Τεχνικές Βαθιάς Μηχανικής Μάθησης για Αναγνώριση Μουσικού Συναισθήματος

Από

Άγγελο Γερουλάνο

Υποβάλλεται

για την εκπλήρωση των προϋποθέσεων λήψης

Μεταπτυχιακού Διπλώματος

στην «Τεχνητή Νοημοσύνη»

στο

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Ιούνιος 2021

Συγγραφέας .....Άγγελος Γερουλάνος.....

ΔΠΜΣ «Τεχνητή Νοημοσύνη»

Ιούνιος 30, 2021

Έγινε αποδεκτό από .....

Θεόδωρος Γιαννακόπουλος
Ερευνητής Β'

Έγινε αποδεκτό από.....

Ηλίας Μαγκλογιάννης
Καθηγητής
Μέλος Εξεταστικής Επιτροπής

Έγινε αποδεκτό από.....

Γεώργιος Βούρος
Καθηγητής
Μέλος Εξεταστικής Επιτροπής

Τεχνικές Βαθιάς Μηχανικής Μάθησης για Αναγνώριση Μουσικού Συναισθήματος

Από

Άγγελο Γερουλάνο

Υποβλήθηκε στο ΔΠΜΣ «Τεχνητή Νοημοσύνη» την 30 Ιουνίου 2021 ως υποχρέωση για την λήψη Μεταπτυχιακού Διπλώματος Σπουδών

Περίληψη

Η μουσική είναι φορέας πολλών και ισχυρών συναισθημάτων. Με την ανάπτυξη της τεχνολογίας και του διαδικτύου η πρόσβαση σε τεράστιου όγκου μουσικό περιεχόμενο είναι άμεση σχεδόν από οπουδήποτε. Παρόλη τη διαθεσιμότητα, η επιλογή μουσικής βάσει συναισθηματικής κατάστασης του ακροατή είναι αρκετά δύσκολη υπόθεση.
Η παρούσα εργασία διερευνά μέσω τεχνικών βαθιάς μηχανικής μάθησης την ικανότητα γνωστών αρχιτεκτονικών CNNs (VGG, AlexNet, DenseNet, Inception, ResNeXt, SqueezeNet) στην αναγνώριση μουσικού συναισθήματος σε συνθήκες έλλειψης δεδομένων, με σετ διαφορετικής προέλευσης και όχι πάντοτε ισορροπημένων. Οι τεχνικές που χρησιμοποιούνται είναι η Μεταφορά Μάθησης και η επαύξηση δεδομένων μέσω Παραγωγικών Ανταγωνιστικών Δικτύων (GANs).
Πριν από αυτό όμως, με κλασική μηχανική μάθηση πραγματοποιείται εξαγωγή χειροποίητων χαρακτηριστικών όλων των ηχητικών δειγμάτων και ταξινόμηση με γνωστούς ταξινομητές (SVM, K-NN, Random Forest, Extra Trees) προκειμένου να υπάρχει σημείο αναφοράς για τα συγκεντρωτικά αποτελέσματα.
Έτσι, τα δείγματα μετατρέπονται σε Mel-spectrograms για να γίνουν είσοδοι στα συνελικτικά δίκτυα τα οποία εκπαιδεύονται με δύο σενάρια Μεταφοράς Μάθησης και δίνουν μοντέλα που δοκιμάζονται σε πειράματα ταξινόμησης συναισθημάτων. Τέλος, με χρήση του StyleGAN2-ADA γίνεται επαύξηση δεδομένων και δημιουργείται ένα νέο τεχνητό σετ που και αυτό με τη σειρά του δοκιμάζεται σε ταξινομήσεις.
Σημείο αναφοράς των πειραμάτων είναι το 360-set της έρευνας των Eerola & Vuoskoski πλήρως ετικετοποιημένο από ειδικούς στον τομέα της μουσικής, γεγονός που το καθιστά αρκετά σπάνιο. Αποτελείται από 360 αποσπάσματα κινηματογραφικής μουσικής διάρκειας 15”-30”, διαχωρισμένα σε Energy (high, medium, low), Valence (positive, neutral, negative), Tension (high, medium, low) και Emotions (anger, fear, happy, sad, tender). Από όσο μπορούμε να γνωρίζουμε η παρούσα είναι η πρώτη εργασία που πραγματοποιεί τόσο εκτεταμένα πειράματα στο συγκεκριμένο σετ.

Λέξεις – κλειδιά: Μεταφορά Μάθησης, Επαύξηση Δεδομένων, StyleGAN2-ADA, Συνελικτικά Νευρωνικά Δίκτυα, Ταξινόμηση Συναισθήματος

Επιβλέπων: Θεόδωρος Γιαννακόπουλος
Ακαδημαϊκή Θέση: Ερευνητής Β’

Deep Learning for Music Emotion Recognition

by

Angelos Geroulanos

Submitted to the II-MSc "Artificial Intelligence" on June 30, 2021, in partial fulfillment of the requirements for the MSc degree

Abstract

Music is a carrier of many powerful emotions. With the growth of technology and internet, huge amounts of music content can be accessed instantly from almost anywhere. Despite the availability, music selection based on the listener's emotional state is quite a difficult task.
This work investigates through deep learning techniques the ability of well-known CNN architectures (VGG, AlexNet, DenseNet, Inception, ResNeXt, SqueezeNet) in music emotion recognition under scarce data conditions, with diverse and not always balanced sets. The techniques used are Transfer Learning and data augmentation via Generative Adversarial Networks (GANs).
But before that, traditional machine learning is used to extract hand-crafted features of all audio samples and classify them using well-known classifiers (SVM, K-NN, Random Forest, Extra Trees) in order to have a reference point for the aggregated results.
Thus, the samples are converted into Mel-spectrograms as inputs to the convolutional networks which are trained with two transfer learning scenarios and yield models that are tested in emotion classification experiments. Finally, data augmentation is performed using StyleGAN2-ADA and a new artificial set is created which in turn is tested in classification experiments.
The ground truth of these experiments is the 360-set of Eerola & Vuoskoski's research fully tagged by experts in the music field, a fact that makes it quite rare. It consists of 360 excerpts of movie music with a duration of 15"-30"', divided into Energy (high, medium, low), Valence (positive, neutral, negative), Tension (high, medium, low) and Emotions (anger, fear, happy, sad, tender). To our knowledge, this is the first study to conduct such extensive experiments on this set.

Keywords: Transfer Learning, Augmentation, Emotions, StyleGAN2-ADA, CNN

Supervisor: Theodoros Giannakopoulos
Title: Researcher B'

Ευχαριστίες

Η παρούσα Διπλωματική εργασία είναι το επιστέγασμα της φοίτησής μου στο ΔΠΜΣ «Τεχνητή Νοημοσύνη» κατά τη διάρκεια της οποίας ήρθα σε επαφή τόσο με τα αντικείμενα του Προγράμματος όσο και με πολύ ενδιαφέροντες ανθρώπους, Διδάσκοντες και Συναδέλφους.

Ευχαριστώ πολύ όλους τους Διδάσκοντες για το υψηλό επίπεδο των διαλέξεών τους αλλά ιδιαιτέρως τους Καθηγητές κκ. Γεώργιο Βούρο, Ηλία Μαγκλογιάννη για την τιμή που μου έκαναν να είναι τα Μέλη της Εξεταστικής Επιτροπής.

Ο κ. Θεόδωρος Γιαννακόπουλος, υπό την άριστη επίβλεψη του οποίου εργάστηκα, διατήρησε ένα ανοικτό κανάλι άμεσης επικοινωνίας καθόλη τη διάρκεια της εκπόνησης της Διπλωματικής μέσα από το οποίο επιλύαμε προβλήματα και... δημιουργούσαμε νέα! Με τον τρόπο του, βοήθησε να διευρύνω τις γνώσεις μου και κυρίως να γνωρίσω ενεργητικά την ερευνητική μεθοδολογία. Οι ευχαριστίες προς το πρόσωπό του δεν μπορούν να εκφραστούν σε λίγες γραμμές.

Στους γονείς μου, Κατερίνα & Γεράσιμο

Περιεχόμενα

Λίστα Εικόνων/ List of Figures

Λίστα Πινάκων/ List of Tables

1 Εισαγωγή

Η παρούσα εργασία χρησιμοποιεί τεχνικές βαθιάς μηχανικής μάθησης προκειμένου να αναγνωρίσει το μουσικό συναίσθημα και να ταξινομήσει αναλόγως τα μουσικά αποσπάσματα που περιέχονται σε δύο σετ ηχητικών δεδομένων. Το ένα σετ περιλαμβάνει 17000 αποσπάσματα τραγουδιών ποπ-ροκ χωρισμένα σε κλάσεις χαρακτηριστικών όπως τα ταξινόμησε το API του Spotify και το ονομάσαμε “big-set”. Το άλλο σετ που είναι και το σημείο αναφοράς της εργασίας μας καθώς είναι ετικετοποιημένο από ειδικούς του χώρου της μουσικής στα πλαίσια μεγάλης ψυχολογικής - μουσικολογικής έρευνας των Eerola & Vuoskoski. Περιλαμβάνει 360 αποσπάσματα κινηματογραφικής μουσικής χωρισμένα σε χαρακτηριστικά τριών και πέντε κλάσεων και το ονομάσαμε “360-set”. Κάναμε εξαγωγή «χειροποίητων» χαρακτηριστικών από τα ηχητικά τους σήματα και τα ταξινομήσαμε με κλασικούς ταξινομητές όπως SVM, Knn, Random Forest, Extra Trees. Στη συνέχεια μετατρέψαμε όλα τα δείγματα και των δύο σετ σε φασματογραφήματα της κλίμακας Mel προκειμένου να γίνουν είσοδοι στα βαθιά συνελικτικά δίκτυα των δοκιμών. Έξι αρχιτεκτονικές (AlexNet, VGG16_bn, Inception v3, DenseNet121, SqueezeNet 1.0, ResNeXt101-32x8d) με ισάριθμα προεκπαιδευμένα στο ImageNet μοντέλα χρησιμοποιήθηκαν στον 1ο κύκλο πειραμάτων για ταξινομήσεις μέσω δύο σεναρίων Μεταφοράς Μάθησης: στο 1ο σενάριο «παγώνουμε» τα βάρη των επιπέδων του δικτύου πλην του επιπέδου του ταξινομητή και στο 2ο σενάριο γίνεται μικρο-ρύθμιση του δικτύου και ανανέωση των βαρών όλων των επιπέδων του.

Ακολουθεί 2ος κύκλος πειραμάτων όπου προεκπαιδεύουμε δικά μας μοντέλα πάνω στο “big-set” και πραγματοποιούμε επαναληπτικές δοκιμασίες ταξινόμησης με τις ίδιες αρχιτεκτονικές. Τέλος, στο 3ο πείραμα δημιουργούμε τεχνητά δεδομένα προς επαύξηση του “360-set”με χρήση Παραγωγικών Ανταγωνιστικών Δικτύων και συγκεκριμένα του StyleGAN2-ADA. Από το νέο σετ που προκύπτει προεκπαιδεύουμε εκ νέου μοντέλα και τα δοκιμάζουμε σε σειρά ταξινομήσεων των συναισθημάτων. Την απόδοση όλων των παραπάνω κύκλων πειραμάτων την αποτιμούμε με το macro-avg-f1-score που έδωσαν οι ταξινομήσεις πάνω στο test-set του σετ αναφοράς.

1.1 Περιγραφή του προβλήματος

Στις μέρες μας η πρόσβαση σε μουσικό περιεχόμενο Διαδικτυακά είναι εξαιρετικά εύκολη. Επίσης, ο όγκος του περιεχομένου αυτού αυξάνεται εκθετικά και είναι άμεσα διαθέσιμος σε όλους. Διάφορες υπηρεσίες διάθεσης ψηφιακής μουσικής (πχ Spotify, Apple Music, κ.α) είναι διαθέσιμες στους υπολογιστές μας, στα κινητά τηλέφωνα, στα «έξυπνα» ηχεία και αλλού. Ωστόσο, όσο και αν η ροή της μουσικής είναι συνεχής, πολλές φορές δεν ταιριάζει με την τρέχουσα συναισθηματική κατάσταση του ακροατή. Σε τέτοιες περιπτώσεις η αναπαραγωγή κάποιας προτεινόμενης λίστας βάσει μουσικού είδους δεν είναι η καλύτερη επιλογή καθώς οι διακυμάνσεις των προκαλούμενων συναισθημάτων που λαμβάνει ο ακροατής μπορεί να είναι πολύ μεγάλες ακόμα και στα κομμάτια του ίδιου είδους ή και του ίδιου συνθέτη. Το βασικό πρόβλημα, πριν ακόμα φτάσουμε να μιλάμε για αλγόριθμους, είναι ότι τα συναισθήματα παρουσιάζουν μεγάλη δυσκολία στην κατάταξή τους λόγω έλλειψης οικουμενικών ορισμών τους. Ενδεικτικά, σε μία έρευνά τους οι Paul & Anne Kleinginna [1] αξιολόγησαν πάνω από 100 ορισμούς του συναισθήματος και ενώ υπήρχαν καλοί ή λιγότερο καλοί ορισμοί, στο τέλος η ορθότητά τους δεν μπορούσε να αποδειχτεί.

Αυτή λοιπόν η «συναισθηματική σύγχυση» [2] [3] [4] που επικρατεί σε διάφορες Επιστήμες είναι αναμενόμενο να μεταφερθεί και στη Μηχανική Μάθηση. Για παράδειγμα, η εξαγωγή χειροποίητων χαρακτηριστικών από ηχητικά σήματα και η οδήγηση αυτών σε έναν κλασικό ταξινομητή π.χ. SVM μπορεί να δώσει πολύ καλά αποτελέσματα στον γενικό διαχωρισμό μουσικών ειδών (π.χ. τζαζ, κλασική μουσική, ροκ, ρεμπέτικο, κ.α.). Δεν είναι καθόλου σίγουρο ότι θα επιτύχει εξίσου αν του ζητηθεί να βρει τα αντίστοιχα χαρούμενα, λυπημένα, τρυφερά, θυμωμένα κλπ κομμάτια από τα ίδια δείγματα. Γιατί; Διότι απουσιάζει το σωστά ορισμένο, ή αλλιώς, το κοινώς αποδεκτό σημείο αναφοράς (ground truth).

Συγκεκριμενοποιώντας τον παραπάνω γενικό προβληματισμό, σχετικά με την αναγνώριση συναισθήματος, στο πλαίσιο της παρούσας εργασίας δημιουργήσαμε μοντέλα Βαθιάς Μηχανικής Μάθησης, δοκιμάσαμε τεχνικές και τα συγκρίναμε έχοντας ως ground truth ένα μικρό σετ μόλις 360 δειγμάτων κινηματογραφικής μουσικής. Η ιδιαιτερότητα του σημείου αναφοράς μας είναι ότι είναι πλήρως ετικετοποιημένο και ταξινομημένο σε μουσικά συναισθήματα,

όπως αυτά αναγνωρίστηκαν από ειδικούς στον τομέα της Μουσικής στα πλαίσια πειραμάτων Ψυχολογικής - Μουσικολογικής μελέτης των Eerola & Vuoskoski [5].

1.2 Σκελετός Διπλωματικής Εργασίας

Κεφάλαιο 1: Εισαγωγή, περιγραφή του προβλήματος και παράθεση σχετικών εργασιών.

Κεφάλαιο 2: Παρουσιάζεται ο βασικός σκελετός των Συνελικτικών Νευρωνικών Δικτύων, οι συναρτήσεις ενεργοποίησης και οι συναρτήσεις κόστους. Επίσης παρουσιάζονται αρκετοί αλγόριθμοι βελτιστοποίησης και μελετώνται οι αρχιτεκτονικές των ΣΝΔ που χρησιμοποιήθηκαν στα πειράματα της εργασίας.

Κεφάλαιο 3: Εισαγωγή στα Παραγωγικά Ανταγωνιστικά Δίκτυα (GANs) και ανάπτυξη της αρχιτεκτονικής του StyleGAN2-ADA, με το οποίο έγινε η επαύξηση δεδομένων στο Πείραμα Γ.

Κεφάλαιο 4: Περιγράφεται όλη η διαδικασία των Πειραμάτων Α, Β, Γ, Χ ξεκινώντας από την προέλευση και προεπεξεργασία των σετ δεδομένων, τη ρύθμιση των υπερπαραμέτρων και την επιλογή μετρικών εκτίμησης απόδοσης. Στη συνέχεια αναλύονται τα σενάρια της Μεταφοράς Μάθησης που χρησιμοποιήθηκαν και τα συγκεντρωτικά αποτελέσματα ανά κύκλο Πειραμάτων. Τέλος, γίνεται σύνοψη των αποτελεσμάτων και εξαγωγή συμπερασμάτων.

1.3 Σχετικές Εργασίες

1.3.1 Αναγνώριση συναισθήματος (360-set)

Στη μελέτη [6] χρησιμοποιήθηκε ως ground truth το 2ο μικρότερο σετ της μελέτης των Eerola & Vuoskoski που περιέχει μόλις 110 δείγματα και αποτελεί μέρος του 360-σετ. Αναλύοντας τα ηχητικά σήματα, έκαναν εξαγωγή χαρακτηριστικών και με SVM τα ταξινόμησαν. Από τα αποτελέσματα, μεταξύ άλλων, επιβεβαίωσαν τα ευρήματα των ψυχολογικών πειραμάτων όπως για παράδειγμα την υψηλή επικάλυψη μεταξύ των συναισθημάτων anger-fear.

