν το πλάτος και ύψος των χαρτών ενεργοποίησης και περιέχουν residual connections[4]. Επίσης, λόγω της υψηλής ανάλυσης, οι συγγραφείς του [84] επέλεξαν να χρησιμοποιήσουν δύο υποδίκτυα στον Generator, ένα που «ασχολείται» με την παραγωγή εικόνων μισής-ανάλυσης από το άλλο και χρησιμοποίησαν residual connections για να ενώσουν την έξοδο του χαμηλής-ανάλυσης υποδικτύου με τις εξόδους κάποιας στρώσεις του άλλου, πριν τον τελικό decoder.

- **3 × PatchGAN Discriminator**: προκειμένου να αυξήσουν ακόμα περισσότερο το feedback που δίνει ο Discriminator οι συγγραφείς του pix2pixHD χρησιμοποιούν τρεις (3) PatchGAN Discriminators καθένας από τους οποίους δέχεται την ίδια εικόνα (ενν. μαζί με την εικόνα της συνθήκης - 6 κανάλια) αλλά υποδειγματολειπτημένη (downsampled). Οι πίνακες εξόδου των τριών δικτύων είναι ίδιας διάστασης και συνδυάζονται παίρνοντας τον μέσο όρο των τιμών τους. Η αύξηση του feedback του Discriminator βοήθησε σημαντικά με την ευστάθεια της εκπαίδευσης των πολύ-υψηλής ανάλυσης συνόλων δεδομένων.

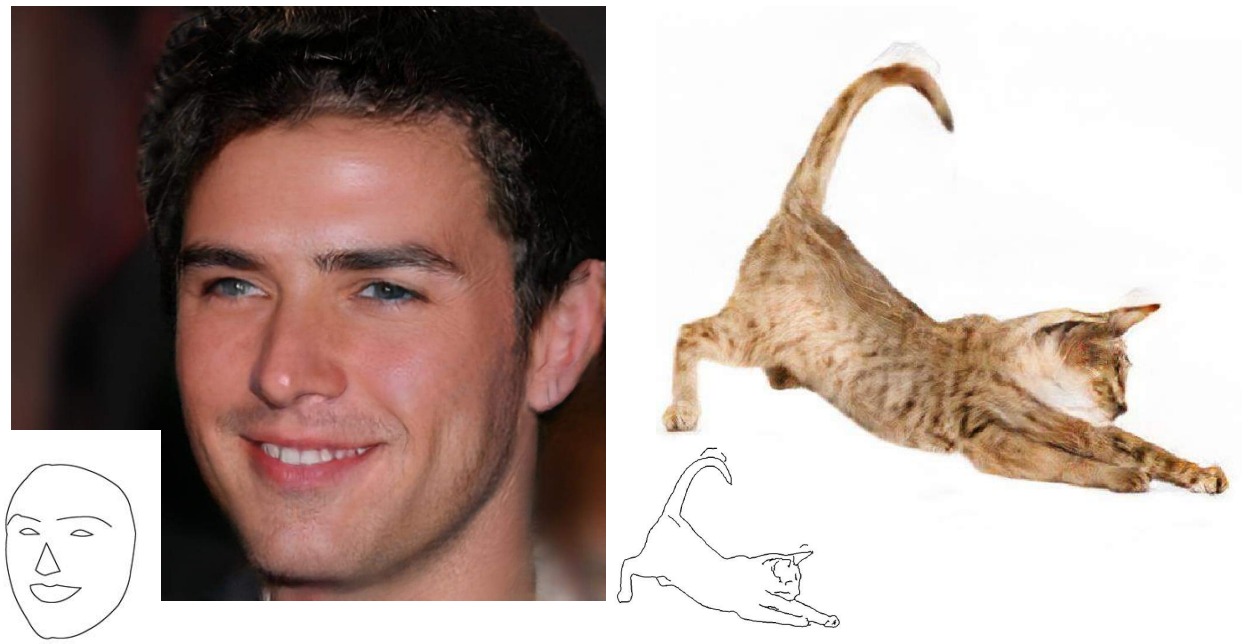
- **Κανοικοποίηση Δείγματος στον Discriminator**: οι συγγραφείς διαπίστωσαν πως η χρήση Κανονικοποίηση Δείγματος (Instance Normalization) στη θέση της Κανονικοποίησης Ομάδας (Batch Normalization) οδήγησε σε καλύτερα παραγόμενα αποτελέσματα, γι΄ αυτό χρησιμοποιούν αυτόν τον τύπο κανονικοποίησης μετά τις συνελικτικές στρώσεις των PatchGAN Discriminators.

- **Αντικατάσταση της Απόστασης Εικονοστοιχείων με Απόσταση Χαρακτηριστικών**: οι συγγραφείς χρησιμοποίησαν μια αρκετά πιο περίπλοκη λογική κανονικοποίησης της συνάρτησης κόστους του Generator. Έτσι, αντί για την (απλή) μέτρηση της απόστασης στον χώρο των εικονοστοιχείων μεταξύ των παραγόμενων και των πραγματικών εικόνων, οι συγγραφείς περνούν αμφότερες τις εικόνες από ένα προεκπαιδευμένο δίκτυο ταξινόμησης εικόνων, το VGG 19 (βλ. υποενότητα 2.1) και μετρούν την απόσταση των διανυσμάτων χαρακτηριστικών στην έξοδο της τελευ-

[4]Οι *residual connections* είναι στην ουσία συνδέσεις που παρακάμπτουν μία οι περισσότερες στρώσεις στο δίκτυο. Οι συνδέσεις αυτές προσθέτουν την είσοδο των στρώσεων αυτών στην έξοδό τους, επιτρέποντας έτσι στο δίκτυο να μάθει ταυτοτικές συναρτήσεις, εάν αυτό βελτιστοποιεί τη συνάρτηση κόστους. Επίσης, βοηθούν σημαντικά τη ροή των gradients προς τα πίσω κατά την εκτέλεση του back-propagation, κάτι που βοηθάει με το πρόβλημα εξαφάνισης παραγώγων και επέτρεψε βαθύτερα δίκτυα.

ταίας συνελικτικής στρώσης του παραπάνω δικτύου. Αυτή η κανονικοποίηση, που ονομάζεται feature loss, οδηγεί σε σημαντικά καλύτερα οπτικά αποτελέσματα ειδικά για μικρού μεγέθους σύνολα δεδομένων.

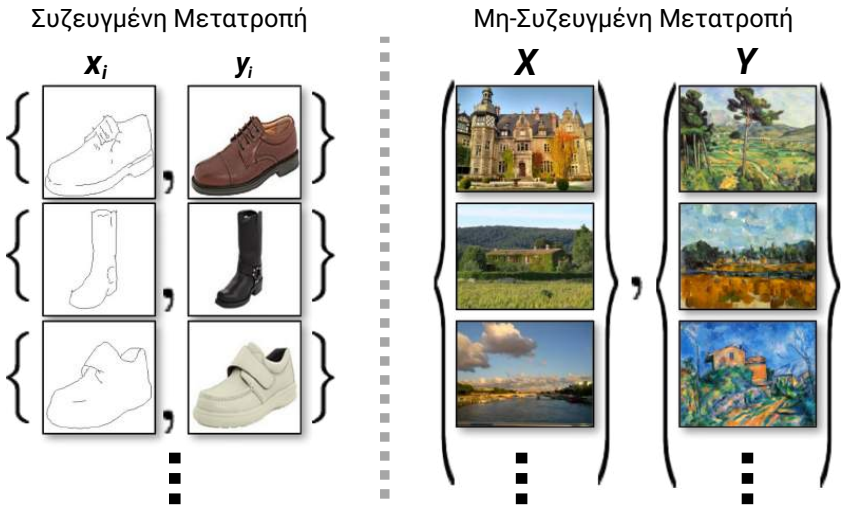
Παρακάτω, στο σχήμα 58, παραθέτουμε μερικές από τις παραγωγές του pix2pixHD όπως δόθηκαν στο [84]. Τέλος, για λόγους πληρότητας, αξίζει να αναφέρουμε ότι οι συγγραφείς του pix2pixHD δύο χρόνια αργότερα παρουσίασαν στο άρθρο τους «*Semantic Image Synthesis with Spatially-Adaptive Normalization*» [107], το state-of-the-art μοντέλο για συζευγμένη μετατροπή εικόνας-σε-εικόνα, το *GauGAN*. Στόχος του μοντέλου αυτού, το όνομα του οποίου αφορμάται από τον ζωγράφο Paul Gauguin, ήταν η δημιουργία ενός εργαλείου επεξεργασίας εικόνων και συγκεκριμένα διαδραστικής μετατροπής ενός χάρτη κατάτμησης που ζωγραφίζει ο χρήστης σε ρεαλιστική εικόνα, *σε πραγματικό χρόνο* (online demo).



Σχήμα 58: Εφαρμογή του μοντέλου pix2pixHD για συζευγμένη μετατροπή σκίτσου σε ρεαλιστική φωτογραφία. Οι φωτογραφίες είναι υποδειγματολειπτημένες για μείωση του μεγέθους, ωστόσο η ανώτερη ποιότητα των παραγόμενων εικόνων είναι και πάλι εμφανής.
Πηγή: «High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs», Wang et al., 2017 [84]

4.3 Μη-Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα

Ακολούθως, περνάμε στην τρίτη και τελευταία κατηγορία εφαρμογών των GANs στα πλαίσια της Παραγωγικής Μοντελοποίησης εικόνων, αυτή της Μη-Συζευγμένης Μετατροπής Εικόνας-σε-Εικόνα (Unpaired Image-to-Image Translation). Αυτή η κατηγορία διαφέρει από την προηγούμενη, δηλαδή τη Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα, καθώς εκεί για κάθε είσοδο στον Generator υπάρχει ξεκάθαρη έξοδος-στόχος, την οποία εκπαιδεύεται να προσομοιώνει. Η ύπαρξη ή δημιουργία, ωστόσο, ζευγών εικόνων (δηλ. επισημασμένου συνόλου δεδομένων) για την εκπαίδευση τέτοιων μοντέλων δεν είναι εύκολη και γενικά είναι αρκετά περιοριστική. Αντιθέτως, στη Μη-Συζευγμένη Μετατροπή δεν υπάρχει αυτή η απαίτηση και τα μοντέλα καλούνται να μάθουν μία αντιστοίχιση μεταξύ χαρακτηριστικών των εικόνων από δύο διαφορετικά πεδία και *χωρίς να είναι ζευγαρωμένες*, κάτι που φαίνεται στο σχήμα 59 παρακάτω. Η μετατροπή, για παράδειγμα, ενός πίνακα του Claude Monet σε μία ρεαλιστική φωτογραφία είναι κάτι πρακτικά αδύνατο να μοντελοποιηθεί σαν πρόβλημα Συζευγμένης Μετατροπής, λόγω της αδυναμίας κατασκευής του αντίστοιχου συνόλου δεδομένων, είναι ωστόσο κάτι που έχει αντιμετωπισθεί επιτυχώς με μοντέλα Μη-Συζευγμένης Μετατροπής, σαν και αυτά που αναφέρουμε σε αυτήν την ενότητα.



Σχήμα 59: Απεικόνιση του πως διαφοροποιείται η Μη-Συζευγμένη από τη Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα σε ότι αφορά τις απαιτήσεις του συνόλου δεδομένων εκπαίδευσης: η πρώτη απαιτεί σύνολο δεδομένων με ζεύγη εικόνων εισόδου-εξόδου, κάτι αρκετά περιοριστικό, ενώ στη δεύτερη κάτι τέτοιο δεν είναι αναγκαίο, επιτρέποντας έτσι μεγαλύτερο εύρος εφαρμογών.
Πηγή: Ανακατασκευή από «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks», Zhu et al., 2017 [84]

Για τη μετατροπή μιας εισόδου σε μία επιθυμητή έξοδο χωρίς αυτή η «επιθυμητή» να

έχει σαφώς καθοριστεί, τα μοντέλα Μη-Συζευγμένης Μετατροπής καλούνται να αφομοιώσουν ποια στοιχεία στο εκάστοτε πεδίο εικόνων αφορούν το *περιεχόμενο* (δηλ. τη σταθερή δομή) αυτών και ποια σχετίζονται με τα επιμέρους *στιλ* (δηλ. τα μεταβλητά χαρακτηριστικά) του κάθε πεδίου. Κατόπιν, καλούνται να μεταφέρουν το περιεχόμενο αντιγράφοντάς το στο πεδίο των εικόνων εξόδου, εφαρμόζοντας παράλληλα τα νέα στιλ Γίνεται φανερό, επομένως πως αυτή η κατηγορία εφαρμογών των GANs είναι σημαντικά δυσκολότερη με περισσότερες προκλήσεις για επιτυχή εκπαίδευσης, με πλεονέκτημα ωστόσο να αποτελεί η δυνατότητα χρησιμοποίησης πολύ μεγαλύτερων συνόλων δεδομένων και συνόλων δεδομένων που κατασκευάζονται πολύ πιο εύκολα και γρήγορα (μη-επιβλέψιμη εκπαίδευση). Σε ότι ακολουθεί, θα αναλύσουμε το μοντέλο CycleGAN, το πρώτο πετυχημένο παράδειγμα GAN για Μη-Συζευγμένη Μετατροπή εικόνας-σε-εικόνα το οποίο χρησιμοποιούμε και στα πλαίσια της παρούσας εργασίας, ενώ στο τέλος της ενότητας θα αναφέρουμε για λόγους πληρότητας και άλλα μοντέλα αυτής της κατηγορίας.

4.3.1 CycleGAN

Ξεκινώντας, θα αναλύσουμε τη δομή και τον τρόπο εκπαίδευσης του CycleGAN, το οποίο παρουσιάστηκε αρχικά από τον Zhou et al. το 2017 στο άρθρο τους «*Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*» [84], ενώ υποβλήθηκαν αρκετές ανανεώσεις με την τελευταία το 2020. Προκειμένου να αντιμετωπίσουν τη μεγάλη δυσκολία της Μη-Συζευγμένης Μετατροπής, οι συγγραφείς του CycleGAN πρότειναν τη χρήση δύο GANs καθένα από τα οποία καλείται να μετατρέψει μία εικόνα από το ένα πεδίο στο άλλο. Η σημαντική καινοτομία τους, ωστόσο, έγκειται στον τρόπο με τον οποίο αναγκάζουν τα δύο GANs να κάνουν έγκυρη μετατροπή μεταξύ των δύο πεδίων: *η Κυκλική Συνοχή (Cycle Consistency)*. Παρακάτω, αναλύουμε βασικά στοιχεία της αρχιτεκτονικής του μοντέλου με έμφαση να δίνεται στο σχηματισμό των συναρτήσεων κόστους για την εκπαίδευση των επιμέρους δικτύων.

Δύο GANs

Όπως αναφέρθηκε, οι συγγραφείς του CycleGAN πρότειναν τη χρήση δύο GANs, καθένα με τον δικό του Generator και Discriminator, προκειμένου να γίνει δυνατή η μη-συζευγμένη μετατροπή εικόνων σε ρεαλιστικές εικόνες του άλλου πεδίου. Έτσι, έστω G ο Generator που μετατρέπει εικόνες από το πρώτο πεδίο (έστω X) στο δεύτερο (έστω Y) και F ο Gene-

rator που μετατρέπει εικόνες από το δεύτερο πεδίο στο πρώτο. Οι συγγραφείς πρότειναν τη χρήση ενός βελτιστοποιητή (optimizer) για κοινή εκπαίδευση και των δύο Generators (δηλ. σε κάθε βήμα θα υπολογίζονται οι παράγωγοι και κατόπιν θα μεταβάλλονται από κοινού οι παράμετροι και των δύο δικτύων). Για τους Discriminators, ωστόσο, πρότειναν να χρησιμοποιηθούν ξεχωριστοί βελτιστοποιητές λόγω των διαφορών που μπορεί να υπάρχουν στο περιεχόμενο και στιλ των εικόνων του εκάστοτε πεδίου.

Έτσι, για τις συναρτήσεις κόστους εκπαίδευσης των δικτύων, δεδομένου ότι οι συγγραφείς (αρχικά) χρησιμοποίησαν την Binary Cross-Entropy, ισχύουν τα εξής:

- Η συνάρτηση κόστους για την εκπαίδευση του πρώτου GAN (Generator $G: X \longrightarrow Y$, Discriminator D_Y), σύμφωνα με την ανάλυση του προηγούμενου κεφαλαίου θα είναι:

$$\mathcal{L}_{cGAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[log(D_Y(y))] + \mathbb{E}_{x \sim p_{data}(x)}[log(1 - D_Y(G(x)))] \tag{4.20}$$

$$J_{D_Y}\left(\vec{\theta_{D_Y}}, \vec{\theta_G}\right) = -\mathcal{L}_{cGAN}(G, D_Y, X, Y) \tag{4.21}$$

$$J_G\left(\vec{\theta_G}, \vec{\theta_{D_Y}}\right) = \mathcal{L}_{cGAN}(G, D_Y, X, Y) \tag{4.22}$$

  όπου όπως και για το pix2pix ο όρος «cGAN» χρησιμοποιείται για να δηλώσει πως έχουμε υπο-συνθήκη παραγωγή και άρα στην είσοδο του εκάστοτε Discriminator θα υπάρχουν κάθε φορά δύο εικόνες συνενωμένες.

- Αντίστοιχα, η συνάρτηση κόστους για την εκπαίδευση του δεύτερου GAN (Generator $F: Y \longrightarrow X$, Discriminator D_X) θα είναι:

$$\mathcal{L}_{cGAN}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_{data}(x)}[log(D_X(x))] + \mathbb{E}_{y \sim p_{data}(y)}[log(1 - D_X(F(y)))] \tag{4.23}$$

$$J_{D_X}\left(\vec{\theta_{D_X}}, \vec{\theta_F}\right) = -\mathcal{L}_{cGAN}(F, D_X, Y, X) \tag{4.24}$$

$$J_F\left(\vec{\theta_F}, \vec{\theta_{D_X}}\right) = \mathcal{L}_{cGAN}(F, D_X, Y, X) \tag{4.25}$$

Λαμβάνοντας υπόψη την από κοινού εκπαίδευση των δύο Generators, η (αντιπαραθετική) συνάρτηση κόστους που καλούνται από κοινού να ελαχιστοποιήσουν είναι η εξής:

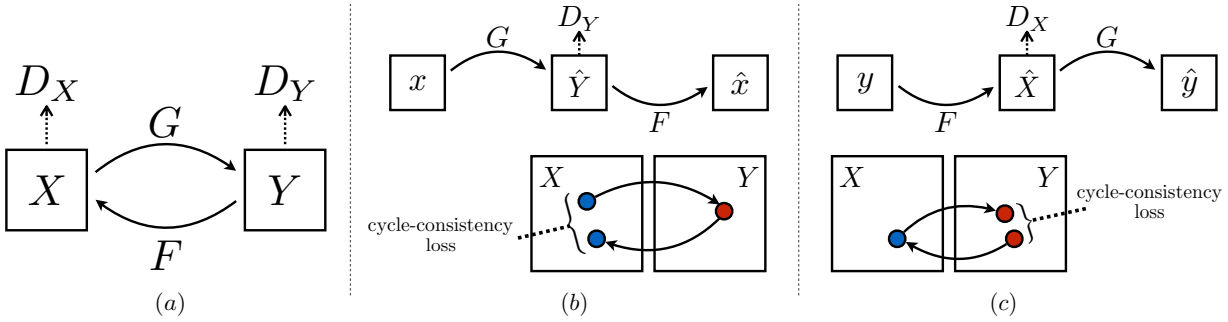
$$J_{gen}\left(\vec{\theta_G}, \vec{\theta_F}, \vec{\theta_{D_Y}}, \vec{\theta_{D_X}}\right) = \mathcal{L}_{cGAN}(G, D_Y, X, Y) + \mathcal{L}_{cGAN}(F, D_X, Y, X) \tag{4.26}$$

$$= \mathbb{E}_{x \sim p_{data}(x)}[log(1 - D_Y(G(x)))] + \mathbb{E}_{y \sim p_{data}(y)}[log(1 - D_X(F(y)))] \tag{4.27}$$

Κυκλική Συνοχή (Cycle Consistency)

Η κυκλική συνοχή, που αποτελεί την κεντρική ιδέα του τρόπου εκπαίδευσης των δικτύων του CycleGAN, προκειμένου να εκτελέσουν μη-συζευγμένη μετατροπή εικόνων διαφορετικών πεδίων, λειτουργεί ως εξής:

τροφοδότησε τον Generator του πρώτου GAN με μια εικόνα από το πρώτο πεδίο, πάρε την έξοδό του (ιδανικά ρεαλιστική και αβίαστα κατατάξιμη στις εικόνες του δεύτερου πεδίου) και τροφοδότησε τον Generator του δεύτερου GAN με την τεχνητή αυτή εικόνα του δεύτερου πεδίου με σκοπό να λάβεις στην έξοδό του μια τεχνητή εικόνα που να μοιάζει στην αρχική. Κατόπιν υπολόγισε την απόσταση μεταξύ της αρχικής εικόνας (που δόθηκε ως είσοδο στον πρώτο Generator) και της εξόδου του δεύτερου Generator με κάποια μετρική απόστασης εικόνων. Επανέλαβε την ίδια διαδικασία προς την άλλη κατεύθυνση, δηλ. συγκρίνοντας μια εικονα από το δεύτερο πεδίο που εισάγεται στον δεύτερο Generator με την έξοδο του πρώτου. Βελτιστοποίησε τα δύο δίκτυα ώστε οι αποστάσεις αυτές να γίνουν ελάχιστες, διασφαλίζοντας παράλληλα ότι οι εικόνες στην έξοδο των δικτύων είναι ρεαλιστικές (μέσω του Discriminator του εκάστοτε πεδίου).



Σχήμα 60: Απεικονιση του τρόπου εκπαίδευσης του CycleGAN.

Πηγή: «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks», Zhu et al., 2017 [84]

Στο σχήμα 60 δίνεται διαγραμματικά ο τρόπος εκπαίδευσης του μοντέλου CycleGAN. Εκεί, αριστερά (a), φαίνονται τα δύο πεδία εικόνων, X και Y, οι δύο Generator, G και F και οι δύο Discriminators, D_X και D_Y. Στο κέντρο (b) φαίνεται ο τρόπος υπολογισμού του κόστους από την Κυκλική Συνοχή (απόσταση x και $\hat{x}$) του πεδίου X και δεξιά (c) (απόσταση y και $\hat{y}$) του πεδίου Y. Στο CycleGAN, η απόσταση αυτή υπολογίζεται στον χώρο των εικονοστοιχείων, π.χ. με την Ευκλείδεια (L_2) απόσταση.

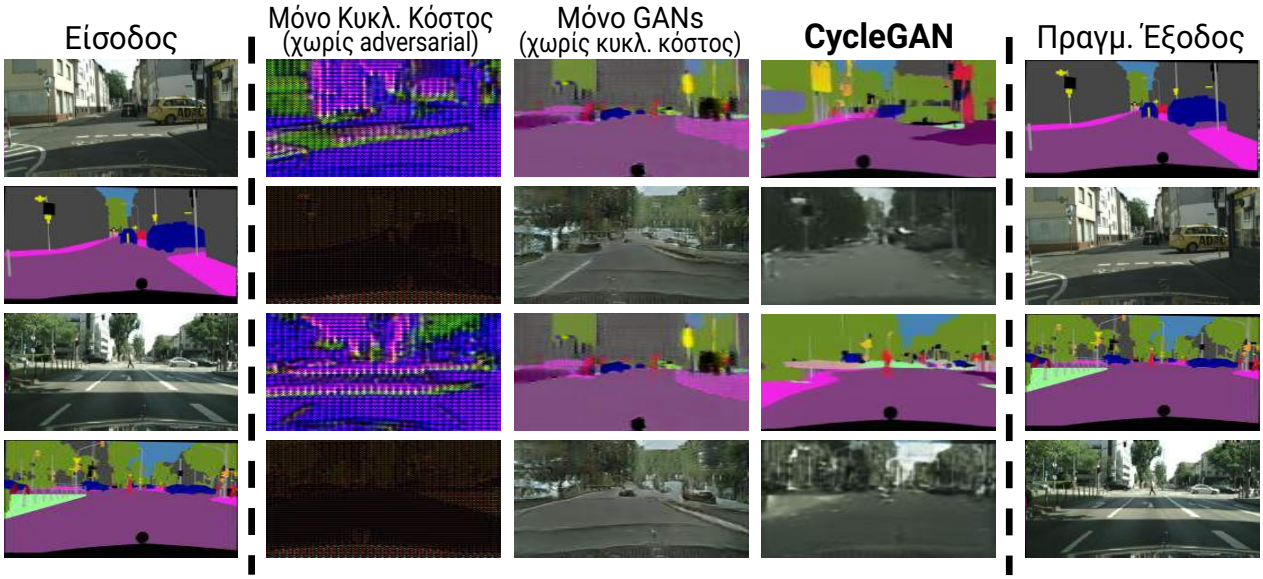
Η όρος «κυκλική» δόθηκε στην παραπάνω διαδικασία καθώς οι διαδοχικές μετατροπές σχηματίζουν ένα κύκλο μεταξύ των πεδίων εικόνων, κάτι που φαίνεται και στο σχήμα 60 που προηγείται. Για την υλοποίηση της κυκλικής αυτής διαδρομής χρησιμοποιώντας τα δύο (2) GANs που αναφέρθηκαν, οι συγγραφείς του [84] πρότειναν σε κάθε βήμα να υπολογίζεται ένας όρος κόστους για την Κυκλική Συνοχή, τοποθετώντας τους δύο Generators σε σειρά, μία φορά τον πρώτο ακολουθούμενο από το δεύτερο ($X \xrightarrow{G} Y \xrightarrow{F} \hat{X}$)

και μία αντίστροφα ($Y \xrightarrow{F} X \xrightarrow{G} \hat{Y}$). Έτσι, όπως φαίνεται και στο σχήμα, σε κάθε βήμα υπολογίζονται δύο αποστάσεις (ή κόστη) τις οποίες τα δίκτυα εκπαιδεύονται να μειώσουν, ενθαρρύνοντας έτσι την Κυκλική Συνοχή. Επομένως, εκτός του όρου αντιπαραθετικού κόστους (adversarial loss) στην από κοινού συνάρτηση κόστους των δύο Generators (σχέση 4.27), προστίθεται και ένας όρος που ενθαρρύνει την κυκλική συνοχή μέσω της απόστασης εικονοστοιχείων των εικόνων στην αρχή και το τέλος του κάθε «κύκλου», με αποτέλεσμα η συνάρτηση κόστους που καλούνται από κοινού να ελαχιστοποιήσουν οι δύο Generators θα είναι:

$$\begin{aligned}
\mathcal{L}_{cyc}(G,F) &= \mathcal{L}_{cyc}\left[X \xrightarrow{G} Y \xrightarrow{F} \hat{X}\right] + \mathcal{L}_{cyc}\left[Y \xrightarrow{F} X \xrightarrow{G} \hat{Y}\right] \\
&= \mathbb{E}_{x\sim p_{data}(x)}\left[\|F(G(x)) - x\|_1\right] + \mathbb{E}_{y\sim p_{data}(y)}\left[\|G(F(y)) - y\|_1\right] \qquad (4.28) \\
J_{gen}\left(\vec{\theta_G}, \vec{\theta_F}, \vec{\theta_{D_Y}}, \vec{\theta_{D_X}}\right) &= \mathcal{L}_{cGAN}(G, D_Y, X, Y) + \mathcal{L}_{cGAN}(F, D_X, Y, X) + \lambda_{cyc} * \mathcal{L}_{cyc}(G,F) \qquad (4.29) \\
&= \mathbb{E}_{x\sim p_{data}(x)}[log\,(1 - D_Y(G(x)))] + \mathbb{E}_{y\sim p_{data}(y)}[log\,(1 - D_X(F(y)))] \\
&+ \lambda_{cyc} * \left[\mathbb{E}_{x\sim p_{data}(x)}\left[\|F(G(x)) - x\|_1\right] + \mathbb{E}_{y\sim p_{data}(y)}\left[\|G(F(y)) - y\|_1\right]\right] \qquad (4.30)
\end{aligned}$$

όπως όπως φαίνεται ως συνάρτηση απόστασης μεταξύ των εικόνων στην αρχή και το τέλος του κάθε κύκλου χρησιμοποιείται η L_1 ή Manhattan (αντί για την L_2 ή Ευκλείδεια, που όπως είναι γνωστό οδηγεί σε πιο θολές εικόνες), ενώ λ_{cyc} είναι το βάρος του όρου κυκλικού κόστους (cyclic loss). Το βάρος αυτό, συμπερίληψης του κυκλικού κόστους, συνήθως είναι αρκετά μεγαλύτερο της μονάδας, ενώ οι συγγραφείς προτείνουν μία τιμή κοντά στο 10, λόγω του ότι είναι αρκετά σημαντικότερο και δυσκολότερο το έργο επίτευξης της Κυκλικής Συνοχής απ΄ ότι της εκμάθησης ρεαλιστικών χαρακτηριστικών μέσω των Discriminators. Κάτι άλλο σχετικά με το βάρος συμπερίληψης του όρου κυκλικού κόστους που διαπίστωσαν οι συγγραφείς στις αφαιρετικές μελέτες (ablation studies) που διεξήγαγαν είναι ότι η έλλειψη του όρου αυτού ($\lambda_{cyc} = 0$) οδηγεί στο πρόβλημα Συρρίκνωσης Ρυθμών σε αμφοτέρους τους Generators, κάτι που φαίνεται στο σχήμα 61 που έπεται.

Όπως φαίνεται από τις αφαιρετικές μελέτες της συνάρτησης κόστους του σχήματος 61 που κατέθεσαν οι συγγραφείς, η μη χρήση των όρων αντιπαραθετικής εκπαίδευσης (δηλ. μη-χρήση και των Discriminators) οδηγεί σε μη-ρεαλιστικά, ποιοτικά υποβαθμισμένα αποτελέσματα. Η μη-χρήση, ωστόσο, του όρου κυκλικής συνοχής, αν και οδηγεί σε καλύτερα αποτελέσματα, αυτά υποφέρουν από Συρρίκνωση Ρυθμών. Στις δύο τελευταίες στήλες φαίνονται η έξοδος του CycleGAN και η πραγματική έξοδος αντίστοιχα, όπου φαίνεται ότι η ταυτόχρονη συμπερίληψη των όρων αντιπαραθετικού και κυκλικού κόστους οδηγεί



Σχήμα 61: Αφαιρετικές μελέτες της συνάρτησης κόστους των Generators κατά την εκπαίδευση τους σε σύνολο δεδομένων χαρτών κατάτμησης-εικόνων (χωρίς ζεύγη εικόνων, ασχέτως που το αρχικό σύνολο δεδομένων ήταν φτιαγμένο για συζευγμένη μετατροπή).
Πηγή: Ανακατασκευή από «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks», Zhu et al., 2017 [84]

σε ρεαλιστικά αποτελέσματα με ποικιλομορφία.

Ταυτοτικό Κόστος (Identity Loss)

Ένας πρόσθετος όρος που δοκίμασαν οι συγγραφείς του CycleGAN κατά τη διάρκεια δοκιμής του μοντέλου σε σύνολο δεδομένων τοπίων με στόχο τη μη-συζευγμένη μετατροπή εικόνων από μία εποχή (π.χ. τοπία το καλοκαίρι) σε μία άλλη (π.χ. τοπία τον χειμώνα), είναι αυτός του Ταυτοτικού Κόστους (Identity Loss). Συγκεκριμένα, οι συγγραφείς ήθελαν να ενθαρρύνουν τους Generators να μάθουν ταυτοτικές συναρτήσεις όταν στην είσοδό τους δίνονται εικόνες από το πεδίο εξόδου (αντί για εισόδου) τους. Μέσω αυτού του όρου, βοήθησαν τους Generators να αντιστοιχίζουν καλύτερα τα χρώματα της εικόνας εισόδου και έτσι να πετυχαίνουν πιο ρεαλιστικά αποτελέσματα. Αναλυτικά, ο όρος που πρότειναν στο [84], καθώς και η νέα συνολική συνάρτηση κόστους που καλούνται από κοινού να ελαχιστοποιήσουν οι δύο Generators, έχουν ως εξής:

$$\begin{aligned}\mathcal{L}_{identity}(G,F) &= \mathcal{L}_{identity}\left[Y \xrightarrow{G} \hat{Y}\right] + \mathcal{L}_{identity}\left[X \xrightarrow{F} \hat{X}\right] \\ &= \mathbb{E}_{y\sim p_{data}(y)}\left[\|G(y)-y\|_1\right] + \mathbb{E}_{x\sim p_{data}(x)}\left[\|F(x)-x\|_1\right] \end{aligned} \tag{4.31}$$

$$J_{gen}\left(\vec{\partial_G}, \vec{\partial_F}, \vec{\partial_{D_Y}}, \vec{\partial_{D_X}}\right) = \mathcal{L}_{cGAN}(G, D_Y, X, Y) + \mathcal{L}_{cGAN}(F, D_X, Y, X) +$$
$$+ \lambda_{cyc} * \mathcal{L}_{cyc}(G, F) \; + \; \lambda_{identity} * \mathcal{L}_{identity}(G, F) \tag{4.32}$$
$$= \mathbb{E}_{x \sim p_{data}(x)}[log\,(1 - D_Y\,(G\,(x)))] + \mathbb{E}_{y \sim p_{data}(y)}[log\,(1 - D_X\,(F\,(y)))]$$
$$+ \lambda_{cyc} * \left[\mathbb{E}_{x \sim p_{data}(x)}\left[\|F\,(G\,(x)) - x\|_1\right] + \mathbb{E}_{y \sim p_{data}(y)}\left[\|G\,(F\,(y)) - y\|_1\right]\right]$$
$$+ \lambda_{identity} * \left[\mathbb{E}_{y \sim p_{data}(y)}\left[\|G\,(y) - y\|_1\right] + \mathbb{E}_{x \sim p_{data}(x)}\left[\|F\,(x) - x\|_1\right]\right] \tag{4.33}$$

με τον συντελεστή συμπερίληψης της identity loss να εξαρτάται έντονα από το σύνολο δεδομένων εκπαίδευσης, με τους συγγραφείς να προτείνουν μία τιμή κοντά στο 5 για το σύνολο δεδομένων που δοκίμασαν τη χρήση της.

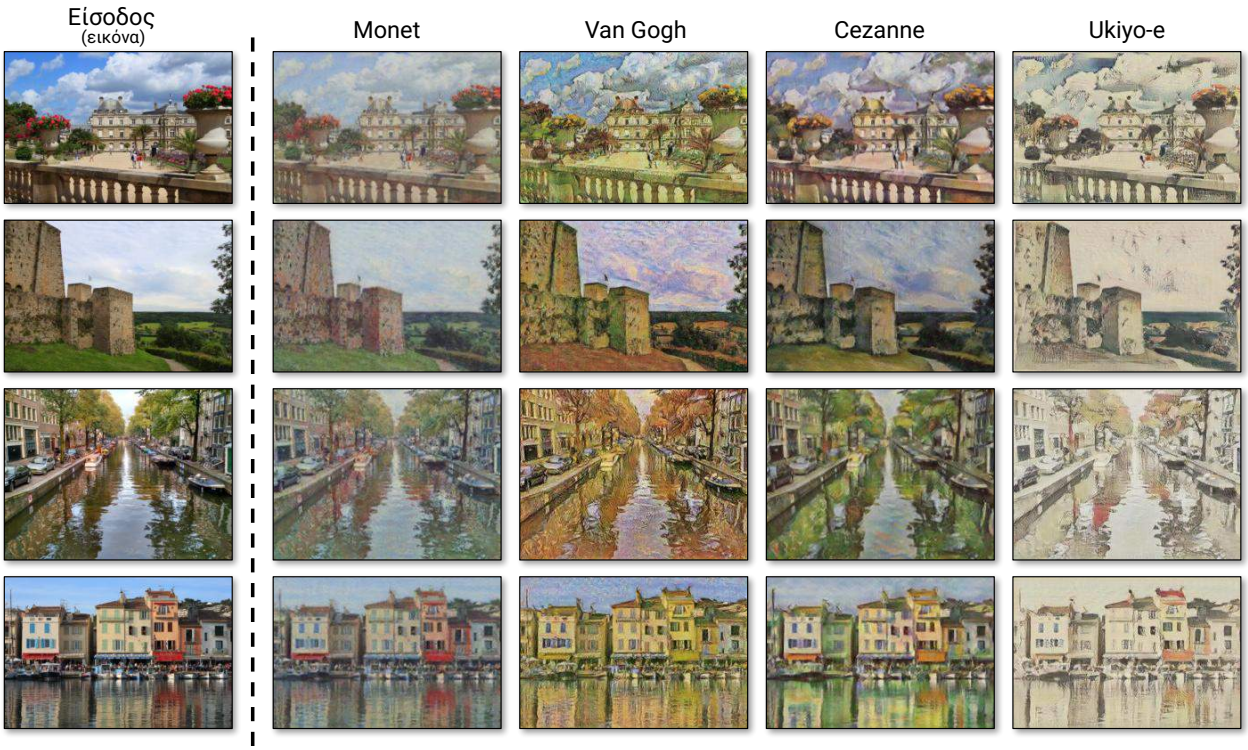
Αρχιτεκτονικη Δικτύων - Αποτελέσματα

Έχοντας πλέον το στήσιμο για την εκπαίδευση των δύο GANs και τις συναρτήσεις κόστους που καλούνται να υλοποιήσουν, μένει η σχεδίαση της αρχιτεκτονικής των δικτύων και η εκπαίδευσή τους. Έτσι, οι συγγραφείς αποφάσισαν οι δύο Generators να είναι πανομοιότυποι, όπως και οι δύο Discriminators. Συγκεκριμένα πρότειναν τις εξής αρχιτεκτονικές:

- **Generators**: δύο (2) συνελικτικές στρώσεις ακολουθούμενες από στρώσεις Κανονικοποίησης Δείγματος και στρώσεις ενεργοποίησης ReLU, αποτελούν το υποδίκτυο του encoder του κάθε Generator. Ακολουθεί το υποδίκτυο στένωσης που αποτελείται από συνελικτικές στρώσεις μοναδιαίου βήματος (δηλ. δεν μεταβάλλουν το πλάτος/ύψος των χαρτών ενεργοποίησης στην είσοδό τους) ακολουθούμενες από όμοιες στρώσεις κανονικοποίησης και ενεργοποίησης, οι οποίες όμως έχουν residual connections μεταξύ των εισόδων και εξόδων τους. Πρότειναν τη χρήση εννέα τέτοιων ομάδων στρώσεων. Τέλος, το υποδίκτυο του κάθε Generator είναι συμμετρικό αυτό του encoder και χρησιμοποιεί ανάστροφες συνελικτικές στρώσεις.
- **Discriminators**: ως Discriminator για κάθε GAN οι συγγραφείς χρησιμοποίησαν τον PatchGAN Discriminator από το pix2pix με ίδια δομή και σχεδιαστικές παραμέτρους όπως αναλύθηκαν εκεί (βλ. σχήμα 56).

Παρακάτω παρουσιάζουμε τα αποτελέσματα εκπαίδευσης των δικτύων σε σύνολο δεδομένων που περιέχει στο ένα πεδίο ρεαλιστικές εικόνες και στο άλλο εικόνες από πίνακες ενός ζωγράφου. Συγκεκριμένα, εκπαιδεύτηκαν τέσσερα (4) διαφορετικά μοντέλα

σε σύνολα δεδομένων εικόνων των αντίστοιχων ζωγράφων, όλα για 100 epochs[5] σταθερού βήματος εκμάθησης (learning rate) ίσου με 0.0002 και 100 όπου το βήμα γραμμικά μειώνονταν στην αρχή του καθενός έως το 0. Επίσης, σε επόμενη έκδοση του άρθρου τους, οι συγγραφείς χρησιμοποιούν ως συνάρτηση αντιπαραθετικής εκπαίδευσης των GANs τη συνάρτηση κόστους Ελαχίστων Τετραγώνων αντί της Binary Cross-Entropy όπως δίνεται στις σχέσεις 4.25 και 4.22, για λόγους που αναφέρθηκαν στην αρχή του προηγούμενου κεφαλαίου. Οι παραγωγές του πρώτου Generator (εικόνες σε πίνακες) του κάθε μοντέλου από τα τέσσερα που εκπαιδεύτηκαν δίνονται στο σχήμα που ακολουθεί, ενώ στο [84] δίνονται αρκετές ακόμα εφαρμογές.



Σχήμα 62: Παραγωγές τεσσάρων (ίδιων) μοντέλων CycleGAN που έχουν εκπαιδευθεί σε σύνολα δεδομένων αποτελούμενα από πραγματικές εικόνες (πρώτο πεδίο - κοινό σε όλα τα σύνολα δεδομένων εκπαίδευσης) και πίνακες γνωστών ζωγράφων. Βλέπουμε, ότι παρόλο που τα μοντέλα εκπαιδεύτηκαν χωρίς ζεύγη εικόνων (unsupervised training), μπορούν να μεταφέρουν επιτυχώς το περιεχόμενο της εικόνας εισόδου στην έξοδο εφαρμόζοντας κατόπιν τα στιλ του αντίστοιχου πεδίου.

Πηγή: Ανακατασκευή από «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks», Zhu et al., 2017 [84]

[5]Ως epoch νοείται η ακολουθία όλων των βημάτων ώστε να περάσουν μέσα από ένα μοντέλο όλα τα δείγματα του συνόλου δεδομένων εκπαίδευσης. Σε κάθε βήμα περνάει μία ομάδα (batch) δειγμάτων.

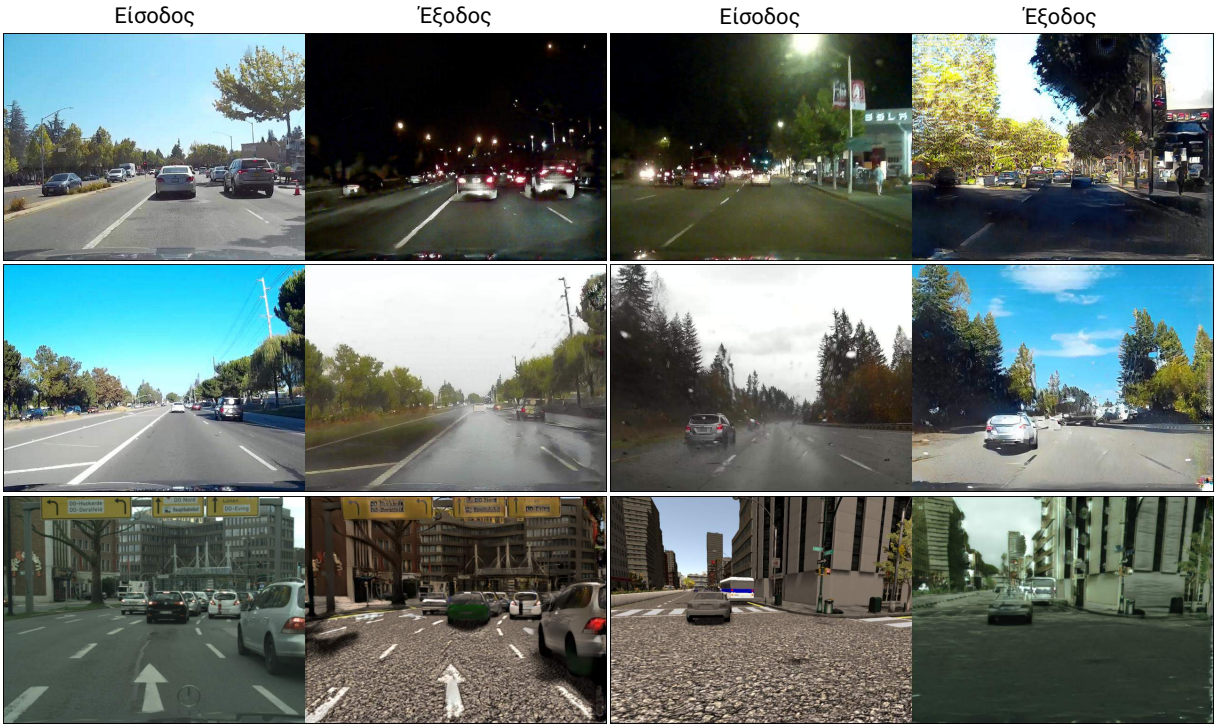
4.3.2 UNIT & MUNIT

Τέλος και για λόγους πληρότητας, αναφέρουμε σε αυτήν υποενότητα άλλα μοντέλα που σχεδιάστηκαν για Μη-Συζευγμένη Μετατροπή εικόνας-σε-εικόνα, τα οποία υλοποιήθηκαν και εκπαιδεύτηκαν από ερευνητές του ίδιου εργαστηρίου της εταιρείας NVIDIA όπως και τα pix2pix και pix2pixHD. Έτσι, αρχικά θα αναφέρουμε συνοπτικά την ιδέα πίσω από το μοντέλο UNIT και αποτελέσματα από την εφαρμογή του, ενώ στο τέλος θα παραθέσουμε και ορισμένα στοιχεία και αποτελέσματα για την αναβάθμισή του, το μοντέλο MUNIT.

UNIT

Βασική ιδέα του μοντέλου UNIT, που παρουσιάστηκε από τον Liu et al. στο άρθρο του «*Unsupervised Image-to-Image Translation Networks*» [79], είναι η ύπαρξη ενός διαμοιραζόμενου λανθάνοντα χώρου μεταξύ δύο πεδίων και η χρήση δικτύων MAK (βλ. υποενότητα 2.3) και GANs για παραγωγή διανυσμάτων αυτού του χώρου και κατόπιν παραγωγή ρεαλιστικών εικόνων. Αυτή η «υβριδική» αρχιτεκτονική έκανε δυνατό τον μετασχηματισμό ενός διανύσματος αυτού του κοινού χώρου σε μία ρεαλιστική εικόνα σε κάθε ένα από τα δύο πεδία, ενθαρρύνοντας κατά την εκπαίδευση του μοντέλου η μετατροπή να γίνεται διατηρώντας κάποια σύζευξη. Έμπνευση των συγγραφέων αποτέλεσε το ότι η μετατροπή μιας εικόνας μεταξύ δύο πεδίων, αφήνει κάποια στοιχεία αυτής αναλλοίωτα, όπως π.χ. τη δομή κάποιων αντικειμένων, τη θέση τους ή το περίγραμμα αυτών. Αυτό που διαφέρει είναι τα επιμέρους στιλ του κάθε πεδίου.

Δεν θα επεκταθούμε περαιτέρω στην ανάλυση, απλώς θα αναφέρουμε πως το UNIT χρησιμοποιεί δύο MAK με coupling μεταξύ τους για την εκμάθηση του λανθάνοντα αυτού χώρου, ενώ για την παραγωγή ρεαλιστικών εικόνων χρησιμοποιεί, όπως και το CycleGAN, δύο GANs (δύο (2) Generators + δύο (2) Discriminators). Στο σχήμα που ακολουθεί φαίνονται μερικές παραγωγές του μοντέλου UNIT, που δικαιώνουν την επιλογή των συγγραφέων να δημιουργήσουν και να εκπαιδεύσουν αυτό το αρκετά σύνθετα μοντέλο, αποτελούμενο από έξι (6) επιμέρους νευρωνικά δίκτυα (δύο Generators, δύο Discriminators και δύο MAK).



Σχήμα 63: Παραγωγές του μοντέλου UNIT.