Στη μελέτη [7] με σημείο αναφοράς πάλι το 360-σετ επιχειρούν την εξαγωγή μοντέλου από ένα νευρωνικό στο στυλ της VGG αρχιτεκτονικής που να μπορεί να δικαιολογήσει μουσικά τις προβλέψεις του μέσω των λεγόμενων mid-level perceptual features. Τα χαρακτηριστικά αυτά όπως είναι η ρυθμική πολυπλοκότητα ή ο αρμονικός χαρακτήρας (μείζων-χαρούμενο, ελάσσων-λυπημένο) έχουν μουσικό νόημα και μπορούν να εντοπιστούν από ακροατές χωρίς μουσικές γνώσεις.

Η μελέτη [8] είναι η τελευταία που εντοπίσαμε να χρησιμοποιεί το 360-σετ. Σε αυτή παρουσιάζεται μια νέα μέθοδος αναγνώρισης μουσικού συναισθήματος που εμπλέκει τα chroma-spectrograms με τις αρχιτεκτονικές των VGG16, AlexNet.

1.3.2 Αναγνώριση συναισθήματος

Στο βιβλίο Music Emotion Recognition [9] μεταξύ άλλων αναφέρεται και μια μέθοδος όπου εστιάζει το ενδιαφέρον της στην προσωποποιημένη αναγνώριση μουσικού συναισθήματος. Δηλαδή, αντί να μοντελοποιεί την όποια αντικειμενικότητα, ζητάει από τον χρήστη να ετικετοποιήσει (μέσω user interface) μέρος της μουσικής συλλογής του ώστε ο αλγόριθμος (regression) να εκπαιδευτεί στην προσωπική συλλογή του.

Στο πείραμα της μελέτης [10] οι εθελοντές καθώς άκουγαν 16 τραγούδια που είχαν επιλέξει από συγκεκριμένη βάση τραγουδιών υποβάλλονταν σε εγκεφαλογράφημα. Μόλις τελείωναν, άκουγαν ξανά τα τραγούδια και τα ετικετοποιούσαν βάσει του διαστατικού μοντέλου συναισθημάτων. Τα δεδομένα μαζί με αυτά του εγκεφαλογραφήματος γίνονταν είσοδοι σε Συνελικτικό

Νευρωνικό Δίκτυο (ΣΝΔ) του οποίου η ικανότητα στην αναγνώριση μουσικού συναισθήματος (Valence & Arousal) μετρήθηκε συγκρινόμενο με ταξινομητή κλασικής μηχανικής μάθησης, έναν SVM, τον οποίο υπερκέρασε.

Στη μελέτη [11] ερευνήθηκαν ταυτόχρονα δύο προβλήματα: της ομοιότητας κάποιων μουσικών έργων με ένα συγκεκριμένο και της ταξινόμησης μουσικού συναισθήματος. Η αναγνώριση συναισθήματος αν και πρόβλημα πολλών κλάσεων, εδώ, αποδομήθηκε σε πρόβλημα πολλαπλών δυαδικών ταξινομήσεων που αντιμετωπίστηκε με SVM ταξινομητή εκπαιδευμένο από τα χειροποίητα χαρακτηριστικά (handcrafted features) που είχαν προηγουμένως εξαχθεί από τα ηχητικά δείγματα.

Στη μελέτη [12] για την αναγνώριση μουσικού συναισθήματος χρησιμοποιήθηκε ΣΝΔ με είσοδο τα φασματογραφήματα των προς ταξινόμηση μουσικών αποσπασμάτων που είχαν εξαχθεί προηγουμένως με αρκετά καλά αποτελέσματα στα δύο σετ που δοκιμάστηκε.

Στη μελέτη [13] σε ένα σετ 1000 τραγουδιών, ετικετοποιημένων με τα χαρακτηριστικά του Σθένος (Valence) και της Διέγερσης (Arousal), πραγματοποιήθηκαν ταξινομήσεις με ΣΝΔ που είχε ως είσοδο τα φασματογραφήματα των τραγουδιών.

1.3.3 Μεταφορά Μάθησης

Στη μελέτη [14], χρησιμοποιείται προεκπαιδευμένο στο ImageNet [15] μοντέλο ΣΝΔ προκειμένου να δοκιμαστεί σε ταξινόμηση μουσικής. Αν και το ImageNet είναι «ξένο» σετ ως προς τα φασματογραφήματα των τραγουδιών, η μελέτη έδειξε ότι η Μεταφορά Μάθησης λειτουργεί σε αυτή την περίπτωση.

Στη μελέτη [16] προτάθηκε ένα προεκπαιδευμένο χαρακτηριστικό ενός συνελικτικού δικτύου (pre-trained convnet feature) όπως ονομάστηκε. Πρόκειται για ένα συνδυαστικό διάνυσμα χαρακτηριστικών από τις ενεργοποιήσεις των χαρτών χαρακτηριστικών (feature maps) των πολλαπλών επιπέδων ενός εκπαιδευμένου συνελικτικού και χρησιμοποιείται ως προεκπαιδευμένο μοντέλο για δοκιμές ταξινόμησης (διαχωρισμού λόγου-μουσικής, πρόβλεψη μουσικού συναισθήματος, διαχωρισμού είδους μουσικής) ή παλινδρόμησης σε άλλα παρόμοια σετ.

Στη μελέτη [17] αναλύονται προεκπαιδευμένα μοντέλα που χρησιμοποιούνται για την αναγνώριση χειρόγραφων αλφάβητων Devanagari με χρήση μεταφοράς μάθησης από ΣΝΔ και συγκεκριμένα των AlexNet, DenseNet, VGG, Inception όπου χρησιμοποιήθηκαν ως εξαγωγείς χαρακτηριστικών (feature extractors). Αποδοτικότερο (σε accuracy) εμφανίστηκε το Inception v3 με το AlexNet εκπαιδεύεται γρηγορότερα και με 1% χαμηλότερο accuracy.

Στη μελέτη [18] η μεταφορά μάθησης χρησιμοποιήθηκε για την ταξινόμηση συναισθήματος μέσα από φωτογραφίες ανθρώπων που εξάχθηκαν από κινηματογραφικές ταινίες. Χρησιμοποιήθηκαν μοντέλα βαθιών συνελικτικών δικτύων προεκπαιδευμένων στο ImageNet με την τεχνική του fine-tuning.

Στη μελέτη [19] προκειμένου να αναγνωριστούν αντικείμενα που περνούν από τον έλεγχο αποσκευών με ακτίνες Χ, χρησιμοποιήθηκε ΣΝΔ προεκπαιδευμένο σε μεγάλο και γενικότερο σετ εικόνων ως αντιστάθμισμα της έλλειψης δεδομένων (εικόνων μέσα από ακτίνες Χ). Το αποτέλεσμα της μεταφοράς μάθησης ήταν να αναγνωρίζεται ένα πιστόλι σε αποσκευή με ορθότητα 98,92%.

1.3.4 Επαύξηση δεδομένων (μέσω GANs)

Στη μελέτη [20] αντιμετωπίστηκε το πρόβλημα έλλειψης δεδομένων, σε πρόβλημα αναγνώρισης συναισθήματος σε ομιλία, με τη χρήση Παραγωγικών Ανταγωνιστικών Δικτύων (GANs). Πιο συγκεκριμένα, βελτίωσαν μια αρχιτεκτονική conditional GAN ώστε να παράγει φασματογραφήματα για την κλάση που μειονεκτούσε. Τα αποτελέσματα σε δύο μεγάλα σετ δεδομένων έδειξαν βελτίωση από 5%-10% όταν χρησιμοποιήθηκε αυτή η μέθοδος.

Στη μελέτη [21] προτείνεται η χρήση CycleGAN ως γεννήτορα εικόνων της κλάσης που υστερεί σε δείγματα, προκειμένου να δοκιμαστεί σε ταξινόμηση (πολλών κλάσεων) συναισθημάτων με ικανοποιητική παρουσία δειγμάτων. Ως ταξινομητής χρησιμοποιήθηκε ένα ΣΝΔ. Επιβεβαιώθηκε αύξηση 5%-10% της απόδοσης της ταξινόμησης όταν χρησιμοποιήθηκε η επαύξηση με το CycleGAN.

Στη μελέτη [22] βελτιώθηκε ένα Παραγωγικό Ανταγωνιστικό Δίκτυο (GAN) προκειμένου να μάθει τα χαρακτηριστικά βλαβών του δέρματος σε διαφορετικά επίπεδα πολυπλοκότητας, ελεγχόμενα. Στη συνέχεια οι τεχνητές εικόνες που

παρήγαγε το GAN εμπλούτισαν το σετ εκπαίδευσης του Πλήρως Συνελικτικού Δικτύου (FCN), για την αυτοματοποιημένη διάγνωση μελανώματος, αυξάνοντας την ποικιλομορφία των χαρακτηριστικών. Η μέθοδος αυτή αξιολογήθηκε στο ISIC 2018 skin lesion segmentation challenge και έδειξε μεγαλύτερη ακρίβεια συγκρινόμενη με τις τρέχουσες μεθόδους τμηματοποίησης δερματικών βλαβών.

2 Από τον ήχο, στη Μηχανική Μάθηση

Όπως είδαμε και στο Εισαγωγικό Κεφάλαιο, ο όγκος της μουσικής που διακινείται μέσω διαδικτύου είναι τεράστιος και πια δεν αρκεί μια απλή αναζήτηση (τίτλου, συνθέτη, χρονολογίας κυκλοφορίας, κ.α.) για να ικανοποιήσει τις ανάγκες ακόμα και ενός απλού, χωρίς πολλές απαιτήσεις, ακροατή. Έτσι, οι εφαρμογές αναπαραγωγής ψηφιακής μουσικής έχουν αρχίσει να προσφέρουν υπηρεσίες βάσει περιεχομένου δηλαδή, μέσω της ανάσυρσης μουσικής πληροφορίας [23] (MIR) αποσπούν ωφέλιμα μοτίβα από τις ηχογραφήσεις τα οποία είναι κατανοητά, χρήσιμα στον ακροατή ώστε να τον βοηθήσουν να επιλέξει ποια μουσική θα ακούσει αναλόγως του είδους, του χαρακτήρα, της ρυθμικότητας και πολλών άλλων χαρακτηριστικών. Ακόμα, μέσω της αναγνώρισης μουσικού συναισθήματος [24], [9] (MER) μπορεί ο ακροατής να επιλέξει ή να δημιουργήσει μια λίστα αναπαραγωγής βάσει συναισθηματικών αποχρώσεων (λύπης, χαράς, θυμού, τρυφερότητας, έντασης, διέγερσης, σθένους, κ.α.). Αφού λοιπόν γίνει εξαγωγή των χαρακτηριστικών με μεθόδους ανάλυσης του ηχητικού σήματος έρχεται η στιγμή της Μηχανικής Μάθησης, όπου οι αλγόριθμοί της (ταξινομητές) εκπαιδεύονται από αυτά και μπορούν να κατατάξουν αναλόγως τα κομμάτια μιας μεγάλης μουσικής βιβλιοθήκης.

2.1 Εξαγωγή ηχητικών χαρακτηριστικών

Προκειμένου από ένα ηχητικό σήμα να εξαχθούν τα χαρακτηριστικά του πρέπει πρώτα από συνεχές (αναλογικό) να μετατραπεί σε διακριτό (ψηφιακό). Έτσι, το αναλογικό σήμα κβαντίζεται (παίρνουμε διακριτές τιμές του πλάτους του) και δειγματοληπτείται (παίρνουμε διακριτές τιμές του στον χρόνο) και είναι έτοιμο για περαιτέρω επεξεργασία. Καθώς ο τομέας της εξαγωγής χαρακτηριστικών (Ε.Χ) δεν είναι ένα καινούριο πεδίο έρευνας, είναι πέραν των σκοπών της παρούσας μελέτης η αναφορά όλων διότι ο αριθμός τους είναι μεγάλος. Θα περιοριστούμε κυρίως σε αυτά που χρησιμοποιήσαμε σαν εισόδους στους ταξινομητές του *Πείραμα Χ*.

2.1.1 Κατηγορίες ηχητικών χαρακτηριστικών

Τα ηχητικά χαρακτηριστικά κατατάσσονται σε τρεις κύριες κατηγορίες [23] ως εξής:

- *Χαμηλού επιπέδου:* Τα χαμηλού επιπέδου χαρακτηριστικά μπορούν να εξαχθούν είτε άμεσα από το ηχητικό σήμα είτε μέσω μετασχηματισμού του π.χ. μέσω μετασχηματισμού Fourier. Δεν έχουν ιδιαίτερο νόημα σε επίπεδο ακροατή όμως η εξαγωγή τους είναι εύκολη και χρησιμοποιούνται ευρέως.
- *Μεσαίου επιπέδου:* Αποτελούνται από χαρακτηριστικά που η αναπαράστασή τους έχει περισσότερο μουσικό νόημα από αυτά του χαμηλού επιπέδου. Τέτοιες αναπαραστάσεις αφορούν στη μελωδική, αρμονική, ηχοχρωματική πλευρά της μουσικής.
- *Υψηλού επιπέδου:* Αυτή η κατηγορία αναφέρεται σε μουσικά χαρακτηριστικά που δεν παράγονται απευθείας από το ηχητικό-μουσικό σήμα. Αφορά περισσότερο σε συμβολικές αναπαραστάσεις της μουσικής όπως είναι η παρτιτούρα που περιγράφει με νότες όλα τα μέρη μιας μουσικής σύνθεσης. Μια άλλη αναπαράσταση είναι αυτή που παράγεται από το MIDI (Musical Instrument Digital Interface) πρωτόκολλο.

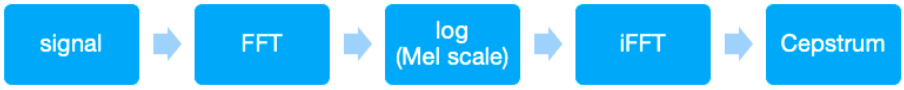
Θα μας απασχολήσουν τα χαρακτηριστικά του χαμηλού επιπέδου.

2.1.2 Χαρακτηριστικά χαμηλού επιπέδου

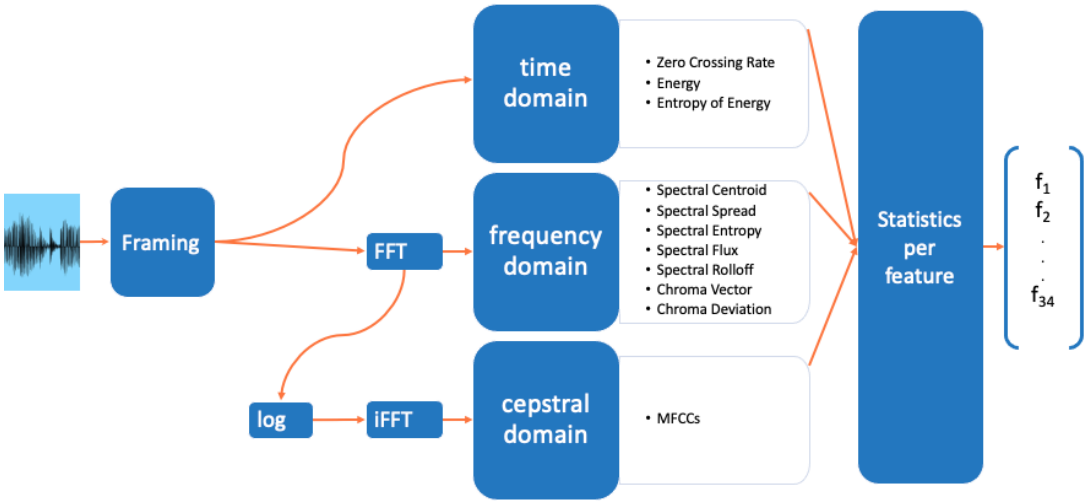
Αναφέρθηκε και προηγουμένως ότι τα χαρακτηριστικά χαμηλού επιπέδου [25], [26] εξάγονται απευθείας από το ηχητικό σήμα και είναι αυτά που μας απασχόλησαν στην εργασία.