Πηγή: Ανακατασκευή από «Unsupervised Image-to-Image Translation Networks», Liu et al., 2017 [79]

MUNIT

Οι συγγραφείς του MUNIT ένα χρόνο αργότερα παρουσίασαν μια αναβαθμισμένη έκδοση του μοντέλου τους. Πρόκειται για το μοντέλο MUNIT και το άρθρο «*Multimodal Unsupervised Image-to-Image Translation*» [90], βασική ιδέα του οποίου είναι η δυνατότητα παραγωγής περισσότερων από μία ρεαλιστικές εικόνες για κάθε διάνυσμα του λανθάνοντα χώρου και σε κάθε ένα από τα δύο πεδία του συνόλου δεδομένων εκπαίδευσης. Για να το πετύχουν αυτό, πρότειναν τη διάσπαση του αρχικού λανθάνοντα χώρου σε δύο (έναν για το περιεχόμενο και έναν για το στιλ), ενώ επίσης διέσπασαν ακόμα περισσότερο τα δίκτυα του μοντέλου τους, καταλήγοντας σε τέσσερις (4) MAK (δύο ανά πεδίο - ένας για παραγωγή του λανθάνοντος διανύσματος περιεχομένου και ένας για το λανθάνον διάνυσμα στιλ της εκάστοτε εικόνας εισόδου). Για την εκπαίδευση του μοντέλου τους, δανείστηκαν τη δομή του Discriminator από το μοντέλο pix2pixHD και τη λογική και αρχιτεκτονική των Generators του UNIT. Ο ενδιαφερόμενος αναγνώστης παραπέμπεται στο σχετικό άρθρο, ενώ ακολούθως παραθέτουμε μερικές χαρακτηριστικές παραγωγές του μοντέλου MUNIT, με τις οποίες ολοκληρώνουμε το παρόν κεφάλαιο.

(α) σκίτσα (ακμές) ⇿ παπούτσια (β) σκίτσα (ακμές) ⇿ τσάντες

Σχήμα 64: Παραγωγές του μοντέλου MUNIT.

Πηγή: Ανακατασκευή από «Multimodal Unsupervised Image-to-Image Translation», Huang et al., 2018 [90]

Κεφάλαιο 5

Εφαρμογή GANs σε Παραγωγή Εικόνων Μόδας - Μεθοδολογία

Περνάμε τώρα στον πυρήνα της παρούσας εργασίας, που είναι η εφαρμογή Generative Adversarial Networks (GANs) σε σύνολα δεδομένων εικόνων μόδας, δηλαδή σύνολα δεδομένων που περιέχουν ρούχα ή/και μοντέλα που τα φορούν και τα διαφημίζουν. Στα πλαίσια αυτής, επομένως, εκπαιδεύτηκαν αρχιτεκτονικές GANs όμοιες με αυτές που περιγράφηκαν στο προηγούμενο κεφάλαιο, ή παραλλαγές αυτών, σε σύνολα δεδομένων εικόνων μόδας. Στο παρόν κεφάλαιο θα περιγράψουμε τόσο τα σύνολα δεδομένων εκπαίδευσης όσο και τον σχεδιασμό και τον τρόπο εκπαίδευσης των μοντέλων που χρησιμοποιήθηκαν, ενώ αφήνουμε τα αποτελέσματα και τις μετρικές αξιολόγησης για το επόμενο κεφάλαιο.

Στην πρώτη ενότητα του κεφαλαίου θα εστιάσουμε στα σύνολα δεδομένων εκπαίδευσης, δηλαδή στις εικόνες μόδας που χρησιμοποιήθηκαν για την εκπαίδευση των μοντέλων που περιγράφονται στην επόμενη ενότητα. Μαζί με τα σύνολα δεδομένων θα αναφερθούμε και στις τεχνικές προεπεξεργασίας αυτών που χρησιμοποιήθηκαν και σε άλλου παραμέτρους εκπαίδευσης, ενώ μαζί με την περιγραφή των μοντέλων που εκπαιδεύτηκαν θα δοθούν όλες οι παράμετροι στησίματος (setup) και εκπαίδευσης αυτών.

5.1 Συνολα Δεδομένων Εικόνων Μόδας

Για την εκπαίδευση των μοντέλων της παρούσας εργασίας χρησιμοποιήθηκαν τέσσερα διακριτά σύνολα δεδομένων (datasets): δύο από την οικογένεια συνόλων δεδομένων DeepFashion, το σύνολο δεδομένων LookBook και το σύνολο δεδομένων handbags2shoes αποτελούμενο από δύο επιμέρους σύνολα δεδομένων, το edge2handbag και το edge2shoe, όπως αναλύονται στις υποενότητες που ακολουθούν. Τα σύνολα δεδομένων αυτά χρησιμοποιήθηκαν κατά την εκπαίδευση των αντίστοιχων μοντέλων με σκοπό την κάλυψη και των τριών κατηγοριών εφαρμογών, δηλαδή για παραγωγή εικόνας από θόρυβο, συζευγμένη μετατροπή εικόνας-σε-εικόνα και μη-συζευγμένη μετατροπή εικόνας-σε-εικόνα.

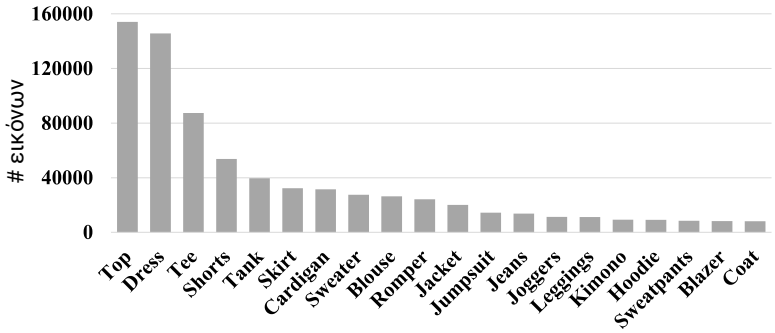
Όλα τα σύνολα δεδομένων που χρησιμοποιήθηκαν περιέχουν σχετικά χαμηλής ανάλυσης εικόνες με τα τρία (3) από τα τέσσερα (4) να περιέχουν εικόνες 64×64, ενώ το ένα περιέχει εικόνες 128×128. Ο λόγος που προχωρήσαμε σε αυτήν την επιλογή είναι ότι με τετοια σύνολα δεδομένων χρειάζονται σημαντικά λιγότεροι και μικρότερης ισχύος υπολογιστικοί πόροι (χώροι αποθήκευσης, χωρητικότητες μνημών RAM και μνημών καρτών γραφικών κλπ.). Επίσης, αυτό μας επέτρεψε να «μικρύνουμε» τα μοντέλα μας, κάνοντας έτσι την εκπαίδευσή τους πιο γρήγορη και πιο ευσταθή. Ακολουθεί μία συνοπτική περιγραφή του κάθε συνόλου δεδομένων εκπαίδευσης που χρησιμοποιήθηκε καθώς και των τεχνικών προεπεξεργασίας αυτών.

5.1.1 DeepFashion

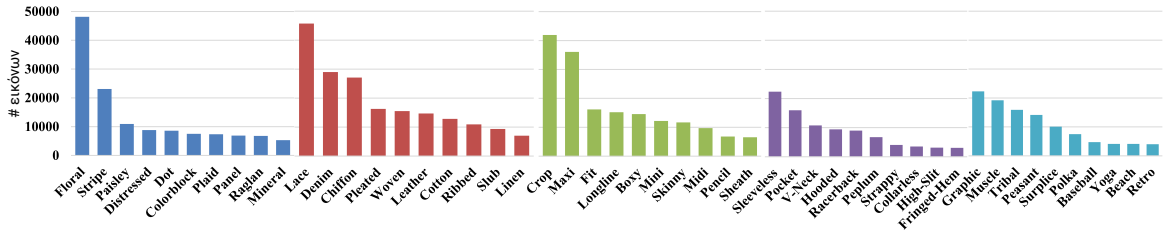
Τα πρώτο σύνολο δεδομένων που χρησιμοποιήθηκε ανήκει στην οικογένεια συνόλων δεδομένων DeepFashion. Τα σύνολα δεδομένων εκπαίδευσης μοντέλων τεχνητής νοημοσύνης DeepFashion παρουσιάστηκαν από τον Liu et al. το 2016 συνοδεύοντας το άρθρο τους «*DeepFashion: Powering Robust Clothes Recognition and Retrieval With Rich Annotations*» [55]. Συνολικά, το DeepFashion αποτελεί το μεγαλύτερο σύνολο δεδομένων εικόνων μόδας, με περισσότερες από 800.000 ποικιλόμορφες εικόνες οι οποίες καλύπτουν ένα ευρύ φάσμα κατηγοριών, από καλά-κεντραρισμένες (well-posed) εικόνες ρούχων από καταστήματα έως φωτογραφίες καταναλωτών χωρίς κανένα περιορισμό (αντληθείσες από κοινωνικά μηχανές αναζήτησης εικόνων). Επιπρόσθετα, το DeepFashion είναι ένα επιμελώς επισημασμένο σύνολο δεδομένων, με κάθε εικόνα να έχει επισημανθεί σχετικά με το σε ποια κατηγορία ρούχων ανήκει (από τις 50 συνολικά του συνόλου δεδομένων), με ένα σύνολο χαρακτηριστικών της (από 1000 συνολικά χαρακτηριστικά), με σημάδια

των ορίων του ρούχου (landmarks) καθώς και με περιγραφικές ετικέτες από τον χρήστη που την ανέβασε (user-generated metadata). Στο σχήμα 65 που ακολουθεί τα παραπάνω φαίνονται με μεγαλύτερη σαφήνεια.

(α) Παράδειγμα εικόνων διαφορετικών κατηγοριών και χαρακτηριστικών στο DeepFashion. Τα χαρακτηριστικά κατηγοριοποιούνται σε πέντε ομάδες: υφή, ύφασμα, σχήμα, μέρος και στιλ.



(β) Αριθμός εικόνων των πρώτων 20 σε πληθυσμό κατηγοριών.



(γ) Αριθμός εικόνων των πρώτων 10 σε πληθυσμό χαρακτηριστικών σε κάθε ομάδα.

Σχήμα 65: Στοιχεία του συνόλου δεδομένων DeepFashion σχετικά με τον τρόπο επισήμανσής του και το διαμοιρασμό των εικόνων στις επιμέρους κατηγορίες / χαρακτηριστικά.

Πηγή: Ανακατασκευή από «DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations», Liu et al., 2016 [55]

Για την καλύτερη οργάνωση και χρήση του συνόλου δεδομένων τους, οι δημιουργεί του DeepFashion πρότειναν την περαιτέρω διάσπασή του σε υποσύνολα δεδομένων για ξεχωριστές κατηγορίες εφαρμογών, τα οποία ονομάζουν Benchmarks. Τα Benchmarks αυτά, επομένως, που συμπεριλαμβάνονται στο DeepFashion είναι τα εξής:

- «**Category and Attribute Prediction Benchmark**»: σύνολο δεδομένων που προορίζεται για ταξινόμηση εικόνων. Περιέχει 63.720 εικόνες με ετικέτες από 50 αμοιβαίως-αποκλειόμενες κατηγορίες και από 1000 χαρακτηριστικά (π.χ αμάνικο/όχι-αμάνικο, μάξι/όχι-μάξι κλπ.). Ένα τέτοιο σύνολο δεδομένων είναι καλώς στημένο για εκπαίδευση ταξινομητών εικόνας, σαν και αυτούς που είδαμε στα Διακριτικά Μοντέλα (βλ. υποενότητα 2.1).

- «**In-Shop Clothes Retrieval Benchmark (ICRB)**»: το δεύτερο σύνολο δεδομένων του DeepFashion και ταυτόχρονα το πρώτο που κάνουμε χρήση, είναι το In-Shop Clothes Retrieval Benchmark (ICRB), το οποίο περιέχει 54.642 εικόνες από 11.735 προϊόντα του ηλεκτρονικού καταστήματος της εταιρείας Forever21. Παρουσιάζουμε λεπτομερώς αυτό το σύνολο δεδομένων στην παράγραφο 5.1.1.1 που ακολουθεί.

- «**Consumer-to-Shop Clothes Retrieval Benchmark (CCRB)**»: πρόκειται για ένα ιδιαίτερο σύνολο δεδομένων το οποίο περιέχει 251.361 ζεύγη εικόνων με την πρώτη εικόνα να είναι το ρούχο φορεμένο σε ένα μοντέλο και την άλλη το ίδιο ρούχο φορεμένα σε κάποιον καταναλωτή σε αυθαίρετες συνθήκες φωτογράφισης. Οι εικόνες έχουν συλλεχτεί από την πλατφόρμα ηλεκτρονικών καταστημάτων Mogujie της Κίνας. Σκοπός του συνόλου δεδομένων είναι η δημιουργία μοντέλων συζευγμένης μετατροπής εικόνων από κοινές φωτογραφίες των κοινωνικών δικτύων σε φωτογραφίες όπου το ρούχο είναι φορεμένο σε ένα μοντέλο στο κατάστημα.

- «**Fashion Image Synthesis Benchmark (FISB)**»: αποτελώντας την τελευταία προσθήκη στα επιμέρους benchmarks του DeepFashion, το Fashion Image Synthesis Benchmark (FISB) περιέχει 78.979 εικόνες, κάθε μία επισημασμένη με μία λεκτική περιγραφή και έναν χάρτη κατάτμησης. Παρουσιάζουμε λεπτομερώς αυτό το σύνολο δεδομένων στην παράγραφο 5.1.1.2 παρακάτω.

Μετά από αυτήν τη σύντομη περιγραφή του συνόλου δεδομένων DeepFashion, ακολουθούν τα επιμέρους σύνολα δεδομένων που χρησιμοποιήσαμε για την εκπαίδευση κάποιων από τα μοντέλα της παρούσας εργασίας. Συγκεκριμένα, στις παραγράφους που ακολουθούν, θα αναλύσουμε τα benchmarks In-Shop Clothes Retrieval και Fashion Image Synthesis,

τα οποία θα χρησιμοποιήσουμε για εκπαίδευση μοντέλου αλλαγής πόζας και παραγωγής ρεαλιστικών εικόνων μόδας αντίστοιχα.

In-Shop Clothes Retrieval Benchmark (ICRB)

Έτσι, αρχικά θα εστιάσουμε στο υποσύνολο δεδομένων του DeepFashion, το In-Shop Clothes Retrieval Benchmark (ICRB). Όπως αναφέρθηκε, αυτό περιέχει 54.642 εικόνες, από 11.735 προϊόντα της εταιρείας Forever21, τα οποία στη συντριπτική τους πλειοψηφία είναι φωτογραφημένα σε ανθρώπους-μοντέλα. Όλες οι φωτογραφίες είναι ποιότητας στούντιο με ουδέτερα συνήθως παρασκήνια και ανάλυσης 256×256 (τουλάχιστον στη χαμηλής-ανάλυσης έκδοση του συνόλου δεδομένων που χρησιμοποιήθηκε στα πλαίσια της παρούσας). Στόχος μας με αυτό το σύνολο δεδομένων ήταν να εκπαιδεύσουμε ένα μοντέλο GAN το οποίο θα αλλάζει την πόζα του ανθρώπου που εικονίζεται στην εικόνα εισόδου. Το μοντέλο αυτό, το PGPG (βλ. υποενότητα 5.2.1) απαιτεί, λοιπόν, εκτός από τη φωτογραφία εισόδου και εξόδου (συζευγμένη μετατροπή) να υπάρχει και πληροφορία για την πόζα της εικόνας εξόδου. Για καλή μας τύχη, οι δημιουργοί του DeepFashion συμπεριέλαβαν στο συγκεκριμένο benchmark και τις πόζες των ανθρώπων που εικονίζονται και μάλιστα πλούσιες περιγραφές αυτών (rich pose annotations) που εξήχθησαν από το μοντέλο DensePose [87] το οποίο έχει ενσωματωθεί πλέον στη σουίτα μοντέλων της Facebook, Detectron [89].

Στο σημείο αυτό να σημειώσουμε ότι η χρήση DensePose εικόνων για την περιγραφή της πόζας μιας εικόνας εισόδου ή της αναμενόμενης εικόνας εξόδου ενός παραγωγικού μοντέλου, περιέχει αρκετά περισσότερη πληροφορία σε σύγκριση με άλλες τεχνικές όπως η εξαγωγή σημείων των αρθρώσεων από μοντέλα συνελικτικών νευρωνικών δικτύων, ακόμα και με καλές εκδοχές αυτών (όπως στο [72]). Ωστόσο οι δημιουργοί του ICRB δεν συμπεριέλαβαν τις πόζες όλων των εικόνων του συνόλου δεδομένων τους, ούτε παρείχαν κάποια πληροφορία σχετικά με ποια προϊόντα ή ποιοι τύποι εικόνων έχουν συνυφασμένη εικόνα πόζας. Το πρώτο βήμα επομένως της προ-επεξεργασίας του συνόλου δεδομένων αποτέλεσε ο διαχωρισμός των εικόνων σε αυτές που έχουν και αυτές που δεν έχουν εικόνα πόζας, κάτι που αναλύεται παρακάτω.

Δομή του συνόλου δεδομένων ICRB

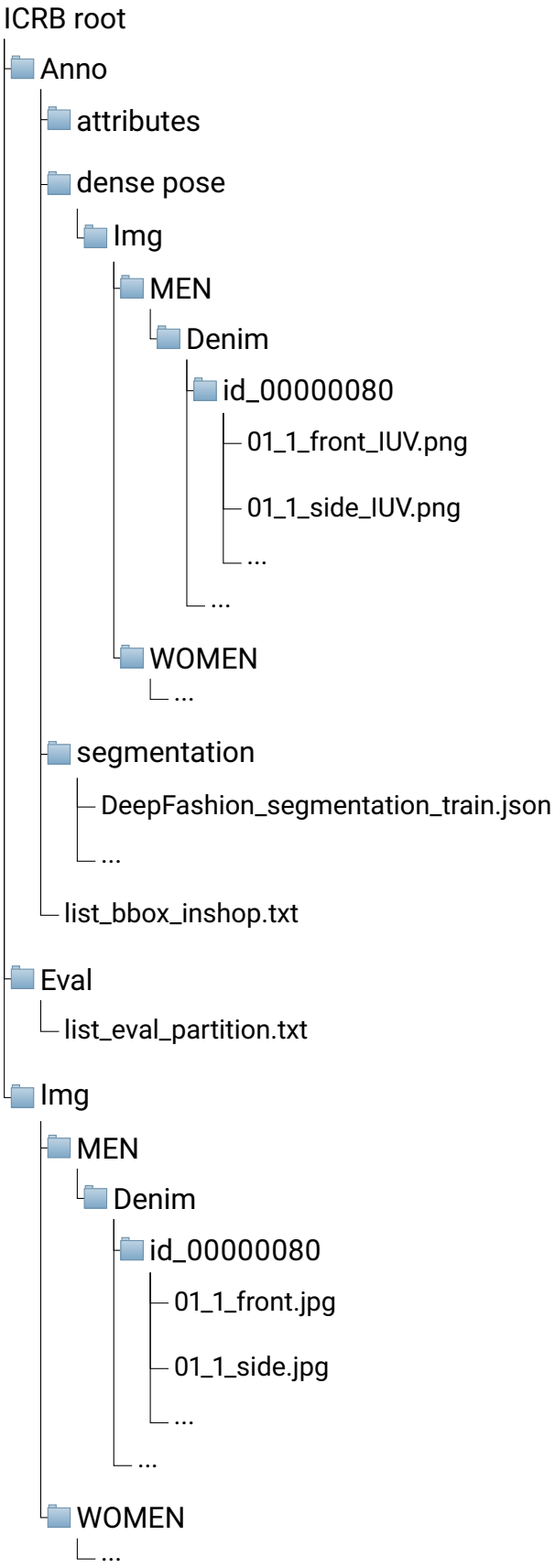
Πριν προχωρήσουμε, ωστόσο, στην παράθεση των βημάτων προ-επεξεργασίας του συνόλου δεδομένων, θεωρούμε σκόπιμο για λόγους καλύτερης κατανόησης να περιγράψουμε στο σημείο αυτό πως είναι δομημένο το σύνολο δεδομένων ICRB, ή τι θα

Σχήμα 66: Δείγματα εικόνων από ένα προϊόν του συνόλου δεδομένων In-shop Clothes Retrieval Benchmark του DeepFashion. Αριστερά φαίνεται το ρούχο φορεμένο σε έναν άνθρωπο-μοντέλο και δεξιά η συνυφασμένη εικόνα πόζας από το μοντέλο DensePose. Αμφότερες περιλαμβάνονται στο σύνολο δεδομένων εκπαίδευσης και είναι ανάλυσης 256×256.
Πηγή: «DeepFashion: Powering Robust Clothes Recognition and Retrieval With Rich Annotations», Liu et al., 2016 [55]

δει ο υποψήφιος χρήστης του όταν το κατεβάσει και το αποσυμπιέσει. Έτσι, αυτό λαμβάνει κανείς όταν αποσυμπιέσει τον αρχικό φάκελο του συνόλου δεδομένων είναι τρεις κεντρικοί υποφάκελοι: ο φάκελος «Anno» που περιέχει τις ετικέτες με κατηγορίες/χαρακτηριστικά/λεκτικές περιγραφές κλπ., ο φάκελος «Eval» που περιέχει ένα αρχείο διαχωρισμού του αρχικού συνόλου δεδομένων σε σύνολα εκπαίδευσης-δοκιμής και, τέλος, ο φάκελος «Img» που περιέχει τις εικόνες χωρισμένες σε φακέλους ανάλογα με την κατηγορία στην οποία ανήκουν και στον κωδικό του προϊόντος του καταστήματος, κάτι που δίνεται γραφικά στο σχήμα 67 παρακάτω.

Στα πλαίσια της παρούσας εργασίας και συγκεκριμένα της εκπαίδευσης του μοντέλου αλλαγής πόζας, χρειαζόμαστε μόνο τις εικόνες του συνόλου δεδομένων καθώς και τις εικόνες πόζας που υπάρχουν στον υποφάκελο *dense pose* του φακέλου *Anno* του συνόλου δεδομένων. Επίσης, πρέπει να δημιουργήσουμε ζεύγη μεταξύ των αντίστοιχων εικόνων, μία διαδικασία που αναλύεται στην παράγραφο προ-επεξεργασίας του συνόλου δεδομένων, ενώ ακολούθως παραθέτουμε ένα δείγμα εικόνων από το σύνολο δεδομένων εκπαίδευσης (δηλ. εικόνες που βρίσκονται σε υποφακέλους του φακέλου *Img*).

Πριν ολοκληρώσουμε την περιγραφή της δομής του συνόλου δεδομένων, θα αναφερθούμε συνοπτικά στον τρόπο εντοπισμού στοιχείων για μία εικόνα από το μονοπάτι αυτής από τη



Σχήμα 67: Δομή φακέλων και αρχείων του In-shop Clothes Retrieval Benchmark.

ρίζα (root) του συνόλου δεδομένων. Έστω λοιπόν η εικόνα στο ακόλουθο μονοπάτι:

`/Img/MEN/Denim/id_00005608/01_3_back.jpg`

τότε, «MEN/Denim» είναι η κατηγορία που ανήκει το προϊόν στον φάκελο του οποίου βρίσκεται η εικόνα, «id_00005608» είναι ο φάκελος του προϊόντος στο όνομα του οποίου δηλώνεται και το ID του προϊόντος στο κατάστημα και «01_3_back.jpg» είναι το όνομα το αρχείου της εικόνας. Στο όνομα αυτό δηλώνεται η ομάδα εντός του προϊόντος που ανήκει (π.χ. οι εικόνες ενός προϊόντος μπορεί να χωρίζονται με βάση το χρώμα ή το μέγεθος), εδώ «01», ακολουθούμενη από έναν αύξοντα αριθμό των στοιχείων της ομάδας, εδώ «3» και τέλος την πόζα στην οποία απεικονίζεται το μοντέλο (ένα από τα «front», «back», «side», «additional»), εδώ «back».

Σχήμα 68: Εικόνα στο μονοπάτι `/Img/MEN/Denim/id_00005608/01_3_back.jpg` του ICRB. Όπως φαίνεται, πρόκειται πράγματι για μία εικόνα που περιέχει ένα ανδρικό παντελόνι, φορεμένο σε μοντέλο με πόζα προς τα πίσω, στοιχεία που δηλώνονται στο μονοπάτι της εικόνας.

Προ-επεξεργασία του συνόλου δεδομένων ICRB: *ICRB Scraper*

Περνάμε ακολούθως στα βήματα προ-επεξεργασίας που χρειάστηκαν ώστε το σύνολο δεδομένων ICRB να μπορεί να χρησιμοποιηθεί για την εκπαίδευση του μοντέλου αλλαγής πόζας (αλλά μερικώς και σε αυτό εξαγωγής ρούχων από μοντέλα, όπως εξηγείται στην υποενότητα 5.1.2). Πριν προχωρήσουμε, θεωρούμε σκόπιμο να αναφερθεί πως η προ-επεξεργασία που κάνουμε σε επίπεδο συνόλου δεδομένων προσπαθούμε κατά το *δυνατό να μην επηρεάζει το αρχικό σύνολο δεδομένων*, μιας και που τα μεγέθη αυτών συνήθως είναι απαγορευτικά για αλλαγές, αλλά να έγκειται σε παραγωγή αρχείων που θα χρησιμοποιηθούν από τον φορτωτή είτε για φιλτράρισμα των εικόνων ή για εφαρμογή μετασχηματισμών αυτών. Για τον σκοπό αυτό δημιουργήθηκε ένας scraper ειδικού σκοπού με τη βασικές λειτουργίες να συνοψίζονται στα εξής βήματα:

1. ΄**Αντληση Εικόνων Πόζας**: επισκέπτεται κάθε τελικό φάκελο του συνόλου δεδομένων, δηλαδή φακέλους με όνομα `id_XXXXXXXX` (όπου `X` ψηφίο του 8-ψήφιου ID του προϊόντος) και για κάθε έναν ελέγχει εάν υπάρχει φάκελος με το ίδιο όνομα στο αντίστοιχο μονοπάτι με τις εικόνες από το DensePose, `/Anno/dense pose/Img/*/*/`. Εάν ναι, αντιγράφει όλες τις εικόνες πόζας και τις επικολλά στον αρχικό φάκελο.

2. **Εμπρόσθιο Πέρασμα**: Ακολούθως, ο scraper επισκέπτεται εκ νέου όλους τους τελικούς φακέλους με σκοπό τη δημιουργία δύο json αρχείων σε κάθε έναν από αυτούς: το `item_info.json` και `item_posable_info.json`. Στο πρώτο αρχειο, αποθηκεύει τα ονόματα και το πλήθος των εικόνων του εκάστοτε προϊόντος, καθώς και τις ομάδες στις οποίες αυτές χωρίζονται και το πλήθος των ομάδων αυτών. Η λογική εδώ είναι ότι για τη δημιουργία ζευγών εικόνων αλλαγής πόζας δεν αρκεί μόνο οι εικόνες να είναι του ίδιου προϊόντος, αλλά πρέπει να ανήκουν και στην ίδια ομάδα αυτού (δηλ. να απεικονίζεται το ίδιο χρώμα, μέγεθος κλπ. του προϊόντος). Στο δεύτερο αρχείο, το οποίο σχηματίζεται μετά το πρώτο, καταγράφονται, για κάθε ομάδα εικόνων, όλα τα ζεύγη φωτογραφιών για τα οποία υπάρχει τόσο η κανονική εικόνα όση και η εικόνα πόζας και για τις δύο, δημιουργώντας έτσι μία λίστα δυάδων (σε κάθε δυάδα υπάρχουν μόνο τα ονόματα αρχείων των κανονικών εικόνων). Η διαδικασία αυτή επαναλαμβάνεται και για τα 11.735 προϊόντα του ICRB.

3. **Οπίσθιο Πέρασμα**: Αφού ολοκληρωθεί το εμπρόσθιο πέρασμα του συνόλου δεδομένων από τον scraper, ξεκινάει το οπίσθιο πέρασμα, στο οποίο αναδρομικά ενώνονται τα αρχεία json του κάθε τελικού φακέλου σχηματίζοντας όλο και μεγαλύτερα τέτοια αρχεία, ίδιας όμως δομής. Συγκεκριμένα, η συνένωση δύο αρχείων json ακολουθεί την εξής λογική:

   3.1. ένωσε τις λίστες με τα ονόματα των αρχείων εικόνων, αφού πρώτα προσθέσεις στην αρχή του ονόματος του κάθε αρχείου το όνομα του φακέλου στον οποίο ανήκει

   3.2. πρόσθεσε τους αντίστοιχους μετρητές των δύο αρχείων

   3.3. ένωσε τις λίστες δυάδων των αρχείων `item_posable_info.json` κατά αντιστοιχία με το πρώτο βήμα

   3.4. αποθήκευσε το αποτέλεσμα σε ένα νέο αρχείο με όνομα `items_info.json` ή

`items_posable_info.json` ανάλογα με το ποια αρχεία συνενώθηκαν

Η αναδρομική εκτέλεση της παραπάνω διαδικασίας οδηγεί στη δημιουργία δύο αρχείων περιγραφής του συνόλου δεδομένων στη ρίζα ή αρχικό φάκελο αυτού: το `/Img/items_info.json` και το `/Img/items_posable_info.json`. Αυτά τα αρχεία περιέχουν αντίστοιχα όλες τις εικόνες και ομάδες εικόνων του συνόλου δεδομένων και όλα τα ζεύγη με εικόνων με διαφορετικές πόζες για τις οποίες υπάρχουν και οι αντίστοιχες εικόνες πόζας.

Θα θέλαμε να σημειώσουμε στο σημείο αυτό πως μερικές από τις εικόνες του συνόλου δεδομένων στο όνομα του αρχείου τους και συγκεκριμένα στο πεδίο που αναγράφεται η πόζα έχουν την τιμή «additional». Αυτές οι εικόνες δυστυχώς δεν παρουσιάζουν συνοχή και ενώ κάποιες είναι όντως μία πρόσθετη πόζα όπως και αυτές που το όνομά τους τελειώνει σε «front» ή «back», στην πλειονότητά τους είναι εικόνες που προβάλλουν διαφορετικά σημεία του σώματος ή σκέτο το ρούχο και γενικά δεν θα ήταν δόκιμο να χρησιμοποιηθούν στην εκπαίδευση του μοντέλου. Για τον λόγο αυτό, αν και οι εικόνες αυτές λαμβάνονται υπόψη από τον scraper, εν τέλει δεν χρησιμοποιούνται στην εκπαίδευση κάποιου μοντέλου. Επίσης, όπως φαίνεται, στον παρακάτω πίνακα δίνονται μερικά στοιχεία του συνόλου δεδομένων πριν και μετά την προ-επεξεργασία αυτού μέσω του ICRB Scraper:

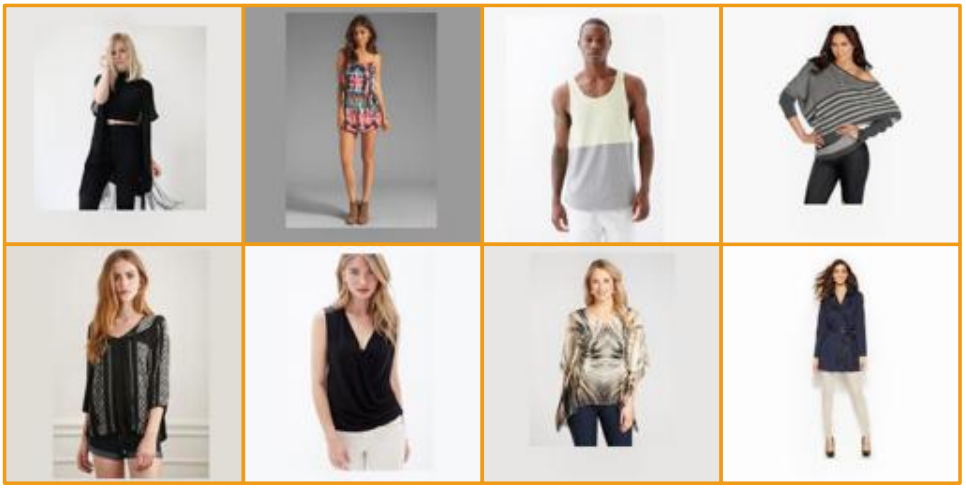
Πίνακας 3: Σύγκριση μεγέθους του συνόλου δεδομένων ICRB πριν και μετά την προ-επεξεργασία.

	# **Εικόνων**	# **Ομάδων**	# **Εικόνων με Πόζα**	# **Ομάδων με Εικόνες με Πόζα**	# **Ζευγών Εικόνων με Πόζα**
Πριν	54.642	-	-	-	-
Μετά	52.675	12.923	46.436	11.184	46.273 (×2)

Fashion Image Synthesis Benchmark (FISB)

Προχωράμε στη συνέχεια στην περιγραφή του δεύτερου υποσυνόλου δεδομένων του DeepFashion που χρησιμοποιήσαμε για την εκπαίδευση μοντέλων στην παρούσα εργασία, του Fashion Image Synthesis Benchmark (FISB). Αυτό το σύνολο δεδομένων που, όπως αναφέρθηκε, περιέχει 78.979 εικόνες ανάλυσης 128×128, δημιουργήθηκε με σκοπό την

εκπαίδευση μοντέλων για παραγωγή νέων εικόνων ρουχισμού. Μαζί με τις εικόνες, οι δημιουργοί δίνουν ένα σύνολο ετικετών, όπως χάρτες κατάτμησης ή λεζάντες, που όμως δεν μας χρειάστηκαν στην παρούσα εργασία. Παρακάτω δίνονται ορισμένες εικόνες από σύνολο δεδομένων FISB για καλύτερη κατανόηση του περιεχομένου του συνόλου δεδομένων.



Σχήμα 69: Τυχαία δείγματα από το (υπο)σύνολο δεδομένων FISB του DeepFashion. Οι εικόνες είναι ανάλυσης 128×128px.

Ως προς τη δομή του συνόλου δεδομένων FISB, αυτή σε αντίθεση με το ICRB είναι πρακτικά ανύπαρκτη. Οι δημιουργοί του απλώς δίνουν σε ένα τεράστιο αρχείο `Img.h5` (τύπου HDF5) μεγέθους 16.5GB όλες τις εικόνες του συνόλου δεδομένων τους αποθηκευμένες σειριακά. Ωστόσο, όπως φαίνεται και στο σχήμα 69 που προηγείται, οι εικόνες δεν είναι σωστά περικομμένες πριν από την αποθήκευσή τους στο αρχείο δεδομένων. Επομένως, κάποια προ-επεξεργασία σε επίπεδου συνόλου δεδομένων απαιτείται και στην περίπτωση του FISB, κάτι που αναλύεται ακολούθως.

Προ-επεξεργασία του συνόλου δεδομένων FISB: *FISB Scraper*

Πριν προχωρήσουμε στη μέθοδο προ-επεξεργασίας του συνόλου δεδομένων FISB που εφαρμόστηκε, θεωρούμε σκόπιμο να αναφερθούμε στον τρόπο χρήσης του στην παρούσα εργασία. Έτσι, όπως αναφέρθηκε και προηγούμενα, με αυτό το σύνολο δεδομένων θέλουμε να εκπαιδεύσουμε ένα μοντέλο τύπου GAN και μάλιστα μία παραλλαγή του StyleGAN (βλ. υποενότητα 4.1.3) για παραγωγή ρεαλιστικών εικόνων μόδας που να μοιάζουν (ιδανικά να είναι αδιάκριτες) από αυτές του σύνολου εκπαίδευσης. Σε μία αρκετά απλουστευμένη προσέγγιση, το μοντέλο καλείται να μάθει την κατανομή πιθανότητας του κάθε εικονοστοιχείου της εικόνας εξόδου ώστε όταν λάβει ένα δείγμα αυτής να παράξει

μία ρεαλιστική εικόνα. Όπως φαίνεται, όμως, στις εικόνες του σχήματος 69 παραπάνω, οι εικόνες, μη-όντας σωστά περικομμένες και μάλιστα χωρίς να υπάρχει μία διάσταση/θέση που να μπορούσαμε να τις περικόψουμε ώστε αυτέ να είναι σωστά κεντραρισμένες, δυσχεραίνουν αρκετά το έργο του μοντέλου και κάνουν επιτακτική την ανάγκη *εύρεσης σημείων περικοπής για κάθε εικόνα ξεχωριστά*.

Έτσι, το πρώτο βήμα προ-επεξεργασίας του σύνολου δεδομένων FISB ήταν η εύρεση των σημείων περικοπής των εικόνων που περιέχει. Για το σκοπό αυτό ακολουθήσαμε την εξής απλή διαδικασία:

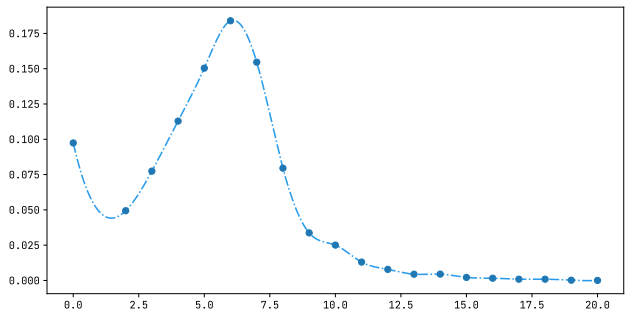
1. **Εύρεση Κάτω Ορίου**: ξεκινάμε από τα τελευταία εικονοστοιχεία της κάθε εικόνας και περικόπτουμε ένα μικρό μακρόστενο ορθογώνιο διαστάσεων 128×2px (πλ×υ) και ελέγχουμε εάν το ορθογώνιο αυτό περιέχει μόνο ουδέτερα χρώματα παρασκηνίου (ελέγχοντας τη διαφορά της ελάχιστης από τη μέγιστη τιμή σε κάθε κανάλι). Για όσο η συνθήκη είναι αληθής (δηλ. το ορθογώνιο ανήκει στο παρασκήνιο) μεγαλώνουμε κατά ένα εικονοστοιχείο και δοκιμάζουμε εκ νέου. Η βασική ιδέα εδώ είναι ότι όταν βρεθεί η πρώτη σειρά εικονοστοιχείων του απεικονιζόμενου αντικειμένου (π.χ. τα παπούτσια ή οι μηροί των ποδιών) σημαίνει πως βρήκαμε την τελευταία σειρά εικονοστοιχείων και άρα η επόμενή της είναι η πρώτη που περικοπεί.

2. **Εύρεση ΄Ανω Ορίου**: ακολούθως, έχοντας δηλαδή καταγράψει το κάτω όριο, επαναλαμβάνουμε μία αντίστοιχη διαδικασία για το επάνω. Παίρνουμε, δηλαδή, μία ζώνη εικονοστοιχείων 128×2px (πλ×υ) και μεγαλώνουμε το ύψος της ζώνης έως ότου βρούμε μη ουδέτερα χρώματα στο ορθογώνιο. Κατόπιν καταγράφουμε την τελευταία σειρά για την οποία δεν βρέθηκε αντικείμενο, ως το άνω όριο της εικόνας.

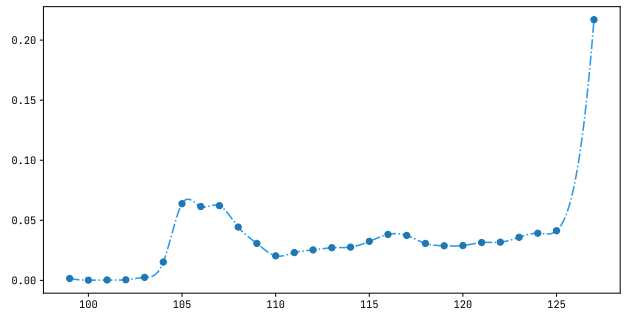
3. **Εύρεση Οριζόντιων Ορίων**: έχοντας το άνω και κάτω όριο της κάθε εικόνας και με δεδομένο ότι θέλουμε αυτές να είναι τετράγωνες, η εύρεση των οριζόντιων ορίων απλώς περικόπτουμε ισομερώς την εικόνα (μετά την άνω και κάτω περικοπή) και καταγράφουμε αυτά τα όρια. Πρέπει να σημειωθεί, ότι αυτό «δουλεύει» διότι οι δημιουργεί του συνόλου δεδομένων έβαλαν τις εικόνες με σωστό κεντράρισμα από αριστερά και δεξιά αλλά με λανθασμένο στο άνω και κάτω όριο.

Επαναλαμβάνοντας αυτή τη διαδικασία για κάθε εικόνα του FISB, δημιουργούμε ένα αρχείο - το ονομάζουμε `crops.json` - το οποίο αποθηκεύουμε στο φάκελο-ρίζα αυτού. Έχει ενδιαφέρον, να απεικονίσουμε γραφικά τη συχνότητα περικοπής ως προς μήκος και

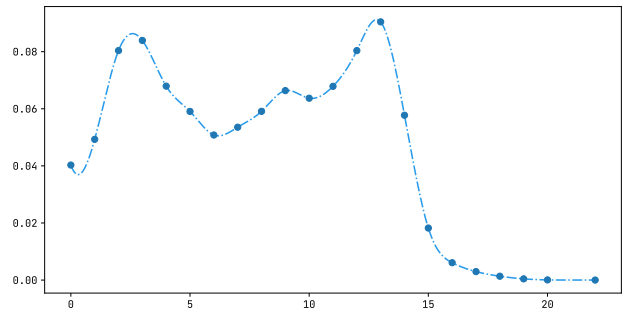
πλάτος της κάθε εικόνας, κάτι που γίνεται στα σχήματα που ακολουθούν όπου φαίνεται ότι ή μόνη λύση για να έχουμε σωστά περικομμένες εικόνες είναι να κάνουμε περικοπή σε κάθε εικόνα ξεχωριστά.



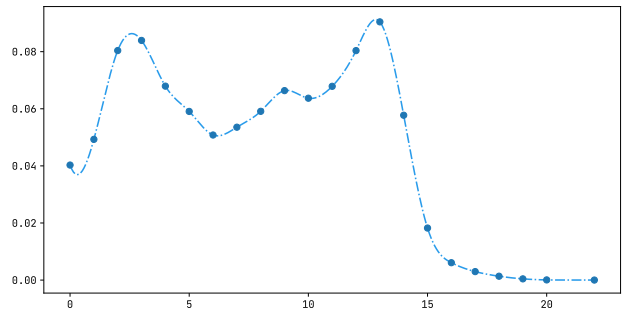
(α) άνω όριο



(β) κάτω όριο



(γ) αριστερό όριο

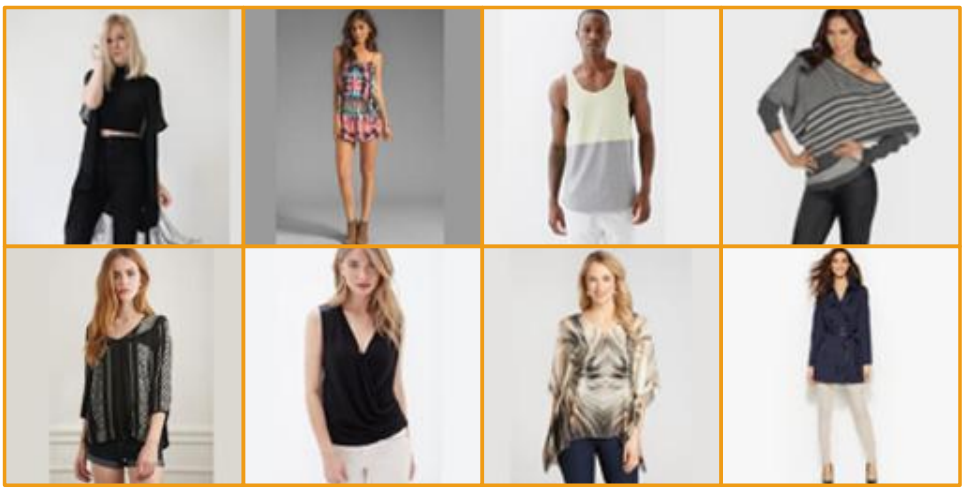


(δ) δεξιό όριο

Σχήμα 70: Συχνότητα (πιθανότητα) ορίων περικοπής σε κάθε μεριά των εικόνων του FISB

Επιπρόσθετα και όπως και κατά την προ-επεξεργασία του ICRB της προηγούμενης παραγράφου, επιλέξαμε να μην αλλάξουμε καθόλου το αρχείο δεδομένων απλώς να προσθέσουμε και εδώ αρχεία μετα-πληροφορίας ώστε να μπορεί να γίνει η προ-επεξεργασία της κάθε εικόνας τη στιγμή της φόρτωσής της από τον φορτωτή δεδομένων (dataloader). Στο σχήμα 71 παρακάτω παραθέτουμε τις ίδιες εικόνες με αυτές του σχήματος 69 οι οποίες έχουν περικοπεί σύμφωνα με τα όρια της εκάστοτε εικόνας από το αρχείο `crops.json`.

Μία ακόμη λειτουργία την οποία υλοποιήσαμε στον scraper του FISB είναι αυτή της ανίχνευσης του χρώματος παρασκηνίου της κάθε εικόνας. Η κεντρική ιδέα πίσω από αυτήν την κίνηση είναι η ίδια όπως και για τη περικοπή εικόνων: η εκπαίδευση ενός μοντέλου παραγωγής εικόνων είναι πιο αργή και σημαντικά πιο ασταθής όταν το σύνολο δεδομένων δεν περιέχει εικόνες με σαφή και συνεκτική δομή. Έτσι, όταν ζητάμε από το μοντέλο μας να παράγει εικόνες δίνοντάς του κάποιες εικόνες με λευκό φόντο, κάποιες με γκρι, κάποιες με καφέ και κάποιες με περισσότερα από ένα χρώματα, στην ουσία



Σχήμα 71: Τυχαία δείγματα από το (υπο)σύνολο δεδομένων FISB του DeepFashion, μετά την περικοπή. Οι εικόνες είναι οι αντίστοιχες του σχήματος 69, επίσης ανάλυσης 128×128px.

απλώς δυσκολεύουμε το έργο του. Για τον σκοπό αυτό, εξάγουμε το χρώμα παρασκηνίου ή σωστότερα την ομάδα χρώματος του παρασκηνίου της κάθε εικόνας του FISB. Η διαδικασία εξαγωγής του χρώματος παρασκηνίου, η οποία επίσης υλοποιείται από τον *FISB Scraper* έχει ως εξής:

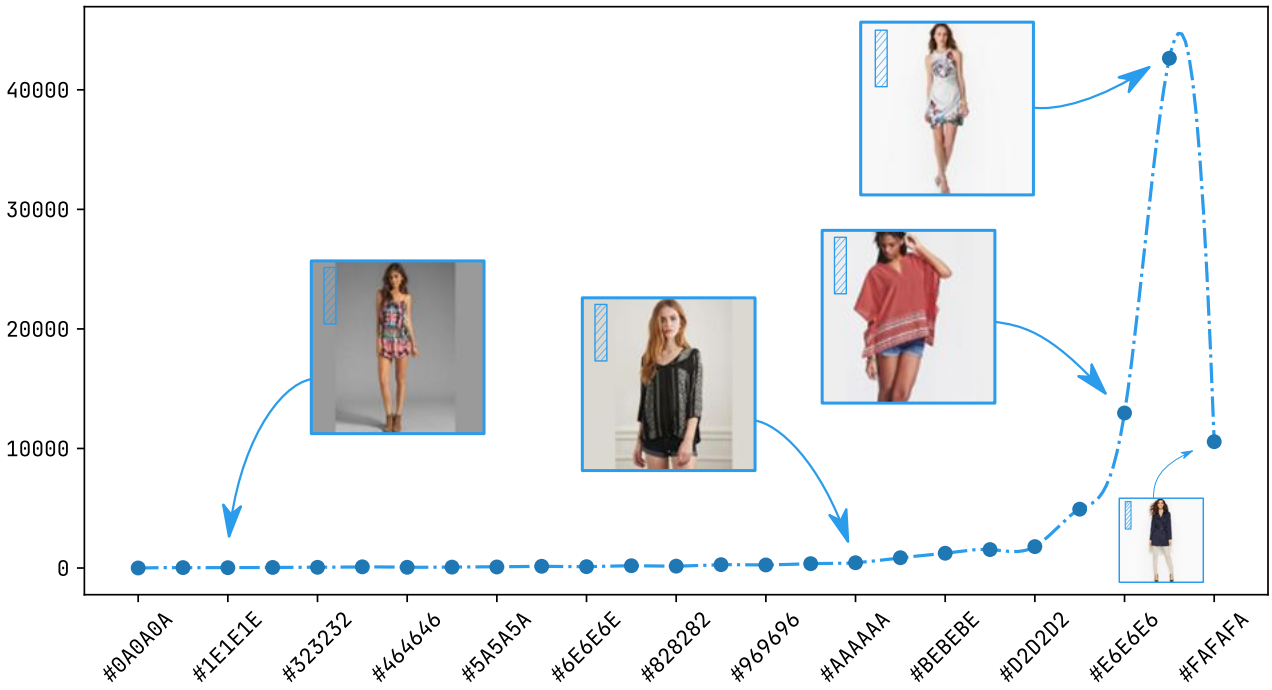
1. **Λωρίδες Δειγματοληψίας**: όπως και παραπάνω αποκόπτουμε ένα στενό ορθογώνιο παραλληλόγραμμο από την εκάστοτε εικόνα, μόνο που αυτή τη φορά η λωρίδα είναι ως προς το ύψος της εικόνας. Συγκεκριμένα, για κάθε εικόνα αποκόπτουμε μία λωρίδα διαστάσεων 15×100px (πλ×υ) από το αριστερό τμήμα (θέση αριστερής-επάνω γωνίας λωρίδας [10,5]px) και μία ίδιων διαστάσεων από το δεξίο τμήμα (θέση αριστερής-επάνω γωνίας λωρίδας [105,5]px).

2. **Δειγματοληψία Χρώματος**: κατόπιν, επιλέγουμε από ποια λωρίδα θα γίνει η δειγματοληψία του χρώματος παρασκηνίου βλέποντας ποια έχει περισσότερο ομοιόμορφα χρώματα. Πιο αναλυτικά, υπολογίζουμε τα διακριτά χρώματα και το πλήθος εικονοστοιχείων ανά χρώμα για κάθε λωρίδα και επιλέγουμε αυτή που έχει τα περισσότερα εικονοστοιχεία με το ίδιο χρώμα. Αυτό είναι και το χρώμα που θεωρούμε ότι κυριαρχεί στο παρασκήνιο της κάθε εικόνας.

3. **Εξαγωγή Ομάδας Χρώματος**: προκειμένου να μην έχουμε πολλά διακριτά χρώματα παρασκηνίου, μετά την εξαγωγή του χρώματος της κάθε ομάδας κβαντίζουμε τις τιμές των χρωμάτων, στρογγυλοποιώντας στην κοντινότερη πεντάδα του RGB χρώματος του κάθε καναλιού. Τέλος, μετατρέπουμε την κβαντισμένη τιμή

χρώματος RGB του παρασκηνίου της κάθε εικόνας σε δεκαεξαδική (hex) τιμή και προσθέτουμε τον δείκτη (index) της εικόνας στη λίστα με αυτά των εικόνων που έχουν τη συγκεκριμένη τιμή χρώματος παρασκηνίου. Ό λόγος που χρησιμοποιήσαμε hex τιμές χρωμάτων είναι διότι έτσι ο χρήστης μπορεί να ζητήσει εικόνες με χρώμα λευκότερο από μία τιμή και ο φορτωτής απλώς να συγκρίνει τις συμβολοσειρές. Αποθηκεύουμε το τελικό λεξικό χρωμάτων-λιστών δεικτών σε ένα αρχείο `backgrounds.json` το οποίο, όπως και προηγούμενα, αποθηκεύουμε στον φάκελο-ρίζα του συνόλου δεδομένων FISB.

Παρακάτω, και για λόγους πληρότητας, παραθέτουμε την κατανομή των χρωμάτων παρασκηνίου που εξήχθησαν με την παραπάνω διαδικασία, όπου ενδεικτικά δίνουμε και κάποιες εικόνες που έχουν το αντίστοιχο χρώμα παρασκηνίου. Όπως φαίνεται στο σχήμα αυτό, οι δημιουργοί του FISB δεν ήταν αρκετά επιμελείς ως προς τη συλλογή εικόνων, κάτι που όπως αποδείχτηκε δυσκόλεψε αρκετά την ευσταθή εκπαίδευση του μοντέλου μας που το χρησιμοποιεί.



Σχήμα 72: Συχνότητα (πιθανότητα) χρώματος παρασκηνίου του (υπο)συνόλου δεδομένων FISB του DeepFashion. Στο σχήμα έχουν προστεθεί και ενδεικτικές εικόνες με τα αντίστοιχα χρώματα παρασκηνίου. Η σκιαγραμμισμένη περιοχή στις εικόνες είναι η περιοχή εξαγωγής του χρώματος παρασκηνίου.

5.1.2 LookBook

Φεύγοντας από το DeepFashion, στην παρούσα παράγραφο θα περιγράψουμε ένα άλλο σύνολο δεδομένων το οποίο χρησιμοποιήθηκε για την εκπαίδευση ενός από τα μοντέλα μας, το LookBook. Το LookBook είναι ένα σύνολο δεδομένων που δημιουργήθηκε και παρουσιάστηκε από τον Yoo et al. στο άρθρο τους «*Pixel-Level Domain Transfer*» [66]. Κύριος σκοπός του άρθρου τους ήταν η παρουσίαση του μοντέλου PixelDTGAN, παραλλαγή του οποίου εκπαιδεύσαμε και στην παρούσα εργασία (βλ. παράγραφο 5.2.2), για συζευγμένη μετατροπή εικόνας ενός ανθρώπου που φοράει ένα ρούχο στην εικόνα του ρούχου σε ουδέτερο παρασκήνιο (δηλ. εξαγωγή ρούχου). Για την καλύτερη εκπαίδευση του μοντέλου τους, οι συγγραφείς του [66] συνέλεξαν εικόνες από πέντε (5) μεγάλα ηλεκτρονικά καταστήματα της Ασίας και συγκεκριμένα τα: *bongjashop.com*, *jogunshop.com*, *stylenanda.com*, *smallman.co.kr* και *wonderplace.co.kr*. Συνολικά, συλλέχθηκαν 84.748 εικόνες από 9.732 προϊόντα ρούχων άνω της μέσης (π.χ. μπλουζάκια, μπουφάν, φούτερ κλπ.). Έτσι, κάθε εικόνα ρούχου σε ουδέτερο παρασκήνιο αντιστοιχίζεται σε κατά μέσο όρο οκτώ (8) φωτογραφίες όπου το ρούχο έχει φορεθεί σε άνθρωπο-μοντέλο, οι οποίες όμως έχουν τραβηχτεί ως επί το πλείστο εκτός στούντιο (δηλ. το παρασκήνιο συνήθως είναι αρκετά θορυβώδες).

Σχήμα 73: Τυχαία δείγματα από το σύνολο δεδομένων LookBook. Δεξιά φαίνεται το προϊόν σε ουδέτερο παρασκήνιο, ενώ αριστερά φαίνονται τέσσερις (4) τυχαίες αντιστοιχίσεις αυτού. Οι εικόνες είναι ανάλυσης 256×256px.

Δομή του συνόλου δεδομένων LookBook

Πριν προχωρήσουμε, ωστόσο, στην παράθεση των βημάτων προ-επεξεργασίας του συνόλου δεδομένων, θεωρούμε σκόπιμο για λόγους καλύτερης κατανόησης να περιγράψου-

με και εδώ (όπως και στο ICRB) πως είναι δομημένο το σύνολο δεδομένων LookBook, ή τι θα δει ο υποψήφιος χρήστης του όταν το κατεβάσει και το αποσυμπιέσει. Έτσι, αυτό λαμβάνει κανείς όταν αποσυμπιέσει τον αρχικό φάκελο του συνόλου δεδομένων είναι ένας κεντρικός υποφάκελος: ο φάκελος «Img» που περιέχει τις εικόνες χωρισμένες σε φακέλους ανάλογα με τον αύξοντα αριθμό του προϊόντος στο οποίο ανήκουν, κάτι που δίνεται γραφικά στο σχήμα 74 παρακάτω. Όπως φαίνεται και εκεί, σε κάθε τελικό φάκελο (με όνομα φακέλου `id_XXXXXXXX` όπου `X` το ψηφίο του αύξοντα αριθμού) υπάρχει η εικόνα του ρούχου σκέτο (αρχείο `flat.jpg`) καθώς και κατά μέσο όρο οκτώ (8) εικόνες (με ονόματα αρχείων `{0-7}.jpg`) των ανθρώπων που φορούν το ρούχο αυτό.

Σχήμα 74: Δομή φακέλων και αρχείων του συνόλου δεδομένων LookBook.

Προ-επεξεργασία του συνόλου δεδομένων LookBook: *PixelDTScraper*

Η προ-επεξεργασία του συνόλου δεδομένων LookBook ήταν αρκετά πιο απλή από αυτήν των συνόλων δεδομένων που προηγήθηκαν. Δημιουργήθηκε και εδώ ένας scraper, ο *PixelDTScraper*, βασικές εργασίες του οποίου ήταν οι ακόλουθες:

1. **Έλεγχος των Προϊόντων**: επίσκεψη όλων των τελικών φακέλων και έλεγχος ότι υπάρχει το αρχείο με το ρούχο σκέτο καθώς και τουλάχιστον μία ακόμη εικόνα (ώστε να βγαίνει ένα τουλάχιστον ζεύγος για συζευγμένη μετατροπή). Πράγματι, μετά την

εκτέλεση αυτού του βήματος διαπιστώθηκε πως από τα 9.732 προϊόντα τα 8.726 πληρούσαν αυτήν τη συνθήκη, ενώ τα υπόλοιπα στα οποία έλειπε η εικόνα με το ρούχο ή αυτή ήταν η μοναδική εικόνα του προϊόντος, αφαιρέθηκαν.

2. **Έλεγχος των Εικόνων**: σε αρκετές περιπτώσεις οι εικόνες δεν ήταν τετράγωνες ή/και ήταν μεγαλύτερης ανάλυσης από 256×256, δείγμα ενός μη-επιμελώς συλλεγμένου συνόλου δεδομένων. Ακολούθως, ο *PixelDTScraper* επισκέπτεται κάθε τελικό φάκελο και για κάθε εικόνα μη-σωστών διαστάσεων κάνει τα εξής:

   2.1. σμίκρυνση της εικόνας (με τήρηση των αναλογιών) ώστε η μεγαλύτερη διάσταση (συνήθως το ύψος) να είναι 256px

   2.2. τοποθέτηση της εικόνας στο κέντρο ενός λευκού καμβά διαστάσεων 256×256px, το οποίο ουσιαστικά προσθέτει λευκές μπάρες ίδιου μεγέθους αριστερά και δεξιά της εικόνας (ή επάνω και κάτω εάν το πλάτος ήταν η μεγαλύτερη διάσταση)

   2.3. αντικατάσταση της αρχικής εικόνας από τη νέα (in-place editing)

3. **Εξαγωγή Πληροφοριών των Προϊόντων**: ως τελευταίο βήμα προ-επεξεργασίας σε επίπεδο συνόλου δεδομένων, ο *PixelDTScraper* εκτελεί μία διαδικασία παρόμοια με αυτήν του *ICRB Scraper* για εξαγωγή πληροφοριών σχετικά με τις εικόνες του συνόλου δεδομένων και αποθήκευσή της στον κεντρικό φάκελο, η οποία συνοψίζεται στα εξής δύο (2) βήματα:

   3.1. **Εμπρόσθιο Πέρασμα**: ο scraper επισκέπτεται εκ νέου όλους τους τελικούς φακέλους με σκοπό τη δημιουργία ενός json αρχείου σε κάθε έναν από αυτούς, το αρχείο `item_dt_info.json`. Στο αρχείο αυτό, αποθηκεύει τα ονόματα και το πλήθος των εικόνων του εκάστοτε προϊόντος. Επίσης, στο αρχείο αυτό αποθηκεύονται τα ζεύγη εικόνων άνθρωπος-προϊόν, δημιουργώντας έτσι μία λίστα δυάδων και το πλήθος αυτών. Η διαδικασία αυτή επαναλαμβάνεται και για τα 8.726 προϊόντα του LookBook.

   3.2. **Οπίσθιο Πέρασμα**: Αφού ολοκληρωθεί το εμπρόσθιο πέρασμα του συνόλου δεδομένων από τον scraper, ξεκινάει το οπίσθιο πέρασμα, στο οποίο αναδρομικά ενώνονται τα αρχεία json του κάθε τελικού φακέλου σχηματίζοντας όλο και μεγαλύτερα τέτοια αρχεία, ίδιας όμως δομής, με τη συνένωση των

αρχείων να ακολουθεί ίδια λογική με αυτή του ICRB Scraper. Η αναδρομική εκτέλεση της παραπάνω διαδικασίας οδηγεί στη δημιουργία ενός αρχείου περιγραφής του συνόλου δεδομένων στη ρίζα ή αρχικό φάκελο αυτού, το `/Img/items_dt_info.json`. Αυτό το αρχείο περιέχει όλες τις εικόνες και τα ζεύγη εικόνων του συνόλου δεδομένων LookBook, που στο σύνολό τους είναι 77.507 εικόνες (8.726 προϊόντων και 68.781 ανθρώπων) και 68.820 ζεύγη εικόνων αντίστοιχα (λόγος που τα ζεύγη είναι λίγο περισσότερα είναι διότι κάποια προϊόντα έχουν περισσότερες από μία εικόνες προϊόντων).

Ακολούθως, μετά τη διαπίστωση ότι και στο σύνολο δεδομένων ICRB του DeepFashion υπάρχουν ορισμένες εικόνες των ρούχων σκέτων και φωτογραφισμένων σε ουδέτερα παρασκήνια (όπως δηλαδή και στο LookBook) αποφασίσαμε να συμπεριλάβουμε και αυτά στο σύνολο δεδομένων εκπαίδευσης LookBook, διαδικασία που αναλύεται ακολούθως.

Επάυξηση Δεδομένων από το DeepFashion ICRB

Μετά από έλεγχο που έγινε στο σύνολο δεδομένων ICRB του DeepFashion διαπιστώθηκε ότι 683 προϊόντα αυτού περιέχουν σκέτα τα ρούχα ενώ παράλληλα έχουν και εικόνες ανθρώπων του φορούν. Αυτές είναι οι περιπτώσεις όπου μέσα στον αντίστοιχο φάκελο προϊόντος του ICRB υπάρχουν εικόνες με όνομα αρχείου `*_flat.jpg`. Τα προϊόντα αυτά, όπως είναι λογικό, ταιριάζουν σημαντικά με αυτά του LookBook και μάλιστα οι άνθρωποι είναι φωτογραφημένοι σε συνθήκες στούντιο, αρκετά καλύτερες δηλαδή από αυτές των εικόνων ανθρώπων του LookBook. Αυτό που έγινε επομένως είναι το εξής:

1. **Μεταφορά Προϊόντων από το ICRB**: τα 683 προϊόντα που εντοπίστηκαν μεταφέρθηκαν στον κεντρικό φάκελο του συνόλου δεδομένων LookBook

2. **Μετανομασία Προϊόντων από το ICRB**: στα προϊόντα δόθηκε ένα όνομα με βάση την τελευταία τιμή του αύξοντα αριθμού των δεικτών των προϊόντων του LookBook. Έτσι, τα νέα προϊόντα υπάρχουν στους φακέλου `id_00008726` έως και `id_00009408`.

3. **Μετονομασία Εικόνων και scraping**: ακολούθως οι εικόνες των ανθρώπων μετονομάστηκαν για να τηρούν την ονοματολογία του LookBook, δηλαδή έλαβαν ονόματα αρχειων `0.jpg`, `1.jpg` κλπ. Κατόπιν, εκτελέστηκε ένα εμπρόσθιο και ένα οπίσθιο πέρασμα του *PixelDTScraper* προκειμένου να συμπεριληφθούν και αυτοί οι νέοι φάκελοι.

Ακολούθως, παραθέτουμε ένα δείγμα από τα προϊόντα του συνόλου δεδομένων ICRB που προστέθηκαν στο LookBook, ενώ κατόπιν δίνουμε μία περίληψη του συνόλου δεδομένων LookBook πριν και μετά την προ-επεξεργασία αυτού.

Σχήμα 75: Τυχαίο δείγμα από τα προϊόντα του συνόλου δεδομένων ICRB του DeepFashion που προστέθηκαν στο LookBook. Δεξιά φαίνεται το προϊόν σε ουδέτερο παρασκήνιο, ενώ αριστερά φαίνονται τέσσερις (4) τυχαίες αντιστοιχίσεις αυτού. Οι εικόνες είναι και εδώ ανάλυσης 256×256px.

Πίνακας 4: Περίληψη του συνόλου δεδομένων LookBook πριν και μετά την προ-επεξεργασία και την προσθήκη των εικόνων από προϊόντα του ICRB του DeepFashion.

	# **Προϊόντων**	# **Εικόνων** **(ανθρώπων + ρούχων)**	# **Ζευγών Εικόνων**
LookBook (Πριν)	9.732	84.748 (75.016 + 9.732)	-
LookBook (Μετά)	8.726	77.507 (68.781 + 8.726)	68.820 (×2)
ICRB (Μετά)	683	3.644 (2.961 + 683)	2.922 (×2)
Συνολικά (Μετά)	9.409	81.151 (71.742 + 9.409)	71.742 (×2)

5.1.3 handbags2shoes

Το τέταρτο και τελευταίο σύνολο δεδομένων που χρησιμοποιήσαμε για την εκπαίδευση μοντέλων στην παρούσα εργασία είναι το σύνολο δεδομένων *handbags2shoes*. Πρόκειται για ένα υβριδικό σύνολο δεδομένων αποτελούμενο από δύο επιμέρους σύνολα και, όπως είναι λογικό, χρησιμοποιήθηκε για μη-συζευγμένη μετατροπή εικόνας-σε-εικόνα. Το σύνολο δεδομένων handbags2shoes έχει προκύψει από τα εξής σύνολα δεδομένων:

1. ***shoes_64.hdf5***: πρόκειται για το σύνολο δεδομένων που δημιουργήθηκε από τους Yu και Grauman για την εκπαίδευση μοντέλων τους στα [37] και [86]. Αποτελείται από 50.025 εικόνες παπουτσιών από τον κατάλογο του ηλεκτρονικού καταστήματος

Zappos.com. Όπως φαίνεται και στο σχήμα 76 που ακολουθεί, αυτό το σύνολο δεδομένων περιλαμβάνει εικόνες από τέσσερις βασικές κατηγορίες υποδημάτων: παπούτσια, σανδάλια, παντόφλες και μπότες. Όλες οι εικόνες του συνόλου δεδομένων shoes_64.hdf5 είναι κεντραρισμένες, σε λευκό φόντο, με τον ίδιο προσανατολισμό (νοτιοδυτικό) και σε ανάλυση 64×64px. Το συνολικό μέγεθος του αρχείου hdf5 που περιέχει τις εικόνες είναι 260MB.

2. ***handbags_64.hdf5***: πρόκειται για το σύνολο δεδόμενων που δημιουργήθηκε από τους Zhu et al. για τις ανάγκες του άρθρου τους «*Generative Visual Manipulation on the Natural Image Manifold*» [67]. Αποτελείται από 137.300 εικόνες τσαντών (κυρίως γυναικείων) από τη μηχανή αναζήτησης της Amazon. Όπως φαίνεται και στο σχήμα 77 παρακάτω, οι εικόνες που περιέχονται σε αυτό το σύνολο δεδομένων είναι κεντραρισμένες, σε λευκό φόντο και σε ανάλυση 64×64px. Το συνολικό μέγεθος του αρχείου hdf5 που περιέχει τις εικόνες είναι 774MB.

Για καλή μας τύχη τα δύο παραπάνω σύνολα δεδομένων, τα οποία από κοινού σχηματίζουν το σύνολο δεδομένων που ονομάζουμε *handbags2shoes*, περιέχουν εικόνες επιμελώς συλλεγμένες και επεξεργασμένες. Δεν απαιτείται επομένως κάποια περαιτέρω προ-επεξεργασία του συνόλου δεδομένων *handbags2shoes* το οποίο νοείται έτοιμο για χρήση από μοντέλα για μη-συζευγμένη μετατροπή εικόνας-σε-εικόνα.

(α) παπούτσια

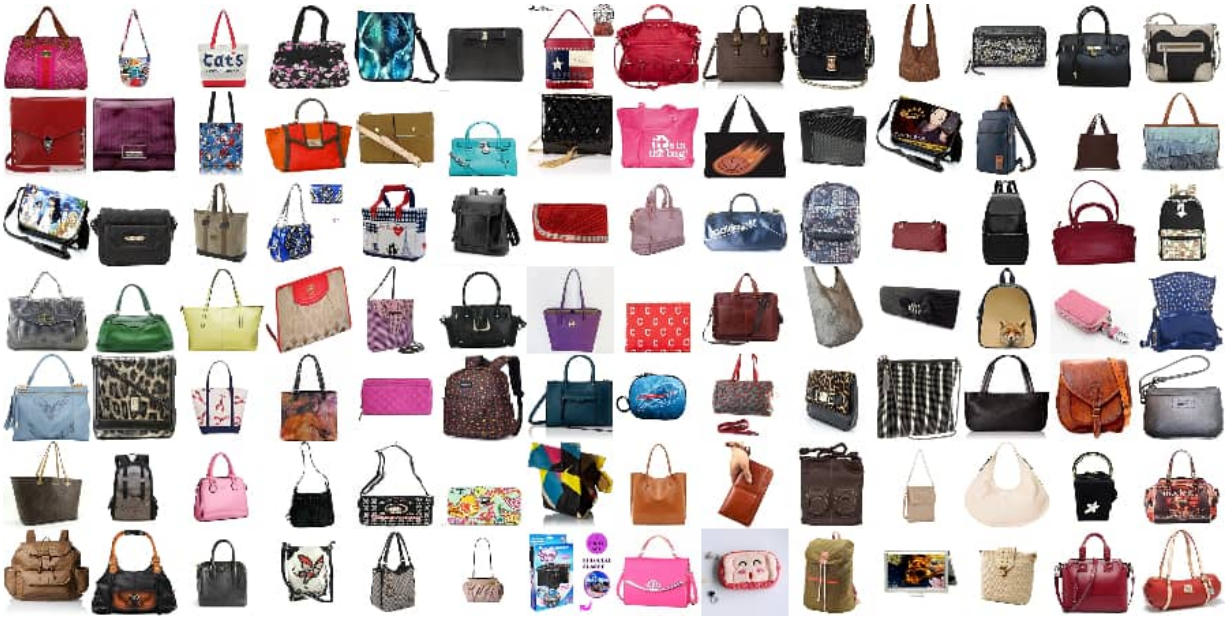
(β) σανδάλια

(γ) παντόφλες

(δ) μπότες

Σχήμα 76: Κατηγορίες υποδημάτων που περιέχονται στο σύνολο δεδομένων *shoes_64.hdf5*. Οι εικόνες είναι ανάλυσης 64×64px.

Πηγή: *http://vision.cs.utexas.edu/projects/finegrained/utzap50k*



Σχήμα 77: Τυχαία δείγματα από το σύνολο δεδομένων handbags_64.hdf5. Οι εικόνες είναι ανάλυσης 64×64px.

Πηγή: *http://efrosgans.eecs.berkeley.edu/iGAN/samples*

5.2 Μοντέλα που εκπαιδεύτηκαν

Ακολούθως, περνάμε στην ενότητα περιγραφής των μοντέλων που εκπαιδεύτηκαν στην παρούσα εργασία, χρησιμοποιώντας για την εκπαίδευσή τους τα σύνολα δεδομένων που περιγράφηκαν στην προηγούμενη ενότητα. Όλα τα μοντέλα αυτά είναι ΒΠΜ τύπου GAN, ενώ καλύπτουν το πλήρες φάσμα εφαρμογών των GANs στην Παραγωγική Μοντελοποίηση εικόνων. Συγκεκριμένα, τα μοντέλα που εκπαιδεύτηκαν καθώς και η κατηγορία εφαρμογής του καθενός από αυτά δίνονται στον πίνακα 5 που ακολουθεί.

Πίνακας 5: Σύνοψη των μοντέλων που εκπαιδεύτηκαν.

Όνομα Μοντέλου	Εφαρμογή	Κατηγορία Εφαρμογής	Σύνολο Δεδομένων	Μοντέλο Βάσης
PGPG (*PoseGAN*)	Αλλαγή Πόζας	Συζευγμένη μετατροπή	DeepFashion ICRB	-
PixelDTGAN	Εξαγωγή Ρούχου	Συζευγμένη μετατροπή	LookBook & DeepFashion ICRB	pix2pix
DiscoGAN	Μεταφορά Στιλ	Μη-Συζευγμένη μετατροπή	handbags2shoes	CycleGAN
StyleGAN	Παραγωγή εικόνων μόδας	Παραγωγή από θόρυβο	DeepFashion FISB	StyleGAN

Όπως φαίνεται και στον πίνακα, στόχος μας με την παρούσα εργασία ήταν *η δημιουργία ενός πολυ-εργαλείου αποτελούμενο από μοντέλα GAN το οποίο θα μπορεί να εκτελεί ρεαλιστικές παραγωγές και μετασχηματισμούς σε εικόνες μόδας και ρουχισμού*. Σε ότι ακολουθεί στο παρόν κεφάλαιο, θα περιγράψουμε τη δομή και τον τρόπο σχεδιασμού και εκπαίδευσης του κάθε μοντέλου ξεχωριστά, θα δώσουμε τις τελικές παραμέτρους καθώς και δείγματα εισόδου-εξόδου για το καθένα. Αφήνουμε την παρουσίαση των καμπύλων εκπαίδευσης, των μετρικών και πληθώρα παραγωγών των μοντέλων για το επόμενο κεφάλαιο, ώστε ο ενδιαφερόμενος αναγνώστης να μπορεί εύκολα και γρήγορα να αναζητήσει τα αποτελέσματα της παρούσας εργασίας. Η περιγραφή των μοντέλων ακολουθεί τη σειρά με την οποία εκπαιδεύτηκαν ή οποία ταυτίζεται με τη σειρά με την οποία τοποθετούνται στον παραπάνω πίνακα.

5.2.1 Αλλαγή Πόζας (PGPG - *PoseGAN*)

Το πρώτο μοντέλο GAN που εκπαιδεύτηκε στα πλαίσια της παρούσας εργασίας είναι μία ελαφρώς παραλλαγμένη εκδοχή του μοντέλου PGPG, που σχεδίασαν και υλοποίησαν ο Ma et al. στο άρθρο τους «*Pose Guided Person Image Generation*» [81]. Βασική ιδέα των δημιουργών του PGPG είναι η δημιουργία ενός μοντέλου GAN για αυτοματοποιημένη αλλαγή πόζας σε μία εικόνα εισόδου σύμφωνα με μία εικόνα πόζας που δίνεται ταυτόχρονα στην είσοδο του μοντέλου. Οι συγγραφείς εφάρμοσαν το μοντέλο τους σε σύνολα δεδομένων εικόνων μόδας με σκοπό την εφαρμογή του σε αντίστοιχα σενάρια του πραγματικού κόσμου (π.χ. αυτόματη αλλαγή πόζας στις εικόνες ενός ηλεκτρονικού καταστήματος για μείωση του κόστους φωτογράφισης).

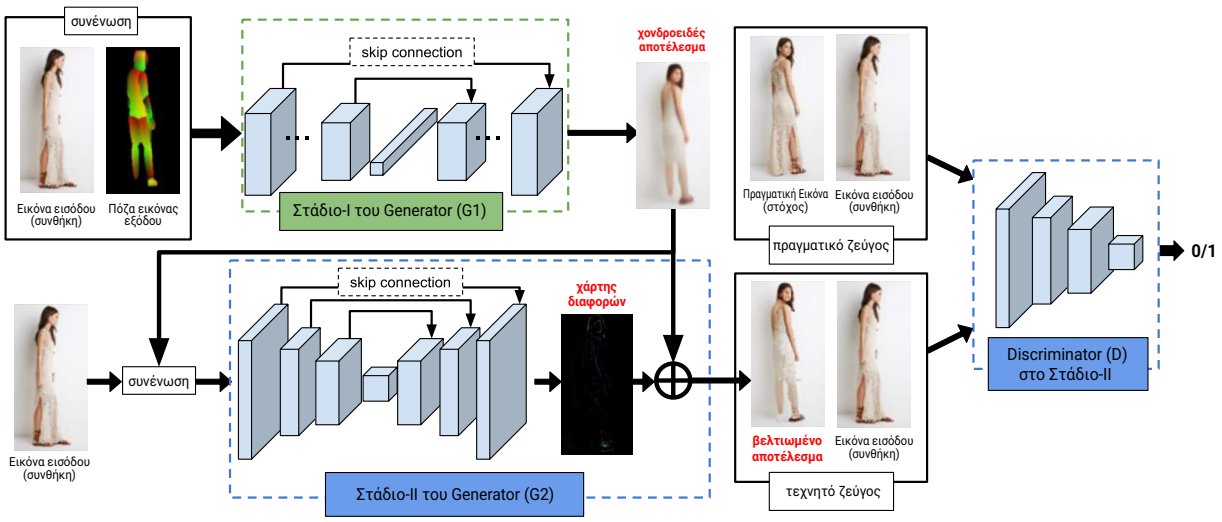
Όπως αναφέρθηκε στην προηγούμενη ενότητα, για την εκπαίδευση του μοντέλου θα χρησιμοποιηθεί το σύνολο δεδομένων ICRB του DeepFashion, αποτελούμενο από ζεύγη εικόνων εισόδου-εξόδου με τις αντίστοιχες εικόνες πόζας της κάθε μίας. Πριν προχωρήσουμε θέτουμε στο σημείο αυτό τη λογική εκπαίδευσης και δοκιμής το μοντέλου:

δοθείσης στην είσοδο μιας εικόνας εισόδου και της πόζας της εικόνας εξόδου, το μοντέλο καλείται να παράξει μία ρεαλιστική εικόνα κατά το δυνατόν πιο κοντά στην πραγματική εικόνα εξόδου, αλλάζοντας έτσι την πόζα του εικονιζόμενου ανθρώπου και ρούχου στην είσοδο

Πρόκειται, επομένως, για μοντέλο Συζευγμένης Μετατροπής εικόνας-σε-εικόνα, και μάλιστα συζευγμένης μετατροπής δυάδας εικόνων σε εικόνα (δηλ. με διπλή συνθήκη). Παρακάτω, αρχικά παραθέτουμε ορισμένα στοιχεία για το μοντέλο PGPG, όπως αυτό παρουσιάστηκε στο [81] και κατόπιν δίνουμε την αρχιτεκτονική του μοντέλου που υλοποιήσαμε στα πλαίσια της εργασίας καθώς και των παρμέτρων εκπαίδευσης αυτού.

Γενική Περιγραφή του Μοντέλου PGPG

Θα ξεκινήσουμε την περιγραφή του μοντέλου αλλαγής πόζας που υλοποιήσαμε από την παράθεση στοιχείων του αρχικού μοντέλου και άρθρου, του PGPG. Το μοντέλο PGPG εισήγαγε μία νέα και αρκετά καινοτόμα αρχιτεκτονική GAN, τουλάχιστον σε ότι αφορά το δίκτυο του Generator. Καθένα από τα δίκτυα του μοντέλου, δηλαδή ο Generator και ο Discriminator αναλύονται στις παραγράφους που ακολουθούν, ενώ η συνολική αρχιτεκτονική του μοντέλου δίνεται στο σχήμα 78 παρακάτω για αναφορά.



Σχήμα 78: Συνολική αρχιτεκτονική του μοντέλου PGPG. Φαίνονται τα δύο στάδια του Generator, ο Discriminator καθώς και οι είσοδοι/έξοδοι όλων των υποδικτύων του μοντέλου.
Πηγή: Ανακατασκευή από «Pose Guided Person Image Generation», Ma et al., 2017 [81]

Generator του PGPG

Η καινοτομία στη σχεδίαση της αρχιτεκτονικής του Generator του μοντέλου PGPG έγκειται στη χρήση δύο διακριτών υποδικτύων τα οποία τοποθετούνται σειριακά και τα οποία από κοινού εκπαιδεύονται για την παραγωγή ρεαλιστικών εικόνων αλλαγής πόζας στην έξοδο που λαμβάνεται από το δεύτερο υποδίκτυο. Έτσι, σε αντίθεση με τη λογική άλλων μοντέλων συζευγμένης μετατροπής, οι συγγραφείς του PGPG σκέφτηκαν ότι το μοντέλο τους θα εκπαιδεύονταν πιο σταθερά και με καλύτερα αποτελέσματα εάν διασπούσαν την εργασία του Generator σε δύο επιμέρους εργασίες που υλοποιούνται από δύο ξεχωριστά υποδίκτυα, ως εξής:

- **Στάδιο-I του Generator (G1)**: το πρώτο στάδιο ή υποδίκτυο του Generator λαμβάνει ως εισόδους την πραγματική εικόνα εισόδου και την πόζα της εικόνας εξόδου *συνενωμένες* (στα κανάλια). Σημειώνεται εδώ πως οι συγγραφείς χρησιμοποίησαν heatmap annotations[1] για τις πόζες, δηλαδή εικόνες με λευκό χρώμα στις αρθρώσεις και μαύρο οπουδήποτε αλλού. Σε αντίθεση, εμείς κάναμε χρήση των πολύ πιο πλούσιων DensePose annotations στα οποία υπάρχει και πληροφορία των επιφανειών του σώματος. Οι συγγραφείς ονομάζουν με «I» την εκάστοτε πραγματική εικόνα

[1]Τα heatmap annotations της πόζας ενός ανθρώπου είναι ένα σύνολο σημείων που βρίσκονται στις θέσεις των αρθρώσεων του σώματος. Για την κωδικοποίησή τους, δημιουργείται μία εικόνα 17 καναλιών, σε κάθε ένα από τα οποία υπάρχει μία λευκή περιοχή στη θέση της αντίστοιχης άρθρωσης και μαύρο αλλού.

και με «P» την εκάστοτε εικόνα πόζας. Έστω, επομένως, ότι έχουμε ένα ζεύγος εικόνων εισόδου-εξόδου, (I_A, I_B) συνοδευόμενες από τις αντίστοιχες εικόνες πόζας, (P_A, P_B). Στην είσοδο του πρώτου σταδίου του Generator δίνονται (συνενωμένες) το ζεύγος εικόνων (I_A, P_B) και το υποδίκτυο G1 καλείται να παράξει μία ρεαλιστική εκδοχή της I_B, έστω $\hat{I_{B_1}}$. Ωστόσο, επειδή αυτό αποτελεί γενικά δύσκολο στόχο, οι συγγραφείς πρότειναν να εκπαιδεύσουν το υποδίκτυο αυτό «πιέζοντάς» το η εικόνα $\hat{I_{B_1}}$ που επιστρέφει να αποτελεί μία χονδροειδή (π.χ. θολή) εκδοχή της πραγματικής I_B αιχμαλωτίζοντας ωστόσο όλη την πληροφορία της πόζας και τη βασική δομή της ζητούμενης εικόνας. Για να το πετύχουν αυτό, το δίκτυο G1 εκπαιδεύεται απλώς με κάποια συνάρτηση κόστους μέτρησης απόστασης στον χώρο των εικονοστοιχείων (π.χ. Manhattan), κάτι που αν και βοηθάει στην ευστάθεια της εκπαίδευσης, γενικά οδηγεί σε πιο θολά και «ασφαλή» αποτελέσματα.

- **Στάδιο-II του Generator (G2)**: στο δεύτερο στάδιο του Generator του PGPG, ή υποδίκτυο G2, οι συγγραφείς δοκίμασαν επίσης κάτι αρκετά εφευρετικό. Δοθείσης εκ νέου της αρχικής (πραγματικής) εικόνας εισόδου καθώς και της εικόνας που παρήγαγε το πρώτο στάδιο, G1, ο G2 δεν καλείται να παράξει μία ρεαλιστική εικόνα απευθείας, αλλά έναν χάρτη (ή εικόνα) διαφορών η οποία προστίθεται στη έξοδο του G1. Η προσθήκη της εξόδου του G2 στην έξοδο G1 πιέζει το υποδίκτυο G2 να μάθει να «αποθορυβοποιεί» τη χονδροειδή έξοδο του G1 και να την κάνει λιγότερο θολή (sharpen), ενώ ταυτόχρονα βοηθάει σημαντικά στη σταθεροποίηση της εκπαίδευσης ειδικά στα αρχικά στάδια αυτής [81]. Επανερχόμενοι στους συμβολισμούς που δόθηκαν στο άρθρο, για το εμπρόσθιο πέρασμα του Generator, θα ισχύει:

$$(I_A, P_B) \longrightarrow \textbf{G1} \longrightarrow \hat{I_{B_1}} \tag{5.1}$$

$$\left(I_A, \hat{I_{B_1}}\right) \longrightarrow \textbf{G2} \longrightarrow \left(+\hat{I_{B_1}}\right) \longrightarrow \hat{I_B} \tag{5.2}$$

όπου με $\hat{I_B}$ συμβολίζεται η τελική έξοδος του Generator.

Ο λόγος διαχωρισμού των δικτύων, επομένως, είναι διότι το δύσκολο έργο αιχμαλώτισης της νέας πόζας δίνεται σε ένα δίκτυο το οποίο εκπαιδεύεται με συνάρτηση κόστους ανακατασκευής (reconstruction loss), κάτι που γίνονταν και στους AK και κάτι γενικά πιο εύκολο και σταθερό, ενώ το έργο «προσθήκης ρεαλισμού» δίνεται σε ένα δίκτυο το οποίο πρέπει να παράγει ρεαλιστικούς χάρτες διαφοράς και όχι ρεαλιστικές εικόνες εξ΄ ολοκλήρου. Ακολούθως, παραθέτουμε τις συναρτήσεις κόστους με τις οποίες εκπαιδεύονται τα υποδίκτυα του Generator και τις οποίες καλούνται να ελαχιστοποιήσουν βελτιστοποιώντας τις παραμέτρους τους. Τονίζουμε στο σημείο αυτό πως τα δύο δίκτυα

εκπαιδεύονται και βελτιστοποιούνται *από κοινού*, από έναν βελτιστοποιητή. Οι συνάρτηση κόστους για το υποδίκτυο G1 του Generator έχουν ως εξής:

$$\mathcal{L}_{G1} = \|[G1\,(I_A, P_B) - I_B] \odot (1 + M_B)\|_1 \tag{5.3}$$

όπου η μάσκα της πόζας P_B, M_B, έχει ίδιες διαστάσεις με την πραγματική εικόνα εξόδου, I_B, αλλά λαμβάνει δυαδικές τιμές: 0 για τα εικονοστοιχεία εκτός του σώματος και 1 για τα υπόλοιπα (τα DensePose annotations έχουν ήδη 0 στο παρασκήνιο, οπότε η M_B υπολογίζεται άμεσα). Η χρήση μάσκας στον υπολογισμό του κόστους ενθαρρύνει το G1 να αγνοεί τις αλλαγές του παρασκηνίου που πιθανόν θα υπάρχουν στη νέα πόζα και να επικεντρώνεται στα σημεία εντός του σώματος. Αυτό, αν και κάνει ακόμα πιο θολές τις παραγόμενες εικόνες από το G1 (ειδικά στο παρασκήνιο), αυξάνει την ευστάθεια του δικτύου και το βοηθάει να μάθει πιο γρήγορα. Η παραπάνω συνάρτηση κόστους του G1, επομένως, μετράει την απόσταση των εικονοστοιχείων της εικόνας στην έξοδό του από την πραγματική χρησιμοποιώντας την απόσταση L_1 ή Manhattan. Πριν αναφέρουμε τη συνάρτηση κόστους του G2 υποδικτύου και επειδή αυτή χρησιμοποιεί και το αντιπαραθετικό (adversarial) κόστος εκπαίδευσης των GANs, θα παραθέσουμε σύντομα τα βασικά στοιχεία του Discriminator.

Discriminator του PGPG

Για το δίκτυο του Discriminator οι συγγραφείς του PGPG χρησιμοποίησαν μία παραλλαγή του Discriminator του DCGAN (βλ. σχήμα 42), που όμως επειδή πρόκειται για υπο-συνθήκη παραγωγή, στην είσοδο του Discriminator εκτός από την πραγματική/τεχνητή εικόνα εισάγεται και η εικόνα-συνθήκη. Η τελευταία, στο PGPG, είναι η αρχική εικόνα εισόδου του Generator, I_A. Έτσι, κάθε φορά ο Discriminator στην είσοδό του θα έχει έξι (6) αντί για τρία (3) κανάλια (οι δύο εικόνες συνενωμένες - βλ. Discriminator του pix2pix, παράγραφος 4.2.1), δηλαδή είτε το ζεύγος εικόνων (I_A, I_B) ή το $\left(I_A, \hat{I_B}\right)$. Ως συνάρτηση κόστους για την εκπαίδευση του Discriminator, οι συγγραφείς του [81], χρησιμοποιήσαν την Binary Cross-Entropy (σχέση 3.5), την οποία συμβολίζουν ως $\mathcal{L}_{bce}\,(\text{inputs}, \text{label})$ (όπου label είναι 0 για τα τεχνητά και 1 για τα πραγματικά δείγματα στον Discriminator).

Έτσι, η συνάρτηση κόστους που προσπαθεί να ελαχιστοποιήσει ο Discriminator του PGPG, θα είναι:

$$\mathcal{L}^{\mathrm{D}}_{adv} = \mathcal{L}_{bce}\left(\mathrm{D}\left(I_A, I_B\right), 1\right) + \mathcal{L}_{bce}\left(\mathrm{D}\left(I_A, G2\left(I_A, \hat{I}_{B1}\right)\right), 0\right), \tag{5.4}$$

Επανερχόμενοι τώρα στη συνάρτηση κόστους που καλείται να ελαχιστοποιήσει το δεύτερο στάδιο του Generator του PGPG, δηλαδή το υποδίκτυο G2, αυτή σύμφωνα με τους

συγγραφείς θα πρέπει να είναι το άθροισμα του αντιπαραθετικού κόστους (που προέρχεται από τον Discriminator) και ενός κόστους ανακατασκευής παρόμοιο με αυτό του G1. Η βασική ιδέα πίσω από τη χρήση του όρου ανακατασκευής στον χώρο των εικονοστοιχείων (L1/Manhattan) είναι ότι η αυτός επιταχύνει την εκπαίδευση του δεύτερου σταδίου (κατ΄ αντιστοιχία με το πρώτο) και ταυτόχρονα αυτή η τάση θόλωσης των εικόνων εξόδου «βοηθάει» με τις μετρικές αξιολόγησης των παραγόμενων εικόνων. Συμπερασματικά, η συνάρτηση κόστους του G2 δίνεται ακολούθως:

$$\mathcal{L}_{adv}^{\mathsf{G}} = \mathcal{L}_{bce}\left(\mathsf{D}\left(I_A, G2\left(I_A, \hat{I}_{B1}\right)\right), 1\right) \tag{5.5}$$

$$\mathcal{L}_{G2} = \mathcal{L}_{adv}^{\mathsf{G}} + \lambda_{recon} * \left\|\left(G2\left(I_A, \hat{I}_{B1}\right) - I_B\right) \odot (1 + M_B)\right\|_1, \tag{5.6}$$

όπου και εδώ εφαρμόζεται η μάσκα της πόζας εξόδου για καλύτερη αξιολόγηση της απόδοσης του μοντέλου συγκεκριμένα κατά την αλλαγή της πόζας, ενώ λ_{recon} είναι το βάρος του κόστους ανακατασκευής, συνήθως (όπως και στο PGPG) αρκετά μεγαλύτερο της μονάδας.

Για λόγους σαφήνειας, τονίζουμε στο σημείο αυτό πως κατά το εμπρόσθιο πέρασμα στον Generator, το G1 υπολογίζει την έξοδό του και το κόστος ανακατασκευής συγκρίνοντάς τη με την αρχική εικόνα και μετά παγώνει. Δηλαδή, το *δεν υπολογίζονται οι παράγωγοι των παραμέτρων του G1 ως προς το κόστος του G2, παρόλο που ο ίδιος βελτιστοποιητής μεταβάλλει από κοινού τις παραμέτρους των δύο υποδικτύων*. Ακολούθως και πριν περάσουμε στις λεπτομέρειες της υλοποίησής μας, παραθέτουμε στο σχήμα 79, παρακάτω, ενδεικτικά αποτελέσματα των συγγραφέων του PGPG μετά την εκπαίδευση του μοντέλου τους στο σύνολο δεδομένων DeepFashion ICRB (στην υψηλής ανάλυσης εκδοχή του, δηλαδή σε εικόνες 256×256px).

Υλοποίηση του Μοντέλου Αλλαγής Πόζας: *PoseGAN*

Στην παράγραφο αυτή θα δώσουμε τη δική μας υλοποίηση για τα δίκτυα του PGPG, ένα μοντέλο που ονομάσαμε *PoseGAN* (credits to prof. P. Mitkas). Δυστυχώς είχαμε πολύ περιορισμένες πληροφορίες για την αρχιτεκτονική και τις παραμέτρους εκπαίδευσης του PGPG από το [81] γι΄ αυτό και το *PoseGAN* παρόλο που εμπνέεται σημαντικά και ακολουθεί τη γενική οργάνωση του PGPG δεν μπορεί να θεωρηθεί ως απλή υλοποίηση του άρθρου. Με αφετηρία, επομένως, τη συνολική ιδέα και δομή του PGPG καθώς και τις ελάχιστες παραμέτρους των συνελικτικών στρώσεων των δικτύων τους, στην παράγραφο αυτή θα ξεκινήσουμε με την περιγραφή του Generator (αρχικά του G1 και κατόπιν του G2) και θα ολοκληρώσουμε με την περιγραφή του δικτύου του Discriminator.

Σχήμα 79: Ενδεικτικές παραγωγές του μοντέλου PGPG μετά από εκπαίδευσή του στο σύνολο δεδομένων ICRB του DeepFashion.