Η διαδικασία της εξαγωγής (Εικόνα 2) αρχίζει με τη λεγόμενη παραθύρωση σε μικρής διάρκειας παράθυρα του αρχικού σήματος (short-term windowing). Το μήκος κάθε τέτοιου παραθύρου (frame) κυμαίνεται από 10ms-100ms αναλόγως εφαρμογής και τύπου σήματος. Τα frames μπορούν και να επικαλύπτονται κάποιες φορές. Στη συνέχεια και για κάθε τέτοιο frame εξάγουμε ένα σετ χαρακτηριστικών. Όταν τα χαρακτηριστικά προκύπτουν απευθείας από το σήμα λέμε ότι αφορούν στο πεδίο του χρόνου. Όταν προκύπτουν μετά από μετασχηματισμό Fourier (Fast Fourier Transform) τότε αφορούν στο πεδίο των συχνοτήτων. Τέλος, τα λεγόμενα cepstral χαρακτηριστικά όπως για παράδειγμα

τα Mel Frequency Cepstral Coefficients (MFCCs) προκύπτουν από το cepstrum (αναγραμματισμός του Spectrum και ορίζεται ως ανεστραμμένος μετ/σμός Fourier του λογάριθμου του spectrum). Εικόνα 1



Εικόνα 1: Από το σήμα, στο cepstrum



Εικόνα 2: Short-term windowing και εξαγωγή χαρακτηριστικών με την pyAudioAnalysis [27]

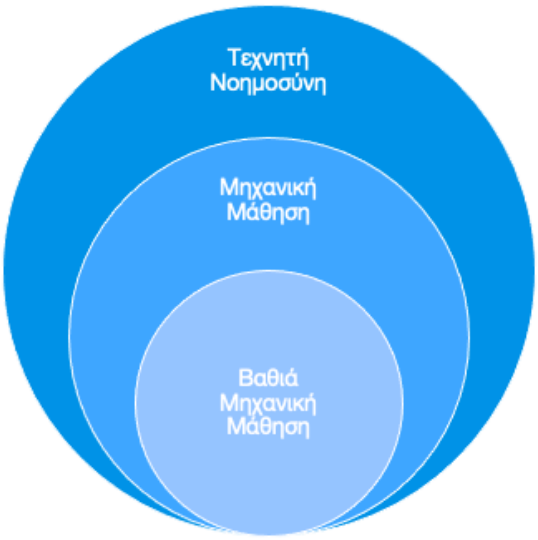
2.2 Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι ένας κλάδος της Τεχνητής Νοημοσύνης (Εικόνα 3) που επιτρέπει στα υπολογιστικά συστήματα να μαθαίνουν από παραδείγματα και να βελτιώνονται από την εμπειρία που αποκτούν χωρίς να έχουν προηγουμένως προγραμματιστεί [28] . Τρεις είναι οι γενικές κατηγορίες που κατατάσσονται τα έργα της Μηχανικής Μάθησης [29], [30]:

- Επιβλεπόμενη μάθηση (Supervised learning): Στα προβλήματα επιβλεπόμενης μάθησης υπάρχουν τα δεδομένα και οι ετικέτες τους. Τα δεδομένα χρησιμοποιούνται ως υλικό εκπαίδευσης στο μοντέλο το οποίο μέσω της συνάρτησής του προσπαθεί να τα ταξινομήσει όσο το δυνατόν πιο κοντά με το σημείο αναφοράς (ground truth). Τελικός σκοπός είναι να μπορεί να χρησιμοποιήσει αυτή τη γνώση και σε άγνωστα δεδομένα.

Υπάρχουν δύο κύριες κατηγορίες επιβλεπόμενης μάθησης: *η ταξινόμηση (classification)*: στην οποία οι ετικέτες έχουν κατηγορική μορφή, δηλαδή το κάθε δεδομένο κατατάσσεται σε συγκεκριμένη κατηγορία και η *παλινδρόμηση (regression):* όπου η έξοδος είναι μια συνεχής μεταβλητή. Μερικοί γνωστοί αλγόριθμοι επιβλεπόμενης μάθησης είναι οι : SVM, K-nn, Naïve Bayes, Neural Networks, Decision Trees, Logistic regression, Random Forest, Linear Regression κ.α.

- Μη επιβλεπόμενη μάθηση (Unsupervised learning): Στη μη επιβλεπόμενη μάθηση δεν υπάρχουν ετικέτες στα δεδομένα επομένως δεν περιμένουμε κάποια πρόβλεψη από το μοντέλο. Αυτό που περιμένουμε είναι να εντοπίσει κάποιες κρυμμένες συσχετίσεις μεταξύ των δεδομένων εισόδου. Γνωστοί αλγόριθμοι μη επιβλεπόμενης μάθησης είναι: k-means clustering, Hierarchical clustering.
- Ενισχυτική μάθηση (Reinforcement learning): Κύρια διαφορά της ενισχυτικής μάθησης από τις άλλες δύο είναι ότι ο αλγόριθμος αλληλεπιδρά με το περιβάλλον του μαθαίνοντας μια στρατηγική ενεργειών προκειμένου να επιτύχει έναν στόχο όπου αναλόγως επιβραβεύεται ή τιμωρείται.

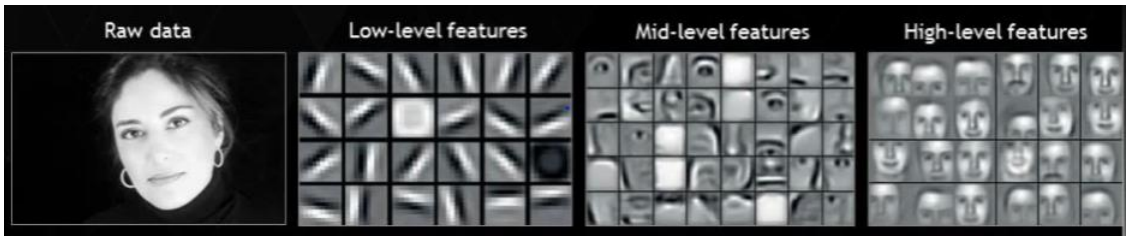


Εικόνα 3: Διάγραμμα Venn της ΤΝ

2.2.1 Βαθιά ή «κλασική» Μηχανική Μάθηση;

Η διαφορά της μηχανικής μάθησης από τη βαθιά μάθηση έγκειται στο γεγονός ότι οι αλγόριθμοι μηχανικής μάθησης καθορίζουν κάποια χαρακτηριστικά μέσα στο σετ δεδομένων. Συνήθως αυτά είναι «χειροποίητα» (handcrafted) επομένως κάποιες φορές αρκετά ευαίσθητα όταν αναπτύσσονται. Για παράδειγμα, [31] θα μπορούσε κάποιος να εξάγει από το ηχητικό σήμα (όπως είδαμε στο 2.1.2) και να χρησιμοποιήσει μερικά MFCCs, ως τέτοια, υποθέτοντας ότι παρέχουν ικανοποιητική πληροφορία για συγκεκριμένη ταξινόμηση και να εκπαιδεύσει έναν ταξινομητή (πχ SVM) ώστε να ετικετοποιήσει ένα σετ δεδομένων βάσει αυτών και μόνο αυτών. Η υπόλοιπη πληροφορία που υπάρχει στο σήμα έμεινε ανεκμετάλλευτη.

Αντίθετα, η βασική ιδέα της βαθιάς μάθησης είναι ότι μαθαίνει όλα τα χαρακτηριστικά με ιεραρχικό τρόπο μέσω των πολλαπλών επιπέδων εκπαίδευσης των νευρωνικών δικτύων απευθείας από τα δεδομένα. Για παράδειγμα, στην Εικόνα 4, το δίκτυο προσπαθεί να αναγνωρίσει ένα πρόσωπο,



Εικόνα 4: Χαμηλού, μέσου, υψηλού επιπέδου χαρακτηριστικά στη Βαθιά Μάθηση

πρώτα θα μάθει τα χαμηλού-επιπέδου χαρακτηριστικά η σύνθεση των οποίων θα οδηγήσει στα μεσαίου-επιπέδου και μετά ακόμα βαθύτερα στα υψηλού-επιπέδου. Μέσω της μη-γραμμικότητας των συναρτήσεων ενεργοποίησης των πολλαπλών επιπέδων του νευρωνικού οι διασυνδέσεις είναι πλήρως «εκπαιδεύσιμες» κάτι που δεν συμβαίνει σε δίκτυο ενός επιπέδου.

3 Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ)

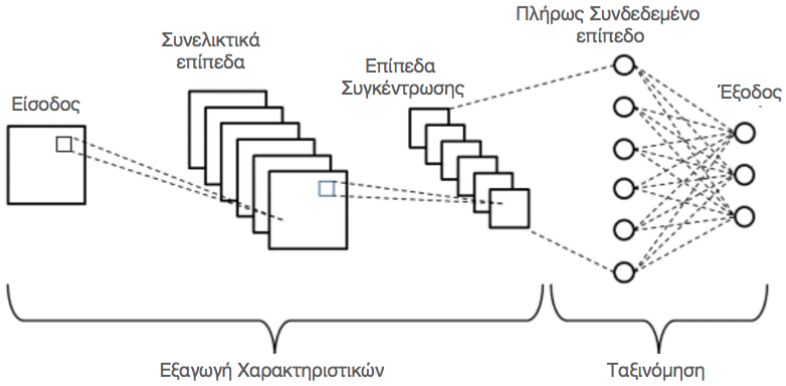
Τα Συνελικτικά Νευρωνικά Δίκτυα (ΣΝΔ) είναι ένας τύπος νευρωνικών δικτύων που απολαμβάνει ευρείας χρήσης και αποδοχής στους τομείς της Υπολογιστικής Όρασης και Επεξεργασίας Εικόνων. Τα ΣΝΔ βρίσκουν εφαρμογή σε έργα Ταξινόμησης Εικόνων, Αναγνώρισης Αντικειμένων, Επεξεργασίας Βίντεο, Φυσικής Επεξεργασίας Γλώσσας, Αναγνώρισης Ομιλίας. Η εξαιρετική ικανότητα εκμάθησης που διαθέτουν οφείλεται κυρίως στην πολλαπλότητα των σταδίων εξαγωγής χαρακτηριστικών τα οποία μαθαίνουν αυτομάτως αναπαραστάσεις από τα δεδομένα των εικόνων.

3.1 Η Αρχιτεκτονική των ΣΝΔ

Η γενική αρχιτεκτονική ενός ΣΝΔ [32], [33] συνδυάζει εναλλασσόμενα επίπεδα συνέλιξης (convolution) και συγκέντρωσης (pooling) ακολουθούμενα από ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα (fully connected ή dense layers). Αναφορά γίνεται και στα ευρέως χρησιμοποιούμενα επίπεδα απόσυρσης (dropout layers).

3.1.1 Συνελικτικό επίπεδο (Convolutional layer)

Το συνελικτικό είναι το πρώτο επίπεδο ενός ΣΝΔ που χρησιμοποιείται για την εξαγωγή των χαρακτηριστικών από την κάθε εικόνα-είσοδο του δικτύου. Η μαθηματική πράξη της συνέλιξης που συμβαίνει μεταξύ των εικονοστοιχείων της εικόνας και ενός φίλτρου παράγει ως αποτέλεσμα μια τροποποιημένη εικόνα με ανανεωμένα εικονοστοιχεία, Εικόνα 5



Εικόνα 5: Γενική Αρχιτεκτονική ΣΝΔ

Φίλτρα(filters) ή πυρήνες (kernels)

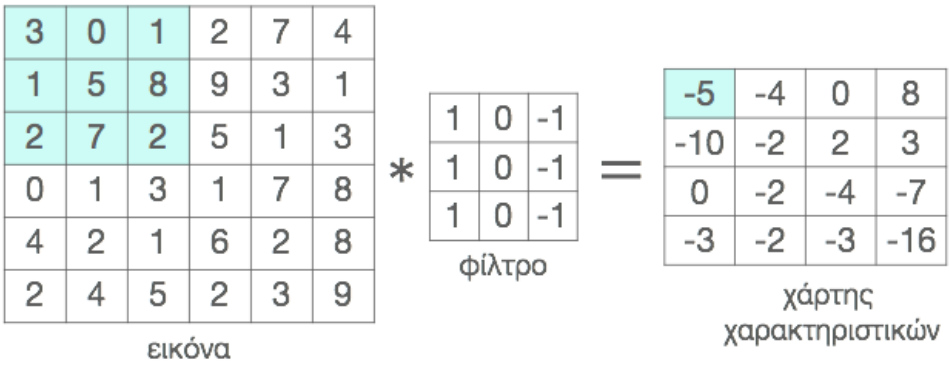
Τα φίλτρα είναι αριθμητικοί πίνακες μικρών συνήθως διαστάσεων και η λειτουργία τους είναι είτε να φιλτράρουν ανεπιθύμητη πληροφορία είτε να ενισχύουν κάποια χαρακτηριστικά μιας εικόνας. Για παράδειγμα η συνέλιξη της ακόλουθης φωτογραφίας με ένα φίλτρο αναγνώρισης ακμών (edge detection) παράγει μια νέα εικόνα, έναν νέο χάρτη χαρακτηριστικών (feature map) όπου τονίζονται οι ακμές του εικονιζόμενου προσώπου, Εικόνα 6

Εικόνα 6: Αποτέλεσμα συνέλιξης με φίλτρο αναγνώρισης ακμών

Συνέλιξη

Βάσει των παραπάνω υποθέτουμε ότι η ακόλουθη Εικόνα 7 απεικονίζει τα εικονοστοιχεία μιας φωτογραφίας σε μορφή πίνακα 6Χ6, το σύμβολο * είναι το σύμβολο της πράξης της συνέλιξης και ο πίνακας 3Χ3 είναι το φίλτρο που χρησιμοποιούμε πχ για αναγνώριση ακμών. Το φίλτρο ολισθαίνει πάνω στον

πίνακα στοιχείο ανά στοιχείο και εκτελούνται κάποιες μαθηματικές πράξεις για να καταλήξουμε στον τελικό πίνακα 4X4 που ονομάζεται χάρτης χαρακτηριστικών (feature map) και είναι η παραλλαγμένη εικόνα.



Εικόνα 7: Η πράξη της συνέλιξης

Έτσι, για τον υπολογισμό του εσωτερικού γινόμενου των πινάκων παίρνουμε το 3X3 φίλτρο το εναποθέτουμε στην πάνω αριστερή περιοχή του 6X6 πίνακα και εκτελούμε τις πράξεις:

$$(3 * 1) + (1 * 1) + (2 * 1) + (0 * 0) + (5 * 0) + (7 * 0) + (1 * -1) + (8 * -1) + (2 * -1) = -5$$

Το -5 είναι το πρώτο (πάνω αριστερά) στοιχείο του χάρτη χαρακτηριστικών. Αντιστοίχως δουλεύουμε και με τα υπόλοιπα στοιχεία, ολισθαίνοντας το φίλτρο ώσπου να συμπληρωθεί ο τελικός πίνακας. Γενικεύοντας την παραπάνω διαδικασία εάν υποθέσουμε ότι η αρχική εικόνα είναι διαστάσεων $n * n$ και το φίλτρο με το οποίο συνελλίσσεται είναι διαστάσεων $f * f$ τότε ο παραγόμενος χάρτης χαρακτηριστικών θα είναι $(n - f + 1) * (n - f + 1)$ διαστάσεων.

Αργότερα, ο χάρτης χαρακτηριστικών θα οδηγηθεί σε άλλα επίπεδα για να μάθει και άλλα χαρακτηριστικά της εικόνας εισόδου.

Υπερπαράμετροι στο Συνελικτικό Επίπεδο

Υπάρχουν κάποιες υπερπαράμετροι οι οποίες με τις κατάλληλες ρυθμίσεις επηρεάζουν το μέγεθος και το βάθος της εξόδου της παραγόμενης εικόνας μετά τη συνέλιξη. Αυτές είναι:

Το πλήθος των φίλτρων (πυρήνων): Αυξάνοντας το πλήθος των φίλτρων σε κάποιο συνελικτικό επίπεδο αυξάνεται ο αριθμός των νευρώνων (κρυφών μονάδων) οπότε και το δίκτυο γίνεται ικανότερο να ανιχνεύσει πολυπλοκότερα μοτίβα.

Το μέγεθος των φίλτρων: Οι τιμές του πίνακα του φίλτρου, τα λεγόμενα βάρη (weights) προκύπτουν κατά την εκπαίδευση του δικτύου και αντιπροσωπεύουν τη σημαντικότητα του χαρακτηριστικού που εκφράζουν στην εικόνα εξόδου. Γενικά, τα μικρότερου μεγέθους φίλτρα συγκεντρώνουν μεγάλο όγκο τοπικής πληροφορίας με συνέπεια να συλλαμβάνουν και τις παραμικρές λεπτομέρειες κάποιου μέρους της εικόνας ενώ τα μεγαλύτερου μεγέθους φίλτρα αποτυπώνουν μια πιο γενική, εποπτική αλλά λιγότερο λεπτομερειακή άποψη της εικόνας. Το σχήμα των φίλτρων είναι σχεδόν πάντοτε τετράγωνο με διαστάσεις π.χ. 3x3 ή 5x5 (συνήθως μονών αριθμών) χωρίς να αποκλείονται και μεγαλύτερες αλλά με σπανιότερη χρήση καθώς χάνουν σημαντικές λεπτομέρειες της εικόνας και κάποιες φορές οδηγούν σε υπερπροσαρμογή.

Βηματισμός (stride): Κατά τη σάρωση του πίνακα της εικόνας εισόδου από το φίλτρο μπορεί να επιλεγεί το βήμα με το οποίο θα γίνεται η σάρωση αυτή τόσο οριζόντια όσο και κάθετα. Δηλαδή ο βηματισμός προσδιορίζει πόσα εικονοστοιχεία θα προσπερνά το φίλτρο κατά τη συνέλιξή του με την εικόνα. Αυτή η διαδικασία έχει ως αποτέλεσμα τη δραστική μείωση του μεγέθους της παραγόμενης εικόνας κατά τον τύπο:

$$\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor$$

όπου n=διάσταση του πίνακα εισόδου, p=μέγεθος του padding (γέμισμα περιθωρίου), f= διάσταση του φίλτρου, s= μέγεθος βηματισμού.