Πηγή: Ανακατασκευή από «Pose Guided Person Image Generation», Ma et al., 2017 [81]

Στάδιο-I του Generator (G1)

Όπως αναφέρθηκε, ο σκοπός του πρώτου σταδίου του Generator, G1, είναι η ενσωμάτωση της πόζας της εικόνας εξόδου με την εικόνα εισόδου (ή εικόνα-συνθήκη). Για την επίτευξη του στόχου αυτού, οι συγγραφείς προτείνουν τη χρήση ενός δικτύου παρόμοιο με το U-Net που χρησιμοποιείται ως Generator στο μοντέλο pix2pix (βλ. υπενότητα 4.2.1). Κάτι που επισημαίνουν είναι ότι επειδή κατά την αλλαγή της πόζας μπορεί να χρειαστεί να μεταφερθεί πληροφορίας από τη μεριά της εικόνας στην άλλη, θα ήταν χρήσιμο στη σημείο στένωσης του δίκτυο σαν αυτά των AK που θα χρησιμοποιηθεί ως G1 να χρησιμοποιηθεί μία πλήρως συνδεδεμένη στρώση.

Η υλοποίηση του υποδικτύου G1 του Generator στην οποία καταλήξαμε για το *PoseGAN* έχει τα ακόλουθα χαρακτηριστικά:

- ✓ βασίζεται στο δίκτυο U-Net, έχοντας οκτώ (8) συνελικτικές στρώσεις με βήμα (stride) 1 στον encoder (όλες με φίλτρα 3×3) και ισόποσες ανάστροφες συνελικτικές στρώσεις στον decoder με skip connections από τις αντίστοιχες στρώσεις του encoder, (κάποιες με φίλτρα 2×2 ενώ άλλες 3×3 όπως εξηγείται παρακάτω) και

ακολουθούμενες από στρώσεις Κανονικοποίησης Ομάδας (βλ. παράγραφο 4.1.1)

- ✓ στο σημείο στένωσης (bottleneck) εφαρμόζει σειριακά: συνελικτική στρώση μείωσης των καναλιών, στρώση flatten, πλήρως συνδεδεμένη στρώση $2560 \longrightarrow 2560$ νευρώνων, στρώση unflatten και συνελικτική στρώση αύξησης των καναλιών[2], ώστε να συμπεριληφθεί πλήρως-συνδεδεμένη
- ✓ εκπαιδεύεται με συνάρτηση κόστους ανακατασκευής στον χώρο των εικονοστοιχείων, L1 ή Manhattan, ωστόσο οι παράμετροί του βελτιστοποιούνται από κοινού με αυτές του υποδικτύου G2

Η αρχιτεκτονική του υποδικτύου G1 μπορεί να συνοψιστεί ως εξής:

***PoseGAN* › G1 › Encoder**

$UF_{6 \longrightarrow 16} \longrightarrow [$

$CONV3_{C_{in} \rightarrow 2*C_{in}} \longrightarrow BN \longrightarrow LReLU_{0.2} \longrightarrow CONV3_{2*C_{in} \rightarrow 2*C_{in}} \longrightarrow BN \longrightarrow LReLU_{0.2} \longrightarrow POOL2$

$] \times 4$

όπου οι τετράγωνες αγκύλες εσωκλείουν τις ομάδες στρώσεων συστολής (contracting blocks) όπως ονομάζονται στο U-Net (τέσσερις (4) χρησιμοποιήθηκαν συνολικά στον encoder του G1 - γενικά είναι μία από τις υπερ-παραμέτρους του μοντέλου). Με UF συμβολίζουμε τη συνελικτική στρώση αύξησης των καναλιών (μοναδιαίο βήμα, φίλτρα 1×1), ενώ $CONVX_{C_{in} \rightarrow C_{out}}$ συμβολίζουμε μία συνελικτική στρώση με φίλτρα X×X, μοναδιαίο βήμα, C_{in} κανάλια εισόδου και C_{out} εξόδου. Με BN συμβολίζουμε μία στρώση κανονικοποίησης ομάδας. Με POOL2 συμβλίζουμε μία στρώση pooling μεγίστου με βήμα 2.

***PoseGAN* › G1 › Bottleneck**

$PJ_{256 \rightarrow 10} \rightarrow FLATTEN_{10 \times 8 \times 8 \rightarrow 2560} \rightarrow FC_{2560 \rightarrow 2560} \rightarrow LReLU_{0.2} \rightarrow$

$UNFLATTEN_{2560 \rightarrow 10 \times 8 \times 8} \rightarrow PJ_{10 \rightarrow 256}$

όπου με $PJ_{C_{in} \rightarrow C_{out}}$ συμβολίζουμε συνελικτική στρώση προβολής των καναλιών (μοναδιαίο βήμα, φίλτρα 1×1) από C_{in} σε C_{out} και με FC συμβολίζουμε την πλήρως-συνδεδεμένη στρώση.

[2] Οι στρώσεις προβολής (αύξησης ή μείωσης) των καναλιών είναι συνελικτικές στρώσεις με μοναδιαίο βήμα και φίλτρο διάστασης 1×1. Αυτές οι στρώσεις επενεργούν σε κάθε θέση του πλέγματος μειώνοντας μονάχα τον αριθμό των χαρτών ενεργοποίησης ή καναλιών της εισόδου.

***PoseGAN* › G1 › Decoder**

$$
\begin{aligned}
&[\\
&\quad UP_{2\times} \longrightarrow CONV2_{C_{in}\to C_{in}/2} \longrightarrow CONCAT\_SKIP\_CON \longrightarrow CONV3_{C_{in}\to C_{in}/2} \longrightarrow \\
&\quad BN \longrightarrow LReLU_{0.2} \longrightarrow CONV2_{C_{in}/2\to C_{in}/2} \longrightarrow BN \longrightarrow LReLU_{0.2}\\
&] \times 4 \longrightarrow PJ_{16\to 3} \longrightarrow TANH
\end{aligned}
$$

όπου οι τετράγωνες αγκύλες εσωκλείουν τις ομάδες στρώσεων επέκτασης (expanding blocks) όπως ονομάζονται στο U-Net (τέσσερις (4) χρησιμοποιήθηκαν συνολικά στον decoder του G1 - γενικά είναι μία από τις υπερ-παραμέτρους του μοντέλου). Επίσης, με $UP_{2\times}$ συμβολίζουμε τη στρώση διγραμμικού (bilinear) upsampling που τη χρησιμοποιούμε σύμφωνα με τα ευρήματα των συγγραφέων των PGGAN, StyleGAN (του προηγούμενου κεφαλαίου) αντί για ανάστροφη συνελικτική, CONCAT_SKIP_CON είναι η skip connection που μεταφέρει $C_{in}/2$ στο πλήθος χάρτες ενεργοποίησης από την έξοδο του αντίστοιχου contracting block του encoder (ίδιου πλάτους και ύψους), ενώ οι τελευταίες δύο στρώσεις χρησιμοποιούνται για την έξοδο εικόνας τριών καναλιών (RGB) και κάθε εικονοστοιχείο στο εύρος [-1,1]. Τέλος, με TANH συμβολίζουμε τη συνάρτηση εξόδου υπερβολικής εφαπτομένης, η οποία έχει σιγμοειδή μορφή και πεδίο τιμών το $[-1.0, 1.0]$.

Στάδιο-II του Generator (G2

Θα αναφέρουμε εκ νέου και συνοπτικά, ότι σκοπός του δεύτερου σταδίου του Generator του PGPG ή, στη δική μας υλοποίηση, του *PoseGAN*, είναι δοθέντων μιας αρχικής εικόνας και μιας θολής εικόνας της εικόνας εξόδου (ή τουλάχιστον μίας που έχει αιχμαλωτίσει την πόζα της εικόνας εξόδου), να παράξει και να επιστρέψει μία ρεαλιστική εικόνα που να μοιάζει στην πραγματική εικόνα εξόδου ώστε να ξεγελάσει τον Discriminator. Για τη διευκόλυνση του έργου του προστίθεται στην έξοδό του η θολή εικόνα εξόδου του G1, κάτι που αναγκάζει το υποδίκτυο G2 να παράγει χάρτες διαφορών για αποθορυβοποίηση και sharpening. Για την υλοποίηση του δεύτερου σταδίου οι συγγραφείς του PGPG χρησιμοποιούν επίσης μία παραλλαγή του U-Net χωρίς όμως πλήρως-συνδεδεμένη στρώση αυτή τη φορά καθώς η πληροφορία της πόζας έχει θεωρητικά αιχμαλωτιστεί.

Στη δική μας υλοποίηση, το G2 του Generator του PoseGAN είναι σχεδόν πανομοιότυπο με τον Generator του pix2pix, δηλαδή πρόκειται για ένα δίκτυο U-Net, με τη σημαντική προσθήκη DropOut [35] στο πρώτο μισό του encoder του G2. Με τη στρώση DropOut νευρώνες απενεργοποιούνται σε κάθε βήμα με τυχαία σειρά τόσο κατά το εμπρόσθιο όσο και κατά το οπίσθιο πέρασμα. Αυτό αναγκάζει το μοντέλο να μάθει πιο εύρωστα χαρακτηριστικά

και γενικά έχει βρεθεί πως βοηθάει με τη γενίκευση (generalization) αυτού. Σημειώνουμε στο σημείο αυτό, πως αν και εν γένει στις εφαμοργές συζευγμένης μετατροπής εικόνας-σε-εικόνα, επειδή υπάρχει ξεκάθαρος στόχος εξόδου, δεν χρεισημοποιείται καμία πηγή θορύβου ή στοχαστικότητας (γι΄ αυτό και δεν εισάγεται τυχαίος θόρυβος στην είσοδο του Generator). Ωστόσο, στα πλαίσια του PoseGAN κρίναμε πως η εισαγωγή αυτής της πηγής στοχαστικότητας θα βοηθούσε το G2 να παράγει πιο ποικιλόμορφες εικόνες ειδικά στα σενάρια όπου η αλλαγή της πόζας οδηγούσε στην εμφάνιση νέων σημείων του σώματος και, κυριότερα, νέων ρούχων.

Υπενθυμίζουμε πως το υποδίκτυο G2 εκπαιδεύεται τόσο με κόστος ανακατασκευής (Manhattan) όσο και με το αντιπαραθετικό κόστος από τις προβλέψεις του Discriminator για τα δείγματά που παράγει. Παρακάτω και αντίστοιχα με το G1, δίνουμε παραστατικά την αρχιτεκτονική του δικτύου του G2.

PoseGAN › G2 › Encoder

$$
\begin{aligned}
&UF_{6\longrightarrow 32} \longrightarrow [\\
&\quad CONV3_{C_{in}\rightarrow 2*C_{in}} \longrightarrow BN \longrightarrow DO \longrightarrow LReLU_{0.2} \longrightarrow \\
&\quad CONV3_{2*C_{in}\rightarrow 2*C_{in}} \longrightarrow BN \longrightarrow DO \longrightarrow LReLU_{0.2} \longrightarrow POOL2 \\
&] \times 6
\end{aligned}
$$

όπου με DO συμβολίζουμε τη στρώση DropOut, ενώ όλα τα μεγέθη είναι ίδια όπως και στον encoder του G1. Επίσης, φαίνεται από την αρχική στρώση αύξησης των καναλιών, $UF_{6\longrightarrow 32}$, ότι στον encoder του G2 χρησιμοποιούμε περισσότερα κανάλια στις συνελικτικές στρώσεις, αλλά κυριότερα χρησιμοποιούμε και δύο ακόμα ομάδες στρώσεων συστολής, για συνολικά έξι (6). Σημειώνεται, ότι η στρώση UF (αύξησης των καναλιών) αποτελείται από φίλτρα διάστασης 7×7.

PoseGAN › G2 › Bottleneck

Δεν χρησιμοποιήθηκε καμία στρώση στο σημείο στένωσης του G2.

Οι 1024 χάρτες ενεργοποίησης διαστάσεων 4×4 περνούν στην επόμενη στρώση.

***PoseGAN* › G2 › Decoder**

$$
\begin{aligned}
&[\\
&\quad \mathrm{UP}_{2\times} \longrightarrow \mathrm{CONV2}_{C_{in}\to C_{in}/2} \longrightarrow \mathrm{CONCAT\_SKIP\_CON} \longrightarrow \mathrm{CONV3}_{C_{in}\to C_{in}/2} \longrightarrow\\
&\quad \mathrm{BN} \longrightarrow \mathrm{LReLU}_{0.2} \longrightarrow \mathrm{CONV2}_{C_{in}/2\to C_{in}/2} \longrightarrow \mathrm{BN} \longrightarrow \mathrm{LReLU}_{0.2}\\
&]\times 6 \longrightarrow \mathrm{PJ}_{32\to 3} \longrightarrow \mathrm{TANH}
\end{aligned}
$$

Τονίζουμε στο σημείο αυτό πως η αύξηση της χωρητικότητας του encoder και αντίστοιχα του decoder του G2 έγινε διότι κρίναμε ότι το έργο του G2 είναι στην πραγματικότητα πιο δύσκολο από του G1 και είναι αυτό που στην τελική θα κριθεί από τις μετρικές αξιολόγησης και, κυρίως, το ανθρώπινο μάτι.

Discriminator του *PoseGAN*

Σε αντίθεση με τους συγγραφείς του PGPG, ο Discriminator που χρησιμοποιήσαμε για την εκπαίδευση του *PoseGAN* είναι ο PatchGAN Discriminator του pix2pix με ίδια δομή και αρχιτεκτονική όπως αυτόν του σχήματος . Η αρχιτεκτονική του δίνεται παραστατικά ακολούθως.

***PoseGAN* › Discriminator**

$$
\begin{aligned}
&\mathrm{UF}_{6\longrightarrow 8} \longrightarrow [\\
&\quad \mathrm{CONV3}_{C_{in}\to 2*C_{in}} \longrightarrow \mathrm{IN} \longrightarrow \mathrm{ReLU} \longrightarrow\\
&]\times 5 \longrightarrow \mathrm{PJ}_{512\to 1}
\end{aligned}
$$

όπου η χρήση πέντε (5) ομάδων στρώσεων συστολής οδηγεί στην παραγωγή πίνακα πιθανοτήτων 2×2 στην έξοδο. *Κάθε στοιχείο του πίνακα εξόδου ουσιαστικά βλέπει ένα μέρος (patch) 32×32 της αντίστοιχης μεριάς της εισόδου*, ενώ με IN συμβολίζουμε την κανονικοποίηση δείγματος. Οι στρώσεις CONV είναι βήματος 2, δηλαδή μειώνουν στο μισό το πλάτος και ύψος των χαρτών ενεργοποίησης (ή καναλιών) στην είσοδό τους.

Παράμετροι εκπαίδευσης και αποτελέσματα

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε ο αλγόριθμος βελτιστοποίησης Adam [31] τόσο για τις παραμέτρους του Generator (τα υποδίκτυά του βελτιστοποιούνται από κοινού) όσο και του PatchGAN Discriminator. Το αρχικό βήμα εκμάθησης ορίσθηκε ίσο με 0.0001 και οι παράμετροι απόσβεσης του αλγορίθμου, β_1 και β_2, ορίσθηκαν ίσες με 0.9 και 0.999 αντίστοιχα (σύμφωνα με το [31]). Ως συνάρτηση αντιπαραθετικού κόστους χρησιμοποιήσαμε τη συνάρτηση κόστους Ελαχίστων Τετραγώνων (βλ. 3.1) ενώ εφαρμόσαμε Κανονικοποίηση Φάσματος στην τελευταία συνελικτική στρώση του Discriminator,

αυτή της προβολής των καναλιών στο ένα κανάλι εξόδου. Χρησιμοποιήσαμε το 10% των εικόνων του αρχικού συνόλου εκπαίδευσης ως σύνολο ελέγχου (test set) και το υπόλοιπο 90% ως σύνολο εκπαίδευσης.

Εκπαιδεύσαμε το μοντέλο *PoseGAN* με τις παραπάνω παραμέτρους για 93 περάσματα του συνόλου δεδομένων (epochs) σε εικόνες από το σύνολο δεδομένων ICRB (DeepFashion) που όμως πρώτα μετατρέψαμε σε ανάλυση 128×128. Πριν την είσοδό τους στο μοντέλο, οι εικόνες μετατρέπονται σε τρισδιάστατους πίνακες και οι τιμές τους κανονικοποιούνται στο δίαστημα $[-1.0, 1.0]$. Αποτελέσματα από την εφαρμογή του *PoseGAN* στη χαμηλής-ανάλυσης εκδοχή του συνόλου δεδομένων ICRB του DeepFashion δίνονται στην πρώτη ενότητα του κεφαλαίου που ακολουθεί, ενότητα 6.1. Ο ενδιαφερόμενος αναγνώστης καλείται να ανατρέξει στην ενότητα αυτή.

5.2.2 Εξαγωγή Ρούχου (PixelDTGAN)

Περνάμε, ακολούθως, στην περιγραφή του δεύτερου μοντέλου που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας. Το μοντέλο αυτό, που ονομάστηκε PixelDTGAN κατά την παρουσίασή του από τον Yoo el al. στο άρθρο τους «*Pixel-Level Domain Transfer*» [66]. Όπως φαίνεται και στον πίνακα 5 που προηγήθηκε, πρόκειται και αυτό για ένα μοντέλο Συζευγμένης Μετατροπής εικόνα-σε-εικόνα το οποίο εκπαιδεύτηκε χρησιμοποιώντας τα ζεύγη εικόνων από το σύνολο δεδομένων LookBook επαυξημένο με αυτά από το ICRB (DeepFashion) για Εξαγωγή Ρούχου από ανθρώπους-μοντέλα. Έτσι, αυτό που θέλουμε να πετύχουμε με την εκπαίδευση του μοντέλου αυτού, είναι δοθείσης μιας εικόνας που απεικονίζεται ένας το άνω μέρος του σώματος ενός ανθρώπου που φοράει ρούχα, να προσπαθήσουμε να εξάγουμε ή καλύτερα να προσεγγίσουμε τα ρούχα αυτά απεικονίζοντάς τα στην έξοδο του μοντέλου.

Το μοντέλο PixelDTGAN γενικά μοιάζει αρκετά με το pix2pix που περιγράφηκε στο προηγούμενο κεφάλαιο και προορίζεται και για παρόμοια εργασία. Δεδομένης, λοιπόν, της ανάλυσης που έχει προηγηθεί, εδώ θα είμαστε αρκετά συνοπτικοί. Θα ξεκινήσουμε με την ανάλυση του προβλήματος (τι ήθελαν να πετύχουν οι συγγραφείς), θα συνεχίσουμε δίνοντας τη δομή των δικτύων που προτάθηκαν στο [66] και θα ολοκληρώσουμε την υποενότητα αυτή δίνοντας τη δική μας προσέγγιση και αρχιτεκτονική των δικτύων που εκπαιδεύσαμε.

Ανάλυση προβλήματος Εξαγωγής Ρούχων (PixelDTGAN)

Όπως αναφέρθηκε, οι συγγραφείς προσπάθησαν μέσω της χρήσης GAN να λύσουν το πρόβλημα συζευγμένης μετατροπής εικόνας με άνθρωπο στην εικόνα του ρούχου που φοράει. Όπως αναφέρουν, ωστόσο, το πρόβλημα αυτό κανονικά δεν έχει μοναδική απάντηση και ιδανικά θα θέλαμε ένα μοντέλο να μας δίνει παραλλαγές του ρούχου που απεικονίζεται ή περισσότερες όψεις αυτού, όπως φαίνεται και στο σχήμα 80 παρακάτω.

Σχήμα 80: Απεικόνιση των πραγματικών εξόδων ή αυτών ενός ιδανικού μοντέλου εξαγωγής ρούχου.
Πηγή: Ανακατασκευή από «Pixel-Level Domain Transfer», Yoo et al., 2016 [66]

Οι συγγραφείς, ωστόσο, εστίασαν στην παραγωγή ενός μόνο ρούχου ή μιας μόνο όψης αυτού από το μοντέλο GAN που ανέπτυξαν καθώς έτσι διευκολύνεται σημαντικά το έργο του Generator. Επιπρόσθετα, το σύνολο δεδομένων που συνέλεξαν για την εκπαίδευση του μοντέλου τους, το LookBook που αναλύθηκε στην προηγούμενη ενότητα, αποτελείται από ζεύγη πολλά-προς-ένα με το «πολλά» να είναι εικόνες ανθρώπων και το «ένα» να είναι εικόνα ρούχου που απεικονίζεται σε όλους αυτούς.

Ένα άλλο σημαντικό σημείο που διαπίστωσαν οι συγγραφείς του [66] είναι, όπως φαίνεται και στο σχήμα παραπάνω, δεν αρκεί ο Generator ενός καλά εκπαιδευμένου GAN να παράγει ρεαλιστικές εικόνες ρούχων στη έξοδό του, αλλά πρέπει αυτές να είναι συνυφασμένες με το ρούχο που φοράει ο άνθρωπος που εικονίζεται στην εικόνα εισόδου (ή συνθήκη) αυτού. Επομένως, πρότειναν τον έλεγχο από ξεχωριστά δίκτυα-Discriminators των παραγόμενων εικόνων με το ένα να μετράει και να δίνει feedback σχετικά με τον ρεαλισμό και το άλλο σχετικά με το αν σχετίζονται οι εικόνες εισόδου-εξόδου σημασιολογικά (δηλ. εάν για είσοδο άνδρα που φοράει μπουφάν η έξοδος είναι εικόνα που μοιάζει με μπουφάν). Πρόκειται για μία καινοτομία των συγγραφέων που σύμφωνα με αυτούς οδήγησε στην πιο γρήγορη εκπαίδευση του μοντέλου τους. Παρακάτω, δίνουν με την αρχιτεκτονική και αναλύουμε τη δομή των προτεινόμενων δικτύων.

Αρχιτεκτονική του μοντέλου (PixelDTGAN)

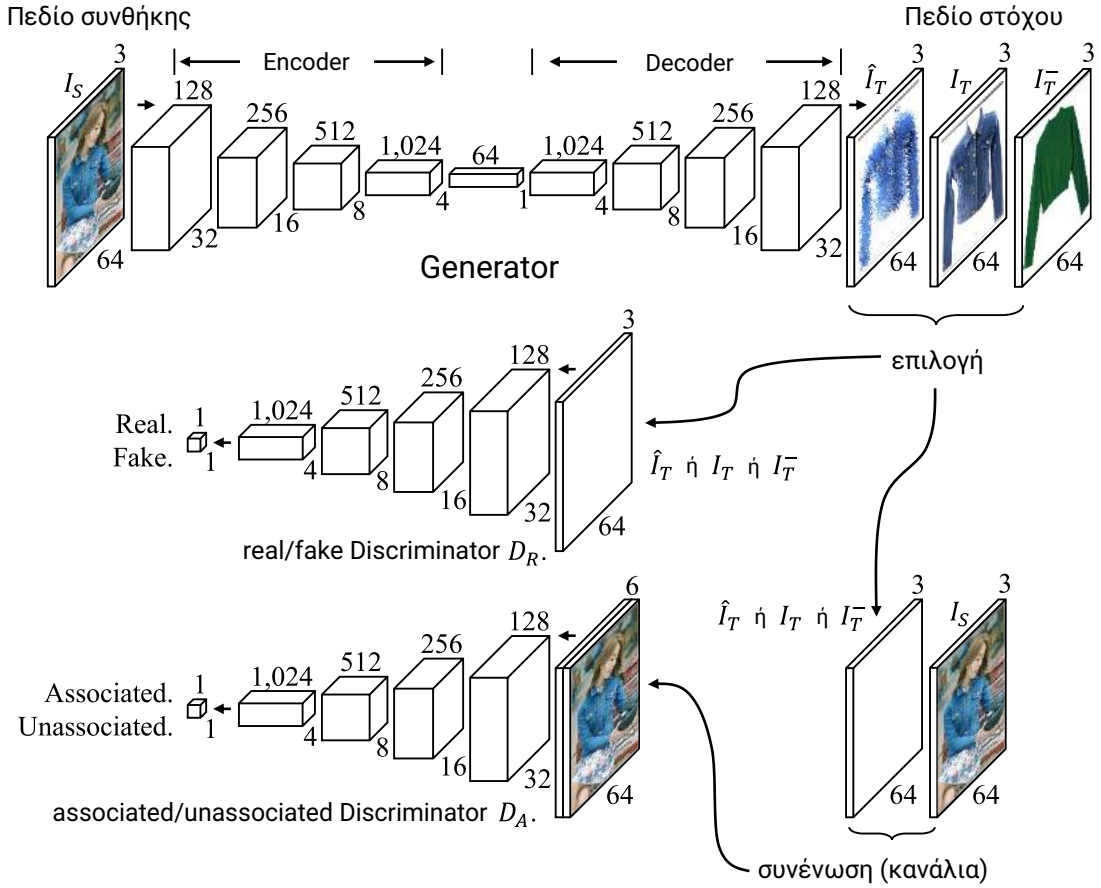
Συνοπτικά, η αρχιτεκτονική του μοντέλου (PixelDTGAN) συνοψίζεται στα ακόλουθα δίκτυα:

- **Generator**: μοιάζει με το δίκτυο U-Net, δηλαδή τον Generator του pix2pix, ωστόσο έχει δύο σημαντικές διαφοροποιήσεις: πρώτον δεν έχει skip connections από τον encoder στον decoder και δεύτερον σε κάθε ομάδα στρώσεων περιέχεται μία (1) αντί για δύο (2) συνελικτικές στρώσεις. Ουσιαστικά, *δεν διακρίνει τίποτα άλλο τον Generator του PixelDTGAN από έναν AK, πέρα από το ότι δεν εκπαιδεύεται για αυτοκωδικοποίηση της εισόδου*. Για την εκπαίδευσή του Generator οι συγγραφείς χρησιμοποίησαν μόνο αντιπαραθετική συνάρτηση κόστους και όχι ανακατασκευής (όπως προηγούμενα).

- **Discriminators**: η καινοτομία και συνεισφορά των συγγραφέων του μοντέλου ήταν, όπως αναφέρθηκε, η χρήση δύο Discriminators, τον D_R (ή real/fake Discriminator όπως τον ονομάζουν) που εκπαιδεύεται να επιστρέφει πιθανότητες ρεαλισμού των εικόνων εισόδους και τον D_A (ή associated/unassociated Discriminator όπως τον ονομάζουν) που εκπαιδεύεται να επιστρέφει πιθανότητες (σημασιολογικής) συσχέτισης μεταξύ των εικόνων εισόδου. Μία άλλη λεπτομέρεια της υλοποίησής τους είναι ότι αν και έχουμε υπο-συνθήκη παραγωγή, οι συγγραφείς επέλεξαν να μην δίνουν στον real/fake Discriminator, D_R, την εικόνα συνθήκης κάτι που κάνουν ωστόσο για τον associated/unassociated Discriminator, D_A. Η λογικής τους εδώ είναι και πάλι ότι κάτι τέτοιο θα επιταχύνει ακόμα περισσότερο την εκπαίδευση ενώ η επιβολή της υπο-συνθήκης παραγωγής γίνεται ταυτόχρονα από το άλλο δίκτυο, δηλαδή *οι Discriminators δουλεύουν συνεργατικά*. Τέλος, για καλύτερη εκπαίδευση του associated/unassociated Discriminator, D_A, οι συγγραφείς δίνουν εκτός από ζεύγη εικόνων συνθήκη-(πραγματική/τεχνητή του πεδίου στόχος) όπου εκπαιδεύεται να βγάλει 1/0 αντίστοιχα και ζεύγη εικόνων συνθήκη-(πραγματική αλλά αταίριαστη του πεδίου στόχος), όπου η εικόνα (πραγματική αλλά αταίριαστη του πεδίου στόχος) προέρχεται από το (πραγματικό) σύνολο δεδομένων του δεύτερου πεδίου, αλλά δεν είναι η σωστή πραγματική εικόνα εξόδου για τη δεδομένη εικόνα εισόδου (ή συνθήκης).

Στο σχήμα 81 παρακάτω δίνουμε την αρχιτεκτονική του μοντέλου όπως προτάθηκε στο [66]. Εκεί, με I_S σημειώνεται η εικόνα εισόδου στον Generator καθώς και η μία από τις δύο εικόνες που εισάγεται στον associated/unassociated Discriminator, D_A, με I_T σημειώνεται

η πραγματική εικόνα εξόδου, ενώ με $\hat{I}_T$ η έξοδος του Generator (που εκπαιδεύεται για να προσεγγίζει την I_T). Τέλος, με I_T^- σημειώνεται μία άλλη πραγματική εικόνα ασυσχέτιστη με την εικόνα εισόδου, ενώ το T δηλώνει ότι οι εικόνες ανήκουν σε αυτές του πεδίου-στόχος.



Σχήμα 81: Πλήρης αρχιτεκτονική του μοντέλου PixelDTGAN. Φαίνονται το δίκτυο του Generator (επάνω), καθώς και οι δύο Discriminators (μέση και κάτω). Επίσης, απεικονίζεται η είσοδος και έξοδος του κάθε δικτύου.

Πηγή: Ανακατασκευή από «Pixel-Level Domain Transfer», Yoo et al., 2016 [66]

Για την εκπαίδευση του μοντέλου τους, οι συγγραφείς χρησιμοποιήσαν τη συνάρτηση κόστους Binary Cross-Entropy σε όλα τα δίκτυα. Επίσης, οι δύο Discriminators του μοντέλου εκπαιδεύονται και βελτιστοποιούνται *ξεχωριστά*. Τέλος, οι συγγραφείς εκπαίδευσαν το μοντέλο τους σε εικόνες χαμηλότερης ανάλυσης από αυτήν του συνόλου δεδομένων και συγκεκριμένα σε ανάλυση 64×64. Ακολουθεί, η δική μας υλοποίηση του PixelDTGAN.

Υλοποίηση του (PixelDTGAN)

Η δική μας υλοποίηση του μοντέλου (PixelDTGAN) δεν αποτελεί πιστή υλοποίηση της παραπάνω αρχιτεκτονικής και αντιγραφή των παραμέτρων εκπαίδευσης. Ο λόγος είναι διττός: κατά πρώτον όταν παρουσιάστηκε το [66] τόσο οι Discriminators όσο και οι Generators δεν ήταν τόσο εξελιγμένοι (π.χ. δεν χρησιμοποιούνταν η στρώση κανονικοποίηση φάσματος ή η συνάρτηση κόστους ελαχίστων τετραγώνων) και κατά δεύτερον ο εκπονητής της παρούσας ήταν ανέκαθεν κατά της αντιγραφής-επικόλλησης εργασιών και υπέρ του πειραματισμού. Έτσι, παρακάτω παραθέτουμε σειριακά τις αρχιτεκτονικές του Generator, του real/fake Discriminator και του associated/unassociated Discriminator, ενώ στο τέλος της υποενότητας παραθέτουμε τις παραμέτρους εκπαίδευσης που χρησιμοποιήθηκαν.

Generator του PixelDTGAN

Ο Generator που υλοποιήσαμε δεν διαφέρει σημαντικά από αυτόν του PixelDTGAN με την ουσιαστικότερη διαφορά να εντοπίζεται στο σημείο στένωσης, όπου το μήκος των δικών μας διανυσμάτων (ή αριθμός καναλιών εάν ειδωθεί ως η έξοδος της προηγούμενης συνελικτικής στρώσης) είναι 100 αντί για 64 που φαίνεται στο παραπάνω σχήμα. Επίσης, στη δική μας υλοποίηση, ο Generator εκπαιδεύεται ταυτόχρονα με τα αντιπαραθετικά κόστη από τους δύο Discriminators **και με κόστος ανακατασκευής L1**, κάτι που όπως κρίναμε από την εκπαίδευση του *PoseGAN* βοηθάει με την ευστάθεια της εκπαίδευσης. Ακολουθεί η αρχιτεκτονική του δικτύου του Generator, η οποία δίνεται χρησιμοποιώντας το ίδιο σύνολο συμβολισμών για τις επιμέρους στρώσεις όπως και στην περιγραφή του *PoseGAN* στην προηγούμενη ενότητα.

PixelDTGAN › Generator › Encoder

$$\begin{aligned}&\mathrm{CONV5}_{3\to 128} \longrightarrow \mathrm{LReLU}_{0.2} \longrightarrow [\\ &\quad \mathrm{CONV3}_{C_{in}\to 2*C_{in}} \longrightarrow \mathrm{BN} \longrightarrow \mathrm{LReLU}_{0.2}\\ &] \times 3 \longrightarrow \mathrm{CONV4}_{1024\to 100} \longrightarrow \mathrm{PN}\end{aligned}$$

όπου οι τετράγωνες αγκύλες εσωκλείουν τις ομάδες στρώσεων συστολής (contracting blocks) (πέντε (5) χρησιμοποιήθηκαν συνολικά στον encoder του Generator συμπεριλαμβανομένης της αρχικής στρώσης και της στρώσης στο σήμειο στένωσης που χρησιμοποιεί Κανονικοποίηση Εικονοστοιχείων - γενικά είναι μία από τις υπερ-παραμέτρους του μοντέλου). Με PN συμβολίζουμε τη στρώση Κανονικοποίηση Εικονοστοιχείων.

PixelDTGAN › Generator › Decoder

$$\text{TCONV4}_{100\to1024} \longrightarrow \text{PN} \longrightarrow \text{ReLU} \longrightarrow [$$
$$\quad \text{TCONV5}_{C_{in}\to C_{in}/2} \longrightarrow \text{BN} \longrightarrow \text{ReLU}$$
$$] \times 3 \longrightarrow \text{TCONV5}_{128\to3} \longrightarrow \text{TANH}$$

όπου με TCONVX συμβολίζουμε μία ανάστροφη συνελικτική στρώση με φίλτρα X×X, ενός όπως φαίνεται και σύμφωνα με το [66], οι ανορθωμένες γραμμικές μονάδες στον decoder δεν είναι leaky όπως στον encoder αλλά κανονικές.

Discriminators του PixelDTGAN

Οι δύο Discriminators είναι πανομοιότυποι στη σχεδίαση. Διαφέρουν μόνο στο ότι ο D_A λαμβάνει στην είσοδό του δύο εικόνες συνενωμένες και άρα έξι (6) κανάλια, ενώ ο D_R δέχεται μία εικόνα και άρα τρία (3) κανάλια. Ως πρότυπο δίκτυο για τους Discriminators χρησιμοποιήθηκε ο PatchGAN, κάτι που γενικά έχει ακολουθηθεί σε όλα τα μοντέλα που εκπαιδεύτηκαν πλην του τελευταίου και κάτι που γενικά εφαρμόζεται όλο και συχνότερη στη σχετική βιβλιογραφία. Σε σύγκριση με τον Discriminator του *PoseGAN*, οι δύο Discriminators εδώ έχουν παρόμοια δομή, ωστόσο εδώ **δοκιμάσαμε 4 ομάδες στρώσεων συστολής (αντί για 5) και 128 κανάλια βάσης (αντί για 16)**. Επίσης, οι δύο Discriminators δεν περιέχουν την αρχική στρώση αύξησης των καναλιών - αντ΄ αυτού ξεκινάνε αμέσως οι ομάδες στρώσεων συστολής. Η κοινή αρχιτεκτονική και των δύο Discriminators δίνεται ακολούθως.

PixelDTGAN › Discriminator

$$\text{CONV3}_{3/6\longrightarrow128} \longrightarrow [$$
$$\quad \text{CONV3}_{C_{in}\to2*C_{in}} \longrightarrow \text{IN} \longrightarrow \text{ReLU} \longrightarrow$$
$$] \times 4 \longrightarrow \text{PJ}_{1024\to1}$$

όπου η χρήση τεσσάρων (4) ομάδων στρώσεων συστολής οδηγεί στην παραγωγή πίνακα πιθανοτήτων 4×4 στην έξοδο. *Κάθε στοιχείο του πίνακα εξόδου ουσιαστικά βλέπει ένα μέρος (patch) 16×16 της αντίστοιχης μεριάς της εισόδου*, ενώ με IN συμβολίζουμε την κανονικοποίηση δείγματος. Οι στρώσεις CONV είναι βήματος 2.

Παράμετροι εκπαίδευσης και αποτελέσματα

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε ο αλγόριθμος βελτιστοποίησης Adam [31] τόσο για τις παραμέτρους του Generator όσο και για αυτές του καθενός PatchGAN Discriminator. Το αρχικό βήμα εκμάθησης ορίσθηκε σε όλους τους βελτιστοποιητές ίσο

με 0.0001 και οι παράμετροι απόσβεσης του αλγορίθμου, β_1 και β_2, ορίσθηκαν ίσες με 0.9 και 0.999 αντίστοιχα (σύμφωνα με το [31]). Ως συνάρτηση αντιπαραθετικού κόστους και στους δύο Discriminators χρησιμοποιήθηκε η συνάρτηση κόστους Ελαχίστων Τετραγώνων (βλ. 3.1) ενώ εφαρμόσαμε Κανονικοποίηση Φάσματος στην τελευταία συνελικτική στρώση και των δύο PatchGAN Discriminators (δηλ. της στρώση προβολής των καναλιών στο ένα κανάλι εξόδου). Χρησιμοποιήσαμε το 10% των εικόνων του αρχικού συνόλου εκπαίδευσης ως σύνολο ελέγχου (test set) και το υπόλοιπο 90% ως σύνολο εκπαίδευσης.

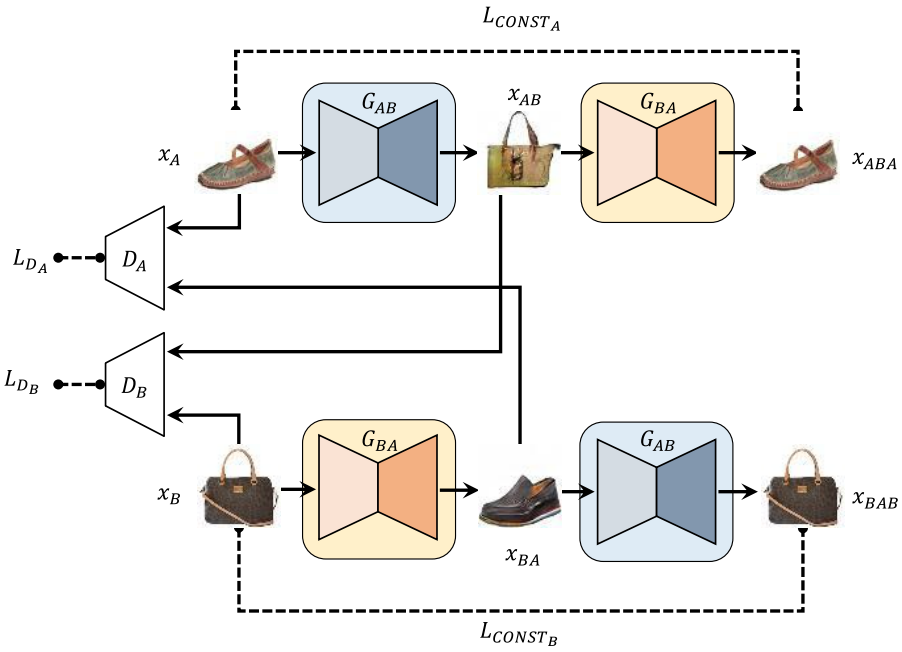
Εκπαιδεύσαμε την υλοποίησή μας του μοντέλου PixelDTGAN με τις παραπάνω παραμέτρους για 348 περάσματα του συνόλου δεδομένων (epochs) σε εικόνες από το σύνολο δεδομένων LookBook + ICRB (DeepFashion) που όμως πρώτα μετατρέψαμε σε ανάλυση 64×64. Πριν την είσοδό τους στο μοντέλο, οι εικόνες μετατρέπονται σε τρισδιάστατους πίνακες και οι τιμές τους κανονικοποιούνται στο διάστημα $[-1.0, 1.0]$. Αποτελέσματα από την εφαρμογή του PixelDTGAN στη χαμηλής-ανάλυσης εκδοχή του συνόλου δεδομένων LookBook και ICRB (DeepFashion) δίνονται στη δεύτερη ενότητα του κεφαλαίου που ακολουθεί, ενότητα 6.2. Ο ενδιαφερόμενος αναγνώστης καλείται να ανατρέξει στην ενότητα αυτή, ενώ ακολούθως παραθέτουμε για σύγκριση τα αποτελέσματα (δηλ. μερικές παραγωγές) του PixelDTGAN όπως παρουσιάστηκαν στο [66]:

Σχήμα 82: Παραγωγές του μοντέλου PixelDTGAN όπως παρουσιάστηκαν στο αντίστοιχο άρθρο.
Πηγή: Ανακατασκευή από «Pixel-Level Domain Transfer», Yoo et al., 2016 [66]

5.2.3 Ταίριασμα Στιλ (DiscoGAN - CycleGAN)

Το τρίτο μοντέλο που υλοποιήσαμε και εκπαιδεύσαμε στα πλαίσια της παρούσας της εργασίας είναι μία παραλλαγή του CycleGAN που αναφέρθηκε στο τέλος του προηγούμενου κεφαλαίου. Θα αναφερθούμε, επομένως, στην παρούσα υποενότητα σε Μη-Συζευγμένη Μετατροπή εικόνας-σε-εικόνα προκειμένου να ταιριάξουμε την οπτική εμφάνιση ή στιλ

μιας εικόνας παπουτσιού που δίνεται στην είσοδο με μια εικόνα τσάντας ή το αντίστροφο. Το μοντέλο (DiscoGAN), στο οποίο οποία θα βασιστούμε για την υλοποίησή μας, παρουσιάστηκε το 2017 από τον Kim et al. στο άρθρο τους «*Learning to Discover Cross-Domain Relations with Generative Adversarial Networks*» [76], στην προσπάθειά τους να εκπαιδεύσουν ένα μοντέλο που αναγνωρίζει συσχετίσεις μεταξύ δύο πεδίων εικόνων χρησιμοποιώντας δύο ξεχωριστά (μη-ζευγαρωμένα) σύνολα δεδομένων εικόνων. Η πλήρης αρχιτεκτονική του μοντέλου DiscoGAN που αναπτύχθηκε δίνεται ακολούθως, ενώ όπως αναφέραμε κατά την ανάλυση των συνόλων δεδομένων, θα χρησιμοποιήσουμε το μοντέλο αυτό αφού το εκπαιδεύσουμε στο σύνολο δεδομένων handbags2shoes **για παραγωγή εικόνων τσαντών που ταιριάζουν με εικόνες παπουτσιών και το αντίστροφο**.



Σχήμα 83: Πλήρης αρχιτεκτονική του μοντέλου DiscoGAN. Φαίνονται τα δύο μοντέλα GAN, (G_{AB}, D_B) και (G_{AB}, D_B), ενδεικτικές είσοδοι και έξοδοι του κάθε δικτύου καθώς και το κόστος κυκλικής συνοχής.

Πηγή: «Learning to Discover Cross-Domain Relations with Generative Adversarial Networks», Kim et al., 2017 [76]

Αν και το άρθρο παρουσιάστηκε σε μεταγενέστερο χρόνο από αυτό του μοντέλου CycleGAN για μη-συζευγμένη μετατροπη, εντούτοις οι συγγραφείς του DiscoGAN δεν αναφέρουν σε κανένα σημείο το πρώτο. Αντιθέτως, παραθέτουν το δικό τους σκεπτικό πίσω από τη χρήση δύο GANs (όπως δηλαδή και στο CycleGAN) και την από-κοινού εκπαίδευσή τους. Το παρόν μοντέλο, στα πλαίσια της παρούσας εργασίας αναπτύχθηκε σύμφωνα με

το CycleGAN όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο και με μικρές αλλαγές από το DiscoGAN, που έχουν να κάνουν κυρίως με τις παραμέτρους των συνελικτικών στρώσεων και όχι με την ουσία των δικτύων - εξάλλου οι δύο αρχιτεκτονικές είναι πολύ κοντά αν όχι η ίδια. Θεωρούμε επομένως αποδεκτό να προσπεράσουμε την ανάλυση των συγγραφέων του [76] και να επικεντρωθούμε απευθείας στην υλοποίησή μας και στον τρόπο εκπαίδευσης του μοντέλου που αναπτύχθηκε.

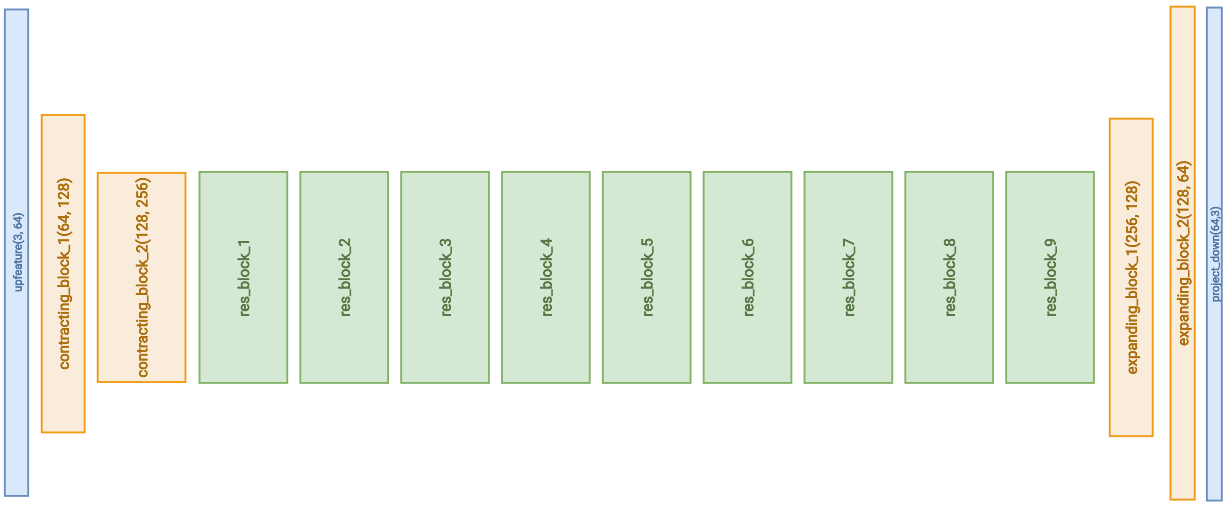
Υλοποίηση του DiscoGAN - CycleGAN

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, το CycleGAN αποτείται από δύο GANs καθένα από τα οποία καλείται κάθε φορά να παράγει μία ρεαλιστική εικόνα από το πεδίο εικόνων της εξόδου του λαμβάνοντας ως συνθήκη μία εικόνα από το πεδίο εικόνων της εισόδου του. Έτσι και σύμφωνα με το σχήμα 83, το μοντέλο που αναπτύξαμε κατ΄ αρχάς θα εκπαιδευτεί από σύνολο δεδομένων αποτελούμενο από δύο διακριτά πεδία εικόνων, το A και το B, τα οποία αν και έχουν θεωρητικά κάποια συγγένεια δεν είναι συλλεγμένα ως ζεύγη εικόνων από κάθε πεδίο. Όπως αναφέρθηκε το σύνολο αυτό θα είναι το handbags2shoes αποτελούμενο από ένα σύνολο εικόνων υποδημάτων και ένα τσαντών χεριού. Έτσι, σχεδιάστηκαν και εκπαιδεύτηκαν δύο (2) Generators, ο G_{AB} που **δεχόμενος παπούτσια παράγει εικόνες τσαντών που ταιριάζουν στιλιστικά** και ο G_{BA} για την αντίστροφη εργασία. Ταυτόχρονα, σχεδιάστηκαν και εκπαιδεύτηκαν δύο (2) Discriminators, ο D_B που προσπαθεί να διακρίνει τις πραγματικές από τις τεχνητές εικόνες τσαντών και ο D_A παπουτσιών. Οι δύο Generators εκπαιδεύονται από κοινού και, **αντίθετα από το CycleGAN και σύμφωνα με το DiscoGAN, εκπαιδεύσαμε από κοινού και τους Discriminators** (κοινός βελτιστοποιητής για τα δύο δίκτυα). Ακολούθως παραθέτουμε τις αρχιτεκτονικές των δικτύων και τις παραμέτρους εκπαίδευσης αυτών.

Αρχιτεκτονική των Generators

Όπως έχουμε τονίσει, οι δύο Generators είναι πανομοιότυπα δίκτυα, τα οποία ακολουθούν τη δομή που παρουσιάστηκε στην αντίστοιχη παράγραφο του μοντέλου CycleGAN. Για υπενθύμιση, αναφέρουμε πως αποτελούνται από τρία μέρη: τον encoder με συνελικτικές στρώσεις μείωσης διαστάσεων και αύξησης του βάθους, του ενδιάμεσου δικτύου (στη βιβλιογραφία αυτό λέγεται και *δίκτυο transformer*) αποτελούμενου από στρώσεις με residual connections οι οποίες όμως δεν μεταβάλλουν τις διαστάσεις των χαρτών ενεργοποίησης και τον decoder ο οποίος περιέχει ανάστροφες συνελικτικές στρώσεις με συμμετρικά (mirrored) χαρακτηριστικά αυτών του encoder. Παρακάτω παραθέτουμε

σχηματικά τη δομή του κάθε Generator, ενώ ακολούθως την περιγράφουμε με τρόπο αντίστοιχο με τα μοντέλα που προηγήθηκαν.



Σχήμα 84: Αρχιτεκτονική των Generators του μοντέλου DiscoGAN που υλοποιήσαμε.

Όπως φαίνεται στο σχήμα, χρησιμοποιήθηκαν εννέα (9) ομάδες στρώσεων residual, σε καθεμία από τις οποίες περιέχονται δύο συνελικτικές στρώσεις ακολουθούμενες από την αντίστοιχη στρώση κανονικοποίησης, ενώ μετά τη πρώτη στρώση κανονικοποίησης οι χάρτες περνούν από ανορθωμένη γραμμική μονάδα. Οι συγγραφείς του CycleGAN αναφέρουν πως αυτές οι στρώσεις είναι που καταφέρνουν να αιχμαλωτίσουν το περιεχόμενο και στιλ του εκάστοτε πεδίου εξόδου και έτσι να μετασχηματίσουν επιτυχώς την εικόνα εισόδου από το ένα πεδίο στο άλλο. Παρακάτω, δίνονται παραστατικά τα στοιχεία της αρχιτεκτονικής του κάθε Generator.

DiscoGAN › Generator

$$
\begin{aligned}
&\mathrm{UF}_{3\to 64} \longrightarrow [\\
&\quad \mathrm{CONV3}_{C_{in}\to 2*C_{in}} \longrightarrow \mathrm{IN} \longrightarrow \mathrm{ReLU}\\
&] \times 2 \longrightarrow (256{\times}16{\times}16) \longrightarrow [\\
&\quad \mathrm{CONV3}_{C_{in}\to C_{in}} \longrightarrow \mathrm{IN} \longrightarrow \mathrm{ReLU} \longrightarrow \mathrm{CONV3}_{C_{in}\to C_{in}} \longrightarrow \mathrm{IN}\\
&] \times 9 \longrightarrow [\\
&\quad \mathrm{TCONV3}_{C_{in}\to 2*C_{in}} \longrightarrow \mathrm{IN} \longrightarrow \mathrm{ReLU}\\
&] \times 2 \longrightarrow \mathrm{PJ}_{64\to 3} \longrightarrow \mathrm{TANH}
\end{aligned}
$$

όπου στο πρώτο σετ των αγκυλών περιλαμβάνονται οι ομάδες στρώσεων συστολής (contracting blocks) του encoder, στο δεύτερο του transformer (residual blocks) και στο

τρίτο του decoder. Σημειώνεται εδώ ότι τα **residual blocks επενεργούν στους χάρτες ενεργοποίησης χωρίς να αλλάζουν τις διαστάσεις τους** έτσι ώστε να μπορεί να γίνει η πρόσθεση στην έξοδο του κάθε block της αντίστοιχης εισόδου του.

Αρχιτεκτονική των Discriminators

Όπως για κάθε δίκτυο Discriminator έτσι και εδώ χρησιμοποιήθηκε ο PatchGAN Discriminator. Συγκεκριμένα, χρησιμοποιήθηκαν τέσσερις (4) ομάδες στρώσεων συστολής και άρα το κάθε δίκτυο Discriminator είναι πανομοιότυπο με αυτό που χρησιμοποιήθηκε στους Discriminators του PixelDTGAN. Ωστόσο τα κανάλια βάσης εδώ είναι 64 σύμφωνα με το σχετικό άρθρο (αντί για 128 που χρησιμοποιήσαμε στο PixelDTGAN ή για 8 στο *PoseGAN*). Ακολουθεί η αρχιτεκτονική των δικτύων και σύντομα σχόλια.

DiscoGAN › Discriminator

$$\text{CONV3}_{3/6 \longrightarrow 64} \longrightarrow [$$
$$\text{CONV3}_{C_{in} \rightarrow 2*C_{in}} \longrightarrow \text{IN} \longrightarrow \text{ReLU} \longrightarrow$$
$$] \times 4 \longrightarrow \text{PJ}_{1024 \rightarrow 1}$$

Η χρήση τεσσάρων (4) ομάδων στρώσεων συστολής οδηγεί και εδώ στην παραγωγή πίνακα πιθανοτήτων 4×4 στην έξοδο. *Κάθε στοιχείο του πίνακα εξόδου ουσιαστικά βλέπει ένα μέρος (patch) 16×16 της αντίστοιχης μεριάς της εισόδου*, ενώ με IN συμβολίζουμε την κανονικοποίηση δείγματος. Οι στρώσεις CONV είναι βήματος 2.

Παράμετροι εκπαίδευσης και αποτελέσματα

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν δύο βελτιστοποιητές: ένας για την από κοινού βελτιστοποίηση των παραμέτρων των δύο Generators και ένας για την από κοινού βελτιστοποίηση των παραμέτρων των δύο Discriminators. Η επιβολή κοινής εκπαίδευσης και στους Discriminators (εκτός από του Generators), σύμφωνα με τους συγγραφείς του DiscoGAN, ενθαρρύνει το μοντέλο να μάθει έναν αντιστρέψιμο 1-1 μετασχηματισμό εικόνων μεταξύ των δύο πεδίων. Και στους δύο βελτιστοποιητές χρησιμοποιήθηκε ως αλγόριθμος βελτιστοποίησης ο Adam [31]. Το αρχικό βήμα εκμάθησης ορίσθηκε σε όλους τους βελτιστοποιητές ίσο με 0.0002 και οι παράμετροι απόσβεσης του αλγορίθμου, β_1 και β_2, ορίσθηκαν ίσες με 0.9 και 0.999 αντίστοιχα (σύμφωνα με το [31]). Ως σύνάρτηση αντιπαραθετικού κόστους και στους δύο Discriminators χρησιμοποιήθηκε η συνάρτηση κόστους Ελαχίστων Τετραγώνων (βλ. 3.1) ενώ εφαρμόσαμε Κανονικοποίηση Φάσματος στην τελευταία συνελικτική στρώση και των δύο PatchGAN Discriminators (δηλ. της στρώση προβολής των καναλιών στο ένα κανάλι εξόδου). Χρησιμοποιήσαμε

το 10% των εικόνων του αρχικού συνόλου εκπαίδευσης ως σύνολο ελέγχου (test set) και το υπόλοιπο 90% ως σύνολο εκπαίδευσης. Τέλος, στο μοντέλο αυτό δοκιμάστηκε ο ακόλουθος **προγραμματιστής του βήματος εκπαίδευσης**: όταν η συνάρτηση κόστους επιπεδώσει ή αρχίζει να ανεβαίνει τότε ο προγραμματιστής κατεβάζει το βήμα στο ένα τοις χιλίοις (1‰) της πρότερής του τιμής και περιμένει για 200 βήματα μέχρι να ξανακάνει τον έλεγχο (ο έλεγχος διαρκεί 10 τουλάχιστον βήματα).

Εκπαιδεύσαμε την υλοποίησή μας του μοντέλου DiscoGAN με τις παραπάνω παραμέτρους για 189 περάσματα του συνόλου δεδομένων (epochs) σε εικόνες από το σύνολο δεδομένων handbags2shoes που περιέχει εικόνες από τσάντες και παπούτσια ανάλυσης 64×64. Πριν την είσοδό τους στο μοντέλο, οι εικόνες μετατρέπονται σε τρισδιάστατους πίνακες και οι τιμές τους κανονικοποιούνται στο διάστημα $[-1.0, 1.0]$. Αποτελέσματα από την εφαρμογή της υλοποίησής μας του DiscoGAN στο σύνολο δεδομένων handbags2shoes δίνονται στη δεύτερη ενότητα του κεφαλαίου που ακολουθεί, ενότητα 6.3. Ο ενδιαφερόμενος αναγνώστης καλείται να ανατρέξει στην ενότητα αυτή, ενώ ακολούθως παραθέτουμε για σύγκριση τα αποτελέσματα (δηλ. μερικές παραγωγές) του DiscoGAN όπως παρουσιάστηκαν στο [76]:

5.2.4 Παραγωγή ρεαλιστικών εικόνων μόδας (StyleGAN)

Το τελευταίο μοντέλο που αναπτύχθηκε και εκπαιδεύτηκε στην παρούσα εργασία δεν θα μπορούσε να είναι άλλο από το StyleGAN. Με το μοντέλο αυτό, το οποίο ανήκει στα GANs που παράγουν εικόνα από θόρυβο, καλύπτουμε όλο το φάσμα των κατηγοριών εφαρμογών των GANs στα πλαίσια της Παραγωγικής Μοντελοποίησης εικόνων. Το μοντέλο εκπαιδεύτηκε στο σύνολο δεδομένων Fashion-Image Synthesis Benchmark (FISB) του DeepFashion το οποίο, όπως αναλύθηκε, αποτελείται από εικόνες ανάλυσης 128×128 με ανθρώπους-μοντέλα να ποζάρουν για φωτογραφίσεις ρούχων.

Έχουμε αναλύσει λεπτομερειακά την πρώτη έκδοση του μοντέλου αυτού (βλ. υποενότητα 4.1.3), την οποία υλοποιήσαμε και εκπαιδεύσαμε, γι αυτό δεν θα επεκταθούμε ιδιαίτερα σε αυτήν την υποενότητα. Έτσι, σε ότι ακολουθεί απλώς θα δώσουμε παραστατικά τα στοιχεία της αρχιτεκτονικής τόσο του Style-based Generator όσο και του PatchGAN Discriminator, επισημαίνοντας όλες τις διαφορές και απλοποιήσεις της υλοποίησής μας σε σύγκριση με το αρχικό.

Style-based Generator

(α) Παπούτσια σε τσάντες (G_{AB})

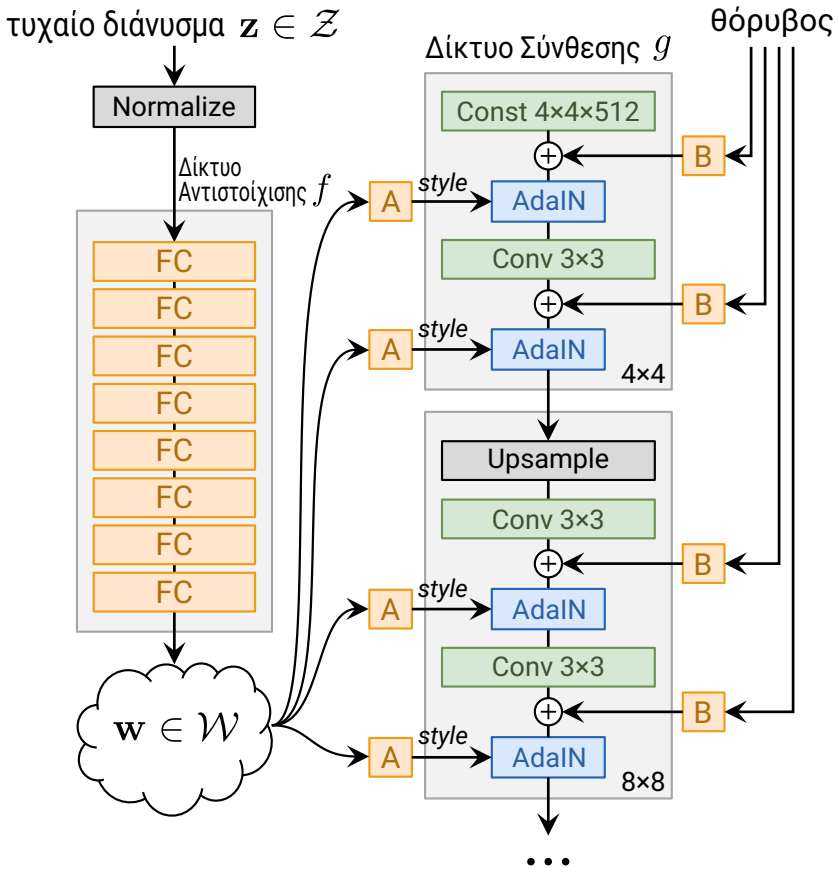
(β) Τσάντες σε παπούτσια (G_{BA})

Σχήμα 85: Παραγωγές του μοντέλου DiscoGAN που εκπαιδεύτηκε στο handbags2shoes, όπως παρουσιάστηκαν στο αντίστοιχο άρθρο.
Πηγή: Ανακατασκευή από «Learning to Discover Cross-Domain Relations with Generative Adversarial Networks», Kim et al., 2017 [76]

Όπως αναφέρθηκε στην υποενότητα 4.1.3 κατά την ανάλυση του StyleGAN [91], αυτό είναι μια παραλλαγή του DCGAN με σειρά καινοτομιών. Ο Generator του μοντέλου καλείται να παράγει ρεαλιστικές και με ποικιλομορφία εικόνες στην έξοδό του, λαμβάνοντας ως είσοδο ένα διάνυσμα τυχαίου θορύβου. Για να το πετύχει αυτό γρήγορα και σταθερά, ξεκινάει να παράγει χαμηλής ανάλυσης εικόνες και σταδιακά αυξάνει το μέγεθός του (διπλασιάζοντάς το), με την προσθήκη ενός ακόμα νέου συνελικτικού block και στα δύο δίκτυα. Το πρώτο μέρος του Generator του StyleGAN, ή Style-based Generator, ε-ίναι το Δίκτυο Αντιστοίχισης Θορύβου, ενώ ακολούθως είναι το Δίκτυο Σύνθεσης που περιλαμβάνει συνελικτικές στρώσεις και στρώσεις Προσαρμοστικής Κανονικοποίησης Δείγματος, σύμφωνα με το παρακάτω σχήμα.

Διαφοροποιήσαμε την υλοποίησή μας του Style-based Generator, προκειμένου να απλοποι-ήσουμε τη σχεδίασή του, σε δύο βασικά σημεία:

1. **Τέσσερις (4) Πλήρως-Συνδεδεμένες Στρώσεις**: στο δίκτυο αντιστοίχισης θορύβου (ή δίκτυο παραγωγής των διανυσμάτων στυλ, $\vec{w}$) αντί για οκτώ (8) πλήρως συνδε-



Σχήμα 86: Αρχιτεκτονικη του Generator του μοντέλου StyleGAN.

δεμένες στρώσεις χρησιμοποιήσαμε τέσσερις (4). Με αυτήν την αλλαγή εξοικονομήσαμε περίπου $1.1M$ παραμέτρους και κάναμε πιο γρήγορη την εκπαίδευση του Generator, σε βάρος του πλέον χειρότερου ξεμπερδέματος του λανθάνοντα χώρου.

2. **Μη-χρήση Equalized Learning-Rate**: το equalized learning-Rate στην ουσία είναι ένα είδος κανονικοποίησης των βαρών (όχι εξόδων) των συνελικτικών στρώσεων, η οποία δουλεύει με τρόπο παρόμοιο με αυτόν της Φασματικής Κανονικοποίησης αλλά με διαφορετικό τύπο. Σύμφωνα με τους συγγραφείς, αυτή είναι χρήσιμη σε κάποιες περιπτώσεις όπου το βήμα εκπαίδευσης είναι ταυτόχρονα πολύ μικρό και πολύ μεγάλο, ωστόσο επειδή θέλαμε να φτιάξουμε από τη αρχή το μοντέλο και όχι να το αντιγράψουμε, αποφασίσαμε να βγάλουμε αυτές τις στρώσεις.

3. **Πιο ομαλές μεταβάσεις**: η παράμετρος μίξης a (βλ. PGGAN) στη δική μας υλοποίηση και όταν διπλασιάζονται τα δίκτυα μεταβαίνει ομαλά από τη τιμή 0 (όπου τότε δεν προσμετρούνται οι έξοδοι των νέων blocks) σε 1. Εμείς χρησιμοποιήσαμε μιας σιγμοειδής μορφή συνάρτηση μετάβασης, ώστε αυτή να γίνει πιο ομαλά. Αντίθετα,

οι συγγραφείς χρησιμοποιούν γραμμική και επίσης με μικρότερα διαστήματα (πιο γρήγορη) απόσβεσης.

Ακολούθως δίνουμε παραστατικά την αρχιτεκτονική της δικής μας υλοποίησης του Generator.

StyleGAN › Generator ▹ Δίκτυο Αντιστοίχισης Θορύβου

$$\vec{z} \longrightarrow [$$
$$\quad FC_{512\to512} \longrightarrow ReLU$$
$$] \times 4 \longrightarrow \vec{w}$$

StyleGAN › Generator ▹ Δίκτυο Σύνθεσης

$$CONST_{512\times4\times4} \longrightarrow AdaIN \longrightarrow CONV3_{512\to1024} \longrightarrow AdaIN \longrightarrow [$$
$$\quad UP_{2\times} \longrightarrow CONV3_{C_{in}\to C_{out}} \longrightarrow InjectNoise \longrightarrow AdaIN \longrightarrow$$
$$\quad CONV3_{C_{in}\to C_{out}} \longrightarrow InjectNoise \longrightarrow AdaIN$$
$$] \times 5 \longrightarrow PJ_{1024\to3}$$

όπου στις στρώσεις $CONV3_{C_{in}\to C_{out}}$ δεν σημειώνεται ξεκάθαρα ο αριθμός καναλιών εξόδου γιατί ακολουθεί έναν μη-κοινό τύπο, InjectNoise είναι η στρώση προσθήκης στοχαστικού θορύβου και AdaIN η στρώση Προσαρμοστικής Κανονικοποίησης Δείγματος, ενώ στην έξοδο δεν χρησιμοποιείται συνάρτηση tanh.

StyleGAN Discriminator

Σε αντίθεση με τους Discriminators των μοντέλων που προηγήθηκαν, το StyleGAN χρησιμοποιεί ένα δίκτυο παρόμοιας αρχιτεκτονικής με τον PatchGAN Discriminator όχι όμως πανομοιότυπης. Η κύρια διαφοροποίησή του έγκειται στο γεγονός ότι οι συγγραφείς του κατέληξαν πως είναι καλύτερο ο Discriminator να βγάζει μία τιμή εξόδου αντί για πίνακα τιμών (όπως δηλαδή στα αρχικά GANs). Έτσι, και στη δική μας υλοποίηση υιοθετούμε το παραπάνω συμπέρασμα και ο Discriminator που υλοποιήσαμε έχει ως εξής: είναι ίδιος με τον PatchGAN Discriminator, δίκτυο που ξέρουμε και εμπιστευόμαστε, αλλά προσθέτουμε μία πλήρως συνδεδεμένη στρώση ακολουθούμενη από σιγμοειδή συνάρτηση ενεργοποίησης στην έξοδο και **μειώνουμ τα κανάλια βάση σε 16**. Επίσης, **δοκιμάσαμε τη μη-χρήση κανονικοποίησης στον Discriminator στο τελευταίο πείραμά μας** για να μπορέσουμε να καταλάβουμε τη διαφορά. Ακολουθεί η αρχιτεκτονική του Discriminator με συμβολισμούς αντίστοιχους όπως προηγούμενα.

StyleGAN › Discriminator

$$\text{CONV3}_{3\longrightarrow 16} \longrightarrow [$$
$$\text{CONV3}_{C_{in}\rightarrow 2*C_{in}} \longrightarrow \text{LReLU}_{0.2} \longrightarrow \text{CONV3}_{2*C_{in}\rightarrow 2*C_{in}} \longrightarrow \text{LReLU}_{0.2} \longrightarrow \text{AVGPOOL2}$$
$$] \times 5 \longrightarrow \text{BatchStd} \longrightarrow \text{FC}_{4608\rightarrow 1}$$

όπου με AVGPOOL2 συμβολίζουμε στρώση pooling μέσου όρου και βήματος 2, BatchStd στρώση Τυπικής Απόκλισης Ομάδας, ενώ η χρήση της πλήρως-συνδεδεμένης στρώσης στην έξοδο οδηγεί στην παραγωγή μίας πιθανότητας στην έξοδο ανά εικόνα. Οι στρώσεις CONV εδώ είναι βήματος 1 αφού για τη μείωση της διάστασης υπάρχει η στρώση pooling.

Παράμετροι εκπαίδευσης και αποτελέσματα

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν δύο βελτιστοποιητές: ένας για τη βελτιστοποίηση των παραμέτρων του Generator και ένας για τη βελτιστοποίηση των παραμέτρων του Discriminator. Και στους δύο βελτιστοποιητές χρησιμοποιήθηκε ως αλγόριθμος βελτιστοποίησης ο Adam [31]. Το αρχικό βήμα εκμάθησης ορίσθηκε σε όλους τους βελτιστοποιητές ίσο με 0.0002 και οι παράμετροι απόσβεσης του αλγορίθμου, β_1 και β_2, ορίσθηκαν ίσες με 0.9 και 0.999 αντίστοιχα (σύμφωνα με το [31]). Ως συνάρτηση αντιπαραθετικού κόστους και στους δύο Discriminators χρησιμοποιήθηκε η **συνάρτηση κόστους Wasseerstein με Ποινή Παραγώγων για επιβολή της συνθήκης 1-Lipschitz** (βλ. 3.1) (με βάρος της ποινής παραγώγων στη συνάρτηση κόστους ίσο με 10). Χρησιμοποιήσαμε το 10% των εικόνων του αρχικού συνόλου εκπαίδευσης ως σύνολο ελέγχου (test set) και το υπόλοιπο 90% ως σύνολο εκπαίδευσης. Τέλος, δοκιμάστηκε και στο μοντέλο αυτό ο **προγραμματιστής του βήματος εκπαίδευσης** του CycleGAN: όταν η συνάρτηση κόστους επιπεδώσει ή αρχίζει να ανεβαίνει τότε ο προγραμματιστής κατεβάζει το βήμα στο ένα τοις χιλίοις (1‰) της πρότερής του τιμής και περιμένει για 200 βήματα μέχρι να ξανακάνει τον έλεγχο (ο έλεγχος διαρκεί 10 τουλάχιστον βήματα).

Εκπαιδεύσαμε την υλοποίησή μας του μοντέλου StyleGAN με τις παραπάνω παραμέτρους για 149 περάσματα του συνόλου δεδομένων (epochs) σε εικόνες από το σύνολο δεδομένων Fashion-Image Synthesis Benchmark (FISB) του (DeepFashion) που περιέχει εικόνες από ανθρώπους-μοντέλα που ποζάρουν σε στούντιο φωτογράφισης ρούχων. Οι εικόνες είναι ανάλυσης 128×128. Πριν την είσοδό τους στο μοντέλο, οι εικόνες μετατρέπονται σε τρισδιάστατους πίνακες και οι τιμές τους κανονικοποιούνται στο διάστημα $[-1.0, 1.0]$. Αποτελέσματα από την εφαρμογή της υλοποίησής μας του StyleGAN στο σύνολο δεδο-

μένων handbags2shoes δίνονται στη δεύτερη ενότητα του κεφαλαίου που ακολουθεί, ενότητα 6.4. Ο ενδιαφερόμενος αναγνώστης καλείται να ανατρέξει στην ενότητα αυτή, ενώ πριν ολοκληρώσουμε την ενότητα αναφέρουμε πως **η εκπαίδευση του μοντέλου δεν έχει ακόμα ολοκληρωθεί (δηλ. μένει να γίνουν και πρόσθετοι πειραματισμοί)**.

Κεφάλαιο 6

Παραγωγές Εικόνων Μόδας και Αξιολόγηση

Στο παρόν, αρκετά μικρότερο, κεφάλαιο θα προχωρήσουμε στην παράθεση των καμπυλών εκπαίδευσης και αποτελεσμάτων των μοντέλων που εκπαιδεύτηκαν. Για κάθε μοντέλο που εκπαιδεύτηκε στα πλαίσια της παρούσας εργασίας, λοιπόν, μετά από μια σύντομη περίληψη βασικών στοιχείων της δομής και των παραμέτρων εκπαίδευσής του, θα παραθέσουμε καμπύλες εξέλιξης των συναρτήσεων κόστους, ενδεικτικές παραγωγές και φυσικά μετρικές αξιολόγησης. Ακολούθως και για όπου αυτό είναι δυνατό, θα συγκρίνουμε τις παραγωγές και μετρικές μας με τις αντίστοιχες του σχετικού άρθρου.

Πριν προχωρήσουμε, θέλουμε να τονίσουμε στο σημείο αυτό κάτι που αφορά τις μετρικές. Η μετρική Structural Similarity Index (SSIM) (ή η παραλλαγή της M-CSSIM - βλ. υποενότητα 3.4), συγκεκριμένα έχει νόημα μόνο στις περιπτώσεις ύπαρξης ζευγών εικόνων για σύγκριση. Στο μοντέλο StyleGAN, για παράδειγμα, το οποίο παράγει εικόνες από θόρυβο χωρίς σαφή έξοδο, η αξιολόγηση με τη μετρική SSIM των παραγόμενων εικόνων δεν αποτελεί και τόσο αξιόπιστη μέτρηση της ποιότητάς του. Παρ΄ όλα αυτά υπολογίζεται σε κάθε μοντέλο και παρατίθεται με τα αντίστοιχα σχόλια σε κάθε περίπτωση. Κάτι άλλο που επίσης πρέπει να σημειωθεί, είναι ότι τα configurations των μοντέλων που δίνονται παρακάτω έχουν προκύψει μετά από πληθώρα δοκιμών, ενώ μεταβάλλουμε ορισμένες παραμέτρους ακόμη και κατά τη διάρκεια της εκπαίδευσης. Ο ενδιαφερόμενος αναγνώστης καλείται να συμβουλευτεί τα αντίστοιχα Jupyter notebooks που υπάρχουν στο αποθετήριο κώδικα της εργασίας (`github.com/achariso/gans-thesis`). Μια

ακόμη διευκρίνιση πριν προχωρήσουμε είναι ότι από το σύνολο μετρικών Precision-Recall-F_1 Score επιλέξαμε σε ορισμένες να δίνουμε την εξέλιξη κατά την εκπαίδευση μόνο για τη συνδυασμένη μετρική $F1_1$ Score για λόγους καλύτερης ανάγνωσης (μιας και που οι τρεις μετρικές ως επί το πλείστο των φορών έχουν πολύ κοντινές τιμές). Επιπρόσθετα, σε όσα κόστη δεν αναφέρεται βάρος αυτό νοείται ίσο με μονάδα.

6.1 Αλλαγή Πόζας (PGPG - *PoseGAN*)

Στην πρώτη ενότητα αυτού του κεφαλαίου θα αναφερθούμε στα αποτελέσματα από την εκπαίδευση του μοντέλου αλλαγή πόζας, *PoseGAN*, στο σύνολο δεδομένων In-shop Clothes Retrieval Benchmark (ICRB) του DeepFashion, καθώς και στην αξιολόγηση αυτών. Θα ξεκινήσουμε παραθέτοντας σε μορφή πίνακα μία σύνοψη του μοντέλου που αναπτύχθηκε, κάτι που κάνουμε για όλα τα μοντέλα που εκπαιδεύτηκαν και αναλύονται στο παρόν κεφάλαιο.

Πίνακας 6: Σύνοψη του μοντέλου *PoseGAN*

Όνομα Μοντέλου	*PoseGAN* $^{GT[1]}$
Κωδικός Configuration	`128_MSE_256_6_4_5_none_1e4_true_false_false`
Εφαρμογή	Αλλαγή πόζας σε ανθρώπους-μοντέλα
Κατηγορία Εφαρμογής	Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα
Κατηγορία Παραγωγής	Υπο-συνθήκη (εικόνα)
Σχετικό Μοντέλο	PGPG
Σχετικό ΄Αρθρο	«*Pose Guided Person Image Generation*» [81]
Αριθμός GANs	ένα (1)
Αριθμός Generators	ένας (1) - αποτελούμενος από δύο υποδίκτυα, G1 και G2
Αριθμός Discriminators	ένας (1)
Τύπος Generator(s)	G1: U-Net με skip connections (4 blocks ανά κατεύθυνση + πλήρως συνδεδεμένη στρώση στη στένωση) G2: U-Net με skip connections (6 blocks ανά κατεύθυνση + DropOut στον encoder)
Τύπος Discriminator(s)	PatchGAN Discriminator (5 blocks, 32×32 receptive field, Κανονικοποίηση Φάσματος)

Κατόπιν, συγκεντρώνουμε σε έναν άλλον πίνακα τις παραμέτρους εκπαίδευσης του μοντέλου *PoseGAN*, πρακτική που επίσης ακολουθούμε για όλα τα μοντέλα του παρόντος κεφαλαίου. Εδώ, είναι ο πίνακας 7 που δίνεται παρακάτω. Εκεί, όπου αναφέρεται L_1Loss εννοείται μετά την εφαρμογή της μάσκας πόζας, M_B, όπως αναφέρθηκε στην ανάλυση του προηγούμενου κεφαλαίου.

Πριν προχωρήσουμε στη παράθεση των καμπύλων εκπαίδευσης, θα δώσουμε στο σημείο αυτό μια ενδεικτική τριπλέτα κάποιας ομάδας με την οποία τροφοδοτεί ο φορτωτής δεδομένων (dataloader) το μοντέλο (ενν. τον Generator και Discriminator) σε κάθε επανάληψη (ή βήμα) του βρόγχου εκπαίδευσης (δίνουμε και μια, τυχαία, από το σύνολο δοκιμής). Ο Generator λαμβάνει την πρώτη εικόνα (συνθήκη) και την πόζα της δεύτερης εικόνας και καλείται να παράξει μια εικόνα που μοιάζει στη δεύτερη. Ο Discriminator λαμβάνει την πρώτη εικόνα συνενωμένη είτε με τη δεύτερη ή με την έξοδο του Generator και εκπαιδεύεται να διακρίνει τις πραγματικές από τις τεχνητές.

(α) Τριπλέτα από τον φορτωτή του συνόλου εκπαίδευσης στο index #1234.

(β) Τριπλέτα από τον φορτωτή του συνόλου δοκιμής στο index #1234.

Σχήμα 87: Ενδεικτικές εικόνες που δίνονται στο μοντέλο από τον φορτωτή δεδομένων.

[1]GT = «GANs Thesis»: συμβολισμός που χρησιμοποιούμε για να διακρίνουμε τα ονόματα των μοντέλων από τα αντίστοιχα ονόματα των σχετικών μοντέλων και άρθρων.

Πίνακας 7: Παράμετροι εκπαίδευσης του μοντέλου *PoseGAN*

Όνομα Μοντέλου	*PoseGAN* [GT]
Κωδικός Configuration	`128_MSE_256_4_6_5_none_1e4_true_false_false`
Συναρτ. Κόστους Generator	G1: L_1 Loss (ανακατασκευής) G2: L_1 Loss (ανακατασκευής, βάρος: $\lambda_{\text{recon}} = 1 \rightarrow 5 \rightarrow 10$) + Ελαχίστων Τετραγώνων (MSE) (αντιπαραθετική)
Συναρτ. Κόστους Discriminator(s)	Ελαχίστων Τετραγώνων (MSE)
Αριθμός βελτ/τών για Generator(s)	ένας (1) (από κοινού εκπαίδευση των υποδικτύων G1 και G2 του Generator)
Αριθμός βελτ/τών για Discriminator(s)	ένας (1)
Τύπος βελτ/τών για Generator(s)	Adam, με παραμέτρους (lr=0.0001, β_1=0.9, β_2=0.999)
Τύπος βελτ/τών για Discriminator(s)	Adam, με παραμέτρους (lr=0.0001, β_1=0.9, β_2=0.999)
Σύνολο δεδομένων εκπαίδευσης	ICRB (DeepFashion)
Μέγεθος συνόλου δεδομένων	46.4K εικόνες με πόζα 92.5K ζεύγη εικόνων αλλαγής πόζας
Ανάλυση εικόνων	128×128px
Μέγεθος ομάδας	48 εικόνες/batch
Αριθμός epochs	**93 epochs** (162.533 επαναλήψεις)
Χρόνος Εκπαίδευσης	περίπου έξι (6) μέρες σε 16GB GPUs
# **Παραμέτρων Generator(s)**	393.9M εκπαιδεύσιμες παράμετροι (G1: 276M, G2: 117.4M)
# **Παραμέτρων Discriminator(s)**	1.6M εκπαιδεύσιμες παράμετροι
# **Παραμέτρων**	**395.1M εκπαιδεύσιμες παράμετροι συνολικά**

Καμπύλες Εκπαίδευσης

Ακολούθως, παραθέτουμε τις καμπύλες εξέλιξης των τιμών των συναρτήσεων κόστους ή καμπύλες εκπαίδευσης όπως αλλιώς ονομάζονται. Όπως φαίνεται και στον πίνακα

παραμέτρων εκπαίδευσης παραπάνω, ο Generator εκπαιδεύεται προσπαθώντας να ελαχιστοποίηση μία συνάρτηση κόστους αποτελούμενη από τους όρους σφαλμάτων ανακατασκευής του G1 (για βελτιστοποίηση των παραμέτρων του) και τους όρους σφαλμάτων ανακατασκευής (με βάρος 10) και σφαλμάτων αντιπαράθεσης του G2. Για τα σφάλματα ανακατασκευής χρησιμοποιήθηκε η απόσταση Manhattan στον χώρο των εικονοστοιχείων εξόδου, ενώ για τα αντιπαραθετικά η συνάρτηση κόστους Ελαχίστων Τετραγώνων. Δεν μπορούμε, επομένως, από πριν να υπολογίσουμε το εύρος τιμών που θα λαμβάνουν οι συναρτήσεις κόστους των δικτύων, παρά μόνο ότι αυτό θα είναι μεγαλύτερο του μηδενός. Ωστόσο, αξίζει να αναφέρουμε ότι από γνήσιο πειραματισμό και περιέργεια «πειράζουμε» το βάρος συμπερίληψης του κόστους ανακατασκευής στο υποδίκτυο G2 του Generator ως εξής:

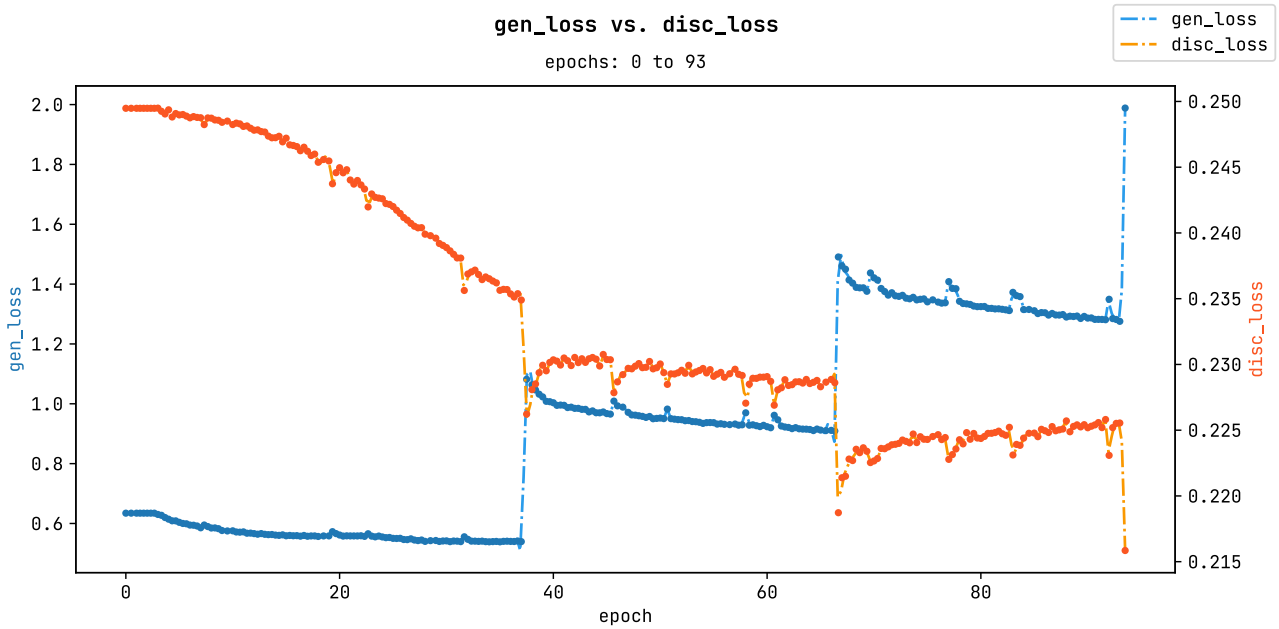
1. **αρχικά λ_{recon}=1**: αρχικά ο G2 δίνει την ίδια βαρύτητα στο σφάλμα ανακατασκευής σε σχέση με το αντιπαραθετικό (που επιστρέφει ο Discriminator)

2. **λ_{recon}=1→5 στο epoch 37**: στο 37ο epoch αλλάζουμε χειροκίνητα το βάρος από 1 σε 5, πιέζοντας έτσι το δίκτυο G2 του Generator να δώσει περισσότερη έμφαση στην εικόνα που του δείχνουμε μέσω της L1. Βλέποντας ότι αυτό βελτιώνει οπτικά τις παραγόμενες εικόνες δοκιμάζουμε εκ νέου αλλαγή του βάρους.

3. **λ_{recon}=1→5 στο epoch 66**: στο 66ο epoch αλλάζουμε και πάλι το βάρος πηγαίνοντάς το στο 10. Εν τέλει (φυσικά μετά από αρκετούς πειραματισμούς) διαπιστώσαμε ότι αν και χειροτερεύουν οι συναρτήσεις κόστους εντούτοις τα οπτικά αποτελέσματα και οι μετρικές αξιολόγησης γίνονται καλύτερες, δικαιώνοντας την επιλογή μας.

Ως συνάρτηση κόστους του Generator στο σχήμα 88 παρακάτω νοείται το άθροισμα των συναρτήσεων κόστους των δύο υποδικτύων αυτού.

Στο σημείο αυτό θέλουμε να σημειώσουμε ότι, όπως θα γίνει κατανοητό στη συνέχεια, τα δίκτυα αντιμάχονται το ένα το άλλο καθώς βελτιώνονται και άρα οι τιμές των συναρτήσεων κόστους δεν μας λένε πολλά σε σχέση με τις παραγωγικές δυνατότητες του μοντέλου. Στο σχήμα παραπάνω απεικονίζονται αυτές οι τιμές και όπως φαίνεται, εάν δει κανείς και τα διαγράμματα εξέλιξης των μετρικών, δεν σχετίζονται με τις τιμές των μετρικών αυτών. Θα μπορούσε για παράδειγμα ο Discriminator και ο Generator να κάνουν τυχαίους περιπάτους γύρω από ελαφρά συγκλίνουσες με το χρόνο τιμές, οι παραγόμενες εικόνες ωστόσο (άρα και οι μετρικές) να γίνονται αισθητά καλύτερες με τον χρόνο. Αυτό ισχύει για όλα τα μοντέλα GAN που εκπαιδεύτηκαν στα πλαίσια της παρούσας εργασίας γι΄ αυτό



Σχήμα 88: Καμπύλες εκπαίδευσης του *PoseGAN*. Φαίνεται η εξέλιξη των συναρτήσεων κόστους του Generator και Discriminator ως προς epochs της εκπαίδευσης και οι απότομες μεταβολές αυτών κατά την αλλαγή του λ_{recon}.

και δεν το επαναλαμβάνουμε στις αντίστοιχες υποενότητες των άλλων μοντέλων.

Ενδεικτικές Παραγωγές

Στη συνέχεια θα δώσουμε μερικές ενδεικτικές παραγωγές του μοντέλου μας, *PoseGAN* ως εξής (η σειρά των στηλών ακολουθεί αυτή του σχετικού άρθρου):

- στην πρώτη στήλη δίνεται η (πραγματική) εικόνα εισόδου ή συνθήκης, όπου απεικονίζεται ένας άνθρωπος σε μία συγκεκριμένη πόζα
- στη δεύτερη στήλη δίνεται η πόζα της εικόνας εξόδου (αυτή συνενωμένη με την εικόνα της πρώτης στήλης εισέρχεται στον Generator, ενώ επίσης χρησιμοποιείται για τον υπολογισμό της μάσκας)
- στην τρίτη στήλη δίνεται η πραγματική εικόνα εξόδου ή εικόνα-στόχος
- στην τέταρτη στήλη δίνεται η έξοδος του G1 (χονδροειδές αποτέλεσμα)
- στην πέμπτη στήλη δίνεται ο χάρτης διαφορών που παράγει ο G2
- στην έκτη και τελευταία στήλη δίνεται η τελική έξοδος του μοντέλου (βελτιωμένο

αποτέλεσμα)

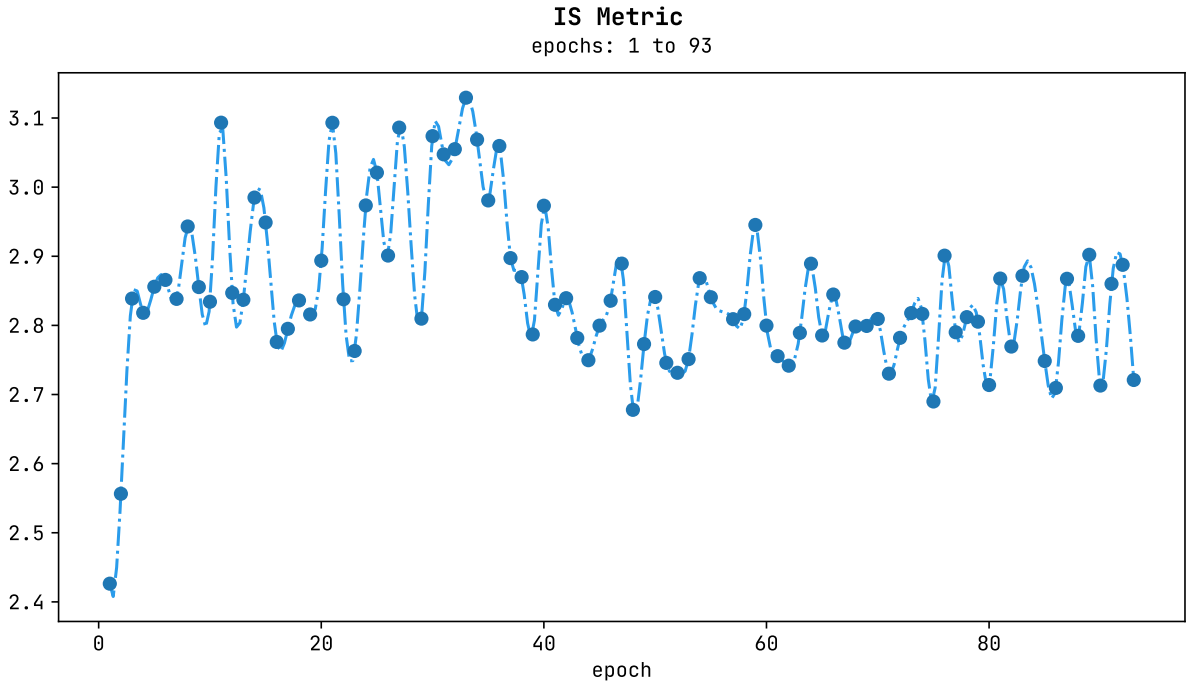
Οι παραγωγές δίνονται στο σχήμα 89 παρακάτω. Όπως φαίνεται εκεί το μοντέλο μας μετά από 93 epochs έχει καταφέρει να αιχμαλωτίσει σε μεγάλο βαθμό τη δομή του συνόλου δεδομένων, με αξιοσημείωτες τις λεπτομέρειες του προσώπου ή των σκιών του σώματος που φαίνονται στις παραχθείσες εικόνες. Επίσης, στην προ-τελευταία στήλη, φαίνεται η μικρή αλλά πολύ σημαντική δουλειά του υποδικτύου G2 στην αποθορυβοποίηση και αύξηση της λεπτομέρειας της χονδροειδούς εικόνας στην έξοδο του G1. Στην υποενότητα που ακολουθεί δίνουμε τις τιμές των μετρικών αξιολόγησης των παραγόμενων εικόνων και τις συγκρίνουμε το σχετικό άρθρο.

Σχήμα 89: Παραγωγές της υλοποίησής μας του μοντέλου PGPG, *PoseGAN*. Όλες οι εικόνες είναι ανάλυσης 128×128, ενώ έχουν συλλεγεί τυχαία από το τελευταίο epoch της εκπαίδευσης.

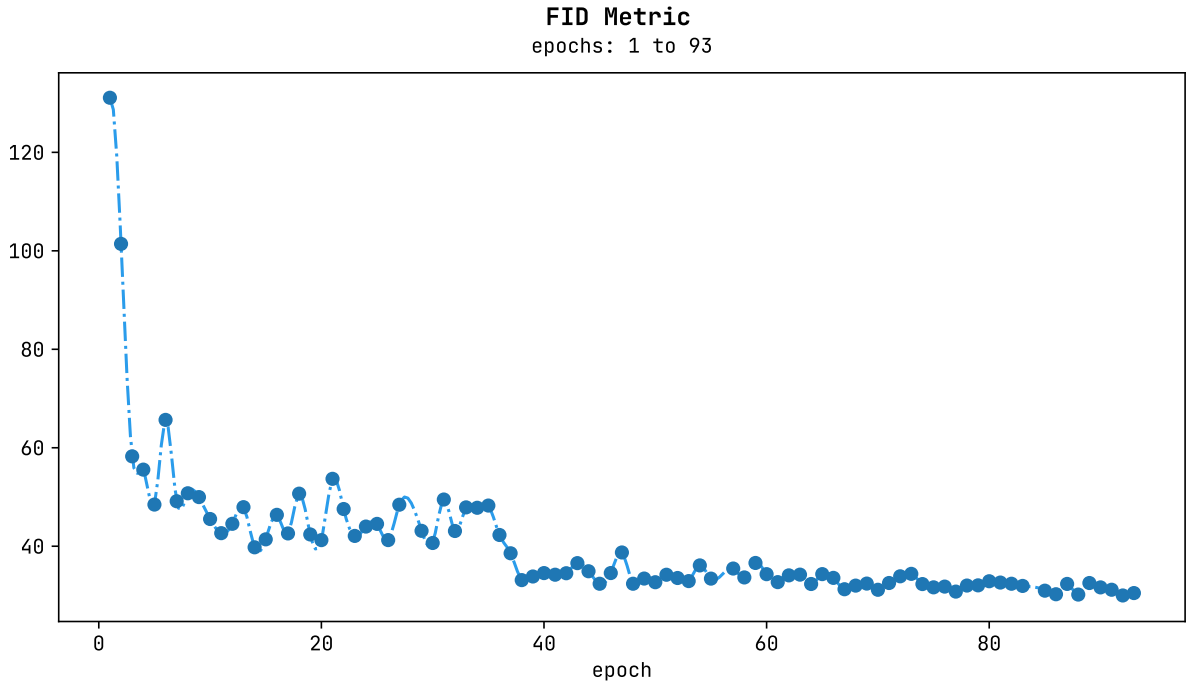
Αξιολόγηση των Παραγωγών

Όπως αναφέραμε στο τέλος του τρίτου κεφαλαίου για την αξιολόγηση των παραγόμενων εικόνων από GANs χρησιμοποιήσαμε τις ακόλουθες μετρικές: Inception Score (IS), Fréchet Inception Distance (FID), Precision-Recall-F_1 Score και Structural Similarity Index (SSIM). Εκεί, αναφέραμε τα πλεονεκτήματα και μειονεκτήματα της κάθε μίας, υιοθετώντας στο τέλος ότι οι μετρικές Precision-Recall-F_1 Score είναι οι πιο αξιόπιστες για μέτρηση της ποιότητας και ποικιλομορφίας των παραγωγών από GANs, κάτι ευρέως αποδεκτό στη σχετική βιβλιογραφία.