Γέμισμα περιθωρίου (padding): Το γέμισμα του περιθωρίου του πίνακα της εικόνας εισόδου με μηδενικά είναι μια ενέργεια που έχει δύο στόχους. Ο πρώτος στόχος είναι η διατήρηση των αρχικών διαστάσεων της εικόνας μετά τη συνέλιξή της με το φίλτρο, κάτι που είναι επιθυμητό στα πολλαπλά συνελικτικά επίπεδα των βαθιών νευρωνικών δικτύων αφού χωρίς το padding θα είχαμε διαδοχικές συρρικνώσεις της εικόνας. Ο δεύτερος στόχος είναι η διατήρηση της πληροφορίας των εικονοστοιχείων που βρίσκονται στα άκρα (οριζόντια και κάθετα) της εικόνας. Κατά τις ολισθήσεις του φίλτρου, τα κεντρικά εικονοστοιχεία περιλαμβάνονται περισσότερες φορές στη συνέλιξη από ότι τα ακραία εικονοστοιχεία της εικόνας. Με την υπερπαράμετρο του γεμίσματος περιθωρίου τα ακραία εικονοστοιχεία μεταφέρονται κεντρικότερα στον πίνακα της εικόνας με συνέπεια να εμφανίζονται συχνότερα στις συνελίξεις του συνελικτικού επιπέδου.

3.1.2 Επίπεδο συγκέντρωσης ή συσσώρευσης (Pooling Layer)

Τα πολλαπλά επίπεδα συνέλιξης που συνήθως έχει ένα ΣΝΔ αυξάνουν σημαντικά τον αριθμό των παραμέτρων (βαρών) που το δίκτυο καλείται να εκπαιδεύσει με κόστος την αύξηση του χρόνου εκπαίδευσης και των υπολογιστικών πόρων που απαιτούνται γι' αυτή. Ο σκοπός του επιπέδου συγκέντρωσης είναι η μείωση του μεγέθους του χάρτη χαρακτηριστικών (όπως αυτός προκύπτει από το επίπεδο συνέλιξης) με ταυτόχρονη όμως διατήρηση της σημαντικής πληροφορίας που υπάρχει ήδη σε αυτόν. Με αυτή την υποδειγματοληψία προκύπτουν νέοι χάρτες χαρακτηριστικών συμπυκνωμένης ανάλυσης. Υπάρχουν αρκετές μέθοδοι συγκέντρωσης είτε δημοφιλείς είτε πρωτοποριακές. Αναφέρουμε τις δύο δημοφιλέστερες καθώς χρησιμοποιούνται στα ΣΝΔ που έγιναν τα πειράματα της παρούσας εργασίας. Αυτές είναι οι μέθοδοι της μέγιστης συγκέντρωσης (max pooling) και της μέσης συγκέντρωσης (average pooling).

Μέγιστη συγκέντρωση (max pooling)

Ο πίνακας της μέγιστης συγκέντρωσης έχει και αυτός υπερπαραμέτρους όπως και το φίλτρο συνέλιξης δηλαδή έχει διάσταση και βηματισμό. Μια σημαντική διαφορά τους είναι ότι το φίλτρο max pooling δεν έχει παραμέτρους προς εκπαίδευση. Έτσι, μόλις πάρει τιμές για τη διάσταση και τον βηματισμό, αυτό που κάνει είναι να ολισθαίνει στον χάρτη χαρακτηριστικών του (προηγούμενου) επιπέδου συνέλιξης και να επιλέγει τη μέγιστη τιμή από κάθε παράθυρο αγνοώντας τις υπόλοιπες με αποτέλεσμα την παραγωγή νέου χάρτη χαρακτηριστικών μικρότερων διαστάσεων. Για τις διαστάσεις του χάρτη χαρακτηριστικών ισχύει ο μαθηματικός τύπος που είδαμε προηγουμένως με τη διαφορά ότι εδώ το p=0.

$$\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor$$

Μέση συγκέντρωση (average pooling)

Το επίπεδο μέσης συγκέντρωσης εκτελεί την υποδειγματοληψία παρόμοια με τον τρόπο του max pooling αλλά αντί για τη μέγιστη επιλέγει τον μέσο όρο των τιμών των εικονοστοιχείων της εκάστοτε περιοχής σάρωσης.

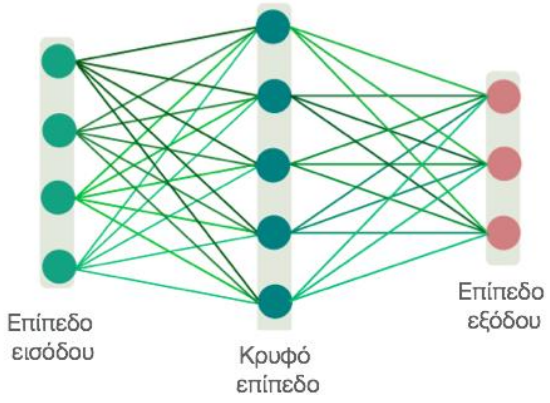
Γενικά, η μέθοδος max pooling χρησιμοποιείται συχνότερα από αυτή του average pooling.

Image flattening (μετατροπή εικόνας σε διάνυσμα)

Η διαδρομή που ακολουθεί η είσοδος (δηλ. η εικόνα) στα συνελικτικά και στα επίπεδα συγκέντρωσης ενός ΣΝΔ μεταμορφώνει διαρκώς την πληροφορία, κατά την εκπαίδευση, σε χάρτες χαρακτηριστικών διαφόρων μεγεθών. Στο τελευταίο επίπεδο, πριν το πλήρως συνδεδεμένο, έχει δημιουργηθεί πλήθος τέτοιων χαρτών δύο διαστάσεων το οποίο πρέπει να μετατραπεί σε διάνυσμα μιας διάστασης προκειμένου να γίνει είσοδος στο πλήρως συνδεδεμένο επίπεδο. Η διαδικασία αυτή ονομάζεται image flattening.

3.1.3 Πλήρως συνδεδεμένο επίπεδο (fully connected layer)

Στην ουσία πρόκειται για ένα πολυεπίπεδο perceptron (MLP) δηλαδή ένα τεχνητό νευρωνικό δίκτυο πρόσθιας τροφοδότησης που περιλαμβάνει ένα ή περισσότερα κρυφά επίπεδα ενδιαμέσως των επιπέδων εισόδου και εξόδου, Εικόνα 8.



Εικόνα 8: MLP με ένα κρυφό (πλήρως συνδεδεμένο) επίπεδο

Μεταξύ των νευρώνων του ίδιου επιπέδου δεν υπάρχει καμία σύνδεση αλλά οι νευρώνες δύο διαδοχικών επιπέδων συνδέονται πλήρως μεταξύ τους. Η δομή του MLP είναι η εξής:

Επίπεδο εισόδου: Είναι το διάνυσμα χαρακτηριστικών, όπως αυτό προέρχεται μετά το image flattening που είδαμε προηγουμένως.

Κρυφό επίπεδο: Ένα η περισσότερα επίπεδα τους νευρώνες των οποίων ούτε ελέγχουμε ούτε βλέπουμε, εξού και η ονομασία «κρυφό». Στις ακμές μεταξύ των νευρώνων ανατίθενται τιμές (βάρη) ανάλογα με τη σημαντικότητα της επιρροής τους στην τελική πρόβλεψη της εξόδου.

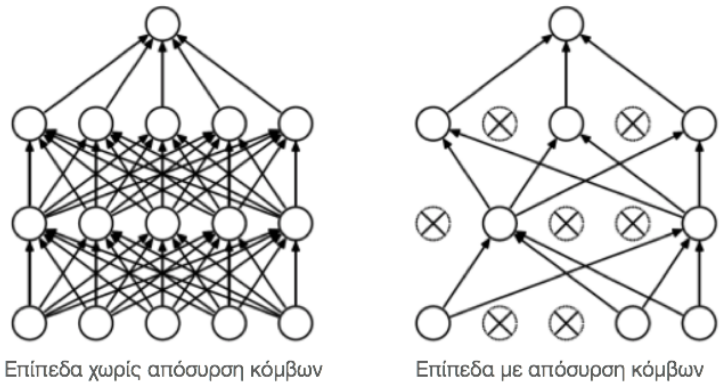
Επίπεδο εξόδου: Είναι το επίπεδο στο οποίο, ανάλογα με τη συνάρτηση ενεργοποίησης που χρησιμοποιούμε, παίρνουμε το αποτέλεσμα της πρόβλεψης από το μοντέλο. Το αποτέλεσμα αυτό θα αποτυπώνεται είτε με πραγματικούς αριθμούς (αν πρόκειται για πρόβλημα παλινδρόμησης) είτε με πιθανότητες (αν πρόκειται για πρόβλημα ταξινόμησης).

3.1.4 Επίπεδο Απόσυρσης (Dropout Layer)

Αν και δεν αποτελεί κύριο συστατικό ενός ΣΝΔ, το επίπεδο απόσυρσης [34] χρησιμοποιείται σε πολλές αρχιτεκτονικές δικτύων συνήθως πριν από το πλήρως συνδεδεμένο επίπεδο (π.χ. VGG, AlexNet, κ.α.). Η τεχνική της απόσυρσης είναι μια μέθοδος αποφυγής της υπερπροσαρμογής στα νευρωνικά δίκτυα.

Η υπερπροσαρμογή συμβαίνει όταν ένα μοντέλο ενώ φαινομενικά έχει εκπαιδευτεί εξαιρετικά καλά, στην πραγματικότητα έχει απλώς απομνημονεύσει όλα τα δεδομένα εκπαίδευσης. Το πρόβλημα αποκαλύπτεται όταν το μοντέλο δοκιμαστεί σε σύνολο δεδομένων στο οποίο δεν έχει εκπαιδευτεί ποτέ όπου και αποτυγχάνει στη γενίκευσή του.

Η τεχνική της απόσυρσης, όπως προδίδει και η ονομασία της, αυτό που κάνει είναι ότι αποσύρει ένα ποσοστό τυχαίων νευρώνων από κάποιο επίπεδο, Εικόνα 9.



Εικόνα 9: Τεχνική της απόσυρσης κόμβων-νευρώνων

Η πιθανότητα της απόσυρσης αποφασίζεται από τον χρήστη, πρόκειται δηλαδή για μια υπερπαράμετρο. Οι αποσυρόμενοι νευρώνες δε συμμετέχουν πλέον στον τρέχοντα γύρο της εκπαίδευσης και αυτό έχει ως αποτέλεσμα τον εξαναγκασμό των υπόλοιπων (ενεργών) νευρώνων του επιπέδου στον εντοπισμό νέων

χαρακτηριστικών που δεν βασίζονται στους προσωρινά (καθώς μπορεί να ενεργοποιηθούν ξανά σε επόμενο γύρο) ανενεργούς νευρώνες. Όσο διαρκεί αυτή η διαδικασία, τα βάρη ανανεώνονται συνεχώς και ως αποτέλεσμα έχουμε ποιοτικότερη εκπαίδευση δηλαδή λιγότερο ευαίσθητη στην υπερπροσαρμογή.

3.2 Συναρτήσεις Ενεργοποίησης (activation functions, nonlinearities)

Οι νευρώνες ενός ΤΝΔ εκτελούν έναν γραμμικό μετασχηματισμό (z) χρησιμοποιώντας τις εισόδους(x_i), τα βάρη(w_i) και την πόλωση(b):

$$z = \sum x_i w_i + b$$

Αυτός ο γραμμικός μετασχηματισμός ονομάζεται και συνάρτηση σταθμισμένου αθροίσματος (weighted sum function). Η συνάρτηση ενεργοποίησης [35] τοποθετείται στο τέλος (στην έξοδο) και μετασχηματίζει τη γραμμικότητα του νευρώνα σε μη γραμμικότητα [36] (nonlinearity) αποφασίζοντας αν ο νευρώνας θα ενεργοποιηθεί ή όχι κανονικοποιώντας την τιμή εξόδου μεταξύ $[0, 1]$ ή $[-1, 1]$ αναλόγως του τύπου της συνάρτησης ενεργοποίησης που χρησιμοποιήθηκε. Έτσι, το δίκτυο μπορεί να προσεγγίσει συναρτήσεις που δεν είναι γραμμικές δηλαδή μπορεί να προβλέψει μια κλάση που διαχωρίζεται από μη-γραμμικά όρια.

Κατά τη διάρκεια της εμπροσθοδιάδοσης του σφάλματος η πληροφορία στους νευρώνες μεταφέρεται από τις εισόδους προς το επίπεδο εξόδου τους:

$$\hat{y} = συνάρτηση\ ενεργοποίησης\ (\sum x_i w_i + b)$$

Στη συνέχεια η τιμή της πρόβλεψης ($\hat{y}$) συγκρίνεται με την πραγματική τιμή (y) για τον υπολογισμό του σφάλματος:

$$σφάλμα = y - \hat{y}$$

και ακολουθεί η ανανέωση των βαρών (w_i) μέσω της οπισθοδιάδοσης (backpropagation) σε όλα τα επίπεδα του δικτύου ώσπου να μειωθεί η διαφορά μεταξύ πρόβλεψης και πραγματικής τιμής και να ολοκληρωθεί έτσι η εκπαίδευσή του.

Οι συναρτήσεις ενεργοποίησης είναι κατά κανόνα:

Διαφορίσιμες (differentiable), για να μπορεί το δίκτυο να ανανεώνει τα βάρη και τις πολώσεις των νευρώνων κατά την οπισθοδιάδοση του σφάλματος, μειώνοντάς το.

Μονότονες (monotonic), καθώς η μονότονη συμπεριφορά της συνάρτησης βοηθάει τη σύγκλιση του νευρωνικού, την ευκολότερη εύρεση ελαχίστου στην κλίση.

Το κριτήριο της μονοτονίας δεν είναι υποχρεωτικό καθώς υπάρχουν συναρτήσεις ενεργοποίησης που δεν το ικανοποιούν. Μια τέτοια συνάρτηση είναι η Mish.

Τελικά, βλέπουμε ότι καθώς οι συναρτήσεις ενεργοποίησης παίζουν καθοριστικό ρόλο στην απόδοση και στον χρόνο εκπαίδευσης των νευρωνικών δικτύων, η προσεκτική επιλογή τους θα οδηγήσει το δίκτυο σε καλύτερη απόδοση, λιγότερο χρόνο εκπαίδευσης και με λιγότερες απώλειες.

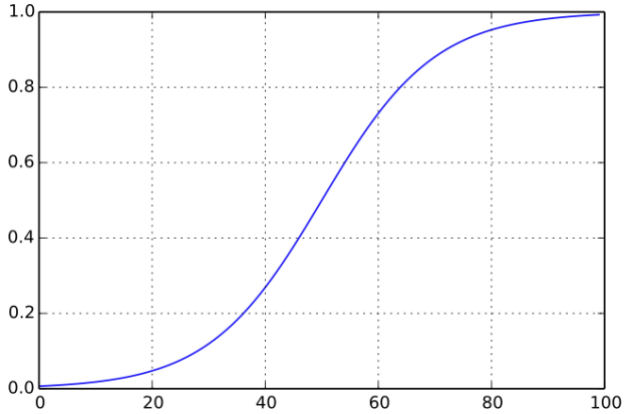
Υπάρχουν πολλά είδη συναρτήσεων ενεργοποίησης, θα αναφέρουμε στη συνέχεια μερικά από τα πιο κοινά:

3.2.1 Σιγμοειδής ή λογιστική συνάρτηση (sigmoid function)

Από τις πιο κοινές συναρτήσεις ενεργοποίησης, χαρακτηρίζεται από το σχήμα "S" της καμπύλης της και ορίζεται από τον τύπο:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Οι τιμές της κυμαίνονται μεταξύ $[0, 1]$ και χρησιμοποιείται στην δυαδική ταξινόμηση (binary classification) για την πρόβλεψη της πιθανότητας μεταξύ δύο κλάσεων όπως φαίνεται στην Εικόνα 10.

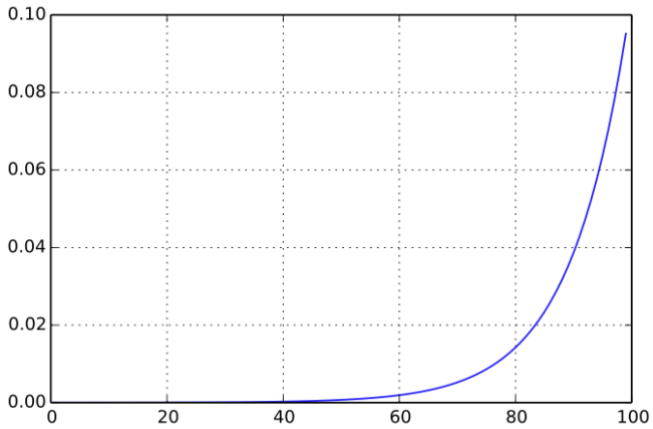


Εικόνα 10: Η σιγμοειδής συνάρτηση

Η σιγμοειδής, αν και παλαιότερα ήταν πολύ διαδεδομένη, έχει αντικατασταθεί από τη ReLU κυρίως στα Βαθιά Νευρωνικά Δίκτυα εξαιτίας του προβλήματος της εξαφάνισης κλίσης [35] (vanishing gradient problem) που προκαλεί. Επειδή, όπως αναφέρθηκε, η σιγμοειδής συμπιέζει τις εισερχόμενες τιμές μεταξύ $[0, 1]$ ακόμα και μεγάλες διακυμάνσεις στις τιμές της εισόδου της συνάρτησης προκαλούν πολύ μικρές αλλαγές στην έξοδο. Αυτό έχει ως αποτέλεσμα την ελαχιστοποίηση της παραγώγου της συνάρτησης σφάλματος (που είναι το γινόμενο όλων των παραγώγων των νευρώνων) κατά την οπισθοδιάδοση και έτσι αφού η κλίση της καθόδου σχεδόν εξαφανίζεται, ποτέ δεν πρόκειται να φτάσει στο επιθυμητό ελάχιστο, δηλαδή η εκπαίδευση του δικτύου αποτυγχάνει. Άλλο ένα μειονέκτημα της συνάρτησης αυτής είναι ότι δεν είναι zero-centered δηλαδή το εύρος τιμών της δεν περιλαμβάνει θετικές και αρνητικές τιμές. Έτσι, η έξοδος ενός νευρώνα με sigmoid πάντοτε θα έχει θετικές τιμές οπότε τα βάρη για να ανανεωθούν χρειάζονται περισσότερες εποχές προκειμένου να εκπαιδευτούν σωστά.