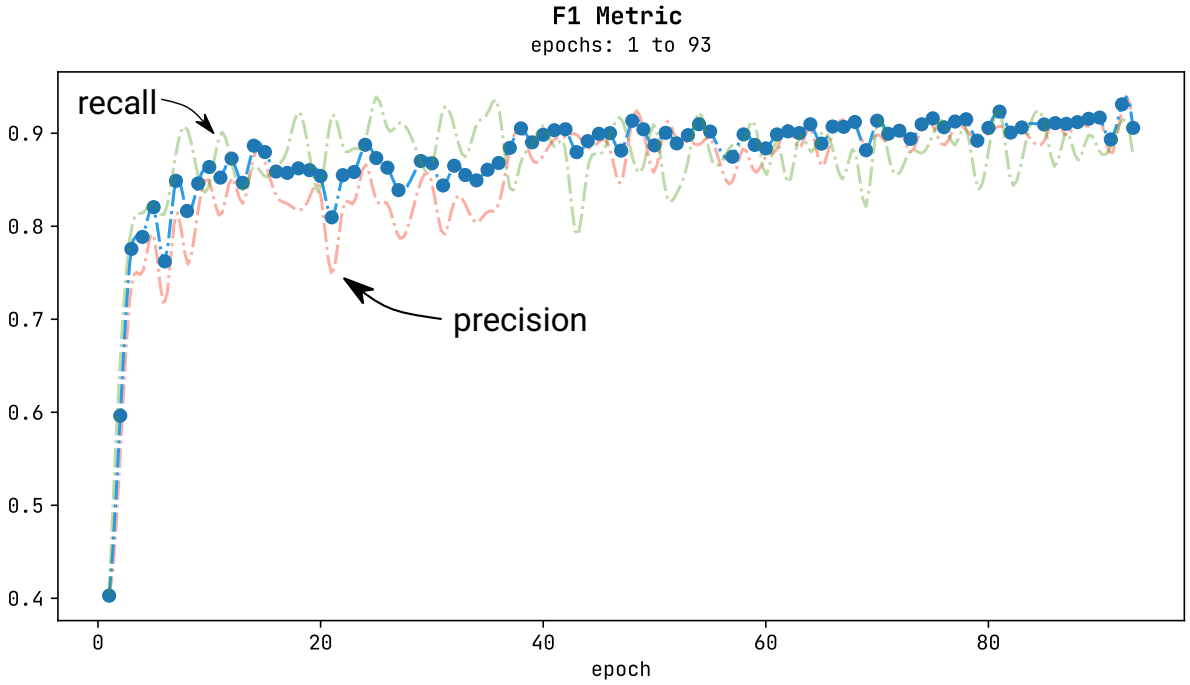
Ακολούθως, παραθέτουμε διαγράμματα εξέλιξης των μετρικών αυτών κατά τη διάρκεια εκπαίδευσης της υλοποίησής μας του μοντέλου PGPG, *PoseGAN*. Πριν την ανάγνωση των διαγραμμάτων, ο αναγνώστης θα πρέπει να είναι ενήμερος ότι για την καταγραφή των μετρικών χρησιμοποιήθηκαν μόλις 1000 τυχαίες εικόνες από το σύνολο δεδομένων δοκιμής και ισάριθμες παραγωγές του Generator. Αντίθετα, οι δημιουργοί των μετρικών (τουλάχιστον του FID και F_1 Score) προτείνουν να χρησιμοποιηθούν τουλάχιστον 10000 εικόνες για την αξιολόγηση. Ο λόγος είναι ότι θα ήταν χρονικά αδύνατο να σταματάμε την εκπαίδευση και να τρέχουμε τις μετρικές με 10000 εικόνες, διαδικασία που διαρκεί περίπου μία ώρα αναλόγως και τον Generator. Συμπερασματικά, ακολούθως δίνουμε τις καμπύλες εξέλιξης των μετρικών οι οποίες όμως πάρθηκαν με λιγότερα δείγματα, ενώ κατόπιν παραθέτουμε σε μορφή πίνακα τις τελικές μετρικές αξιολόγησης του μοντέλου (οι οποίες υπολογίσθηκαν από 10000 εικόνες).



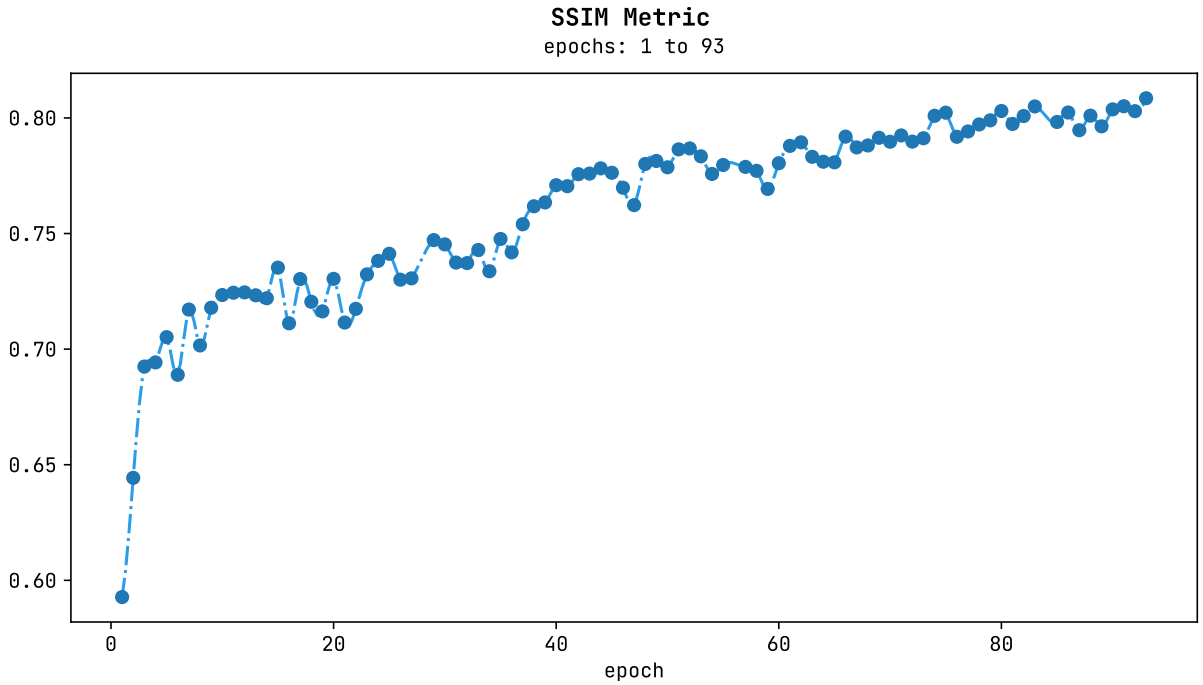
Σχήμα 90: Καμπύλη εξέλιξης της μετρικής Inception Score (IS) κατά τη διάρκεια εκπαίδευσης του μοντέλου *PoseGAN*.



Σχήμα 91: Καμπύλη εξέλιξης της μετρικής Fréchet Inception Distance (FID) κατά τη διάρκεια εκπαίδευσης του μοντέλου *PoseGAN*.



Σχήμα 92: Καμπύλη εξέλιξης της μετρικής F_1 Score κατά τη διάρκεια εκπαίδευσης του μοντέλου *PoseGAN*.



Σχήμα 93: Καμπύλη εξέλιξης της μετρικής Structural Similarity Index (SSIM) κατά τη διάρκεια εκπαίδευσης του μοντέλου *PoseGAN*.

Όπως φαίνεται στα παραπάνω σχήματα, όλες οι μετρικές φαίνεται να βελτιώνονται απότομα στην αρχή, ενώ από ένα σημείο και ύστερα (περίπου στο epoch=40) φαίνονται να συγκλίνουν προς μία τιμή. Αναλυτικότερα, για τη κάθε μετρική παραθέτουμε τα εξής σχόλια:

- **Inception Score**: από το πρώτο διάγραμμα φαίνεται πως το IS αρχικά αυξάνεται απότομα, ενώ από το 40ό epoch, που εικόνες (οπτικά) αρχίζουν να γίνονται όλο και πιο ρεαλιστικές, φαίνεται να μειώνεται και να παρουσιάζει **συμπεριφορά τυχαίου περίπατου γύρω από την τιμή 2.80**. Η τιμή αυτή είναι σχετικά «καλή» συγκριτικά με το αρχικό paper όπου δίνουν 3.01(το μεγαλύτερο το καλύτερο). Ωστόσο, συγκρίνοντας το διάγραμμα του IS με αυτά των υπόλοιπων μετρικών και ιδιαίτερα με αυτό της FID και F1 που θεωρούνται πιο αξιόπιστες και σταθερές, επιβεβαιώνουμε τα ευρήματα σε διάφορες δουλειές στη βιβλιογραφία σχετικά με την αστάθεια της μετρικής του Inception Score, ακόμη περισσότερο όταν το Inception μοντέλο έχει εκπαιδευθεί σε διαφορετικά δεδομένα. Σε κάθε περίπτωση, φαίνεται και εδώ μια θετική εξέλιξη κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **Fréchet Inception Distance (FID)**: από το σχήμα 91 παραπάνω φαίνεται ότι η μετρική FID συγκλίνει μονότονα προς μια τιμή κοντά στο 20, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή. Τιμές FID κάτω από 30 είναι γενικά σημάδι καλής σχεδίασης και επιτυχούς εκπαίδευσης, καθώς τα state-of-the-art μοντέλα εκθέτουν μετρικές FID στο εύρος 10-15 (Ιούλιος 2021).

- **F_1 Score**: γενικότερα η μετρική F_1 Score στα πλαίσια της αξιολόγησης GANs, έχει δειχθεί ότι είναι πιο σταθερή και αξιόπιστη από τις υπόλοιπες. Φαίνεται και εδώ, από το σχήμα 92 παραπάνω, ότι η μετρική συγκλίνει μονότονα προς μια τιμή κοντά στο 0.91, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή. Εντυπωσιακό στοιχείο αποτελεί ότι από το epoch 50 και μετά, το F1 Score, δεν φαίνεται να βελτιώνεται ουσιαστικά, κάτι που όμως δεν ανταποκρίνεται απόλυτα στην ανθρώπινη κρίση του εκπονητή κοιτάζοντας τις εικόνες και κάτι που μπορεί να οφείλεται και στο μικρό του δείγματος των εικόνων που χρησιμοποιούνται για τον υπολογισμό της μετρικής.

- **Structural Similarity Index (SSIM)**: φαίνεται και για αυτή τη μετρική αξιολόγησης, από το σχήμα 93 παραπάνω, ότι η μετρική συγκλίνει μονότονα προς μια τιμή κοντά στο 0.82, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα (και υπεροχή σε σύγκριση με το αρχικό) του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

Τελική αξιολόγηση και σύγκριση με το άρθρο

Για την τελική αξιολόγηση του μοντέλου χρησιμοποιήθηκαν οι παραπάνω μετρικές τόσο σε 10000 εικόνες από από το σύνολο δεδομένων δοκιμής, όσο και σε 10000 από το σύνολο δεδομένων εκπαίδευσης. Τα αποτελέσματα εμφανίζονται στον πίνακα που ακολουθεί καθώς και συγκρίσεις αυτών με τα αντίστοιχα του αρχικού paper (οι παύλες υποδηλώνουν πως οι συγγραφείς του άρθρου δεν είχαν παραθέσει τις αντίστοιχες μετρικές).

Πίνακας 8: Συγκρίσεις μετρικών του *PoseGAN* με του PGPG από το σχετικό άρθρο. Όπως φαίνεται σε όλες τις μετρικές το μοντέλο μας παρουσιάζει υπεροχή με εξαίρεση την οριακά χειρότερη μετρική Inception Score στο test set.

	PGPG από [81]	***PoseGAN*^{GT}**
FID	-	16.19 (train) - 26.50 (test)
IS	3.09	3.83 (train) - 2.96 (test)
SSIM	0.762	0.803 (train) - 0.769 (test)
Precision	-	0.886(train) - 0.835 (test)
Recall	-	0.880(train) - 0.864 (test)
F1	-	0.882(train) - 0.849 (test)

6.2 Εξαγωγή Ρούχου (PixelDTGAN)

Περνάμε στη συνέχεια στην επόμενη ενότητα αυτού του κεφαλαίου, όπου θα αναφερθούμε στα αποτελέσματα από την εκπαίδευση του μοντέλου PixelDTGAN στο σύνολο δεδομένων LookBook επαυξημένο με ζεύγη εικόνων από το In-shop Clothes Retrieval Benchmark (ICRB) του DeepFashion, με σκοπό την αυτοματοποιημένη εξαγωγή του ρούχου που φοράει ο εκάστοτε εικονιζόμενος άνθρωπος. Θα ξεκινήσουμε παραθέτοντας σε μορφή πίνακα μία σύνοψη του μοντέλου που αναπτύχθηκε, όπως και στο προηγούμενο

έγινε και θα γίνει και στα επόμενα μοντέλα.

Πίνακας 9: Σύνοψη του μοντέλου PixelDTGAN

Όνομα Μοντέλου	PixelDTGAN ^{GT}
Κωδικός Configuration	`pxldtgan_default`
Εφαρμογή	Εξαγωγή ρούχου του ανθρώπου στην εικόνα εισόδου
Κατηγορία Εφαρμογής	Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα
Κατηγορία Παραγωγής	Υπο-συνθήκη (εικόνα)
Σχετικό Μοντέλο	PixelDTGAN
Σχετικό ΄Αρθρο	«*Pixel-Level Domain Transfer*» [66]
Αριθμός GANs	ένα (1)
Αριθμός Generators	ένας (1)
Αριθμός Discriminators	δύο (2) - real/fake (D_R) και associated/unassociated (D_A)
Τύπος Generator(s)	U-Net χωρίς skip connections (5 blocks ανά κατεύθυνση, με διάνυσμα 100 στοιχείων στο σημείο στένωσης)
Τύπος Discriminator(s)	2 × PatchGAN Discriminator (4 blocks, 16×16 receptive field, Κανονικοποίηση Φάσματος)
# **Παραμέτρων Generator(s)**	37.7M εκπαιδεύσιμες παράμετροι
# **Παραμέτρων Discriminator(s)**	D_R: 6.2M εκπαιδεύσιμες παράμετροι D_A: 6.2M εκπαιδεύσιμες παράμετροι
# **Παραμέτρων**	**50.1M εκπαιδεύσιμες παράμετροι συνολικά**

Κατόπιν, συγκεντρώνουμε σε έναν άλλον πίνακα τις παραμέτρους εκπαίδευσης του μοντέλου PixelDTGAN, πρακτική που ακολουθήσαμε στο προηγούμενο και θα ακολουθήσουμε και στα επόμενα μοντέλα του παρόντος κεφαλαίου. Για το PixelDTGAN, είναι ο πίνακας 10 που δίνεται παρακάτω. Τονίζεται εκ νέου ότι στη δική μας υλοποίηση χρησιμοποιούμε και κόστος ανακατασκευής εκτός από το αντιπαραθετικό, καθώς παρατηρήσαμε ότι αυτό δουλεύει καλά στο προηγούμενο μοντέλο, επίσης συζευγμένης μετατροπής εικόνας-σε-εικόνα.

Πίνακας 10: Παράμετροι εκπαίδευσης του μοντέλου PixelDTGAN

Όνομα Μοντέλου	PixelDTGAN ^{GT}
Κωδικός Configuration	`pxldtgan_default`
Συναρτ. Κόστους Generator	L_1 Loss (ανακατασκευής, βάρος: $\lambda_{recon} = 4$ εκπαιδεύσιμο) + Ελαχίστων Τετραγώνων (MSE) (αντιπαραθετική)
Συναρτ. Κόστους Discriminator(s)	Ελαχίστων Τετραγώνων (MSE)
Αριθμός βελτ/τών για Generator(s)	ένας (1)
Αριθμός βελτ/τών για Discriminator(s)	δύο (2) (ξεχωριστή εκπαίδευση των D_R και D_A)
Τύπος βελτ/τών για Generator(s)	Adam, με παραμέτρους (lr=0.0001, β_1=0.9, β_2=0.999)
Τύπος βελτ/τών για Discriminator(s)	Adam, με παραμέτρους (lr=0.0001, β_1=0.9, β_2=0.999)
Σύνολο δεδομένων εκπαίδευσης	LookBook + ICRB (DeepFashion)
Μέγεθος συνόλου δεδομένων	81.2K εικόνες (ανθρώπων + ρούχων) 71.7K ζεύγη εικόνων εξαγωγής ρούχου
Ανάλυση εικόνων	64×64px
Μέγεθος ομάδας	256 εικόνες/batch
Αριθμός epochs	**348 epochs** (88.200 επαναλήψεις)
Χρόνος Εκπαίδευσης	περίπου δώδεκα (12) ημέρες σε 12GB & 16GB GPUs

Πριν προχωρήσουμε στη παράθεση των καμπύλων εκπαίδευσης, θα δώσουμε στο σημείο αυτό μια ενδεικτική δυάδα εικόνων από κάποια ομάδα (batch) με την οποία τροφοδοτεί ο φορτωτής δεδομένων (dataloader) το μοντέλο (ενν. τον Generator και τους Discriminators) σε κάθε επανάληψη (ή βήμα) του βρόγχου εκπαίδευσης. Ο Generator λαμβάνει την πρώτη εικόνα (συνθήκη) ενός ανθρώπου που φοράει ένα ρούχο και καλείται να παράξει μια εικόνα που έχει μόνο το ρούχο σε ουδέτερο παρασκήνιο, που να μοιάζει δηλαδή στη δεύτερη εικόνα. Ο associated/unassociated Discriminator λαμβάνει την πρώτη εικόνα συνενωμένη είτε με τη δεύτερη ή με την έξοδο του Generator και εκπαιδεύεται να διακρίνει εάν οι εικόνες είναι πραγματικές και συσχετισμένες μεταξύ τους. Αντίστοιχα,

ο real/fake Discriminator λαμβάνει είτε μόνη της την πραγματική ή μόνη της την τεχνητή και καλείται να τις διακρίνει.

(α) Δυάδα από τον φορτωτή του συνόλου εκπαίδευσης στο index #1234.

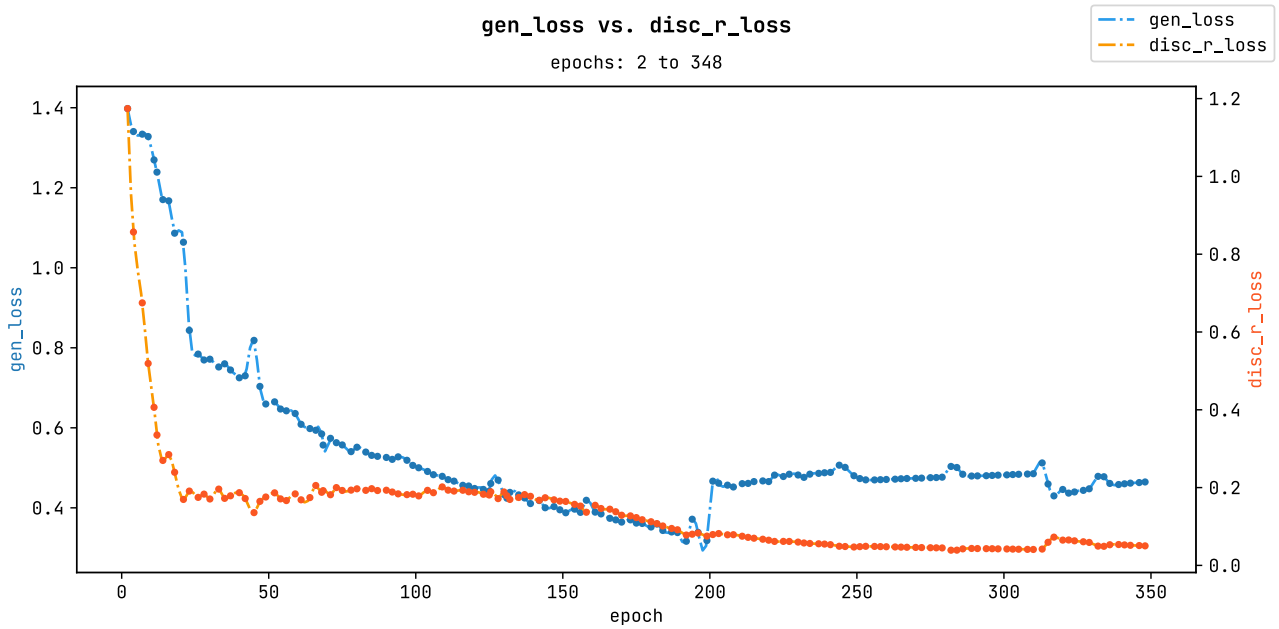
(β) Δυάδα από τον φορτωτή του συνόλου εκπαίδευσης στο index #21211.

Σχήμα 94: Ενδεικτικές εικόνες που δίνονται στο μοντέλο PixelDTGAN από τον φορτωτή δεδομένων. Οι εικόνες είναι ανάλυσης 64×64.

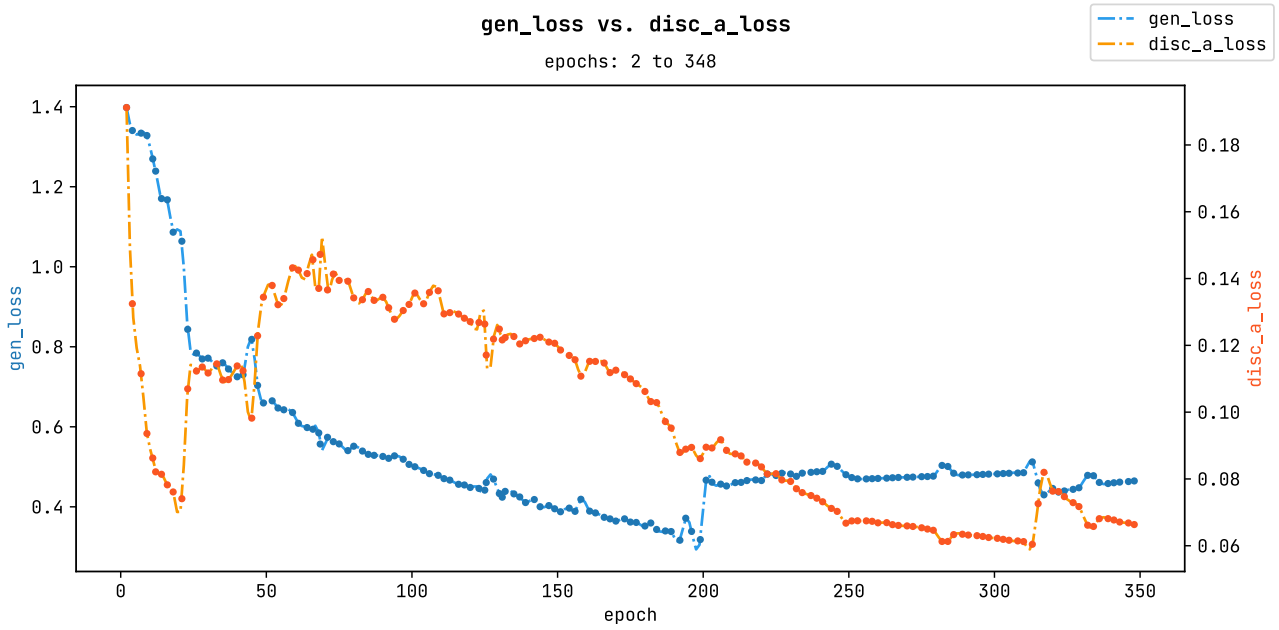
Καμπύλες Εκπαίδευσης

Ακολούθως, παραθέτουμε τις καμπύλες εξέλιξης των τιμών των συναρτήσεων κόστους ή καμπύλες εκπαίδευσης όπως αλλιώς ονομάζονται. Όπως φαίνεται και στον πίνακα παραμέτρων εκπαίδευσης παραπάνω, ο Generator εκπαιδεύεται προσπαθώντας να ελαχιστοποίηση μίας συνάρτηση κόστους αποτελούμενη από τους όρους σφαλμάτων ανακατασκευής (με αρχικό βάρος 4 - εκπαιδεύσιμο) και σφαλμάτων αντιπαράθεσης. Για τα σφάλματα ανακατασκευής χρησιμοποιήθηκε η απόσταση Manhattan στον χώρο των εικονοστοιχείων εξόδου, ενώ για τα αντιπαραθετικά η συνάρτηση κόστους Ελαχίστων Τετραγώνων. Δεν μπορούμε, επομένως, από πριν να υπολογίσουμε το εύρος τιμών που θα λαμβάνουν οι συναρτήσεις κόστους των δικτύων, παρά μόνο ότι αυτό θα είναι μεγαλύτερο του μηδενός.

Παρακάτω, παραθέτουμε τα διαγράμματα εξέλιξης των συναρτήσεων κόστους των δικτύων. Συγκεκριμένα, δίνουμε ένα διάγραμμα των συναρτήσεων κόστους των Discriminators (συγκριτικό) καθώς και αυτής του Generator συγκριτικά με του κάθε Discriminator (gen vs. disc_a & gen vs. disc_r). Επίσης, αναφέρουμε και εδώ τη χειροκίνητη αλλαγή του βάρους ανακατασκευής που έγινε κατά το epoch 186: καθώς το αφήσαμε εκπαιδεύσιμη παράμετρο χωρίς περιορισμούς, αυτό είχε μονότονα μειούμενη τάση, με αποτέλεσμα να γίνει αρνητικό. Διαπιστώθηκε το σφάλμα μας και το επαναφέραμε στο 0 και επιβάλλαμε ReLU πριν την εφαρμογή του, για το υπόλοιπο της εκπαίδευσης.

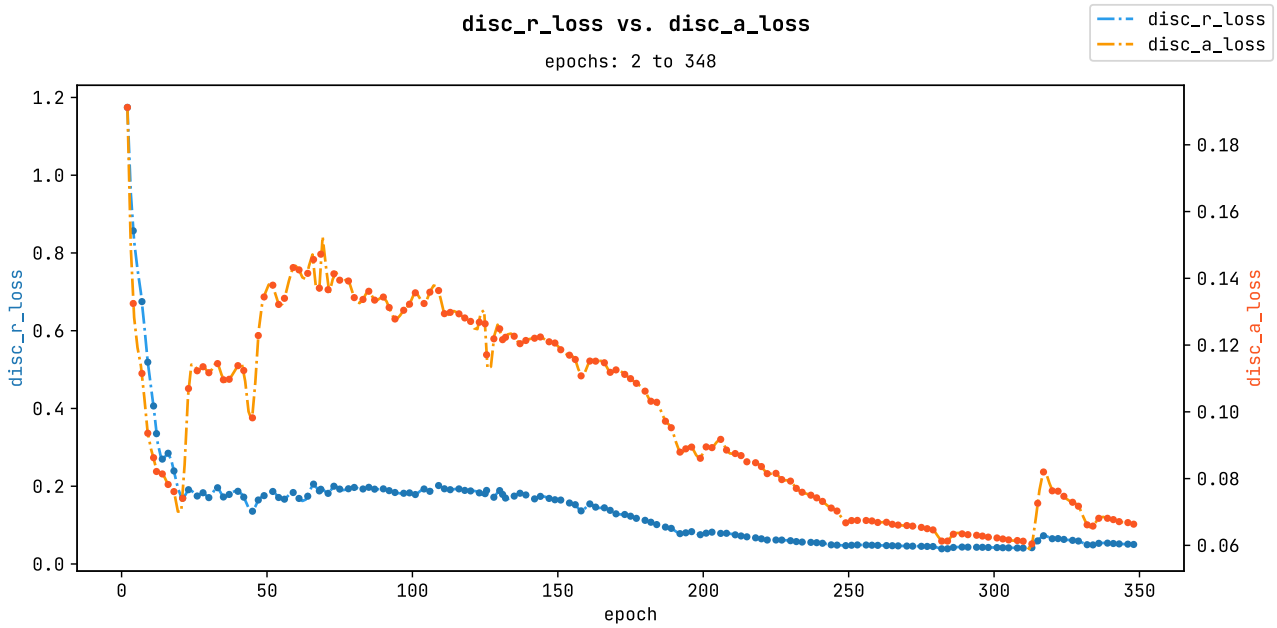


Σχήμα 95: Καμπύλες εκπαίδευσης του *PixelDTGAN*: Generator vs. real/fake Discriminator.



Σχήμα 96: Καμπύλες εκπαίδευσης του *PixelDTGAN*: Generator vs. associated/unassociated Discriminator.

Είναι εμφανές στα παραπάνω σχήματα το σημείο της χειροκίνητης αλλαγής του λ_{recon} στο epoch 186.



Σχήμα 97: Καμπύλες εκπαίδευσης του *PixelDTGAN*: real/fake Discriminator vs. associated/unassociated Discriminator.

Ενδεικτικές Παραγωγές

Στη συνέχεια θα δώσουμε μερικές ενδεικτικές παραγωγές του μοντέλου μας εξαγωγής ρούχου, PixelDTGAN ως εξής (η σειρά των στηλών ακολουθεί αυτή του σχετικού άρθρου):

- στην πρώτη στήλη δίνεται η (πραγματική) εικόνα εισόδου ή συνθήκης, όπου απεικονίζεται ένας άνθρωπος φορώντας το προς-εξαγωγή ρούχο
- στη δεύτερη στήλη δίνεται η πραγματική εικόνα εξόδου ή εικόνα-στόχος
- στην τρίτη και τελευταία στήλη δίνεται η τελική έξοδος του μοντέλου

Οι παραγωγές δίνονται στο σχήμα 98 παρακάτω. Όπως φαίνεται εκεί το μοντέλο μας μετά από 348 epochs έχει καταφέρει να αιχμαλωτίσει σε μεγάλο βαθμό τη δομή του συνόλου δεδομένων των ρούχων, με τις παραχθείσες εικόνες να δίνουν σωστά αποτελέσματα ως προς το χρώμα και το ύφασμα τουλάχιστον του ρούχου-στόχος. Αποτελεί πεποίθηση του εκπονητή της παρούσας εργασίας ότι η περαιτέρω αύξηση της χωρητικότητας του Generator θα μπορούσε να βελτιώσει τις παραγωγές και να αυξήσει τη λεπτομέρειά του (όπως π.χ. σχέδια ρούχων, τσαλακώματα κλπ.). Στην υποενότητα που ακολουθεί δίνουμε τις τιμές των μετρικών αξιολόγησης των παραγόμενων εικόνων και τις συγκρίνουμε το

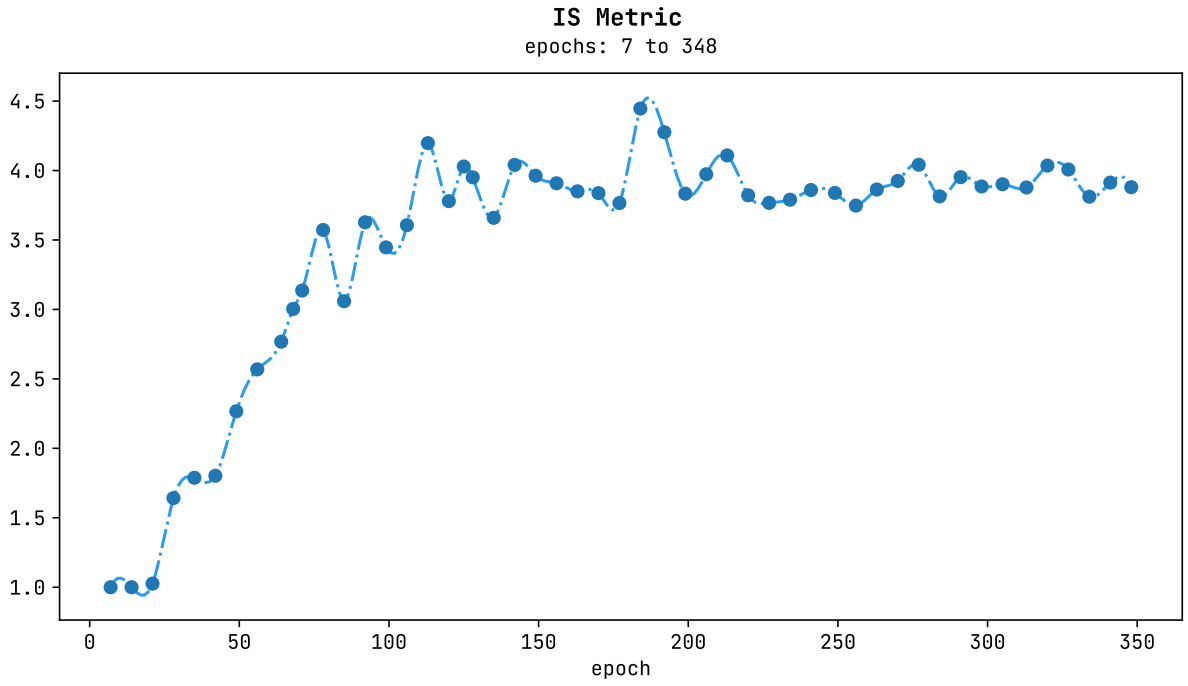
σχετικό άρθρο.

Σχήμα 98: Παραγωγές της υλοποίησής μας του μοντέλου PixelDTGAN. Όλες οι εικόνες είναι ανάλυσης 64×64, ενώ έχουν συλλεγεί τυχαία από το τελευταίο epoch της εκπαίδευσης.

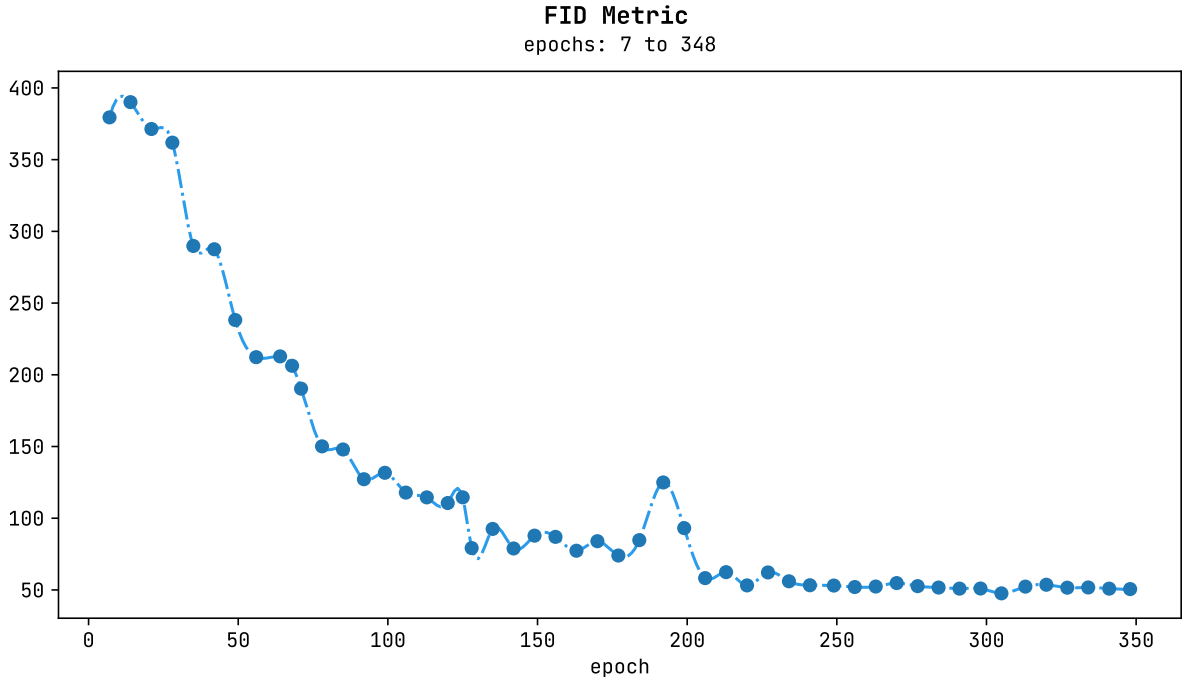
Αξιολόγηση των Παραγωγών

Ακολούθως, παραθέτουμε διαγράμματα εξέλιξης των μετρικών αξιολόγησης (όπως και πριν: IS, FID, F1 & SSIM) κατά τη διάρκεια εκπαίδευσης της υλοποίησής μας του μοντέλου PixelDTGAN. Πριν την ανάγνωση των διαγραμμάτων, ο αναγνώστης θα πρέπει να είναι ενήμερος ότι για την καταγραφή των μετρικών χρησιμοποιήθηκαν και εδώ μόλις 1000 τυχαίες εικόνες από το σύνολο δεδομένων δοκιμής και ισάριθμες παραγωγές του Generator. Για πιο αξιόπιστες μετρικές, ο αναγνώστης παραπέμπεται στην επόμενη υποενότητα, όπου παραθέτουμε σε μορφή πίνακα τις τελικές μετρικές αξιολόγησης του μοντέλου οι οποίες υπολογίσθηκαν από 10000 εικόνες.

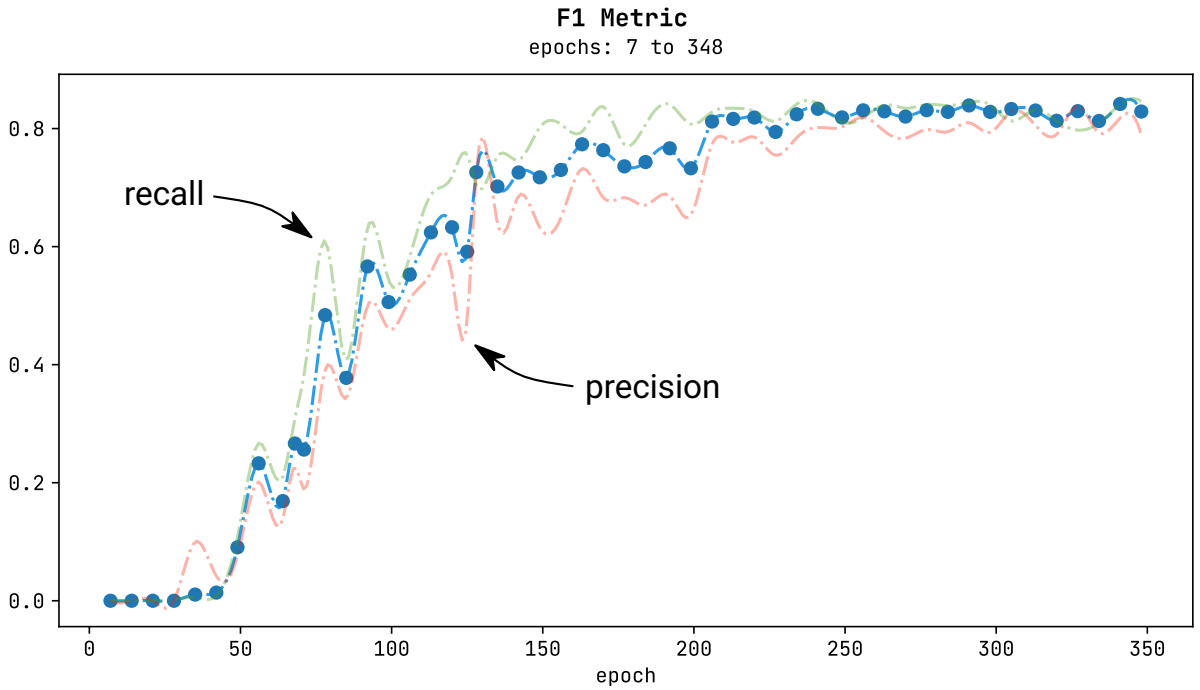
Όπως φαίνεται στα διαγράμματα αυτά, όλες οι μετρικές φαίνεται να βελτιώνονται και εδώ γρήγορα στην αρχή, ενώ από ένα σημείο και ύστερα (περίπου στο epoch=230) φαίνονται να συγκλίνουν προς μία τιμή. Πριν προχωρήσουμε σε περαιτέρω σχολιασμό των διαγραμμάτων, αναφέρουμε εδώ ότι η απότομη μεταβολή των μετρικών στο epoch 186 οφείλεται στο ότι **εκεί χειροκίνητα αλλάξαμε την τιμή του βάρους συμπερίληψης του όρου ανακατασκευής, λ_{recon}, από περίπου -1 που είχε φτάσει, σε 0** και το αφήσαμε να εκπαιδεύεται με ReLU. Ο λόγος είναι ότι θέλαμε να βοηθήσουμε κατ΄ αυτόν τον τρόπο τον



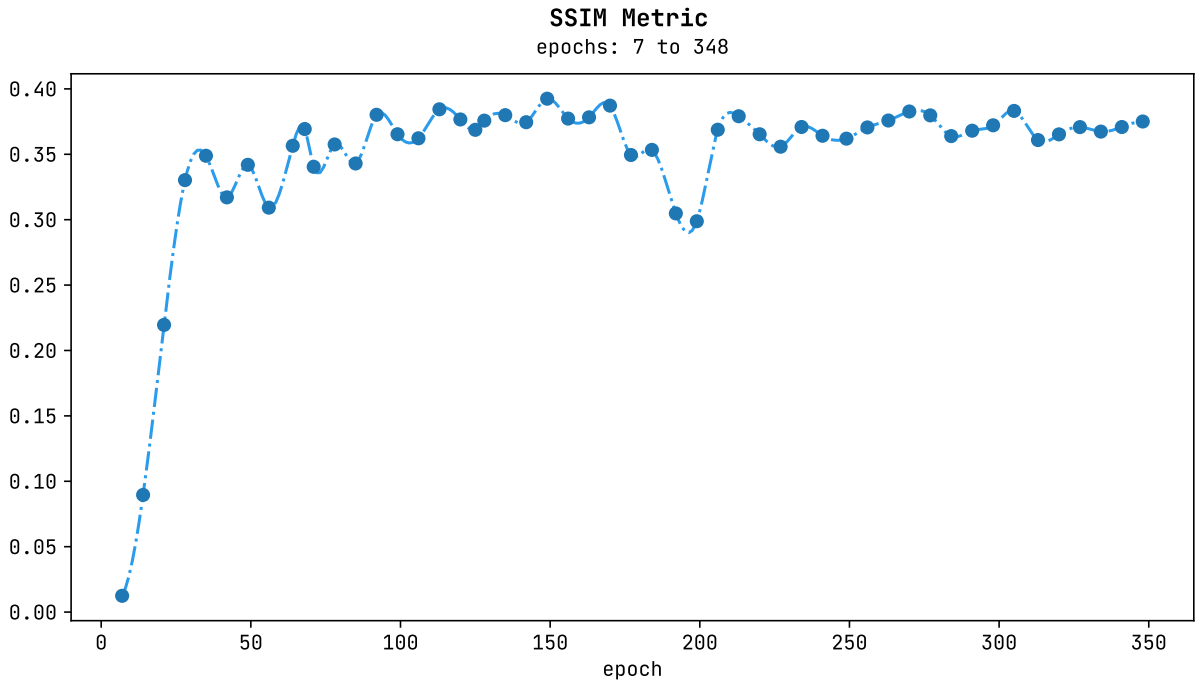
Σχήμα 99: Καμπύλη εξέλιξης της μετρικής Inception Score (IS) κατά τη διάρκεια εκπαίδευσης του μοντέλου PixelDTGAN.



Σχήμα 100: Καμπύλη εξέλιξης της μετρικής Fréchet Inception Distance (FID) κατά τη διάρκεια εκπαίδευσης του μοντέλου PixelDTGAN.



Σχήμα 101: Καμπύλη εξέλιξης της μετρικής F_1 Score κατά τη διάρκεια εκπαίδευσης του μοντέλου PixelDTGAN.



Σχήμα 102: Καμπύλη εξέλιξης της μετρικής Structural Similarity Index (SSIM) κατά τη διάρκεια εκπαίδευσης του μοντέλου PixelDTGAN.

Generator «δείχνοντάς» του όλο και περισσότερο τις εικόνες στόχους. Αναλυτικότερα, για τη κάθε μετρική παραθέτουμε τα εξής σχόλια:

- **Inception Score**: από το πρώτο διάγραμμα φαίνεται ότι η μετρική IS παρουσιάζει συμπεριφορά τυχαίου περίπατου γύρω από τη τιμή 4.0. Η τιμή αυτή είναι αρκετά «καλή» σε σχέση με αντίστοιχα μοντέλα στη βιβλιογραφία (δεδομένου ότι το ImageNET έχει λίγες τάξεις με εικόνες ανθρώπων και άρα το $N_{classes}$ που είχαμε αναφέρει ως μέγιστη τιμή σίγουρα είναι πολύ μικρότερο του 1000). Ωστόσο, συγκρίνοντας το παραπάνω διάγραμμα με αυτά των υπόλοιπων μετρικών και ιδιαίτερα με αυτό της FID και F1 που θεωρούνται πιο αξιόπιστες και σταθερές, επιβεβαιώνουμε τα ευρήματα από διάφορες δουλειές στη βιβλιογραφία σχετικά με την αστάθεια της μετρικής του Inception Score, ακόμη περισσότερο όταν το Inception μοντέλο έχει εκπαιδευθεί σε διαφορετικά δεδομένα. Σε κάθε περίπτωση, φαίνεται και εδώ μια θετική εξέλιξη κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **Fréchet Inception Distance (FID)**: από το σχήμα 100 παραπάνω φαίνεται ότι η μετρική συγκλίνει μονότονα προς μια τιμή κοντά στο 49, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **F_1 Score**: γενικότερα η μετρική F_1 Score στα πλαίσια της αξιολόγησης GANs, έχει δειχθεί ότι είναι πιο σταθερή και αξιόπιστη από τις υπόλοιπες. Φαίνεται και εδώ, από το σχήμα 101 παραπάνω, ότι η μετρική συγκλίνει μονότονα προς μια τιμή κοντά στο 0.82, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **Structural Similarity Index (SSIM)**: φαίνεται και για αυτή τη μετρική αξιολόγησης, από το σχήμα 102 παραπάνω, ότι η μετρική συγκλίνει προς μια τιμή κοντά στο 0.38, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα (και υπεροχή σε σύγκριση με το αρχικό) του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

Τελική αξιολόγηση και σύγκριση με το άρθρο

Για την τελική αξιολόγηση του μοντέλου χρησιμοποιήθηκαν οι παραπάνω μετρικές τόσο σε 10000 εικόνες από από το σύνολο δεδομένων δοκιμής, όσο και σε 10000 από το σύνολο δεδομένων εκπαίδευσης. Τα αποτελέσματα εμφανίζονται στον πίνακα που ακολουθεί καθώς και συγκρίσεις αυτών με τα αντίστοιχα του αρχικού paper (οι παύλες υποδηλώνουν πως οι συγγραφείς του άρθρου δεν είχαν παραθέσει τις αντίστοιχες μετρικές).