3.2.2 Κανονικοποιημένη εκθετική συνάρτηση (softmax function)

Η softmax, Εικόνα 11, είναι μια γενικευμένη μορφή της σιγμοειδούς συνάρτησης και χρησιμοποιείται στο επίπεδο εξόδου ενός νευρωνικού δικτύου που επιλύει πρόβλημα ταξινόμησης πολλών κλάσεων.



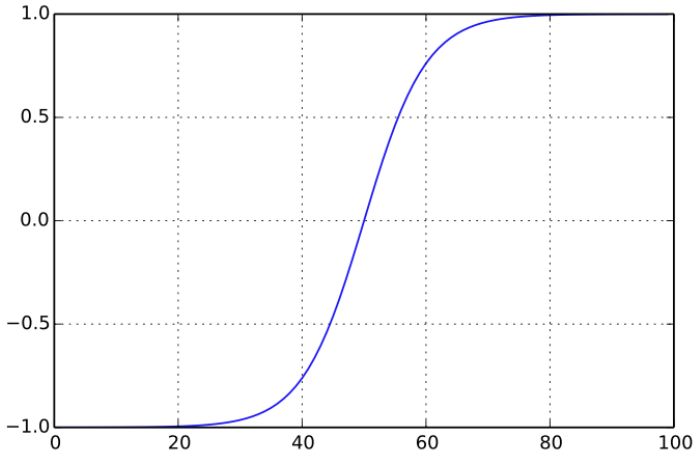
Εικόνα 11: Η συνάρτηση softmax

Οι τιμές που μπορεί να πάρει είναι μεταξύ $[0, 1]$ με το άθροισμα των πιθανοτήτων να ισούται με 1. Η κανονικοποιημένη εκθετική συνάρτηση εκφράζεται από τον τύπο:

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

3.2.3 Συνάρτηση υπερβολικής εφαπτομένης (hyperbolic tangent function, tanh)

Η συνάρτησης της υπερβολικής εφαπτομένης έχει τα πλεονεκτήματα της σιγμοειδούς και επιπλέον είναι zero-centered με συνέπεια να επιτυγχάνει καλύτερη απόδοση στην εκπαίδευση σε ΣΝΔ, Εικόνα 12.



Εικόνα 12: Η συνάρτηση tanh

Η συνάρτηση εκφράζεται από τον τύπο:

$$\sigma(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Η tanh αποτελεί μια τροποποιημένη έκδοση της sigmoid αφού η υπερβολική εφαπτομένη μπορεί να εκφραστεί ως $\tan(x) = 2sigmoid(2x) - 1$. Το εύρος τιμών της εκτείνεται στο διάστημα $[-1, 1]$ οπότε παράγει ως έξοδο αρνητικές, θετικές και μηδενικές τιμές. Με αυτόν τον τρόπο εξαλείφει και το zero-centered πρόβλημα που αναφέρθηκε νωρίτερα. Όμως και σε αυτή τη συνάρτηση παραμένει το πρόβλημα της εξαφάνισης κλίσης λόγω της συμπίεσης των τιμών εισόδου στο $[-1, 1]$.

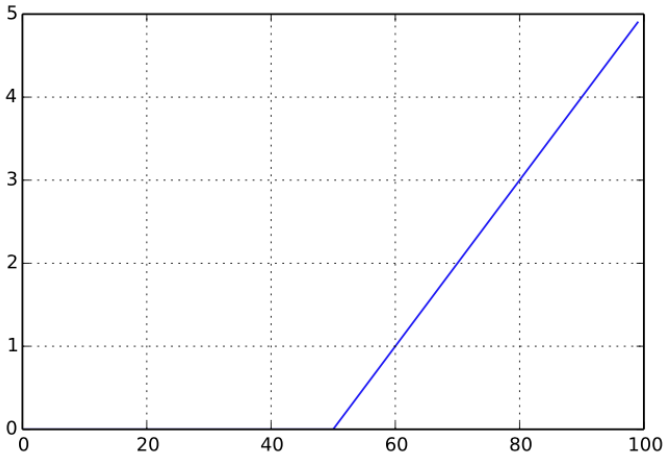
3.2.4 Συνάρτηση διορθωμένης γραμμικής μονάδας (Rectified Linear Unit function, ReLU)

Η συνάρτηση διορθωμένης γραμμικής μονάδας παραμένει, από τον καιρό που προτάθηκε, μια εξαιρετικά δημοφιλής συνάρτηση ενεργοποίησης. Η ReLU [37] έχει πάρα πολύ μικρό υπολογιστικό κόστος εξαιτίας των απλών υπολογισμών χωρίς εκθετικά. Επίσης, στα νευρωνικά δίκτυα έχει καλύτερη απόδοση και γενίκευση από ότι οι tanh και sigmoid και χρησιμοποιείται ευρέως στα κρυφά επίπεδα των βαθιών νευρωνικών δικτύων.

Η συνάρτηση διορθωμένης γραμμικής μονάδας εκφράζεται από τη σχέση:

$$f(x) = \max(0, x) = \begin{cases} x, & \gamma\iota\alpha\ x \geq 0 \\ 0, & \gamma\iota\alpha\ x < 0 \end{cases}$$

Δηλαδή η συνάρτηση ενεργοποιεί τον νευρώνα μόνο αν η είσοδος είναι θετικός αριθμός και τότε έχει γραμμική σχέση με τη μεταβλητή εξόδου. Αν η είσοδος είναι αρνητική ή μηδενική τότε η έξοδος είναι πάντα μηδέν, Εικόνα 13



Εικόνα 13: Η συνάρτηση ReLU

Ένα πλεονέκτημα της ReLU είναι ότι εισάγει την αδράνεια στα κρυφά επίπεδα καθώς οι τιμές της συμπιέζονται μεταξύ $[0, +\infty)$. Επίσης, δεν πάσχει από το πρόβλημα της εξαφάνισης κλίσης (όπως οι sigmoid και tanh) αφού η παράγωγος είναι πάντα 0 ή 1. Όμως, όπως αναφέρθηκε παραπάνω, κάθε είσοδος που είναι αρνητική παράγει 0 στην έξοδο που σημαίνει ότι ο αντίστοιχος νευρώνας δεν θα ενεργοποιηθεί. Έτσι, κατά τη διαδικασία της οπισθοδιάδοσης τα βάρη και οι πολώσεις των ανενεργών νευρώνων δε θα ενημερωθούν και επομένως δε

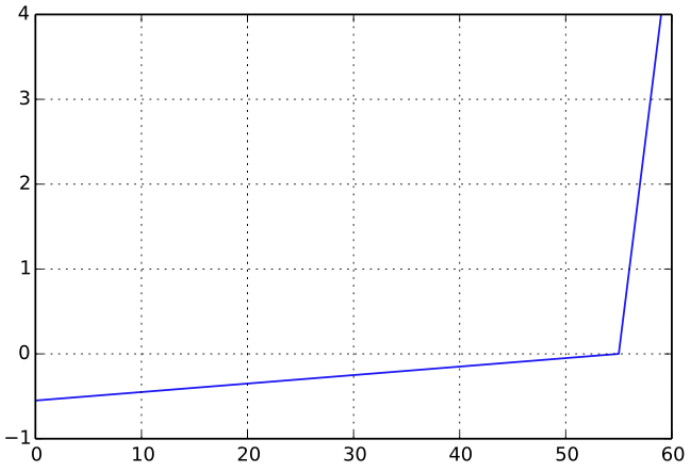
συνεισφέρουν στη διαδικασία της εκπαίδευσης δημιουργώντας το λεγόμενο dying ReLU problem [38] ή πρόβλημα του «νεκρού νευρώνα» . Το πρόβλημα αυτό αντιμετωπίζεται με τη συνάρτηση Leaky ReLU που θα δούμε στη συνέχεια.

3.2.5 Συνάρτηση διαρρέουσας διορθωμένης γραμμικής μονάδας (Leaky ReLU, LReLU)

Πρόκειται για μια βελτιωμένη έκδοση της ReLU που εισάγει μια μικρή αρνητική κλίση («διαρροή») για να λύσει το πρόβλημα του «νεκρού νευρώνα» και εκφράζεται από τη σχέση:

$$f(x) = \begin{cases} 0.01x, & για\ x < 0 \\ x, & για\ x \geq 0 \end{cases}$$

Όταν η τιμή ενός νευρώνα είναι αρνητική τότε η τιμή πολλαπλασιάζεται με το 0.01 ώστε να αποφευχθεί η απενεργοποίησή του. Για τιμή x=0 η αριστερή παράγωγος της LReLU [37] είναι 0.01 ενώ η δεξιά παράγωγος είναι 1, Εικόνα 14.



Εικόνα 14: Η συνάρτηση Leaky ReLU

Αφού αυτές οι δύο παράγωγοι δεν είναι ίσες (για x=0) τότε η συνάρτηση δεν είναι παραγωγίσιμη για x=0. Αυτό για το θετικό μέρος τιμών δεν είναι πρόβλημα (επειδή η κλίση είναι πάντοτε 1) αλλά για το αρνητικό μέρος είναι πρόβλημα καθώς η τιμή 0.01 της κλίσης είναι πολύ κοντά στο 0 εγκυμονώντας τον κίνδυνο εμφάνισης τους προβλήματος της εξαφάνισης κλίσης.

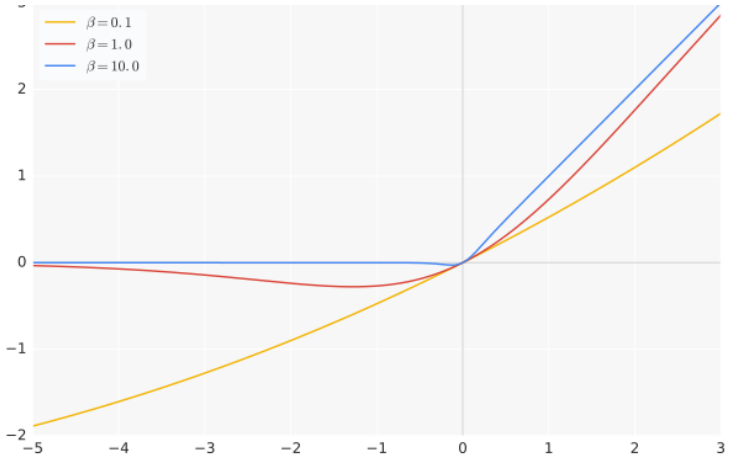
3.2.6 Συνάρτηση Swish

Η Swish [39] είναι η πιο πρόσφατα παρουσιασμένη συνάρτηση ενεργοποίησης που σχολιάζεται σε αυτή την εργασία. Προτάθηκε το 2017 από την ομάδα της Google Brain και λόγω της απλότητάς της αλλά και της ομοιότητάς της με την ReLU την κάνει «εύκολη» αντικαταστάτρια της ReLU στα επίπεδα των νευρωνικών δικτύων.

Η Swish είναι μη φραγμένη επάνω και φραγμένη κάτω όπως η ReLU. Αντίθετα με τη ReLU είναι ομαλή, μη μονότονη συνάρτηση και αυτή η ιδιότητά της την κάνει να ξεχωρίζει από τις πιο κοινές συναρτήσεις ενεργοποίησης. Όπως είδαμε προηγουμένως η ReLU, αλλά και μεταγενέστερες παραλλαγές της όπως η LReLU ή η SELU δεν καταφέρνουν να απαλλαγούν από την εμφάνιση του προβλήματος της εξαφάνισης κλίσης. Φαίνεται όμως ότι η υψηλή απόδοση της Swish σε ΒΝΔ συγκρινόμενη με αυτές των ReLU και sigmoid αποδίδεται κατά ένα μέρος στην ελαχιστοποίηση της εξαφάνισης κλίσης κατά την οπισθοδιάδοση. Η Swish εκφράζεται από τον τύπο:

$$f(x) = x\ sigmoid(\beta x)$$

Όπου β, μια σταθερά ή μια εκπαιδεύσιμη παράμετρος (υπερπαράμετρος). Αν η β = 1 τότε η συνάρτηση γίνεται ισοδύναμη με την Sigmoid-weighted Linear Unit μια παραλλαγή της sigmoid που παρουσιάστηκε για την ενισχυτική μάθηση (reinforcement learning). Αν β = 0 η Swish γίνεται η συνάρτηση $f(x) = \frac{x}{2}$ και αν η παράμετρος β τείνει στο άπειρο τότε η Swish λειτουργεί όπως η ReLU, Εικόνα 15.



Εικόνα 15: Η συνάρτηση swish για διάφορες τιμές της β

3.3 Συναρτήσεις κόστους (cost functions, loss functions)

Η συνάρτηση κόστους ή σφάλματος είναι ένας τρόπος μέτρησης που υπολογίζει το σφάλμα δηλαδή υπολογίζει το πόσο κοντά ή όχι είναι η πρόβλεψη (prediction) που έχει κάνει το νευρωνικό δίκτυο κατά την εκπαίδευσή του σε σχέση με την πραγματικότητα (ground truth).

Η επιλογή της κατάλληλης συνάρτησης κόστους είναι καίριος παράγοντας που επηρεάζει την ακρίβεια του νευρωνικού δικτύου. Η τιμή του σφάλματος θα χρησιμοποιηθεί στη συνέχεια από κάποια συνάρτηση βελτιστοποίησης η οποία μέσω ρυθμίσεως κάποιων παραμέτρων της (optimization) θα οδηγήσει στη βελτίωση της ακρίβειας του μοντέλου.

Οι συναρτήσεις σφάλματος κατατάσσονται σε δύο κύριες κατηγορίες αναλόγως του έργου που καλείται να αποδώσει το μοντέλο, αναφέρονται μερικές εξ'αυτών:

- Παλινδρόμησης:
    1. *Συνάρτηση μέσου τετραγωνικού σφάλματος (Mean Squarred Error Loss)*

Μετράει το μέσο τετραγωνικό σφάλμα μεταξύ της προβλεπόμενης και της πραγματικής τιμής της εξόδου και η τιμής είναι πάντοτε θετική:

$$C(w,b) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y_i} - y_i)^2$$

$$\text{όπου } w = \text{πίνακας } \beta\text{αρών}, b = \text{πολώσεις},$$
$$N = \text{πλήθος δειγμάτων εκπαίδευσης},$$
$$\hat{y_i} = \text{πρόβλεψη εξόδου}, y = \text{πραγματική τιμή εξόδου}$$

Ένα πλεονέκτημα της MSE είναι ότι εγγυάται ότι το μοντέλο δεν θα έχει προβλέψεις ακραίων τιμών (outliers) με μεγάλα σφάλματα καθώς τους προσθέτει περισσότερο βάρος σε σχέση με τα μικρά σφάλματα. Μερικές φορές αυτή η ευαισθησία της MSE στις ακραίες τιμές (λόγω τετραγώνου στις τιμές σφάλματος) μπορεί να φανεί σαν μειονέκτημα σε περιπτώσεις που δεν θέλουμε να λάβουμε υπόψη μας τα outliers.

2. *Συνάρτηση μέσου απόλυτου σφάλματος (Mean Absolute Error Loss)*

Μετράει τη μέση απόλυτη διαφορά μεταξύ της προβλεπόμενης και της πραγματικής τιμής, έχει πάντοτε θετική τιμή και εκφράζεται από τον τύπο:

$$C(w,b) = \frac{1}{N}\sum_{i=1}^{N}|\hat{y_i} - y_i|$$

Το πλεονέκτημα της MAE είναι ότι όλα τα λάθη (και των ακραίων τιμών) έχουν το ίδιο βάρος με συνέπεια η τιμή που δίνει μας δίνει μια πιο ισότιμη μέτρηση της απόδοσης του μοντέλου. Όμως, αν μας ενδιαφέρουν οι προβλέψεις και των ακραίων τιμών τότε η MAE δεν είναι τόσο αποτελεσματική καθώς το βάρος των outliers θα «ζυγίζει» το ίδιο με τις υπόλοιπες τιμές και αυτό μπορεί να δώσει ένα φαινομενικά καλό μοντέλο αλλά στην πραγματικότητα θα έχει πολλές αστοχίες πρόβλεψης. Βλέπουμε δηλαδή ότι το πλεονέκτημα της MAE είναι το μειονέκτημα της MSE και το αντίστροφο.

- Ταξινόμησης:
  1. *Συνάρτηση σφάλματος δυαδικής διασταυρωμένης εντροπίας (Binary Cross Entropy Loss)*

Η συνάρτηση της binary cross-entropy είναι πολύ διαδεδομένη σε προβλήματα δυαδικής ταξινόμησης καθώς ποσοτικοποιεί τη διαφορά μεταξύ δύο κατανομών πιθανοτήτων όπως φαίνεται και από τον τύπο της:

$$C(w,b) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

$$\text{όπου } y = \text{πραγματική πιθανότητα}, \hat{y} = \text{πιθανότατα πρόβλεψης}$$

2. *Συνάρτηση σφάλματος πολλών κλάσεων διασταυρωμένης εντροπίας (Categorical Cross Entropy Loss),*

όταν η ταξινόμηση αφορά μεγαλύτερο αριθμό κλάσεων:

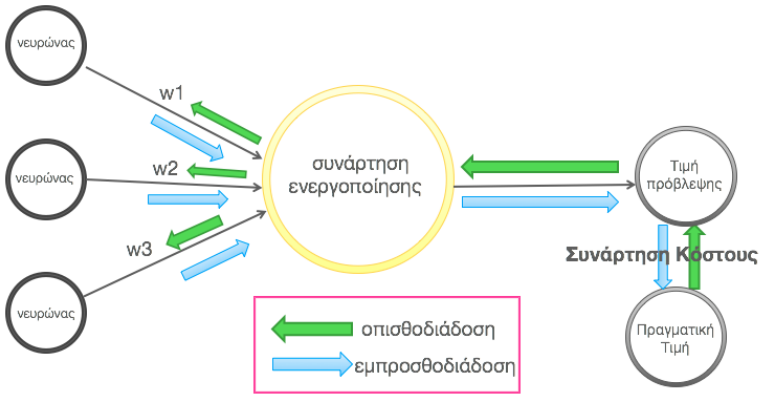
$$C(w,b) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log\ \hat{y}_{ik}$$

$$\text{Όπου } k = \text{μετρητής κλάσεων}, i = \text{ μετρητής δειγμάτων},$$

$$\text{όπου } y = \text{πραγματική πιθανότητα}, \hat{y} = \text{πιθανότατα πρόβλεψης}$$

3.4 Αλγόριθμος εκπαίδευσης ΤΝΔ

Ένα ΤΝΔ προσπαθεί επαναληπτικά, να μειώσει την τιμή της συνάρτησης κόστους προκειμένου η ανανέωση των βαρών των νευρώνων κάθε επιπέδου να οδηγήσει στην εκμάθηση καλύτερων χαρακτηριστικών, στη δημιουργία καλύτερου μοντέλου, Εικόνα 16.



Εικόνα 16: Σχηματική αναπαράσταση εκπαίδευσης ενός ΤΝΔ

Αυτή η επαναληπτική διαδικασία περιγράφεται με τον αλγόριθμο που ακολουθεί:

1ο βήμα Τυχαία αρχικοποίηση (με τιμές κοντά στο μηδέν) των βαρών των νευρώνων.

2ο βήμα Εισαγωγή των πρώτων τιμών του σετ δεδομένων στις εισόδους των νευρώνων του επιπέδου εισαγωγής (input layer).

$$\boldsymbol{input}\ x\text{: } \textit{θέσε τη συνάρτηση ενεργοποίησης } \alpha^l \textit{ για το επίπεδο εισόδου}$$

3ο βήμα Εμπροσθοδιάδοση (forward propagation): Με κατεύθυνση από τα αριστερά (είσοδος) προς τα δεξιά (έξοδος) η πληροφορία διαδίδεται από το επίπεδο εισόδου στο επίπεδο εξόδου μέσω των κρυφών επιπέδων για να υπολογιστεί η πρόβλεψη.

$$\boldsymbol{forward\ prop}\text{: } \textit{για κάθε επίπεδο } l = 2,3,\dots,L \textit{ υπολόγισε}$$
$$z^l = w^l a^{l-1} + b^l \textit{ και } a^l = \sigma(z^l)$$

4ο βήμα Σύγκριση της πρόβλεψης με το πραγματικό αποτέλεσμα και υπολογισμός του σφάλματος (κόστους).

$$\boldsymbol{output\ error}\ \delta^L\text{: } \textit{Υπολόγισε το διάνυσμα } \delta^L = \nabla_a C \odot \sigma'(z^L)$$

$$\textit{όπου } \nabla_a C = \textit{διάνυσμα από τις μερικές παραγώγους } \frac{\partial C}{\partial a_j{}^L} \textit{ ή ο ρυθμός αλλαγής}$$

της συνάρτησης κόστους C σε σχέση με τις ενεργοποιήσεις της εξόδου

5ο βήμα Οπισθοδιάδοση του σφάλματος [4ο] (backpropagation): Με κατεύθυνση από δεξιά (έξοδος) προς τα αριστερά (είσοδος) το λάθος διαδίδεται και γίνεται διόρθωση των βαρών των νευρώνων αναλόγως με το ποσοστό «ευθύνης» του καθενός.

$$\boldsymbol{backpropagation\ of\ error}\text{: } \textit{Για κάθε } l = L-1, L-2, \dots, 2$$
$$\textit{υπολόγισε } \delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

6ο βήμα Επανάληψη των βημάτων 1 έως 5 έως ότου όλες οι τιμές του σετ εκπαίδευσης διοχετευθούν στο ΤΝΔ δηλαδή μέχρι την ολοκλήρωση μιας εποχής.

$$\boldsymbol{output}\text{: } \textit{Η κλίση της συνάρτησης κόστους δίνεται από τους τύπους}\text{:}$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l \textit{ και } \frac{\partial C}{\partial b_j^l} = \delta_j^l\ ,$$

Όπου περιγράφεται ο ρυθμός μεταβολής του κόστους σε σχέση με τα όποια βάρη και τις όποιες πολώσεις των νευρώνων του δικτύου.

7ο βήμα Επανάληψη των προηγούμενων βημάτων για την εκπαίδευση του ΤΝΔ στον επιθυμητό αριθμό εποχών.

Στη συνέχεια, οι τιμές της κλίσης της συνάρτησης κόστους, δηλαδή οι μερικές παράγωγοι των βαρών και των πολώσεων θα χρησιμοποιηθούν από κάποιον αλγόριθμο βελτιστοποίησης, όπως θα δούμε παρακάτω.

3.5 Αλγόριθμοι βελτιστοποίησης (Optimizing Algorithms)

Προηγουμένως παρουσιάστηκε ο γενικός αλγόριθμος εκπαίδευσης ενός ΤΝΔ και είδαμε ότι η συνάρτηση κόστους βοηθάει το δίκτυο να κατανοήσει την επιτυχία ή όχι της εκπαίδευσης στα τρέχοντα δείγματα με τις αντίστοιχες τιμές των παραμέτρων τους (βαρών και πολώσεων). Προκειμένου το μοντέλο να μειώσει τις απώλειες διορθώνοντας τις παραμέτρους, χρησιμοποιεί κάποια συνάρτηση βελτιστοποίησης (optimizer) η οποία χρησιμοποιεί παραγώγους για να αντιληφθεί τον αντίκτυπο των μικρο-αλλαγών των βαρών και πολώσεων των νευρώνων στη συνάρτηση κόστους μέσω της κατεύθυνσης της κλίσης.

3.5.1 Αλγόριθμος καθόδου κλίσης (Gradient Descent Algorithm)

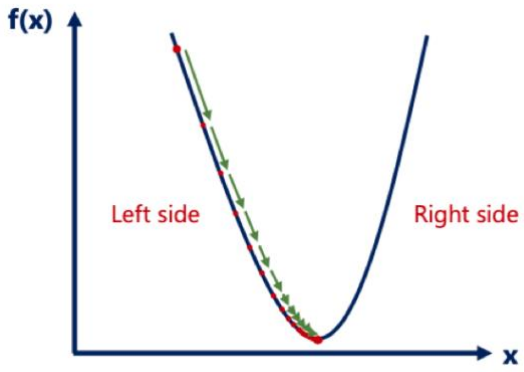
Ο αλγόριθμος της καθόδου κλίσης [41] είναι ένας αλγόριθμος βελτιστοποίησης που ελαχιστοποιεί μια κυρτή συνάρτηση κόστους επαναληπτικά κινούμενος προς την πιο απότομη κάθοδο όπως ορίζεται από την κλίση, ανανεώνοντας τις παραμέτρους του μοντέλου.

Αν στην Εικόνα 17, ο κάθετος άξονας συμβολίζει μια συνάρτηση κόστους f(x) και ο οριζόντιος άξονας κάποια παράμετρο x (πχ βάρος, πόλωση) του μοντέλου τότε μπορούμε να βρούμε το ελάχιστο αυτής της συνάρτησης εφαρμόζοντας τον τύπο:

$$x_{i+1} = x_i - af'(x_i)$$

όπου α = ρυθμός εκπαίδευσης (θετικός αριθμός). Η παράγωγος του $x_i, f'(x_i)$ δείχνει την κλίση της συνάρτησης στο x_i. Αν η παράγωγος είναι αρνητική τότε βρισκόμαστε στην αριστερή πλευρά της καμπύλης οπότε το $x_{i+1} > x_i$ και η επόμενη επανάληψη θα είναι προς τη δεξιά πλευρά. Αν η παράγωγος είναι θετική τότε βρισκόμαστε στη δεξιά πλευρά της καμπύλης και έτσι η αφαίρεση θετικού αριθμού από το x_i θα δώσει μικρότερο αριθμό οπότε στην επόμενη επανάληψη θα κινηθεί προς τα αριστερά. Βλέπουμε λοιπόν ότι είτε έτσι είτε αλλιώς,

πλησιάζουμε το ελάχιστο της συνάρτησης κόστους και όταν η παράγωγος μηδενιστεί τότε θα το έχουμε φτάσει.

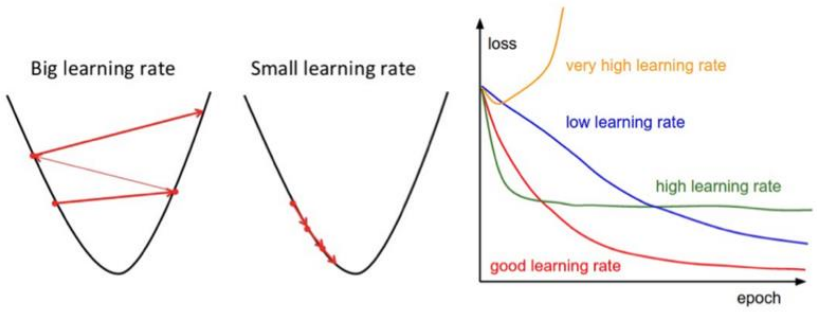


Εικόνα 17: Κάθοδος κλίσης

Αυτός ο αλγόριθμος ισχύει σε ιδανικές συνθήκες καθώς υπάρχουν κάποιοι παράγοντες που επηρεάζουν και καθορίζουν το πότε και το αν θα φτάσουμε στο πραγματικό ελάχιστο (και όχι σε κάποιο τοπικό). Ένας τέτοιος παράγοντας είναι ο ρυθμός εκπαίδευσης.

Ρυθμός εκπαίδευσης (learning rate)

Είναι το μέγεθος του βήματος της καθόδου που κάνει το δίκτυο προς το ελάχιστο της συνάρτησης κόστους. Αποτελεί μια από τις σημαντικότερες υπερπαραμέτρους που ρυθμίζουμε κατά την εκπαίδευση του δικτύου καθώς μεγάλος ρυθμός σημαίνει ταχύτερη εκπαίδευση αφού η κάθοδος γίνεται ταχύτερα ενώ, μικρός ρυθμός σημαίνει πιο αργή εκπαίδευση. Όμως όσο μεγαλύτερα είναι τα βήματα τόσο πιθανότερο είναι να μη φτάσουμε ποτέ στο ελάχιστο διότι συνεχώς θα ταλαντευόμαστε στις πλευρές της καμπύλης, χάνοντας το κατώτερο σημείο της. Από την άλλη, αν τα βήματα είναι πολύ μικρά είναι πολύ πιθανό να φτάσουμε στο ελάχιστο της καμπύλης αλλά το κόστος σε χρόνο είναι πολύ μεγάλο καθώς η εκπαίδευση μπορεί να διαρκέσει ακόμα και εβδομάδες, Εικόνα 18.



Εικόνα 18: Ιδανικός ρυθμός εκπαίδευσης

3.5.2 Παραλλαγές του αλγόριθμου καθόδου κλίσης

Υπάρχουν τρεις παραλλαγές [41] της gradient descent που διαφέρουν ως προς τον όγκο των δεδομένων που χρησιμοποιούνται για τον υπολογισμό της κλίσης της συνάρτησης κόστους.

Αυτές είναι οι batch gradient descent, stochastic gradient descent (SGD) και mini-batch gradient descent.

Αλγόριθμος καθόδου κλίσης παρτίδας (Batch gradient descent, BGD)

Στην ουσία πρόκειται για τον παραδοσιακό αλγόριθμο της καθόδου κλίσης που είδαμε παραπάνω και εκφράστηκε από τη σχέση:

$$\theta = \theta - af'(\theta),$$

$$\text{όπου } \theta \text{ οι παράμετροι ολόκληρου του σετ}, \alpha = \text{ρυθμός εκπαίδευσης}$$

Ο BGD εγγυάται τη σύγκλιση στο ολικό ελάχιστο όταν η συνάρτηση είναι κυρτή ενώ σε αντίθετη περίπτωση όταν δηλαδή η συνάρτηση δεν είναι κυρτή συχνά εγκλωβίζεται σε κάποιο τοπικό ελάχιστο. Επίσης, καθώς για κάθε μια ανανέωση των παραμέτρων x απαιτείται υπολογισμός των κλίσεων ολόκληρου του σετ εκπαίδευσης ο αλγόριθμος batch gradient descent είναι αργός και όχι ικανοποιητική επιλογή για μεγάλα σετ που δε χωρούν στη μνήμη.

Αλγόριθμος στοχαστικής καθόδου κλίσης (Stochastic gradient descent, SGD)

Ο SGD είναι ίσως ο πιο πολυχρησιμοποιημένος αλγόριθμος βελτιστοποίησης τόσο στη Μηχανική Μάθηση όσο και στη Βαθιά Μηχανική Μάθηση. Η διαφορά

του από την κάθοδο κλίσης παρτίδας είναι ότι η ανανέωση των παραμέτρων γίνεται ανά δείγμα και στοχαστικά δηλαδή με τυχαίο τρόπο:

$$\theta = \theta - af'(\theta; x_i; y_i),$$

$$\text{όπου } x_i = \text{δείγμα}, y_i = \text{ετικέτα δείγματος}$$

Η στοχαστικότητα σε συνδυασμό με την ανά δείγμα ανανέωση των παραμέτρων κάνει τον SGD πολύ ταχύτερο σε σχέση με τον BGD. Όμως η υψηλή διακύμανση των συχνών ανανεώσεων προκαλεί μια ασταθή κατάβαση με κίνηση στυλ (προς το όποιο ελάχιστο) «ζιγκ-ζαγκ» που τελικά ωφελεί την όλη διαδικασία όταν η συνάρτηση κόστους δεν είναι κυρτή διότι με αυτό το «ζιγκ-ζαγκ» ανακαλύπτει νέα και ίσως καλύτερα τοπικά ελάχιστα. Από την άλλη, όταν η συνάρτηση είναι κυρτή, αντί να συγκλίνει στο ολικό ελάχιστο αναπηδά συνεχώς πολύ κοντά γύρω του. Το πρόβλημα λύνεται εύκολα με την αργή μείωση του ρυθμού εκπαίδευσης και τότε ο SGD ηρεμεί στο ολικό ελάχιστο όπως και ο BGD.

Αλγόριθμος καθόδου κλίσης μίνι-παρτίδας (mini-batch gradient descent, MB-GD)

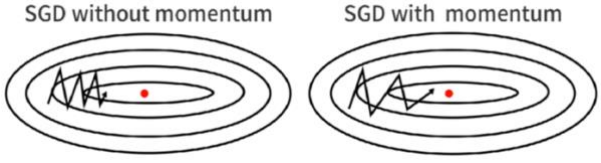
Ο MB-GD συνδυάζει με επιτυχία τα πλεονεκτήματα των δύο προαναφερθέντων αλγόριθμων και κάνει ανανέωση για κάθε μίνι-παρτίδα που αποτελείται από n δείγματα εκπαίδευσης:

$$\theta = \theta - af'(\theta; x_{i:i+n}; y_{i:i+n})$$

Ο διαχωρισμός του σετ εκπαίδευσης σε μίνι-παρτίδες βοηθά τον MB-GD να συγκλίνει με λιγότερες επαναλήψεις και πιο αποτελεσματικά από τον BGD.

Ορμή (Momentum)

Είδαμε προηγουμένως ότι ο SGD πλησιάζει το ελάχιστο της συνάρτησης κόστους κινούμενος με «ζιγκ-ζαγκ». Αυτές οι ταλαντώσεις καθυστερούν τη σύγκλιση, δεν επιτρέπουν μεγαλύτερους ρυθμούς εκπαίδευσης και έτσι μπορεί να κάνουν τον αλγόριθμο να αναπηδά συνεχώς γύρω από το ελάχιστο χωρίς να το φτάνει. Η τεχνική της ορμής επιταχύνει τον SGD προς τη σωστή κατεύθυνση και μειώνει τις ταλαντώσεις της κίνησης, Εικόνα 19



Εικόνα 19: Επίπτωση της ορμής στη Στοχαστική Κάθοδο Κλίσης

Μπορεί να παρομοιαστεί με τη ρίψη μιας μπάλας από ύψωμα: Όσο η μπάλα κυλάει προς τα κάτω, τόσο η ορμή της μεγαλώνει και επιταχύνει συνεχώς. Αντίστοιχα και ο παράγοντας της ορμής (συμβολίζεται με γ και συνήθως παίρνει τιμή 0.9) αυξάνει για διαστάσεις των οποίων οι κλίσεις «δείχνουν» προς τη σωστή κατεύθυνση και έτσι οι παράμετροι ανανεώνονται ταχύτερα.