Πίνακας 11: Συγκρίσεις μετρικών του PixelDTGAN^{GT} με του PixelDTGAN από το σχετικό άρθρο. Όπως φαίνεται σε όλες τις μετρικές το μοντέλο μας παρουσιάζει υπεροχή αν και οι συγγραφείς επέλεξαν να μην αξιολογήσουν εξονυχιστικά το μοντέλο τους.

	PixelDTGAN από [66]	**PixelDTGAN^{GT}**
FID	-	36.04 (test)
IS	-	4.04 (test)
SSIM	0.21	0.381 (test)
Precision	-	0.781 (test)
Recall	-	0.814 (test)
F1	-	0.798 (test)

6.3 Ταίριασμα Στιλ (DiscoGAN - CycleGAN)

Το τρίτο μοντέλο που υλοποιήθηκε και εκπαιδεύτηκε στα πλαίσια της παρούσας εργασίας είναι μια παραλλαγή του CycleGAN το DiscoGAN. Για να τα ξεχωρίζουμε, **ονομάζουμε με CycleGAN τη δική μας υλοποίηση** έως το τέλος της παρούσας υποενότητας. Όπως αναφέρθηκε αυτό αποτελείται από δύο GANs, δηλαδή δύο Generators και δύο Discriminators. Το μοντέλο εκπαιδεύτηκε για Μη-Συζευγμένη Μετατροπή εικόνας-σε-εικόνα στο σύνολο handbags2shoes και άρα εκπαιδεύτηκε ώστε ο ένας Generator όταν δέχεται στην είσοδό του μία εικόνα τσάντας χεριού να δίνει στην έξοδό του μία ρεαλιστική εικόνα ενός παπουτσιού όμοιου στιλιστικά/εμφανισιακά και το αντίστροφο με τον άλλον Generator. Αντίστοιχα υπάρχουν δύο Discriminators, ένας για διάκριση πραγματικών/τεχνητών εικόνων για κάθε ένα από τα δύο πεδία εικόνων του handbags2shoes. Στόχος επομένως με την εκπαίδευση αυτού του μοντέλου είναι το ταίριασμα στιλ μεταξύ εικόνων παπουτσιών και εικόνων τσαντών, ενώ για **την επίτευξη αυτού του ψευδο-αντιστρέψιμου μετασχη-**

ματισμού εκπαιδεύουμε από κοινού τα δίκτυα των Generators και από κοινού αυτά των Discriminators. Θα ξεκινήσουμε και αυτήν την υποενότητα παραθέτοντας σε μορφή πίνακα μία σύνοψη του μοντέλου που αναπτύχθηκε, όπως έγινε και στα προηγούμενα μοντέλα.

Πίνακας 12: Σύνοψη του μοντέλου CycleGAN

Όνομα Μοντέλου	CycleGAN ^{GT}
Κωδικός Configuration	`discogan`
Εφαρμογή	Ταίριασμα στιλ μεταξύ παπουτσιών και τσαντών (μετατροπή από το ένα πεδίο στο άλλο)
Κατηγορία Εφαρμογής	Μη-Συζευγμένη Μετατροπή Εικόνας-σε-Εικόνα
Κατηγορία Παραγωγής	Υπο-συνθήκη (εικόνα)
Σχετικό Μοντέλο	DiscoGAN
Σχετικό ΄Αρθρο	«*Learning to Discover Cross-Domain Relations with Generative Adversarial Networks*» [76]
Αριθμός GANs	δύο (2)
Αριθμός Generators	δύο (2) - G_{AB} (παπούτσια ⟶ τσάντες) και G_{BA} (τσάντες ⟶ παπούτσια)
Αριθμός Discriminators	δύο (2) - D_B (τσάντες) και D_A (παπούτσια)
Τύπος Generator(s)	2 × δίκτυο encoder-transformer-decoder (2 blocks στον encoder & decoder, και 9 blocks στον transformer)
Τύπος Discriminator(s)	2 × PatchGAN Discriminator (4 blocks, 16×16 receptive field, Κανονικοποίηση Φάσματος)
# **Παραμέτρων Generator(s)**	G_{AB}: 11.4M εκπαιδεύσιμες παράμετροι G_{BA}: 11.4M εκπαιδεύσιμες παράμετροι
# **Παραμέτρων Discriminator(s)**	D_B: 1.6M εκπαιδεύσιμες παράμετροι D_A: 1.6M εκπαιδεύσιμες παράμετροι
# **Παραμέτρων**	**25.9M εκπαιδεύσιμες παράμετροι συνολικά**

Κατόπιν, συγκεντρώνουμε σε έναν άλλον πίνακα τις παραμέτρους εκπαίδευσης του μοντέλου CycleGAN, πρακτική που ακολουθήσαμε στα προηγούμενα και θα ακολουθήσουμε και στο επόμενο μοντέλο του παρόντος κεφαλαίου. Για το CycleGAN, είναι ο πίνακας 13 που δίνεται παρακάτω. Επαναλαμβάνουμε στο σημείο αυτό, πως το κόστος για την

από κοινού εκπαίδευση των Generators αποτελείται από τα κόστη κυκλικής συνοχής και ταυτοτικά (που χρησιμοποιούν συναρτήσεις σφάλματος ανακατασκευής, εδώ L1) και από τα αντιπαραθετικά κόστη τα οποία εδώ υπολογίζονται με τη συνάρτηση κόστους Ελαχίστων Τετραγώνων.

Πίνακας 13: Παράμετροι εκπαίδευσης του μοντέλου CycleGAN

Όνομα Μοντέλου	CycleGAN ^{GT}
Κωδικός Configuration	`discogan`
Συναρτ. Κόστους Generator	2 × Κόστος Κυκλικής Συνοχής (L_1 Loss, με βάρος $\lambda_{cycle} = 10$) + 2 × Ταυτοτικό Κόστος (L_1 Loss, με βάρος $\lambda_{identity} = 5$) + 2 × Αντιπαραθετικό Κόστος Ελαχίστων Τετραγώνων (MSE)
Συναρτ. Κόστους Discriminator(s)	2 × Ελαχίστων Τετραγώνων (MSE)
Αριθμός βελτ/τών για Generator(s)	ένας (1)
Αριθμός βελτ/τών για Discriminator(s)	ένας (1) (από κοινού εκπαίδευση των D_A και D_B)
Τύπος βελτ/τών για Generator(s)	Adam, με παραμέτρους (lr=0.0002, β_1=0.9, β_2=0.999) LR Scheduler: OnPlateau(factor: 0.99, cooldown: 200) ανά βήμα (όχι epoch)
Τύπος βελτ/τών για Discriminator(s)	Adam, με παραμέτρους (lr=0.0002, β_1=0.9, β_2=0.999) LR Scheduler: OnPlateau(factor: 0.99, cooldown: 200) ανά βήμα (όχι epoch)
Σύνολο δεδομένων εκπαίδευσης	handbags2shoes
Μέγεθος συνόλου δεδομένων	138.8K εικόνες τσαντών + 50K εικόνες παπουτσιών 71.7K ζεύγη εικόνων εξαγωγής ρούχου
Ανάλυση εικόνων	64×64px
Μέγεθος ομάδας	32 εικόνες/batch
Αριθμός epochs	**189 epochs** (142.200 επαναλήψεις)
Χρόνος Εκπαίδευσης	περίπου τρείς (3) ημέρες σε 24GB GPU

Πριν προχωρήσουμε στη παράθεση των καμπύλων εκπαίδευσης, θα δώσουμε στο σημείο αυτό μια ενδεικτική δυάδα εικόνων από κάποια ομάδα (batch) με την οποία τροφοδοτεί

ο φορτωτής δεδομένων (dataloader) το μοντέλο (ενν. τους Generators και τους Discriminators) σε κάθε επανάληψη (ή βήμα) του βρόγχου εκπαίδευσης. Ο G_{AB} λαμβάνει την εικόνα του (αληθινού ή τεχνητού από τον άλλο) παπουτσιού (συνθήκη) και καλείται να παράξει μια ρεαλιστική εικόνα τσάντας που να ταιριάζει με το παπούτσι εισόδου, ενώ ο G_{BA} το αντίστροφο. Ο D_B λαμβάνει την εικόνα της (πραγματικής ή τεχνητής) τσάντας ενώ ο D_A εκπαιδεύονται να διακρίνουν εάν οι εικόνες είναι πραγματικές ή τεχνητές. Σημειώνουμε επίσης πως παρόλο που έχουμε υπο-συνθήκη παραγωγή οι Discriminators τροφοδοτούνται μόνο με μία εικόνα, ώστε να διατηρήσουν την ιδιότητά τους ως domain Discriminators.

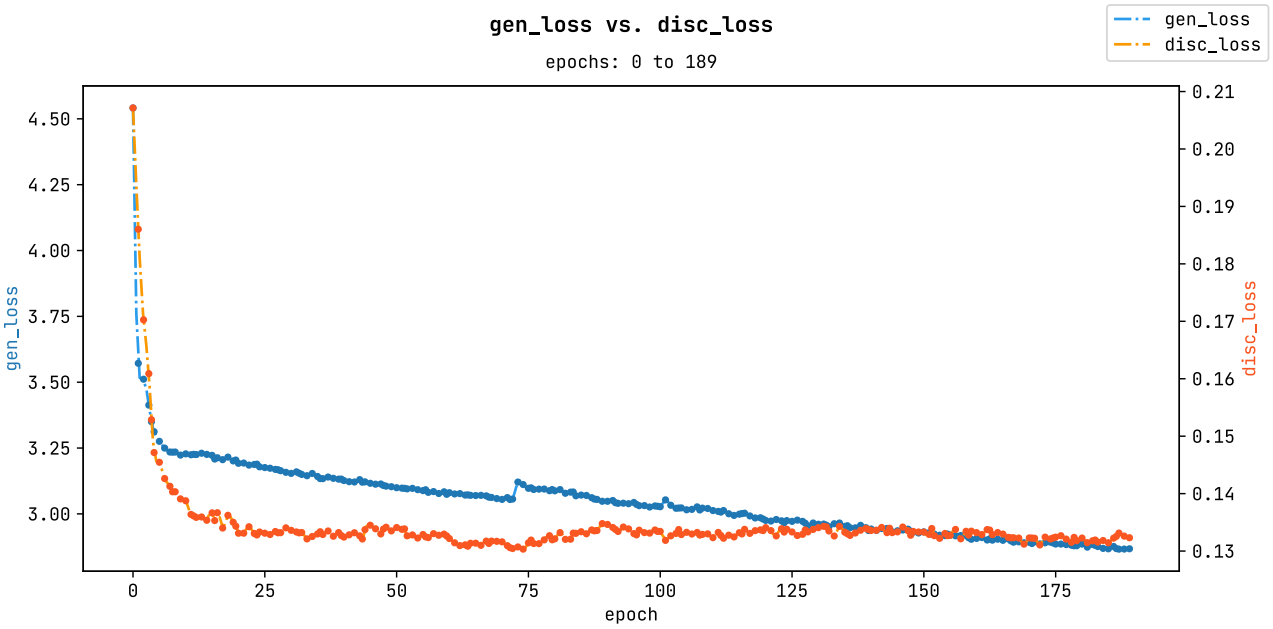
(α) Δυάδα από τον φορτωτή του συνόλου εκπαίδευσης στο index #1.

(β) Δυάδα από τον φορτωτή του συνόλου εκπαίδευσης στο index #2.

Σχήμα 103: Ενδεικτικές εικόνες που δίνονται στο μοντέλο CycleGAN από τον φορτωτή δεδομένων. Οι εικόνες αριστερά προέρχονται από το handbags_64.hdf5, οι δεξιά από το shoes_64.hdf5, ενώ όλες είναι ανάλυσης 64×64.

Καμπύλες Εκπαίδευσης

Ακολούθως, παραθέτουμε και εδώ για λόγους πληρότητας τις καμπύλες εξέλιξης των συναρτήσεων κόστους των βελτιστοποιητών ή καμπύλες εκπαίδευσης. Παρακάτω, παραθέτουμε τα διαγράμματα εξέλιξης των συναρτήσεων κόστους των δικτύων. Συγκεκριμένα, δίνουμε, στο ίδιο διάγραμμα, τη συνάρτηση κόστους των Discriminators συγκριτικά με αυτήν των Generators.



Σχήμα 104: Καμπύλες εκπαίδευσης του *CycleGAN*: Generator vs. real/fake Discriminator.

Είναι εμφανές από το σχήμα 104 ότι κάποια χειροκίνητη αλλαγή, πιθανότατα του βήματος εκμάθησης, συνέβη γύρω στο 73ο epoch, για την οποία δυστυχώς δεν έχει κρατηθεί καμία πληροφορία.

Ενδεικτικές Παραγωγές

Στη συνέχεια θα δώσουμε μερικές ενδεικτικές παραγωγές του μοντέλου μας εξαγωγής ρούχου, CycleGAN ως εξής (η σειρά των στηλών ακολουθεί αυτή του σχετικού άρθρου):

- στην πρώτη γραμμή δίνονται οι είσοδοι στον εκάστοτε Generator: πρώτα δίνουμε παραγωγές του G_{BA} (τσάντες ⟶ παπούτσια) και ακολούθως του G_{AB} (παπούτσια ⟶ τσάντες)
- στη δεύτερη γραμμή δίνονται οι έξοδοι του μοντέλου
- κάθε στήλη είναι και μία ξεχωριστή παραγωγή

Οι παραγωγές δίνονται στα σχήματα 105 και 106 παρακάτω. Όπως φαίνεται εκεί το μοντέλο μας μετά από 189 epochs έχει καταφέρει να αιχμαλωτίσει σε μεγάλο βαθμό τη δομή αμφοτέρων των συνόλων δεδομένων των ρούχων και έτσι μπορεί επιτυχώς να

μεταφέρει μια εικόνα από το ένα πεδίο στο άλλο. Αποτελεί πεποίθηση του εκπονητή της παρούσας εργασίας ότι η περαιτέρω αύξηση της χωρητικότητας των Generators και κυρίως **η συνέχιση της εκπαίδευσης** θα μπορούσε να βελτιώσει τις παραγωγές και να αυξήσει τη λεπτομέρειά τους (όπως π.χ. υφή, σχέδια κλπ.). Στην υποενότητα που ακολουθεί δίνουμε τις τιμές των μετρικών αξιολόγησης των παραγόμενων εικόνων.

Σχήμα 105: Παραγωγές της υλοποίησής μας του μοντέλου CycleGAN από τον Generator G_{AB} (παπούτσια ⟶ τσάντες). Όλες οι εικόνες είναι ανάλυσης 64×64, ενώ έχουν συλλεγεί τυχαία από το τελευταίο epoch της εκπαίδευσης.

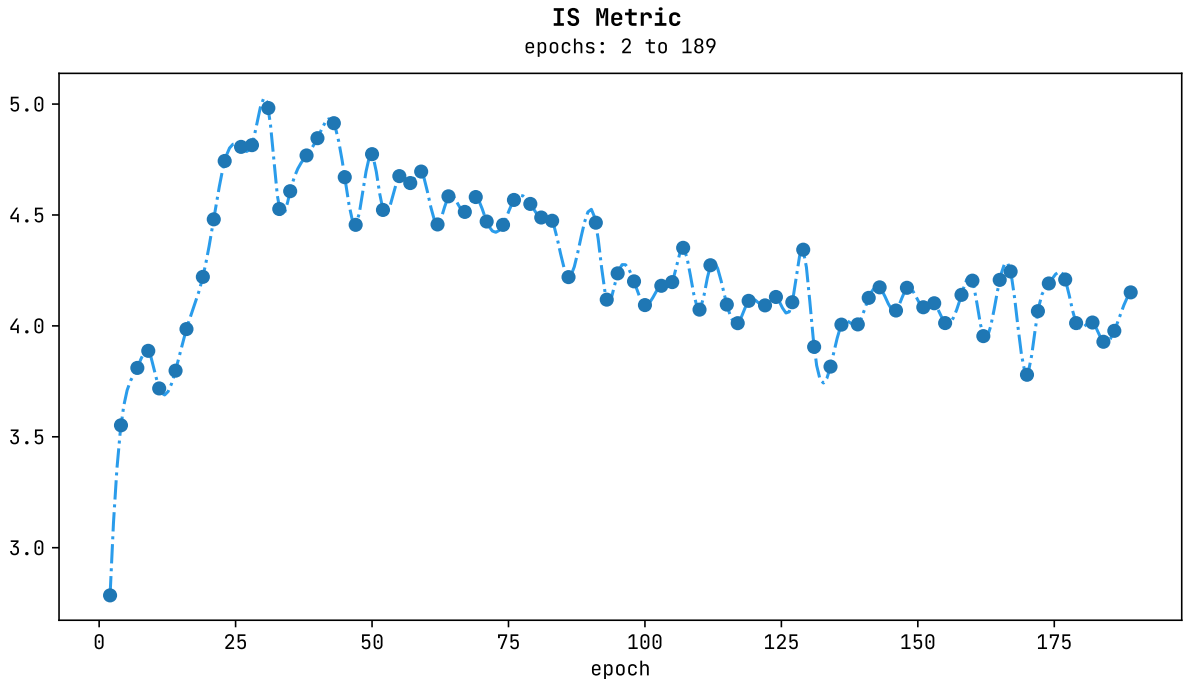
Αξιολόγηση των Παραγωγών

Ακολούθως, παραθέτουμε διαγράμματα εξέλιξης των μετρικών αξιολόγησης (όπως και πριν: IS, FID, F1 & SSIM) κατά τη διάρκεια εκπαίδευσης της υλοποίησής μας του μοντέλου CycleGAN. Πριν την ανάγνωση των διαγραμμάτων, ο αναγνώστης θα πρέπει να είναι ενήμερος ότι για την καταγραφή των μετρικών χρησιμοποιήθηκαν και εδώ μόλις 1000 τυχαίες εικόνες από το σύνολο δεδομένων δοκιμής και ισάριθμες παραγωγές του Generator. Για πιο αξιόπιστες μετρικές, ο αναγνώστης παραπέμπεται στην επόμενη υποενότητα, όπου παραθέτουμε σε μορφή πίνακα τις τελικές μετρικές αξιολόγησης του μοντέλου οι οποίες υπολογίσθηκαν από 10000 εικόνες.

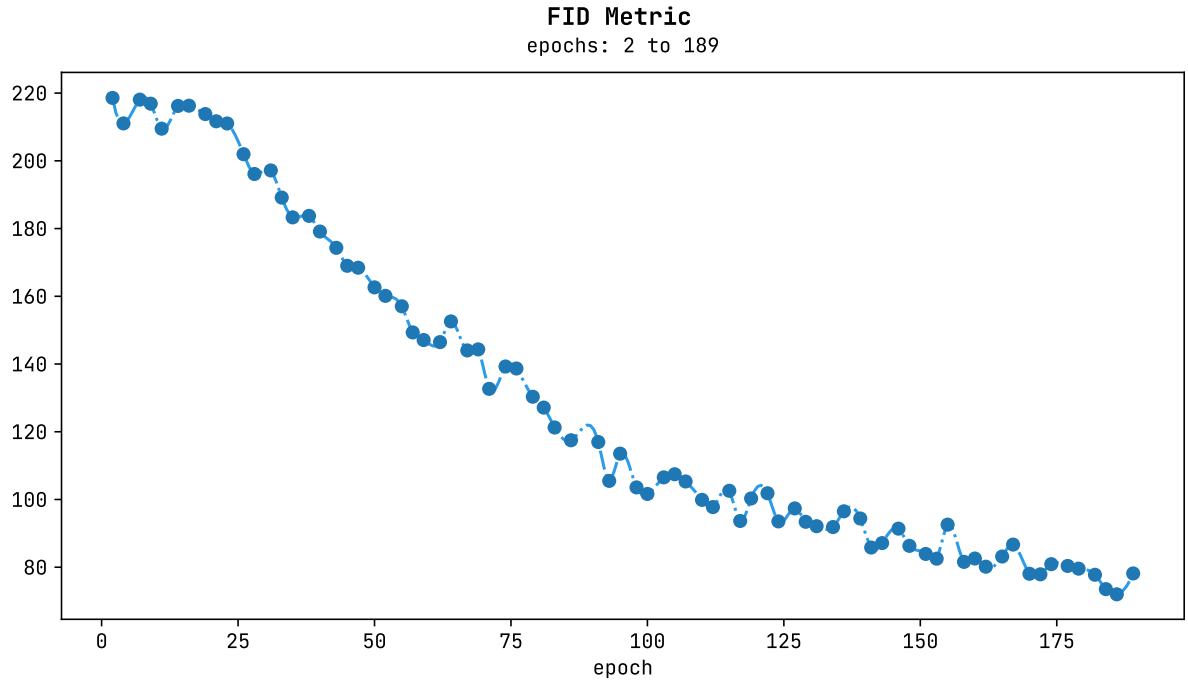
Σχήμα 106: Παραγωγές της υλοποίησής μας του μοντέλου CycleGAN από τον Generator G_{BA} (τσάντες ⟶ παπούτσια). Όλες οι εικόνες είναι ανάλυσης 64×64, ενώ έχουν συλλεγεί τυχαία από το τελευταίο epoch της εκπαίδευσης.

Όπως φαίνεται στα διαγράμματα αυτά, όλες οι μετρικές πλην του Inception Score φαίνεται να βελτιώνονται με τον χρόνο. Και εδώ η βελτίωση γίνεται γρήγορα στην αρχή, ενώ από ένα σημείο και ύστερα (περίπου στο epoch=150 και μετά) φαίνονται να συγκλίνουν προς μία τιμή. Αναλυτικότερα, για τη κάθε μετρική παραθέτουμε τα εξής σχόλια:

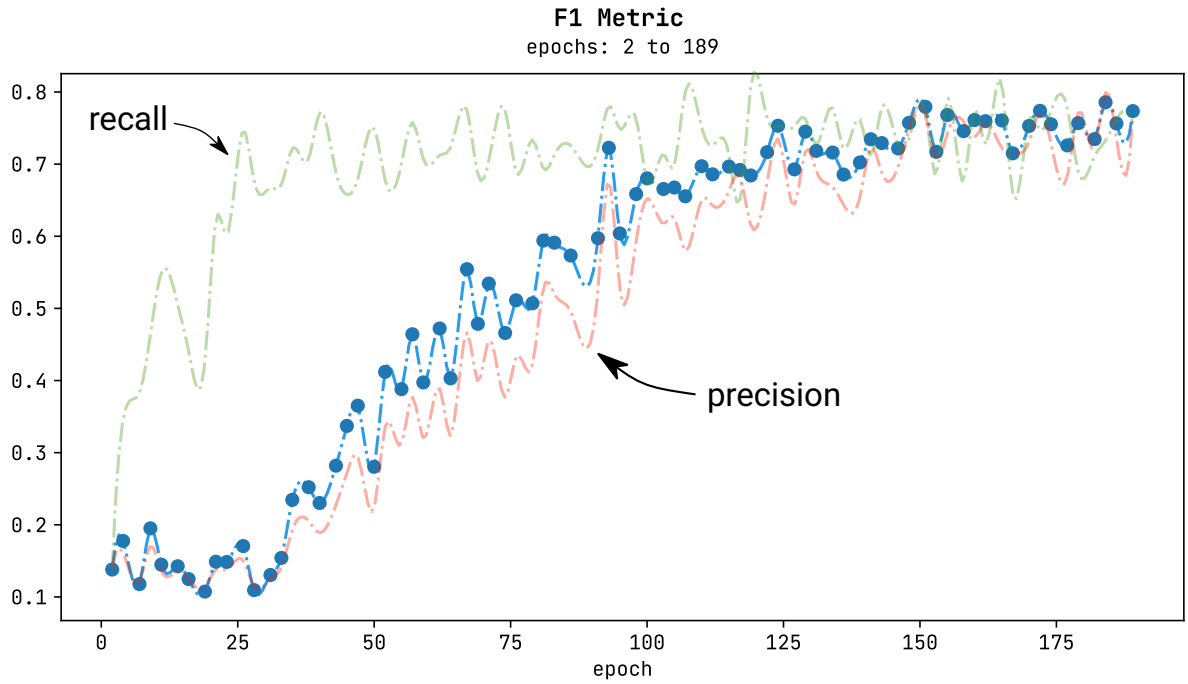
- **Inception Score**: από το πρώτο διάγραμμα φαίνεται ότι η μετρική IS παρουσιάζει συμπεριφορά τυχαίου περίπατου γύρω από τη τιμή 4.0. Η τιμή αυτή είναι αρκετά «καλή» σε σχέση με αντίστοιχα μοντέλα στη βιβλιογραφία (δεδομένου ότι το ImageNET έχει λίγες τάξεις με εικόνες ανθρώπων και άρα το $N_{classes}$ που είχαμε αναφέρει ως μέγιστη τιμή σίγουρα είναι πολύ μικρότερο του 1000). Ωστόσο, συγκρίνοντας το παραπάνω διάγραμμα με αυτά των υπόλοιπων μετρικών και ιδιαίτερα με αυτό της FID και F1 που θεωρούνται πιο αξιόπιστες και σταθερές, επιβεβαιώνουμε τα ευρήματα από διάφορες δουλειές στη βιβλιογραφία σχετικά με την αστάθεια της μετρικής του Inception Score, ακόμη περισσότερο όταν το Inception μοντέλο έχει εκπαιδευθεί σε διαφορετικά δεδομένα. Σε κάθε περίπτωση, φαίνεται και εδώ μια θετική εξέλιξη κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη



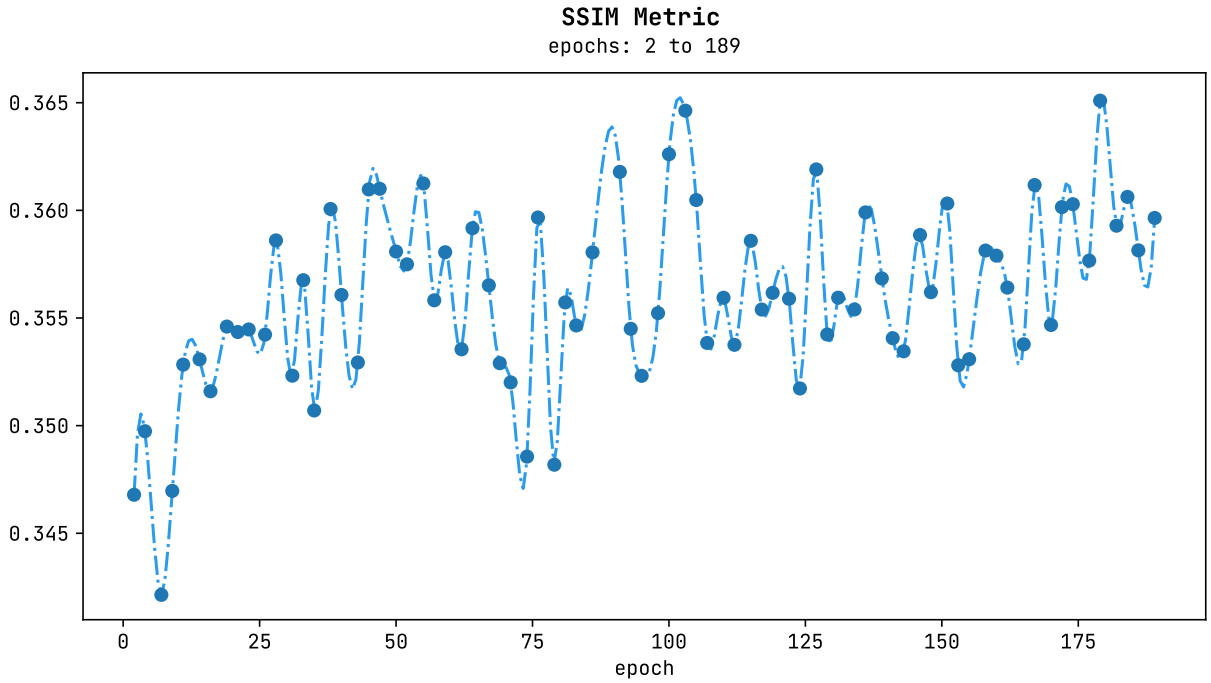
Σχήμα 107: Καμπύλη εξέλιξης της μετρικής Inception Score (IS) κατά τη διάρκεια εκπαίδευσης του μοντέλου CycleGAN.



Σχήμα 108: Καμπύλη εξέλιξης της μετρικής Fréchet Inception Distance (FID) κατά τη διάρκεια εκπαίδευσης του μοντέλου CycleGAN.



Σχήμα 109: Καμπύλη εξέλιξης της μετρικής F_1 Score κατά τη διάρκεια εκπαίδευσης του μοντέλου CycleGAN.



Σχήμα 110: Καμπύλη εξέλιξης της μετρικής Structural Similarity Index (SSIM) κατά τη διάρκεια εκπαίδευσης του μοντέλου CycleGAN.

συγκεκριμένη εφαρμογή.

- **Fréchet Inception Distance (FID)**: από το σχήμα 108 παραπάνω φαίνεται ότι η μετρική έχει μονότονη πτωτική τάση, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή. Είναι πεποίθηση του εκπονητή ότι η συνέχιση της εκπαίδευσης του μοντέλου θα οδηγούσε σε ακόμη καλύτερες τιμές της μετρικής αυτής. Εδώ φαίνεται να σταματάει περίπου στο 70.

- **F_1 Score**: γενικότερα η μετρική F_1 Score στα πλαίσια της αξιολόγησης GANs, έχει δειχθεί ότι είναι πιο σταθερή και αξιόπιστη από τις υπόλοιπες. Φαίνεται και εδώ, από το σχήμα 109 παραπάνω, ότι η μετρική έχει μονότονα αυξητική τάση η οποία σταματάει στα πλαίσια της εκπαίδευσής μας (λόγω πόρων) σε μία τιμή κοντά στο 0.78. Η τάση αυτή αδιαμφισβήτητα αποτελεί αφενός σημάδι ευσταθούς εκπαίδευσης και αφετέρου ένδειξη της αποτελεσματικότητας του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **Structural Similarity Index (SSIM)**: φαίνεται και για αυτή τη μετρική αξιολόγησης, από το σχήμα 110 παραπάνω, ότι η μετρική έχει συμπεριφορά τυχαίου περιπάτου γύρω από μια τιμή κοντά στο 0.36, νούμερο γενικά όχι και τόσο καλό. Ωστόσο, τονίζουμε και πάλι εδώ πως δεν ενδείκνυται η σύγκριση μοντέλων που δεν έχουν ξεκάθαρο στόχο στην έξοδό τους με τη μετρική SSIM και άρα **παρατίθεται εδώ καθαρά για λόγους πληρότητας**.

Τελική αξιολόγηση του μοντέλου

Για την τελική αξιολόγηση του μοντέλου χρησιμοποιήθηκαν οι παραπάνω μετρικές τόσο σε 10000 εικόνες από από το σύνολο δεδομένων δοκιμής, όσο και σε 10000 από το σύνολο δεδομένων εκπαίδευσης. Τα αποτελέσματα εμφανίζονται στον πίνακα που ακολουθεί χωρίς ωστόσο να γίνεται εδώ κάποια σύγκριση με το σχετικό άρθρο, μιας και που οι συγγραφείς αυτού δεν παρέθεσαν καμία μετρική αξιολόγησης (παρόλο που τα αποτελέσματά τους οπτικά φαίνονται καλύτερα σίγουρα λόγω και της πολύ περισσότερης εκπαίδευσης).

Πίνακας 14: Τελικές μετρικές αξιολόγησης του CycleGAN^{GT} (δικής μας υλοποίησης).

	CycleGAN^{GT}
FID	65.78 (train) - 66.06 (test)
IS	4.48 (train) - 4.42 (test)
SSIM	0.355 (train) - 0.353 (test)
Precision	0.607 (train) - 0.607 (test)
Recall	0.920 (train) - 0.918 (test)
F1	0.731 (train) - 0.730 (test)

6.4 Παραγωγή ρεαλιστικών εικόνων μόδας (StyleGAN)

Στην τέταρτη και τελευταία ενότητα αυτού του κεφαλαίου παραθέτουμε αποτελέσματα και μετρικές από την εν εξελίξει εκπαίδευση της υλοποίησής μας του μοντέλου StyleGAN. Όπως έχουμε αναφέρει πρόκειται για ένα αρκετά πολύπλοκο μοντέλο με καινοτόμα αρχιτεκτονική και state-of-the-art αποτελέσματα. Έχουν γίνει κάποιες απλοποιήσεις στην υλοποίησή μας όπως περιγράψαμε στην αντίστοιχη υποενότητα του προηγούμενου κεφαλαίου. Θα ξεκινήσουμε εδώ παραθέτοντας σε μορφή πίνακα μία σύνοψη του μοντέλου που αναπτύχθηκε, όπως κάνουμε για όλα τα προηγούμενα μοντέλα που και αναλύθηκαν στο παρόν κεφάλαιο.

Κατόπιν, συγκεντρώνουμε σε έναν άλλον πίνακα τις παραμέτρους εκπαίδευσης του μοντέλου StyleGAN, πρακτική που επίσης ακολουθήσαμε για όλα τα μοντέλα του παρόντος κεφαλαίου. Εδώ, είναι ο πίνακας 16 που δίνεται παρακάτω.

Πριν προχωρήσουμε στην παράθεση των καμπύλων εκπαίδευσης, καμπύλων εξέλιξης των μετρικών και των τελικών αξιολογήσεων, τονίζουμε για μία ακόμα φορά στο σημείο αυτό ότι **το μοντέλα μας είναι ακόμα σε πειραματικό στάδιο**, με την έννοια ότι ακόμα δοκιμάζονται configurations παρατηρούνται λάθη κλπ. Ωστόσο, παραθέτουμε έως αυτό το χρονικό σημείο την **τρέχουσα configuration στην οποία έχουμε εντοπίσει ότι κάτι πάει λάθος με τον Discriminator**, αλλά δυστυχώς είναι πάγια θέληση του εκπονητή να ορκιστεί και άρα να παραδώσει την παρούσα αναφορά.

Πίνακας 15: Σύνοψη του μοντέλου StyleGAN

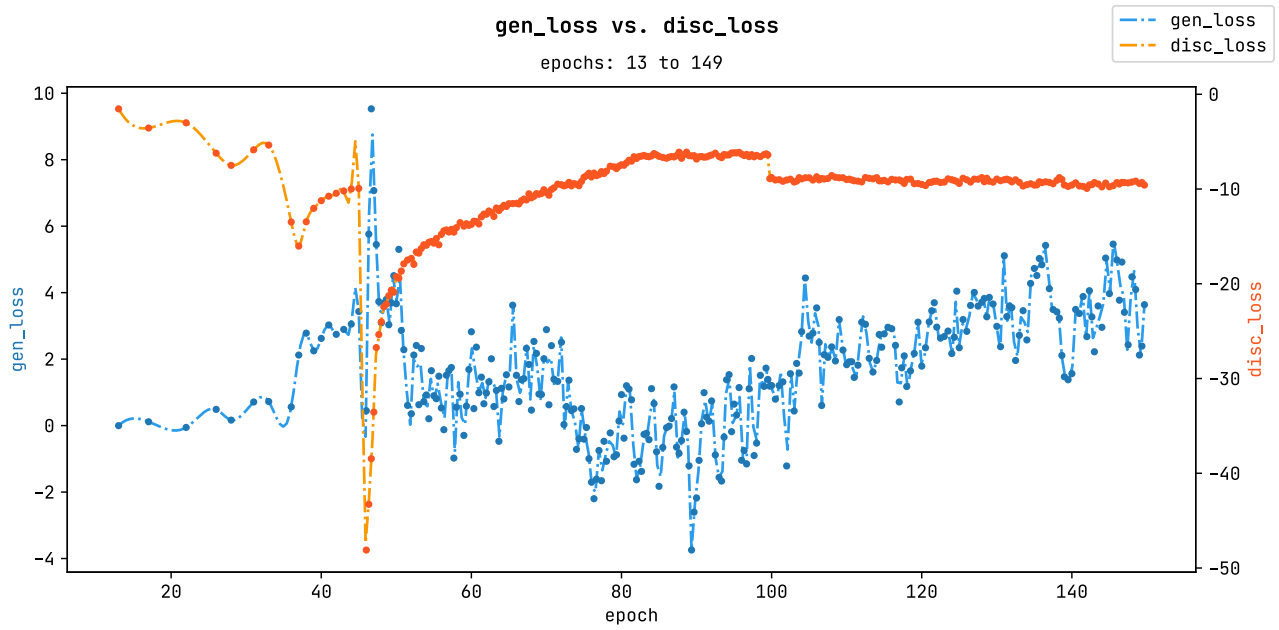
Όνομα Μοντέλου	StyleGAN ^{GT}
Κωδικός Configuration	`default_z512`
Εφαρμογή	Παραγωγή ρεαλιστικών εικόνων ανθρώπων σε φωτογραφίσεις μόδας
Κατηγορία Εφαρμογής	Παραγωγή Εικόνας από Θόρυβο
Κατηγορία Παραγωγής	χωρίς συνθήκη
Σχετικό Μοντέλο	StyleGAN
Σχετικό ΄Αρθρο	«*A Style-Based Generator Architecture for Generative Adversarial Networks*» [91]
Αριθμός GANs	ένα (1)
Αριθμός Generators	ένας (1) - αποτελούμενος από το δίκτυο αντιστοίχισης και το δίκτυο σύνθεσης
Αριθμός Discriminators	ένας (1)
Τύπος Generator(s)	Style-based με σταδιακή αύξηση progressive growing (5 blocks + έγχυση θορύβου + επιβολή στιλ στις στρώσεις AdaIN)
Τύπος Discriminator(s)	StyleGAN Discriminator με σταδιακή αύξηση (5 blocks, 128×128 receptive field)

Πίνακας 16: Παράμετροι εκπαίδευσης του μοντέλου StyleGAN

Όνομα Μοντέλου	StyleGAN ^{GT}
Κωδικός Configuration	`default_z512`
Συναρτ. Κόστους Generator	Wasserstein (WGAN) (αντιπαραθετική)
Συναρτ. Κόστους Discriminator(s)	Wasserstein (WGAN) + Ποινή Παραγώγων με βάρος λ_{GP}=10
Αριθμός βελτ/τών για Generator(s)	ένας (1)
Αριθμός βελτ/τών για Discriminator(s)	ένας (1)
Τύπος βελτ/τών για Generator(s)	Adam, με παραμέτρους (lr=0.0002, β_1=0.9, β_2=0.999) LR Scheduler: OnPlateau(factor: 0.99, cooldown: 100) ανά βήμα (όχι epoch)
Τύπος βελτ/τών για Discriminator(s)	Adam, με παραμέτρους (lr=0.0001, β_1=0.9, β_2=0.999) LR Scheduler: OnPlateau(factor: 0.99, cooldown: 100) ανά βήμα (όχι epoch)
Σύνολο δεδομένων εκπαίδευσης	FISB (DeepFashion)
Μέγεθος συνόλου δεδομένων	79K εικόνες (χρησιμοποιήθηκαν 53.2K εικόνες με χρώμα παρασκηνίου πιο λευκό από `F0F0F0` (hex))
Ανάλυση εικόνων	128×128px
Μέγεθος ομάδας	28 εικόνες/batch (στην ανάλυση 128×128)
Αριθμός epochs	**149 epochs** (186.750 επαναλήψεις)
Χρόνος Εκπαίδευσης	περίπου επτά (7) μέρες σε 24GB GPUs
# **Παραμέτρων Generator(s)**	58.5M εκπαιδεύσιμες παράμετροι (G1: 276M, G2: 117.4M)
# **Παραμέτρων Discriminator(s)**	11.3M εκπαιδεύσιμες παράμετροι
# **Παραμέτρων**	**69.8M εκπαιδεύσιμες παράμετροι συνολικά**

Καμπύλες Εκπαίδευσης

Ακολούθως, παραθέτουμε και εδώ για λόγους πληρότητας τις καμπύλες εξέλιξης των συναρτήσεων κόστους των βελτιστοποιητών ή καμπύλες εκπαίδευσης. Παρακάτω, παραθέτουμε τα διαγράμματα εξέλιξης των συναρτήσεων κόστους των δικτύων. Συγκεκριμένα, δίνουμε, στο ίδιο διάγραμμα, τη συνάρτηση κόστους των Discriminators συγκριτικά με αυτήν των Generators.



Σχήμα 111: Καμπύλες εκπαίδευσης του *StyleGAN*: Generator vs. real/fake Discriminator.

Ενδεικτικές Παραγωγές

Στη συνέχεια θα δώσουμε μερικές ενδεικτικές παραγωγές του μοντέλου μας εξαγωγής ρούχου, StyleGAN ως εξής:

- στην πρώτη γραμμή δίνονται τρεις (3) τυχαίες πραγματικές εικόνες από το σύνολο δεδομένων
- στη δεύτερη γραμμή δίνονται τρεις (3) τυχαίες παραγωγές του Generator **ασυσχέτιστες με τις πραγματικές εικόνες** (βάζουμε τις πραγματικές για αναφορά)

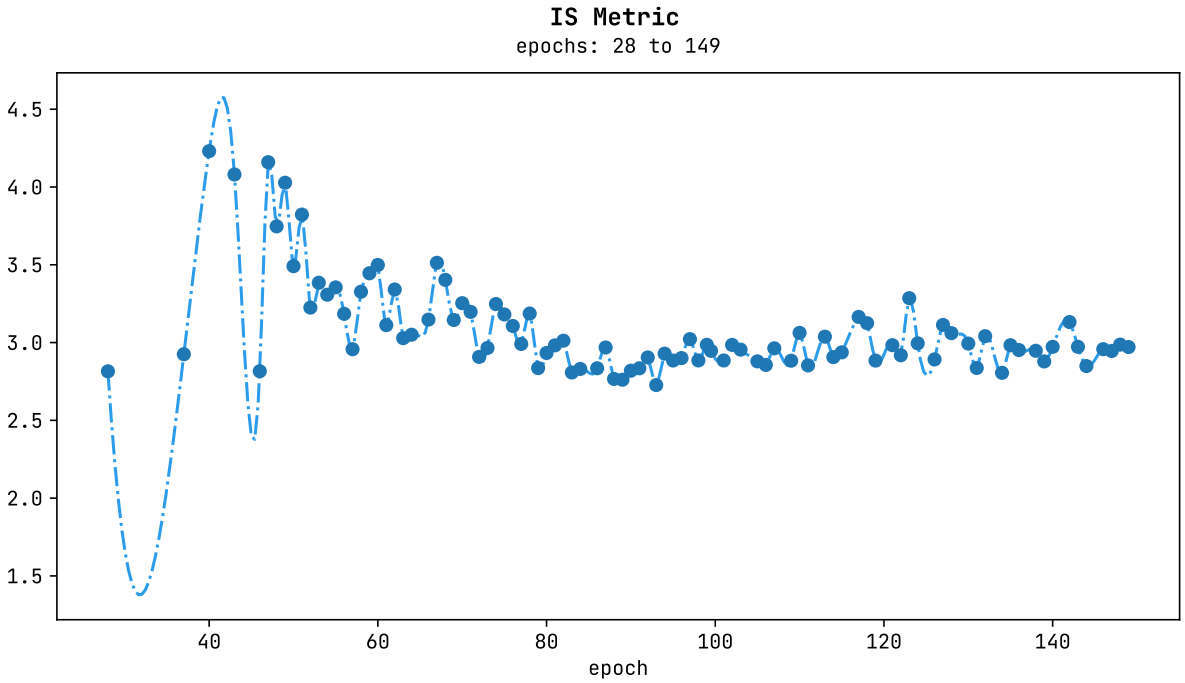
- κάθε στήλη είναι και μία ξεχωριστή παραγωγή

Οι παραγωγές δίνονται στα σχήματα 112 παρακάτω. Όπως φαίνεται εκεί το μοντέλο μας μετά από 149 epochs έχει καταφέρει να αιχμαλωτίσει σε μεγάλο βαθμό τη δομή συνόλου δεδομένων και έτσι μπορεί επιτυχώς να παράγει μία εικόνα με ρεαλιστικά χαρακτηριστικά προσώπου και σώματος, ενώ αρχίζει και καταλαβαίνει τις πόζες (αφού όλες σχεδόν οι εικόνες είναι φωτογραφίσεις προϊόντων μόδας). Αποτελεί πεποίθηση του εκπονητή της παρούσας εργασίας ότι **η συνέχιση της εκπαίδευσης** θα μπορούσε να βελτιώσει τις παραγωγές και να αυξήσει τη λεπτομέρειά τους. Στην υποενότητα που ακολουθεί δίνουμε τις τιμές των μετρικών αξιολόγησης των παραγόμενων εικόνων.

Σχήμα 112: Παραγωγές της υλοποίησής μας του μοντέλου StyleGAN από τον Style-based Generator. Όλες οι εικόνες είναι ανάλυσης 128×128, ενώ έχουν συλλεγεί τυχαία από το τελευταίο epoch της εκπαίδευσης.

Αξιολόγηση των Παραγωγών

Ακολούθως, παραθέτουμε διαγράμματα εξέλιξης των μετρικών αξιολόγησης (όπως και πριν: IS, FID, F1 & SSIM) κατά τη διάρκεια εκπαίδευσης της υλοποίησής μας του μοντέλου StyleGAN. Πριν την ανάγνωση των διαγραμμάτων, ο αναγνώστης θα πρέπει να είναι ενήμερος ότι για την καταγραφή των μετρικών χρησιμοποιήθηκαν και εδώ μόλις 1000 τυχαίες εικόνες από το σύνολο δεδομένων δοκιμής και ισάριθμες παραγωγές του Generator. Για

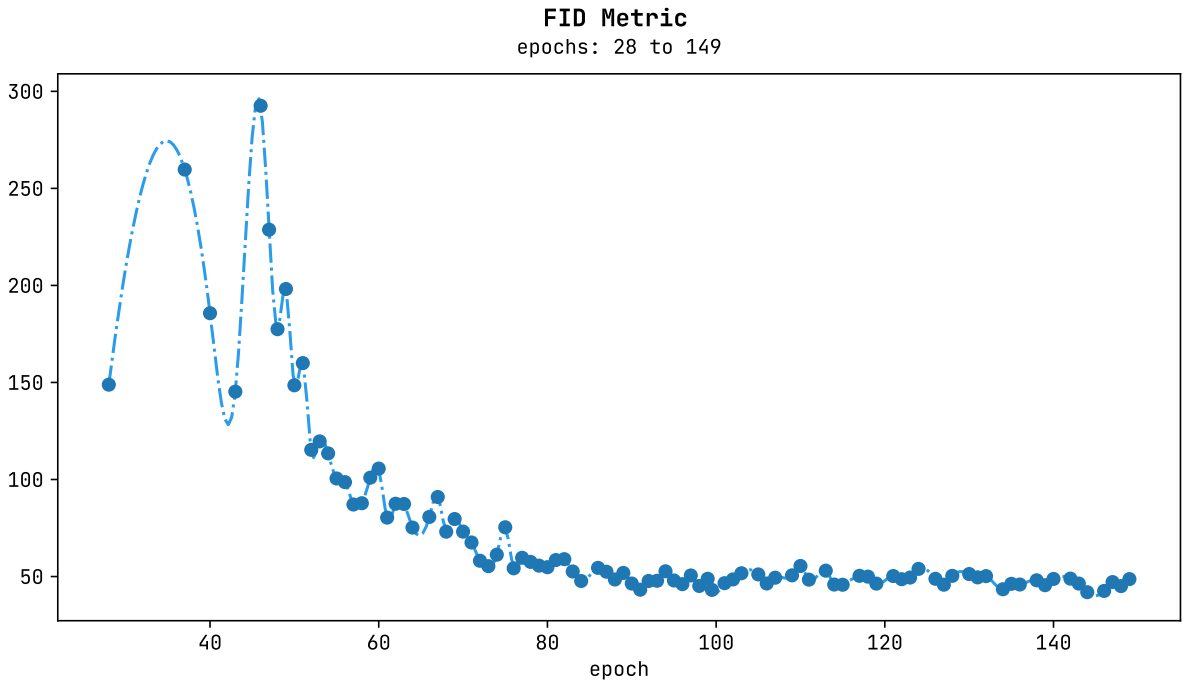


Σχήμα 113: Καμπύλη εξέλιξης της μετρικής Inception Score (IS) κατά τη διάρκεια εκπαίδευσης του μοντέλου StyleGAN.