Αλγόριθμος AdaGrad

Ο AdaGrad προσαρμόζει τον ρυθμό εκμάθησης στις παραμέτρους που σημαίνει ότι παράμετροι που σχετίζονται με συχνά εμφανιζόμενα χαρακτηριστικά ανανεώνονται λιγότερο σε σχέση με τις παραμέτρους των οποίων τα χαρακτηριστικά δεν εμφανίζονται συχνά. Με άλλα λόγια, παράμετροι με μεγάλες κλίσεις θα έχουν μικρό ρυθμό εκμάθησης ενώ παράμετροι με μικρές κλίσεις έχουν μεγαλύτερο ρυθμό εκμάθησης:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\alpha}{\sqrt{G_{t,ii} + \epsilon}} g_{t,i}$$

Δηλαδή ο AdaGrad μεταβάλλει τον γενικό ρυθμό εκμάθησης α σε κάθε χρονική στιγμή t και για κάθε παράμετρο θ βασιζόμενος στις περασμένες κλίσεις των θ.

Το κύριο πλεονέκτημα του AdaGrad είναι ότι δεν χρειάζεται χειροκίνητη ρύθμιση του ρυθμού εκμάθησης και συνήθως επιλέγεται η τιμή 0.01. Όμως καθώς ο παρονομαστής κατά την εκπαίδευση συνεχώς αυξάνει (επειδή κάθε όρος που προστίθεται είναι θετικός) σταδιακά συρρικνώνει το α με συνέπεια να καθυστερεί υπερβολικά η εκπαίδευση ή και να σταματάει.

Αλγόριθμος AdaDelta

Ο AdaDelta [41] είναι μια βελτιωμένη έκδοση του AdaGrad που εξαλείφει το πρόβλημα της συρρίκνωσης του ρυθμού εκμάθησης καθώς περιορίζει τον αριθμό των προηγούμενων κλίσεων σε ένα σταθερό παράθυρο μεγέθους w. Αντί να αποθηκεύονται οι w προηγούμενες κλίσεις ως άθροισμα τετραγώνων, το άθροισμα των κλίσεων αναδρομικά ορίζεται ως μια μειούμενη μέση τιμή όλων των προηγούμενων κλίσεων:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$$

Δηλαδή η τρέχουσα μέση τιμή $E[g^2]_t$ στο χρονικό σημείο εξαρτάται μόνο από την προηγούμενο μέση τιμή και την τρέχουσα κλίση. Συνήθως η τιμή που δίνεται στο γ είναι 0.9.

Εάν γράψουμε τον τύπο του SGD με όρους της παράμετρου $\Delta\theta_t$:

$$\Delta\theta_t = -\alpha g_{t,i}$$
$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

ο AdaDelta παίρνει τη μορφή:

$$\Delta\theta_t = -\frac{a}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Ένα πλεονέκτημα του AdaDelta είναι ότι δε χρειάζεται προκαθορισμένη τιμή του ρυθμού εκπαίδευσης α καθώς (σύμφωνα με τους εμπνευστές του αλγόριθμου, ο παραπάνω τύπος μπορεί να πάρει τη μορφή:

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

Όπου παρατηρούμε ότι εξαλείφεται ο ρυθμός εκπαίδευσης α.

Αλγόριθμος RMSprop

Ο RMSprop [41] είναι ένας αλγόριθμος που πρωτοπαρουσιάστηκε από τον Geoff Hinton σε μια διάλεξή του στη δημοφιλή πλατφόρμα διαδικτυακών μαθημάτων Coursera. Αναπτύχθηκε περίπου την ίδια χρονική περίοδο με τον AdaDelta, ανεξάρτητα, με τον ίδιο στόχο: να μειώσει το πρόβλημα του AdaGrad με τη ραγδαία και συνεχή μείωση του ρυθμού εκπαίδευσης. Έτσι, ο RMSprop διευθετεί το πρόβλημα διατηρώντας έναν κυλιόμενο μέσο όρο των τετραγώνων των κλίσεων και ρυθμίζοντας τις ανανεώσεις των βαρών βάσει αυτού του μεγέθους:

$$E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$$
$$\Delta\theta_{t+1} = \theta_t - \frac{a}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Με προτεινόμενες τιμές (από τον Hinton) γ = 0.9 και α = 0.001.

Αλγόριθμος Adam (Adaptive Moment Estimation)

Ο Adam [42] είναι ένας ακόμη αλγόριθμος βελτιστοποίησης που υπολογίζει προσαρμοστικούς ρυθμούς εκπαίδευσης για κάθε παράμετρο. Λειτουργεί όπως οι AdaDelta και RMSprop αλλά επιπλέον διατηρεί μια εκθετικά μειούμενη μέση τιμή των προηγούμενων κλίσεων m_t

παρόμοια με τον παράγοντα της ορμής (momentum). Ενώ η ορμή, όπως είδαμε προηγουμένως, ενεργεί σαν μια μπάλα που κυλάει σε μια πλαγιά, ο Adam συμπεριφέρεται σαν μια βαριά μπάλα με τριβή που υπερπηδά το τοπικό ελάχιστο και ηρεμεί στο επίπεδο ελάχιστο της επιφάνειας του σφάλματος. Η ανανέωση των βαρών γίνεται βάσει του τύπου:

$$w_t = w_{t-1} - \eta \frac{\widehat{m_t}}{\sqrt{\hat{u}} + \epsilon}$$

με

$$\widehat{m}_t = \frac{\widehat{m}_t}{1 - \beta_1^t}$$

$$\hat{u}_t = \frac{\hat{u}_t}{1 - \beta_2^t}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$u_t = \beta_2 u_{t-1} + (1 - \beta_2) g_t^2$$

όπου m_t = μέσος όρος για την κλίση g_t, u_t=μέσος όρος για το τετράγωνο της κλίσης g_t^2,

η= ρυθμός εκπαίδευσης και οι προτεινόμενες τιμές από τους εμπνευστές του αλγόριθμου είναι η = 0.001, β_1=0.9, β_2=0.999, $\epsilon = 10^{-8}$.

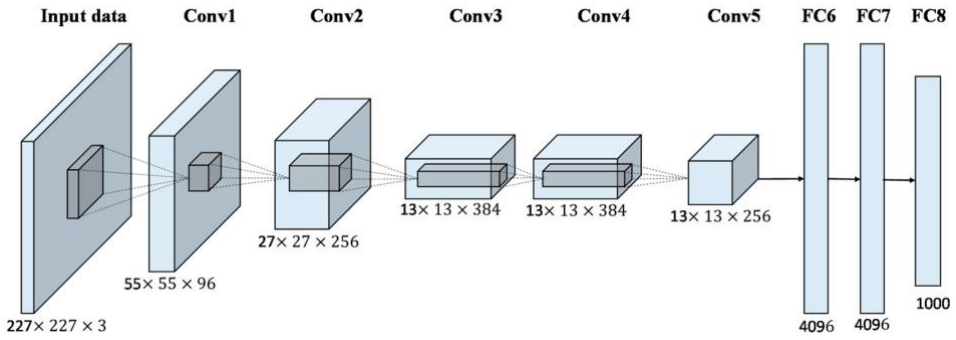
Ο Adam είναι ένας υπολογιστικά αποδοτικός αλγόριθμος, εύκολος στην υλοποίηση και οι απαιτήσεις του για μνήμη είναι περιορισμένες. Λειτουργεί καλά σε μεγάλα σετ δεδομένων με μεγάλες παραμέτρους καθώς και σε προβλήματα με θορυβώδεις ή αραιές (sparse) κλίσεις.

3.6 Αρχιτεκτονικές Βαθιών ΣΝΔ

Για τη διεξαγωγή των πειραμάτων της παρούσας εργασίας εκμεταλλευτήκαμε τις δυνατότητες 6 γνωστών αρχιτεκτονικών ΣΝΔ και συγκεκριμένα των ResNeXt101_32x8d, AlexNet, VGG16_bn, SqueezeNet1.0, DenseNet121, Inception_v3, τις οποίες θα παρουσιάσουμε στη συνέχεια.

3.6.1 AlexNet

Το AlexNet [43] κέρδισε τον διαγωνισμό ταξινόμησης εικόνας ILSVRC (ImageNet Large Scale Visual Recognition Challenge) το έτος 2012. Η αρχιτεκτονική του αποτελείται από οκτώ επίπεδα με τα πέντε πρώτα να είναι συνελικτικά ενώ τα τρία τελευταία είναι πλήρως συνδεδεμένα με το τελευταίο εξ' αυτών να είναι μια συνάρτηση ενεργοποίησης softmax. Τα συνελικτικά επίπεδα έχουν μέγεθος πυρήνων (kernel size) 11x11, 5x5 και 3x3, επίπεδα μέγιστης συγκέντρωσης (max pooling) για υποδειγματοληψία των εικόνων(downsampling) και επίπεδα απόσυρσης (dropout) για την αποφυγή της υπερπροσαρμογής. Στην Εικόνα 20 βλέπουμε την απεικόνιση της αρχιτεκτονικής του δικτύου.



Εικόνα 20: AlexNet

Στο AlexNet χρησιμοποιήθηκαν ορισμένα χαρακτηριστικά που δεν είχαν χρησιμοποιηθεί ξανά σε ΣΝΔ, όπως για παράδειγμα στο LeNet. Έτσι, το AlexNet χρησιμοποιεί τη συνάρτηση ενεργοποίησης ReLU για το για το μη γραμμικό μέρος (non linearities) σε αντίθεση με τις tanh και sigmoid που ήταν οι συνήθεις επιλογές σε πιο παραδοσιακά Νευρωνικά Δίκτυα. Η ReLU χρησιμοποιήθηκε στα κρυφά επίπεδα του δικτύου με συνέπεια την επιτάχυνση της εκπαίδευσης διότι η συνάρτηση αυτή μειώνει την πιθανότητα εμφάνισης του λεγόμενου προβλήματος της εξαφάνισης κλίσης (vanishing gradient) καθώς η παράγωγός της δεν είναι

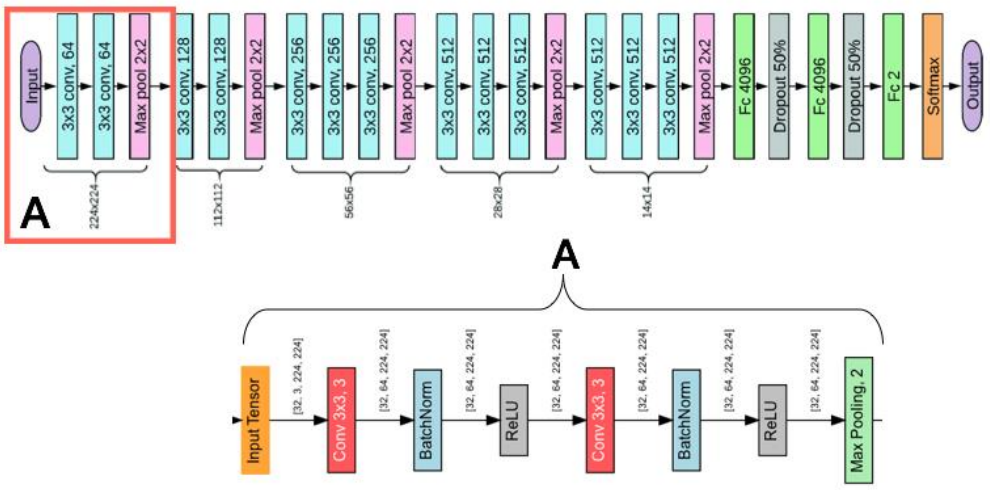
τόσο μικρή όσο των tanh, sigmoid και έτσι τα ανανεωμένα βάρη (κατά την οπισθοδιάδοση) δεν εξαφανίζονται.

Στο AlexNet, χρησιμοποιήθηκε για πρώτη φορά σε μεγάλη κλίμακα, η τεχνική του dropout στα δύο πλήρως συνδεδεμένα επίπεδα του δικτύου. Οι νευρώνες του επιπέδου που αποσύρονται δε συμμετέχουν στην εκπαίδευση ούτε κατά την εμπροσθοδιάδοση ούτε στην οπισθοδιάδοση. Με αυτόν τον τρόπο ο κάθε νευρώνας εξαναγκάζεται σε εκμάθηση πιο χρήσιμων χαρακτηριστικών συναρτήσει πολλών άλλων τυχαίων υποσυνόλων νευρώνων.

3.6.2 VGG16 (batch normalization)

Το δίκτυο VGG [44] κατασκευάστηκε το 2014 (όπου και κατέκτησε τη 2η θέση στο ILSVRC της ίδιας χρονιάς) από το Visual Geometry Group at Oxford University. Τα δομικά του στοιχεία είναι ίδια με αυτά των LeNet και AlexNet με τη διαφορά ότι το VGG είναι ακόμα βαθύτερο δίκτυο με περισσότερα συνελικτικά, συγκέντρωσης και πλήρως συνδεδεμένα επίπεδα. Το VGG εμφανίζεται βελτιωμένο σε σχέση με το AlexNet καθώς έχει αντικαταστήσει τα μεγάλα φίλτρα (των 11x11 και 5x5 στα πρώτα 2 συνελικτικά επίπεδα) με πολλαπλά 3x3 φίλτρα απόσυρσης.

Τα VGG είναι μια οικογένεια παρόμοιων αρχιτεκτονικών με τα νούμερα δίπλα στο VGG να υπονοούν τον αριθμό των επιπέδων τους. Έτσι υπάρχουν τα VGG11, VGG13, VGG16, VGG19. Στα πειράματα της παρούσας εργασίας χρησιμοποιήσαμε το VGG16 στην batch-norm εκδοχή του, Εικόνα 21:



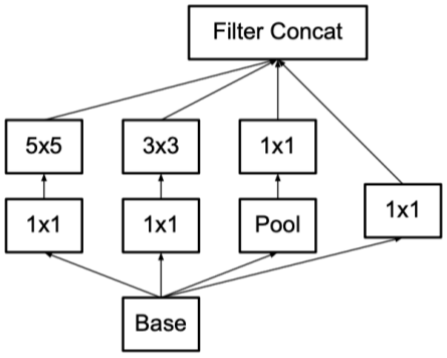
Εικόνα 21: VGG16_bn, όπου μετά από κάθε Conv ακολουθεί BatchNorm

Batch Normalization: Είναι ο μηχανισμός που μειώνει την εσωτερική μετατόπιση μεταβλητής (*Internal Covariance Shift*). Η εσωτερική μετατόπιση μεταβλητής είναι η αλλαγή των κατανομών των εσωτερικών κόμβων σε ένα βαθύ δίκτυο κατά τη διάρκεια της εκπαίδευσης. Η εξάλειψή της υπόσχεται γρηγορότερη εκπαίδευση. Αυτό επιτυγχάνεται μέσα από την κανονικοποίηση των μέσων και των διακυμάνσεων των εισόδων των επιπέδων. Επίσης μειώνει την εξάρτηση των παραγώγων στην κλίμακα των παραμέτρων ή των αρχικών τους τιμών. Αυτό μας επιτρέπει να δουλέψουμε με υψηλότερους ρυθμούς εκπαίδευσης χωρίς τον κίνδυνο της απόκλισης. Επίσης, το batch-norm εξομαλύνει το μοντέλο μειώνοντας έτσι την ανάγκη για dropout.

3.6.3 Inception v3

Η αρχιτεκτονική των Inception δικτύων πρωτοπαρουσιάστηκε από τους Szegedy et al. 2015a με το GoogLeNet (Inception-v1) και στα επόμενα χρόνια ακολούθησαν και άλλες εκδόσεις. Στην παρούσα εργασία χρησιμοποιήσαμε την έκδοση Inception-v3 [45] (Szegedy et al. 2015b), επομένως θα είναι και αυτή που ακολούθως θα αναλυθεί.

Το Inception-v3 μέσω της τμηματοποίησης των μεγάλων συνελίξεων των προηγούμενων εκδόσεων προσπαθεί να μειώσει το υπολογιστικό κόστος χωρίς να επηρεάζει τη γενίκευση του μοντέλου. Έτσι, τα μεγάλα φίλτρα των προηγούμενων μοντέλων (στα inception modules) όπως 5x5, 7x7 αντικαταστάθηκαν με μικρότερα 1x7, 1x5 και συνελίξεις 1x1 χρησιμοποιήθηκαν πριν από αυτά ως στενωποί (bottlenecks), Εικόνα 22.



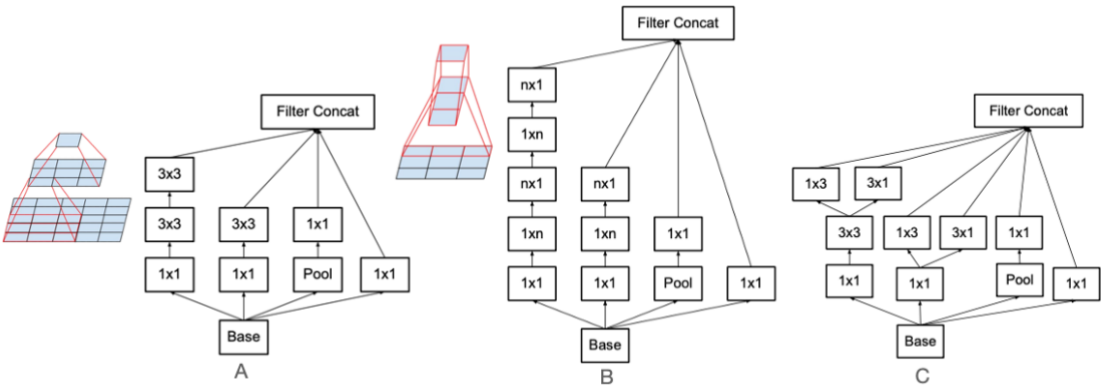
Εικόνα 22: Το πρωτότυπο inception module, του Inception-v1

Οι στενωποί μέσω της διαδικασίας split – transform – merge οδηγούν την είσοδο σε 3-4 διαφορετικούς χάρτες χαρακτηριστικών μικρότερων ή ίσων διαστάσεων με αυτόν της εισόδου και στη συνέχεια τους οδηγούν μέσω 3x3 ή 5x5 συνελίξεων σε μικρότερους χάρτες τριών διαστάσεων. Στον ακόλουθο Πίνακας 1 εμφανίζεται ο σκελετός του Inception-v3 με τα Inception modules A, B, C.