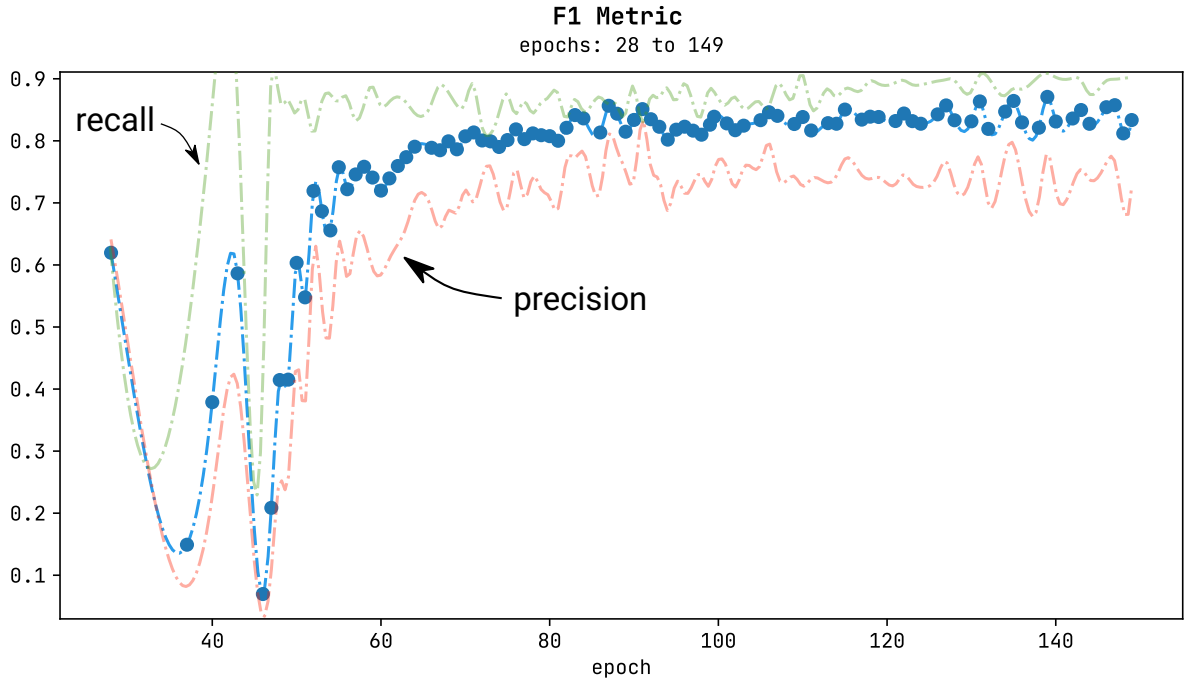
πιο αξιόπιστες μετρικές, ο αναγνώστης παραπέμπεται στην επόμενη υποενότητα, όπου παραθέτουμε σε μορφή πίνακα τις τελικές μετρικές αξιολόγησης του μοντέλου οι οποίες υπολογίσθηκαν από 10000 εικόνες.

Όπως φαίνεται στα διαγράμματα αυτά, όλες οι μετρικές πλην του Inception Score φαίνεται να βελτιώνονται με τον χρόνο. Στην αρχή φαίνονται τα σημεία αύξησης της διάστασης των δικτύων από τις έντονες μεταβολές των μετρικών αξιολόγησης, ενώ από ένα σημείο και ύστερα (περίπου στο epoch=90 και μετά) φαίνονται να συγκλίνουν προς μία τιμή. Αναλυτικότερα, για τη κάθε μετρική παραθέτουμε τα εξής σχόλια:

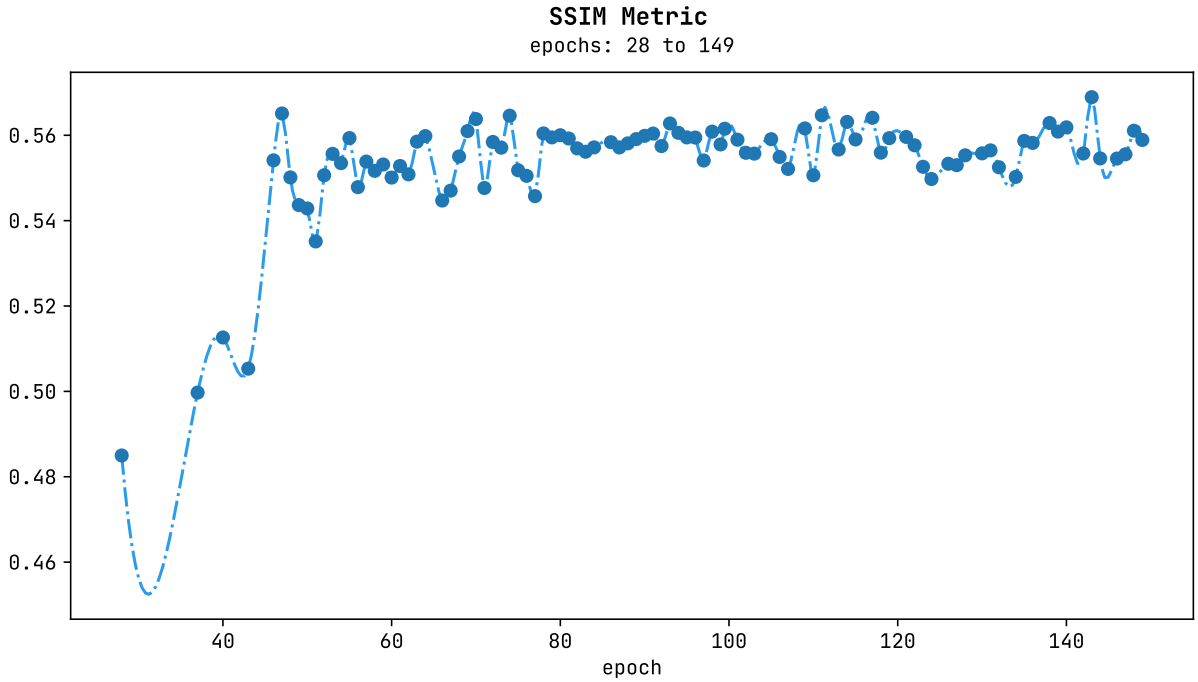
- **Inception Score**: από το πρώτο διάγραμμα φαίνεται ότι η μετρική IS παρουσιάζει συμπεριφορά τυχαίου περίπατου γύρω από τη τιμή 2.9. Η τιμή αυτή είναι δεν είναι «καλή» σε σχέση με αντίστοιχα μοντέλα στη βιβλιογραφία αλλά αποδεκτή δεδομένου ότι το ImageNET έχει λίγες τάξεις με εικόνες ανθρώπων και άρα το $N_{classes}$ που είχαμε αναφέρει ως μέγιστη τιμή σίγουρα είναι πολύ μικρότερο του 1000. Ωστόσο, συγκρίνοντας το παραπάνω διάγραμμα με αυτά των υπόλοιπων μετρικών και ιδιαίτερα με αυτό της FID και F1 που θεωρούνται πιο αξιόπιστες και σταθερές, επιβεβαιώνουμε τα ευρήματα από διάφορες δουλειές στη βιβλιογραφία σχετικά



Σχήμα 114: Καμπύλη εξέλιξης της μετρικής Fréchet Inception Distance (FID) κατά τη διάρκεια εκπαίδευσης του μοντέλου StyleGAN.



Σχήμα 115: Καμπύλη εξέλιξης της μετρικής F_1 Score κατά τη διάρκεια εκπαίδευσης του μοντέλου StyleGAN.



Σχήμα 116: Καμπύλη εξέλιξης της μετρικής Structural Similarity Index (SSIM) κατά τη διάρκεια εκπαίδευσης του μοντέλου StyleGAN.

με την αστάθεια της μετρικής του Inception Score, ακόμη περισσότερο όταν το Inception μοντέλο έχει εκπαιδευθεί σε διαφορετικά δεδομένα. Σε κάθε περίπτωση, φαίνεται και εδώ μια θετική εξέλιξη κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **Fréchet Inception Distance (FID)**: από το σχήμα 114 παραπάνω φαίνεται ότι η μετρική έχει μονότονη πτωτική τάση, κάτι που αφενός αποτελεί σημάδι ευσταθούς εκπαίδευσης και αφετέρου δείχνει την αποτελεσματικότητα του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή. Από το 90ό epoch φαίνεται να επιπεδώνει, κάτι που δεν ανταποκρίνεται ωστόσο στα οπτικά αποτελέσματα και την ανθρώπινη κρίση. Είναι πεποίθηση του εκπονητή ότι η συνέχιση της εκπαίδευσης του μοντέλου θα οδηγούσε σε ακόμη καλύτερες τιμές της μετρικής αυτής. Εδώ φαίνεται να σταματάει περίπου στο 40.

- **F_1 Score**: γενικότερα η μετρική F_1 Score στα πλαίσια της αξιολόγησης GANs, έχει δειχθεί ότι είναι πιο σταθερή και αξιόπιστη από τις υπόλοιπες. Φαίνεται και εδώ, από το σχήμα 115 παραπάνω, ότι η μετρική έχει μονότονα αυξητική τάση η οποία

σταματάει στα πλαίσια της εκπαίδευσής μας (λόγω πόρων) σε μία τιμή κοντά στο 0.3. Η τάση αυτή αδιαμφισβήτητα αποτελεί αφενός σημάδι ευσταθούς εκπαίδευσης και αφετέρου ένδειξη της αποτελεσματικότητας του μοντέλου που αναπτύχθηκε για τη συγκεκριμένη εφαρμογή.

- **Structural Similarity Index (SSIM)**: φαίνεται και για αυτή τη μετρική αξιολόγησης, από το σχήμα 116 παραπάνω, ότι η μετρική έχει συμπεριφορά τυχαίου περιπάτου γύρω από μια τιμή κοντά στο 0.56, νούμερο γενικά αρκετά καλό. Ωστόσο, τονίζουμε και πάλι εδώ πως δεν ενδείκνυται η σύγκριση μοντέλων που δεν έχουν ξεκάθαρο στόχο στην έξοδό τους με τη μετρική SSIM και άρα **παρατίθεται εδώ καθαρά για λόγους πληρότητας**.

Τελική αξιολόγηση του μοντέλου

Για την τελική αξιολόγηση του μοντέλου χρησιμοποιήθηκαν οι παραπάνω μετρικές τόσο σε 10000 εικόνες από από το σύνολο δεδομένων δοκιμής, όσο και σε 10000 από το σύνολο δεδομένων εκπαίδευσης. Τα αποτελέσματα εμφανίζονται στον πίνακα που ακολουθεί χωρίς ωστόσο να γίνεται εδώ κάποια σύγκριση με το σχετικό άρθρο, μιας και που οι συγγραφείς αυτού παρέθεσαν μόνο **το FID (ίσο με 8.53) αλλά για διαφορετικό σύνολο δεδομένων**.

Πίνακας 17: Τελικές μετρικές αξιολόγησης του StyleGAN^{GT} (δικής μας υλοποίησης).

	StyleGAN^{GT}
FID	34.21 (train) - 33.01 (test)
IS	2.979 (train) - 2.987 (test)
SSIM	0.562 (train) - 0.561 (test)
Precision	0.783 (train) - 0.779 (test)
Recall	0.864 (train) - 0.824 (test)
F1	0.831 (train) - 0.801 (test)

Κεφάλαιο 7

Σύνοψη και Μελλοντικές Επεκτάσεις

Κάπου εδώ ολοκληρώνεται η παρούσα εργασία, έχοντας εκπαιδεύσει τέσσερα (4) μοντέλα GAN που καλύπτουν το πλήρες εύρος των κατηγοριών εφαρμογών GANs στα πλαίσια της Παραγωγικής Μοντελοποίησης εικόνας. Αυτά αποτελούν και την πρώτη έκδοση του «πολυ-εργαλείου» μας. Ακολουθεί μια συνοπτική περίληψη της παρούσας διπλωματικής εργασίας καθώς και πιθανές μελλοντικές προεκτάσεις αυτής.

Σύνοψη της Εργασίας

Έτσι, σε ό,τι προηγήθηκε, αρχικά προχωρήσαμε σε μία ανάλυση σχετικών μεθόδων και τεχνικών του γενικότερου ερευνητικού κλάδου της Παραγωγικής Μοντελοποίησης (κεφάλαιο 2). Κατόπιν, επικεντρωθήκαμε στα Generative Adversarial Networks, αναφερόμενοι τόσο σε παραμέτρους εκπαίδευσης αυτών και αξιολόγησης της απόδοσής τους (κεφάλαιο 3), όσο και για σχετικές υλοποιήσεις που έχουν παρουσιαστεί στη βιβλιογραφία αλλά και στην πράξη (κεφάλαιο 4). Προχωρώντας στη δική μας υλοποίηση και μεθοδολογία, στο κεφάλαιο 5, παρουσιάσαμε στοιχεία τόσο για τα σύνολα δεδομένων που χρησιμοποιήθηκαν και τις μεθόδους προ-επεξεργασίας αυτών, όσο και για τα μοντέλα GAN τα οποία σχεδιάστηκαν, υλοποιήθηκαν και εκπαιδεύτηκαν σε αυτά τα σύνολα δεδομένων. Αφήσαμε την παράθεση όλων των αποτελεσμάτων και καμπύλων εκπαίδευσης και εξέλιξης των μετρικών σε ένα ξεχωριστό κεφάλαιο, το κεφάλαιο 6, ώστε ο ενδιαφερόμενος αναγνώστης να μπορεί άμεσα να ανατρέξει στα αποτελέσματα της δουλειάς μας.

Ως αποτέλεσμα και συνεισφορά της παρούσας εργασίας, τέσσερα (4) μοντέλα GAN καθώς και οι μέθοδοι επεξεργασίας σχετικών συνόλων δεδομένων εικόνων μόδας, δίνονται

ελεύθερα και ανοιχτά στο αποθετήριο κώδικα της εργασίας. Τα μοντέλα αυτά από κοινού συνθέτουν το ευφυές πολυ-εργαλείο παραγωγής και εξεργασίας εικόνων μόδας το οποίο χρησιμοποιεί GANs (κατά κύριο λόγο) *για επιλογή πόζας και στιλ σε εφαρμογές σχεδιασμού μόδας*. Θεωρούμε σκόπιμο στο σημείο αυτό να αναφέρουμε για μία ακόμη φορά πως αν και τα μοντέλα που εκπαιδεύτηκαν βασίζονται σε αντίστοιχα μοντέλα της βιβλιογραφίας, προσπαθήσαμε να μην χρησιμοποιήσουμε έτοιμες υλοποιήσεις (όπου υπήρχαν) αλλά να σχεδιάσουμε και να υλοποιήσουμε μοντέλα GAN από την αρχή - το οποίο σε ορισμένες περιπτώσεις οδήγησε σε ουσιαστικά διαφορετικές υλοποιήσεις και παραγόμενα αποτελέσματα.

Μελλοντικές Επεκτάσεις

Ως τελευταίες σημειώσεις, αφήνονται πιθανές μελλοντικές προεκτάσεις τόσο γιατί είναι στις επιδιώξεις του εκπονητή να τις υλοποιήσει όσο και για κάθε ενδιαφερόμενο αναγνώστη.

Οι μελλοντικές προεκτάσεις, λοιπόν, των μοντέλων που αναπτύχθηκαν και εκπαιδεύτηκαν στα πλαίσια της παρούσας εργασίας, είναι μεταξύ άλλων οι ακόλουθες:

- Εκπαίδευση ταξινομητών εικόνας (π.χ. Inception) σε σύνολα δεδομένων εικόνων μόδας (όπως το σύνολο δεδομένων IMaterialist του Kaggle), για πιο αντιπροσωπευτικές και αξιόπιστες μετρικές αξιολόγησης. Εναλλακτικά, εκπαίδευση του Inception v3 σε dataset εικόνων μόδας που χρησιμοποιήθηκαν (π.χ. το Category and Attribute Prediction Benchmark του DeepFashion και επαναξιολόγηση των όλων των μετρικών όλων των μοντέλων.

- Υλοποίηση και εκτέλεση μιας πρόσθετης μετρικής αξιολόγησης των παραγόμενων: Perceptual Loss [52].

- Δοκιμές Style mixing στο StyleGAN: εύρεση ποιες στρώσεις επηρεάζουν ποια ρούχα/σημεία του σώματος. Μείξη ρούχων και προσπάθεια αλλαγής πόζας μέσω του StyleGAN.

- Εκπαίδευση του StyleGAN με συνάρτηση κόστους Ελαχίστων Τετραγώνων (MSE) και Κανονικοποίηση Φάσματος.

- Περισσότερη εκπαίδευση (ενν. για περισσότερα epochs) σε όλα τα μοντέλα με

πιθανή εξαίρεση το *PoseGAN*.

- Δοκιμή του StyleGAN v2 στο ίδιο σύνολο δεδομένων με το StyleGAN, αλλά και σε άλλα παρεμφερή σύνολα δεδομένων.
- Δοκιμή του MUNIT στο ίδιο σύνολο δεδομένων με το CycleGAN και σύγκριση των αποτελεσμάτων/μετρικών αξιολόγησης.

Παράρτημα Α

Ακρωνύμια και συντομογραφίες

Ελληνικά Ακρωνύμια

ΑΚ Αυτόματοι Κωδικοποιητές

ΒΠΜ Βαθιά Παραγωγικά Μοντέλα

ΕΝΝ Επαναλαμβανόμενα Νευρωνικά Δίκτυα

ΚΑΚ Κανονισμένοι Αυτόματοι Κωδικοποιητές

ΜΑΚ Μεταβλητοί Αυτόματοι Κωδικοποιητές

ΠΜ Παραγωγική Μοντελοποίηση

ΠΣ Πλήρως Συνδεδεμένη

ΣΝΔ Συνελικτικά Νευρωνικά Δίκτυα

ΤΜ Τυχαίες Μεταβλητές

ΤΝΔ Τεχνητά Νευρωνικά Δίκτυα

Αγγλικά Ακρωνύμια και Συντομογραφίες

AE Autoencoder

AI Artificial Intelligence

BCE Binary Cross-Entropy

CCRB Consumer-to-Shop Clothes Retrieval Benchmark (DeepFashion)

CGAN Conditional Generative Adversarial Network

CNN Convolutional Neural Network

DAE Denoising Autoencoder

DCGAN Deep Convolutional Generative Adversarial Network

DLSS Deep Learning Super-Sampling (NVIDIA)

DNN Deep Neural Network

EMD Earth Mover's Distance

FC Fully Connected

FFHQ Flickr-Faces-HQ Dataset

FID Fréchet Inception Distance

FISB Fashion-Images Synthesis Benchmark (DeepFashion)

FVSBN Fully-Visible Sigmoid Belief Network

GAN Generative Adversarial Network

GDA Gaussian Discriminant Analysis

GP Gradient Penalty

ICRB In-Shop Clothes Retrieval Benchmark (DeepFashion)

ILSVRC ImageNet Large-Scale Visual Recognition Challenge

IS Inception Score

KL Kullback-Leibler Divergence

LAPGAN Laplacian Pyramid of Generative Adversarial Networks

LReLU Leaky Rectified Linear Unit

LSGAN Least-Squares Generative Adversarial Network

LoC Lines of Code

MADE Masked Autoregressive Density Estimator

MSSIM Mean Structural Similarity Index

MNIST Modified National Institute of Standards and Technology

MSE Mean Square Error

NADE Neural Autoregressive Distribution Estimator

PGGAN Progressively Growing Generative Adversarial Network

PPL Perceptual Path Length

RAE Regularized Autoencoder

ReLU Rectified Linear Unit

RGB Red-Green-Blue (image color channels)

RNN Recurrent Neural Network

SDAE Stacked Denoising Autoencoder

SGD Stochastic Gradient Descent

SN-GAN Spectral-Normalized Generative Adversarial Network

SSIM Structural Similarity Index

SVD Singular Value Decomposition

VAE Variational Autoencoder

VGG Visual Geometry Group (Oxford University)

VQ-VAE Vector-Quantized Variational Autoencoder

WGAN Wasserstein Generative Adversarial Network

Βιβλιογραφικές Αναφορές

[1] M. M. Fréchet, «Sur quelques points du calcul fonctionnel», *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, vol. 22, no. 1, pp. 1–72, Dec. 1906, ISSN: 0009-725X. DOI: 10.1007/BF03018603. [Online]. Available: https://doi.org/10.1007/BF03018603.

[2] R. A. FISHER, «THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS», *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936. DOI: https://doi.org/10.1111/j.1469-1809.1936.tb02137.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-1809.1936.tb02137.x. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

[3] A. M. Turing, «Computing Machinery and Intelligence», *Mind*, vol. 59, no. 236, pp. 433–460, 1950, ISSN: 00264423, 14602113. [Online]. Available: http://www.jstor.org/stable/2251299.

[4] S. Kullback and R. A. Leibler, «On Information and Sufficiency», *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. DOI: 10.1214/aoms/1177729694. [Online]. Available: https://doi.org/10.1214/aoms/1177729694.

[5] Y. Lecun, *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models)*, English (US). Universite P. et M. Curie (Paris 6), Jun. 1987.

[6] H. Bourlard and Y. Kamp, «Auto-Association by Multilayer Perceptrons and Singular Value Decomposition», *Biol. Cybern.*, vol. 59, no. 4–5, pp. 291–294, Sep. 1988, ISSN: 0340-1200. DOI: 10.1007/BF00332918. [Online]. Available: https://doi.org/10.1007/BF00332918.

[7] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. USA: Kluwer Academic Publishers, 1991, ISBN: 0792391810.

[8] G. E. Hinton and R. S. Zemel, «Autoencoders, Minimum Description Length and Helmholtz Free Energy», in *Proceedings of the 6th International Conference on Neural*

Information Processing Systems, ser. NIPS'93, Denver, Colorado: Morgan Kaufmann Publishers Inc., 1993, pp. 3–10.

[9] C. Cortes and V. Vapnik, «Support-vector networks», *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, ISSN: 1573-0565. DOI: 10.1007/BF00994018. [Online]. Available: https://doi.org/10.1007/BF00994018.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, «Gradient-Based Learning Applied to Document Recognition», in *Proceedings of the IEEE*, vol. 86, 1998, pp. 2278–2324. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665.

[11] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, «Object Recognition with Gradient-Based Learning», in *Shape, Contour and Grouping in Computer Vision*, Berlin, Heidelberg: Springer-Verlag, 1999, p. 319, ISBN: 3540667229.

[12] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, «Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data», in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289, ISBN: 1558607781.

[13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, «Image quality assessment: from error visibility to structural similarity», *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: 10.1109/TIP.2003.819861.

[14] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, «Efficient Learning of Sparse Representations with an Energy-Based Model», in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06, Canada: MIT Press, 2006, pp. 1137–1144.

[15] M. A. Ranzato, Y.-L. Boureau, and Y. LeCun, «Sparse Feature Learning for Deep Belief Networks», in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07, Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, pp. 1185–1192, ISBN: 9781605603520.

[16] J. Deng, W. Dong, R. Socher, L.-j. Li, K. Li, and L. Fei-fei, «Imagenet: A large-scale hierarchical image database», in *In CVPR*, 2009.

[17] J. W. Gibbs, *Elementary Principles in Statistical Mechanics: Developed with Especial Reference to the Rational Foundation of Thermodynamics*, ser. Cambridge Library Collection - Mathematics. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511686948.

[18] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, «Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion», *Journal of Machine Learning Research*, vol. 11, no. 110, pp. 3371–3408, 2010. [Online]. Available: http://jmlr.org/papers/v11/vincent10a.html.

[19] X. Glorot, A. Bordes, and Y. Bengio, «Deep Sparse Rectifier Neural Networks», in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, Eds., ser. Proceedings of Machine Learning Research, vol. 15, Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323. [Online]. Available: http://proceedings.mlr.press/v15/glorot11a.html.

[20] H. Larochelle and I. Murray, «The Neural Autoregressive Distribution Estimator», in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, Eds., ser. Proceedings of Machine Learning Research, vol. 15, Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 29–37. [Online]. Available: http://proceedings.mlr.press/v15/larochelle11a.html.

[21] T. Lattimore and M. Hutter, «No Free Lunch versus Occam's Razor in Supervised Learning», *arXiv e-prints*, arXiv:1111.3846, arXiv:1111.3846, Nov. 2011. arXiv: 1111.3846 [cs.LG].

[22] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, «Contractive Auto-Encoders: Explicit Invariance during Feature Extraction», in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11, Bellevue, Washington, USA: Omnipress, 2011, pp. 833–840, ISBN: 9781450306195.

[23] A. Krizhevsky, «Learning Multiple Layers of Features from Tiny Images», *University of Toronto*, May 2012.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks», in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[25] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, Version 20121115, Nov. 2012. [Online]. Available: http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html.

[26] Γραββάνης, Γεώργιος Α. and Γιαννουτάκης, Κωνσταντίνος Μ., *Προγραμματισμός με τη Χρήση MATLAB*. Εκδόσεις Παπασωτηρίου, 2012.

[27] Y. Bengio, L. Yao, G. Alain, and P. Vincent, «Generalized Denoising Auto-Encoders as Generative Models», *arXiv e-prints*, arXiv:1305.6663, arXiv:1305.6663, May 2013. arXiv: 1305.6663 [cs.LG].

[28] A. Makhzani and B. Frey, «k-Sparse Autoencoders», *arXiv e-prints*, arXiv:1312.5663, arXiv:1312.5663, Dec. 2013. arXiv: 1312.5663 [cs.LG].

[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, «Generative Adversarial Networks», *arXiv e-prints*, arXiv:1406.2661, arXiv:1406.2661, Jun. 2014. arXiv: 1406.2661 [stat.ML].

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, «Explaining and Harnessing Adversarial Examples», *arXiv e-prints*, arXiv:1412.6572, arXiv:1412.6572, Dec. 2014. arXiv: 1412.6572 [stat.ML].

[31] D. P. Kingma and J. Ba, «Adam: A Method for Stochastic Optimization», *arXiv e-prints*, arXiv:1412.6980, arXiv:1412.6980, Dec. 2014. arXiv: 1412.6980 [cs.LG].

[32] M. Mirza and S. Osindero, «Conditional Generative Adversarial Nets», *arXiv e-prints*, arXiv:1411.1784, arXiv:1411.1784, Nov. 2014. arXiv: 1411.1784 [cs.LG].

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, «ImageNet Large Scale Visual Recognition Challenge», *arXiv e-prints*, arXiv:1409.0575, arXiv:1409.0575, Sep. 2014. arXiv: 1409.0575 [cs.CV].

[34] K. Simonyan and A. Zisserman, «Very Deep Convolutional Networks for Large-Scale Image Recognition», *arXiv e-prints*, arXiv:1409.1556, arXiv:1409.1556, Sep. 2014. arXiv: 1409.1556 [cs.CV].

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, «Dropout: A Simple Way to Prevent Neural Networks from Overfitting», *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, «Going Deeper with Convolutions», *arXiv e-prints*, arXiv:1409.4842, arXiv:1409.4842, Sep. 2014. arXiv: 1409.4842 [cs.CV].

[37] A. Yu and K. Grauman, «Fine-Grained Visual Comparisons with Local Learning», in *Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014.

[38] E. Denton, S. Chintala, A. Szlam, and R. Fergus, «Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks», *arXiv e-prints*, arXiv:1506.05751, arXiv:1506.05751, Jun. 2015. arXiv: 1506.05751 [cs.CV].

[39] Z. Gan, R. Henao, D. Carlson, and L. Carin, «Learning Deep Sigmoid Belief Networks with Data Augmentation», in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, G. Lebanon and S. V. N. Vishwanathan, Eds., ser. Proceedings of Machine Learning Research, vol. 38, San Diego, California, USA: PMLR, May 2015, pp. 268–276. [Online]. Available: http://proceedings.mlr.press/v38/gan15.html.

[40] M. Germain, K. Gregor, I. Murray, and H. Larochelle, «MADE: Masked Autoencoder for Distribution Estimation», *arXiv e-prints*, arXiv:1502.03509, arXiv:1502.03509, Feb. 2015. arXiv: 1502.03509 [cs.LG].

[41] K. He, X. Zhang, S. Ren, and J. Sun, «Deep Residual Learning for Image Recognition», *arXiv e-prints*, arXiv:1512.03385, arXiv:1512.03385, Dec. 2015. arXiv: 1512.03385 [cs.CV].

[42] S. Ioffe and C. Szegedy, «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift», *arXiv e-prints*, arXiv:1502.03167, arXiv:1502.03167, Feb. 2015. arXiv: 1502.03167 [cs.LG].

[43] Z. Liu, P. Luo, X. Wang, and X. Tang, «Deep Learning Face Attributes in the Wild», in *Proceedings of International Conference on Computer Vision (ICCV)*, Dec. 2015.

[44] A. Radford, L. Metz, and S. Chintala, «Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks», *arXiv e-prints*, arXiv:1511.06434, arXiv:1511.06434, Nov. 2015. arXiv: 1511.06434 [cs.LG].

[45] O. Ronneberger, P. Fischer, and T. Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation», *arXiv e-prints*, arXiv:1505.04597, arXiv:1505.04597, May 2015. arXiv: 1505.04597 [cs.CV].

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, «Rethinking the Inception Architecture for Computer Vision», *arXiv e-prints*, arXiv:1512.00567, arXiv:1512.00567, Dec. 2015. arXiv: 1512.00567 [cs.CV].

[47] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, «LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop», *arXiv e-prints*, arXiv:1506.03365, arXiv:1506.03365, Jun. 2015. arXiv: 1506.03365 [cs.CV].

[48] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, «InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adver-

sarial Nets», *arXiv e-prints*, arXiv:1606.03657, arXiv:1606.03657, Jun. 2016. arXiv: 1606.03657 [cs.LG].

[49] V. Dumoulin and F. Visin, «A guide to convolution arithmetic for deep learning», *arXiv e-prints*, arXiv:1603.07285, arXiv:1603.07285, Mar. 2016. arXiv: 1603.07285 [stat.ML].

[50] I. Goodfellow, «NIPS 2016 Tutorial: Generative Adversarial Networks», *arXiv e-prints*, arXiv:1701.00160, arXiv:1701.00160, Dec. 2016. arXiv: 1701.00160 [cs.LG].

[51] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, «Image-to-Image Translation with Conditional Adversarial Networks», *arXiv e-prints*, arXiv:1611.07004, arXiv:1611.07004, Nov. 2016. arXiv: 1611.07004 [cs.CV].

[52] J. Johnson, A. Alahi, and L. Fei-Fei, «Perceptual Losses for Real-Time Style Transfer and Super-Resolution», *arXiv e-prints*, arXiv:1603.08155, arXiv:1603.08155, Mar. 2016. arXiv: 1603.08155 [cs.CV].

[53] J. Larson and J. Angwin, *Machine Bias*, May 2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[54] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, «Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network», *arXiv e-prints*, arXiv:1609.04802, arXiv:1609.04802, Sep. 2016. arXiv: 1609.04802 [cs.CV].

[55] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, «DeepFashion: Powering Robust Clothes Recognition and Retrieval With Rich Annotations», in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[56] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, «Least Squares Generative Adversarial Networks», *arXiv e-prints*, arXiv:1611.04076, arXiv:1611.04076, Nov. 2016. arXiv: 1611.04076 [cs.CV].

[57] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, «Unrolled Generative Adversarial Networks», *arXiv e-prints*, arXiv:1611.02163, arXiv:1611.02163, Nov. 2016. arXiv: 1611.02163 [cs.LG].

[58] A. Odena, V. Dumoulin, and C. Olah, «Deconvolution and Checkerboard Artifacts», *Distill*, 2016. DOI: 10.23915/distill.00003. [Online]. Available: http://distill.pub/2016/deconv-checkerboard.

[59] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, «Improved Techniques for Training GANs», *arXiv e-prints*, arXiv:1606.03498, arXiv:1606.03498, Jun. 2016. arXiv: 1606.03498 [cs.LG].

[60] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, «Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning», *arXiv e-prints*, arXiv:1602.07261, arXiv:1602.07261, Feb. 2016. arXiv: 1602.07261 [cs.CV].

[61] D. Ulyanov, A. Vedaldi, and V. Lempitsky, «Instance Normalization: The Missing Ingredient for Fast Stylization», *arXiv e-prints*, arXiv:1607.08022, arXiv:1607.08022, Jul. 2016. arXiv: 1607.08022 [cs.CV].

[62] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, «WaveNet: A Generative Model for Raw Audio», *arXiv e-prints*, arXiv:1609.03499, arXiv:1609.03499, Sep. 2016. arXiv: 1609.03499 [cs.SD].

[63] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, «Pixel Recurrent Neural Networks», *arXiv e-prints*, arXiv:1601.06759, arXiv:1601.06759, Jan. 2016. arXiv: 1601.06759 [cs.CV].

[64] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, «Conditional Image Generation with PixelCNN Decoders», *arXiv e-prints*, arXiv:1606.05328, arXiv:1606.05328, Jun. 2016. arXiv: 1606.05328 [cs.CV].

[65] R. A. Yeh, C. Chen, T.-Y. Lim, M. Hasegawa-Johnson, and M. Do, «Semantic Image Inpainting with Perceptual and Contextual Losses», *ArXiv*, vol. abs/1607.07539, 2016.

[66] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, «Pixel-Level Domain Transfer», *arXiv e-prints*, arXiv:1603.07442, arXiv:1603.07442, Mar. 2016. arXiv: 1603.07442 [cs.CV].

[67] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, «Generative Visual Manipulation on the Natural Image Manifold», *arXiv e-prints*, arXiv:1609.03552, arXiv:1609.03552, Sep. 2016. arXiv: 1609.03552 [cs.CV].

[68] M. Arjovsky and L. Bottou, «Towards Principled Methods for Training Generative Adversarial Networks», *arXiv e-prints*, arXiv:1701.04862, arXiv:1701.04862, Jan. 2017. arXiv: 1701.04862 [stat.ML].

[69] M. Arjovsky, S. Chintala, and L. Bottou, «Wasserstein GAN», *arXiv e-prints*, arXiv:1701.07875, arXiv:1701.07875, Jan. 2017. arXiv: 1701.07875 [stat.ML].

[70] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. The MIT Press, 2017.

[71] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, «Improved Training of Wasserstein GANs», *arXiv e-prints*, arXiv:1704.00028, arXiv:1704.00028, Mar. 2017. arXiv: 1704.00028 [cs.LG].

[72] K. He, G. Gkioxari, P. Dollár, and R. Girshick, «Mask R-CNN», *arXiv e-prints*, arXiv:1703.06870, arXiv:1703.06870, Mar. 2017. arXiv: 1703.06870 [cs.CV].

[73] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, «GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium», *arXiv e-prints*, arXiv:1706.08500, arXiv:1706.08500, Jun. 2017. arXiv: 1706.08500 [cs.LG].

[74] T. Karras, T. Aila, S. Laine, and J. Lehtinen, «Progressive Growing of GANs for Improved Quality, Stability, and Variation», *arXiv e-prints*, arXiv:1710.10196, arXiv:1710.10196, Oct. 2017. arXiv: 1710.10196 [cs.NE].

[75] K. Kawaguchi, L. Pack Kaelbling, and Y. Bengio, «Generalization in Deep Learning», *arXiv e-prints*, arXiv:1710.05468, arXiv:1710.05468, Oct. 2017. arXiv: 1710.05468 [stat.ML].

[76] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, «Learning to Discover Cross-Domain Relations with Generative Adversarial Networks», *arXiv e-prints*, arXiv:1703.05192, arXiv:1703.05192, Mar. 2017. arXiv: 1703.05192 [cs.CV].

[77] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, «On Convergence and Stability of GANs», *arXiv e-prints*, arXiv:1705.07215, arXiv:1705.07215, May 2017. arXiv: 1705.07215 [cs.AI].

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks», *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 0001-0782. DOI: 10.1145/3065386. [Online]. Available: https://doi.org/10.1145/3065386.

[79] M.-Y. Liu, T. Breuel, and J. Kautz, «Unsupervised Image-to-Image Translation Networks», *arXiv e-prints*, arXiv:1703.00848, arXiv:1703.00848, Mar. 2017. arXiv: 1703.00848 [cs.CV].

[80] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, «Are GANs Created Equal? A Large-Scale Study», *arXiv e-prints*, arXiv:1711.10337, arXiv:1711.10337, Nov. 2017. arXiv: 1711.10337 [stat.ML].

[81] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, «Pose Guided Person Image Generation», *arXiv e-prints*, arXiv:1705.09368, arXiv:1705.09368, May 2017. arXiv: 1705.09368 [cs.CV].

[82] A. Rangamani, A. Mukherjee, A. Basu, T. Ganapathy, A. Arora, S. Chin, and T. D. Tran, «Sparse Coding and Autoencoders», *arXiv e-prints*, arXiv:1708.03735, arXiv:1708.03735, Aug. 2017. arXiv: 1708.03735 [cs.LG].

[83] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, «Neural Discrete Representation Learning», *arXiv e-prints*, arXiv:1711.00937, arXiv:1711.00937, Nov. 2017. arXiv: 1711.00937 [cs.LG].

[84] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, «High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs», *arXiv e-prints*, arXiv:1711.11585, arXiv:1711.11585, Nov. 2017. arXiv: 1711.11585 [cs.CV].

[85] Y. Yoshida and T. Miyato, «Spectral Norm Regularization for Improving the Generalizability of Deep Learning», *arXiv e-prints*, arXiv:1705.10941, arXiv:1705.10941, May 2017. arXiv: 1705.10941 [stat.ML].

[86] A. Yu and K. Grauman, «Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images», in *International Conference on Computer Vision (ICCV)*, Oct. 2017.

[87] R. Alp Güler, N. Neverova, and I. Kokkinos, «DensePose: Dense Human Pose Estimation In The Wild», *arXiv e-prints*, arXiv:1802.00434, arXiv:1802.00434, Feb. 2018. arXiv: 1802.00434 [cs.CV].

[88] A. Brock, J. Donahue, and K. Simonyan, «Large Scale GAN Training for High Fidelity Natural Image Synthesis», *arXiv e-prints*, arXiv:1809.11096, arXiv:1809.11096, Sep. 2018. arXiv: 1809.11096 [cs.LG].

[89] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He, *Detectron*, https://github.com/facebookresearch/detectron, 2018.

[90] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, «Multimodal Unsupervised Image-to-Image Translation», *arXiv e-prints*, arXiv:1804.04732, arXiv:1804.04732, Apr. 2018. arXiv: 1804.04732 [cs.CV].

[91] T. Karras, S. Laine, and T. Aila, «A Style-Based Generator Architecture for Generative Adversarial Networks», *arXiv e-prints*, arXiv:1812.04948, arXiv:1812.04948, Dec. 2018. arXiv: 1812.04948 [cs.NE].

[92] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler, «Deep Fluids: A Generative Network for Parameterized Fluid Simulations», *arXiv e-prints*, arXiv:1806.02071, arXiv:1806.02071, Jun. 2018. arXiv: 1806.02071 [cs.LG].

[93] A. Kunz, *Leveraging Deep Learning to Fix Images*, Feb. 2018. [Online]. Available: https://research.adobe.com/news/leveraging-deep-learning-to-fix-images/.

[94] F.-F. Li, J. Johnson, and S. Yeung, *CS231n: Convolutional Neural Networks for Visual Recognition*, 2018. [Online]. Available: http://cs231n.stanford.edu/.

[95] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, «ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing», *arXiv e-prints*, arXiv:1803.01837, arXiv:1803.01837, Mar. 2018. arXiv: 1803.01837 [cs.CV].

[96] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, «Spectral Normalization for Generative Adversarial Networks», *arXiv e-prints*, arXiv:1802.05957, arXiv:1802.05957, Feb. 2018. arXiv: 1802.05957 [cs.LG].

[97] K. Nazeri, E. Ng, and M. Ebrahimi, «Image Colorization with Generative Adversarial Networks», *arXiv e-prints*, arXiv:1803.05400, arXiv:1803.05400, Mar. 2018. arXiv: 1803.05400 [cs.CV].

[98] L. Weng, «Flow-based Deep Generative Models», *lilianweng.github.io/lil-log*, 2018. [Online]. Available: http://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html.

[99] 2019. [Online]. Available: https://thispersondoesnotexist.com/.

[100] S. Ermon and A. Grover, *CS236 - Deep Generative Models*, 2019. [Online]. Available: https://deepgenerativemodels.github.io/.

[101] M. Hu and J. Li, «Exploring Bias in GAN-based Data Augmentation for Small Samples», *arXiv e-prints*, arXiv:1905.08495, arXiv:1905.08495, May 2019. arXiv: 1905.08495 [cs.LG].

[102] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, «Analyzing and Improving the Image Quality of StyleGAN», *arXiv e-prints*, arXiv:1912.04958, arXiv:1912.04958, Dec. 2019. arXiv: 1912.04958 [cs.CV].

[103] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila, «Improved Precision and Recall Metric for Assessing Generative Models», *arXiv e-prints*, arXiv:1904.06991, arXiv:1904.06991, Apr. 2019. arXiv: 1904.06991 [stat.ML].

[104] J. Langr and V. Bok, *GANs in action: deep learning with generative adversarial networks*. Manning Publications, 2019.

[105] T. T. Nguyen, Q. V. H. Nguyen, C. M. Nguyen, D. Nguyen, D. Thanh Nguyen, and S. Nahavandi, «Deep Learning for Deepfakes Creation and Detection: A Survey», *arXiv e-prints*, arXiv:1909.11573, arXiv:1909.11573, Sep. 2019. arXiv: 1909.11573 [cs.CV].

[106] F. Noé, S. Olsson, J. Köhler, and H. Wu, «Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning», *Science*, vol. 365, no. 6457, aaw1147, aaw1147, Sep. 2019. DOI: 10.1126/science.aaw1147. arXiv: 1812.01729 [stat.ML].

[107] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, «Semantic Image Synthesis with Spatially-Adaptive Normalization», *arXiv e-prints*, arXiv:1903.07291, arXiv:1903.07291, Mar. 2019. arXiv: 1903.07291 [cs.CV].

[108] Y. Shen, J. Gu, X. Tang, and B. Zhou, «Interpreting the Latent Space of GANs for Semantic Face Editing», *arXiv e-prints*, arXiv:1907.10786, arXiv:1907.10786, Jul. 2019. arXiv: 1907.10786 [cs.CV].

[109] A. Burnes, *NVIDIA DLSS 2.0: A Big Leap In AI Rendering*, Mar. 2020. [Online]. Available: https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/.

[110] D. Chu, I. Demir, K. Eichensehr, J. G. Foster, M. Green, K. Lerman, F. Menczer, C. O'Connor, E. Parson, L. Ruthotto, A. Sahai, J. Sotelo, and L. Venturi, «White Paper : DEEP FAKERY — An Action Plan», 2020.

[111] M. T. García-Ordás, J. A. Benítez-Andrades, I. García-Rodríguez, C. Benavides, and H. Alaiz-Moretón, «Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational Autoencoders for Unbalancing Data», *Sensors*, vol. 20, no. 4, 2020, ISSN: 1424-8220. DOI: 10.3390/s20041214. [Online]. Available: https://www.mdpi.com/1424-8220/20/4/1214.

[112] S. Jandial, A. Chopra, K. Ayush, M. Hemani, A. Kumar, and B. Krishnamurthy, «SieveNet: A Unified Framework for Robust Image-Based Virtual Try-On», *arXiv e-prints*, arXiv:2001.06265, arXiv:2001.06265, Jan. 2020. arXiv: 2001.06265 [cs.CV].

[113] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, «PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models», *arXiv e-prints*, arXiv:2003.03808, arXiv:2003.03808, Mar. 2020. arXiv: 2003.03808 [cs.CV].

[114] P. Salehi, A. Chalechale, and M. Taghizadeh, «Generative Adversarial Networks (GANs): An Overview of Theoretical Model, Evaluation Metrics, and Recent Developments», *arXiv e-prints*, arXiv:2005.13178, arXiv:2005.13178, May 2020. arXiv: 2005.13178 [cs.CV].

[115] S. Shen, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer, «PowerNorm: Rethinking Batch Normalization in Transformers», *arXiv e-prints*, arXiv:2003.07845, arXiv:2003.07845, Mar. 2020. arXiv: 2003.07845 [cs.CL].

[116] S. Zhang, D. Cheng, D. Jiang, and Q. Kou, «Least Squares Relativistic Generative Adversarial Network for Perceptual Super-Resolution Imaging», *IEEE Access*, vol. 8, pp. 185 198–185 208, 2020. DOI: 10.1109/ACCESS.2020.3030044.

[117] Ψαρράκος, Παναγιώτης Ι., *Θέματα Ανάλυσης Πινάκων*, Jun. 2020. [Online]. Available: http://www.math.ntua.gr/~ppsarr/Topics_in_Matrix_Analysis.pdf.

[118] AMD, Apr. 2021. [Online]. Available: https://www.amd.com/en/technologies/radeon-software-fidelityfx.

[119] *Fréchet distance*, May 2021. [Online]. Available: https://en.wikipedia.org/wiki/Fr%5C%C3%5C%A9chet_distance.

[120] A. Ramesh, J. Jay Wang, and G. Goh, *DALL·E: Creating Images from Text*, Jan. 2021. [Online]. Available: https://openai.com/blog/dall-e/.

[121] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, «Zero-Shot Text-to-Image Generation», *arXiv e-prints*, arXiv:2102.12092, arXiv:2102.12092, Feb. 2021. arXiv: 2102.12092 [cs.CV].

[122] S. Zhou, E. Zelikman, and E. Zhou, *Generative Adversarial Networks (GANs) Specialization*, Feb. 2021. [Online]. Available: https://www.deeplearning.ai/program/generative-adversarial-networks-gans-specialization/.

[123] Amazon. [Online]. Available: https://www.mturk.com/.

[124] A. Amidi and S. Amidi, *Supervised Learning cheatsheet*. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-supervised-learning.

[125] *Common Problems | Generative Adversarial Networks | Google Developers*. [Online]. Available: https://developers.google.com/machine-learning/gan/problems.