Πίνακας 1: Inception-v3, με τα Inception modules A,B,C

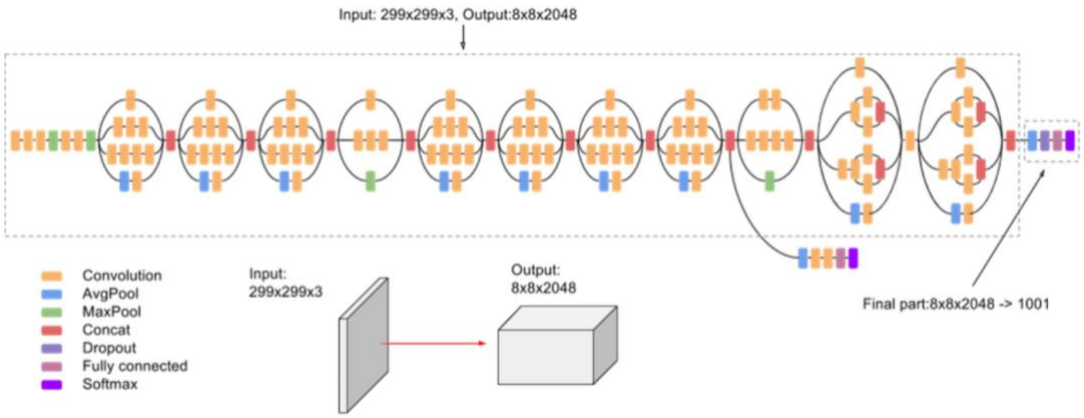
type	patch size/stride or remarks	input size	
conv	3×3/2	299×299×3	
conv	3×3/1	149×149×32	
conv padded	3×3/1	147×147×32	
pool	3×3/2	147×147×64	
conv	3×3/1	73×73×64	
conv	3×3/2	71×71×80	
conv	3×3/1	35×35×192	
3×Inception	As in figure 5	35×35×288	module A
5×Inception	As in figure 6	17×17×768	module B
2×Inception	As in figure 7	8×8×1280	module C
pool	8 × 8	8 × 8 × 2048	
linear	logits	1 × 1 × 2048	
softmax	classifier	1 × 1 × 1000	

Στην Εικόνα 23 βλέπουμε ένα inception module A όπου η συνέλιξη 5x5 της αρχικής δομής έχει αντικατασταθεί από δύο 3x3 συνελίξεις. Ένα inception module B όπου συνέλιξη 7x7 αντικαθίσταται από 2 συνελίξεις 1x7 και 7x1 (στο εικονιζόμενο παράδειγμα n=7 και οι συνελίξεις 7x7 είναι τρείς). Ένα inception module C όπου χρησιμοποιείται για τις υψηλών διαστάσεων αναπαραστάσεις και αντί τα φίλτρα να τοποθετούνται σε στοίβα τοποθετούνται δίπλα-δίπλα.



Εικόνα 23: Inception-v3 modules A,B,C

Παρόλο που το δίκτυο τελικά έχει βάθος 48 επιπέδων, Εικόνα 24, το υπολογιστικό κόστος είναι μόνο 2.5 υψηλότερο από το Inception-v1 και πολύ πιο αποτελεσματικό από το VGG.

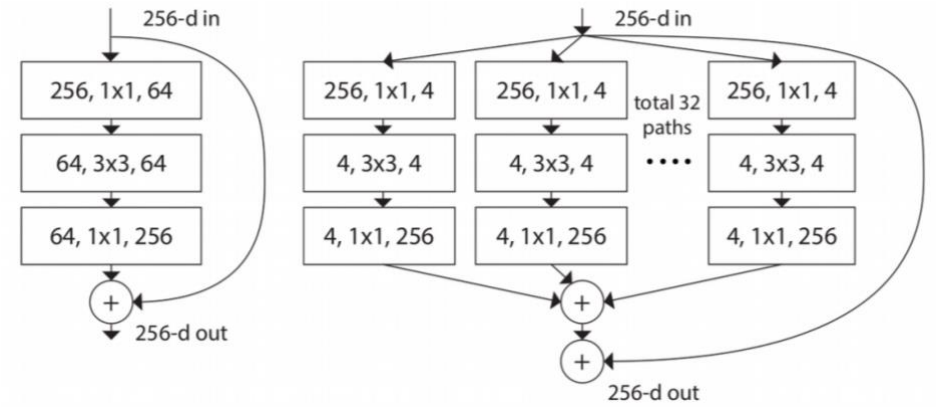


Εικόνα 24: Η αρχιτεκτονική του Inception-v3

3.6.4 ResNeXt 101_32x8d

Το ResNeXt [46] παρουσιάστηκε το 2017 στο Συνέδριο CVPR (Computer Vision and Pattern Recognition) από τους Saining Xie et al. και είναι ένα υπερσύγχρονο μοντέλο στον τομέα της ταξινόμησης στο ImageNet. Δουλεύει καλύτερα από το ResNet καθώς είναι μια αναβαθμισμένη έκδοσή του και συνδυάζει τις στοίβες των συνελικτικών επιπέδων του VGG και την ιδέα του split-transform-merge που χρησιμοποιούν τα inception μοντέλα.

Στο ResNeXt υπάρχουν 4 επίπεδα και κάθε επίπεδο έχει μερικά residual blocks. Κάθε τέτοιο υπολειμματικό μπλοκ έχει αυξημένο πλάτος όπου περισσότερα φίλτρα δημιουργούν πολλαπλά παράλληλα μονοπάτια και το σύνολο αυτών των μονοπατιών το ονόμασαν πληθικότητα (cardinality). Ουσιαστικά παίρνουμε ένα υπολειμματικό μπλοκ με στένωση και το κάνουμε λιγότερο βαθύ αλλά περισσότερο πλατύ όπως φαίνεται στην Εικόνα 25.

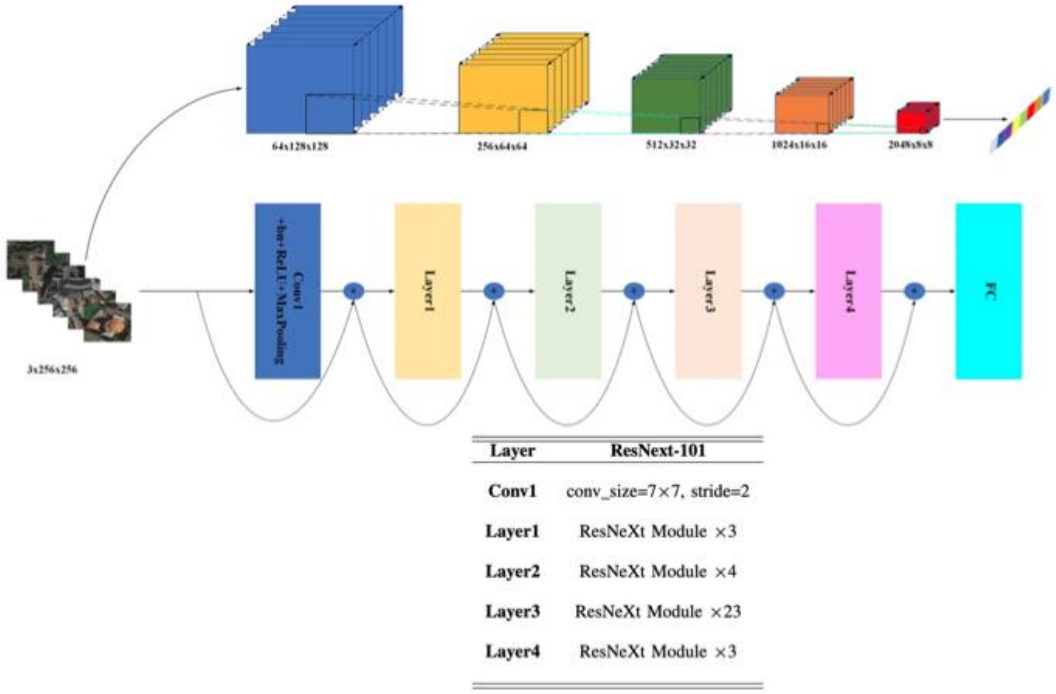


Εικόνα 25: Residual bottleneck (ResNet) (αρ), ResNeXt block (δε)

Όπως παρατηρούμε στο αριστερό μέρους της Εικόνα 25 έχουμε ένα υπολειμματικό μπλοκ με στένωση (του Resnet) και στο δεξί μέρος τον ισοδύναμο μετασχηματισμό του μπλοκ όπως εκφράζεται στο ResNeXt. Η διαδικασία έχει ως εξής: Ο αρχικός πίνακας όπου από $256 \overset{1x1}{\rightrightarrows} 64$ χωρίζεται σε 32 μικρότερους πίνακες όπου από $256 \overset{1x1}{\rightrightarrows} 4$. Στη συνέχεια αυτό το αποτέλεσμα (κάθε πίνακα) οδηγείται στο επόμενο επίπεδο όπου έχουμε μετασχηματισμούς $4 \overset{3x3}{\rightrightarrows} 4$ πάλι χωρισμένους σε 32 πίνακες. Αυτός ο μετασχηματισμός είναι πολύ φτηνότερος υπολογιστικά από τον αντίστοιχο (στο 2ο επίπεδο) του bottleneck του ResNet. Το αποτέλεσμα των συνελίξεων οδηγείται στο επόμενο επίπεδο των 32 πινάκων όπου από τις 4 διαστάσεις μεγεθύνουμε πάλι στις 256 δηλαδή μετασχηματίζουμε από $4 \overset{1x1}{\rightrightarrows} 256$. Τέλος όλα τα παραπάνω συνενώνονται και επιπλέον προστίθεται και η είσοδος (256-d in) και αυτό θα είναι η έξοδος 256-d out.

Το ResNeXt ενώ έχει τον ίδιο αριθμό παραμέτρων με ένα αντίστοιχο ResNet, τα χαρακτηριστικά που εξάγονται από το ResNeXt έχουν καλύτερη απόδοση στο ImageNet classification task σε σχέση με αυτά του ResNet που σημαίνει ότι έχουν πολύ ισχυρότερη ικανότητα. Επιπλέον το ResNeXt-101_32x8d

(δηλ 101 layers, cardinality=32, bottleneck width=8), που χρησιμοποιούμε και στα πειράματα της παρούσας εργασίας, πετυχαίνει κορυφαία απόδοση στο ImageNet χωρίς fine-tuning και χωρίς έξτρα δεδομένα εκπαίδευσης. Η αρχιτεκτονική του εν λόγω δικτύου φαίνεται στην Εικόνα 26.

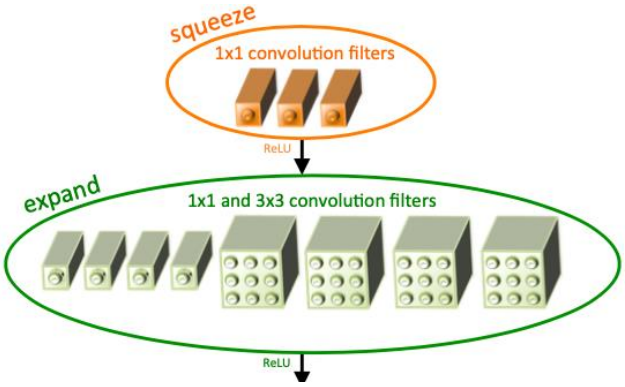


Layer	ResNext-101
Conv1	conv_size=7×7, stride=2
Layer1	ResNeXt Module ×3
Layer2	ResNeXt Module ×4
Layer3	ResNeXt Module ×23
Layer4	ResNeXt Module ×3

Εικόνα 26: Αρχιτεκτονική ResNeXt-101

3.6.5 SqueezeNet 1.0

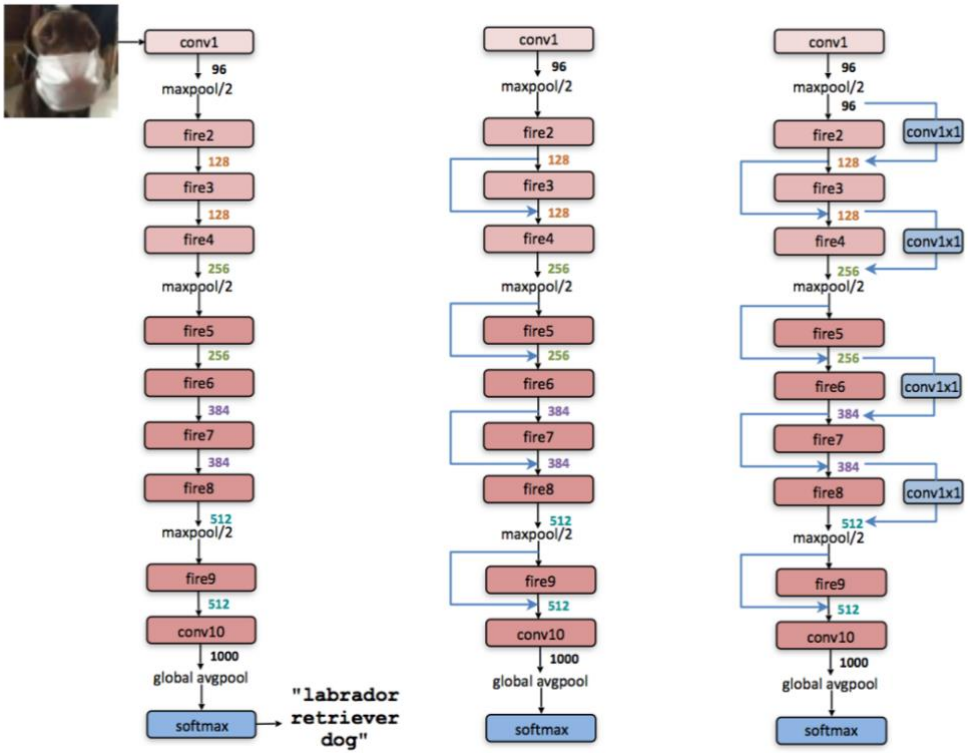
Η αρχιτεκτονική του SqueezeNet [47] παρουσιάστηκε στο άρθρο των F. Iandola et al. το 2016 οι οποίοι πρότειναν ένα μικρό δίκτυο το οποίο έφτανε τις επιδόσεις του AlexNet στο ImageNet με 50x λιγότερες παραμέτρους. Επίσης, με τεχνικές συμπίεσης του μοντέλου κατάφεραν να το συμπιέσουν σε λιγότερο από 0.5MB που είναι 510x μικρότερο από το AlexNet.



Εικόνα 27: Fire module του δικτύου SqueezeNet

Το βασικό δομικό στοιχείο του SqueezeNet είναι το λεγόμενο Fire module. Το Fire module, όπως φαίνεται Εικόνα 27, αποτελείται από ένα squeeze συνελικτικό επίπεδο που έχει μόνο 1x1 φίλτρα το οποίο οδηγείται σε ένα διευρυμένο επίπεδο που έχει μια μίξη από φίλτρων 1x1, 3x3. Το Fire module έχει τρεις υπερπαραμέτρους όπου η ρύθμισή τους βοηθά το squeeze επίπεδο να μειώσει τον αριθμό των εισόδων στα 3x3 φίλτρα βάσει τριών στρατηγικών.

Η αρχιτεκτονική του SqueezeNet, Εικόνα 28, ξεκινά με ένα αυτόνομο συνελικτικό επίπεδο (conv1), που ακολουθείται από 8 Fire modules (fire2-9) και καταλήγει σε ένα τελικό συνελικτικό (conv10). Η αύξηση του αριθμού των φίλτρων γίνεται σταδιακά ανά Fire module και από την αρχή προς το τέλος.

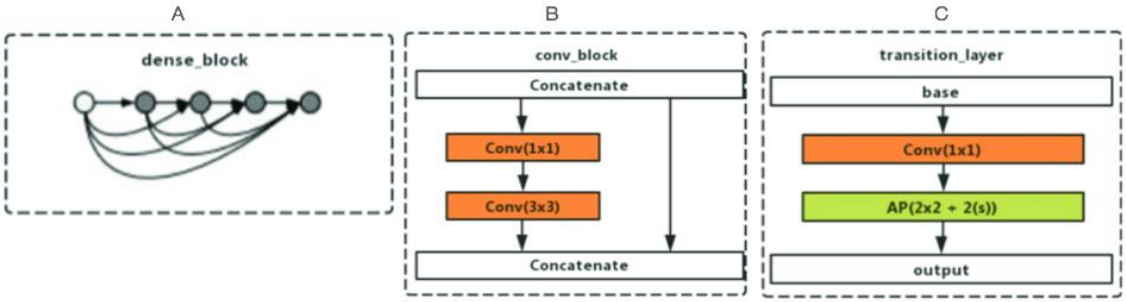


Εικόνα 28: Η αρχιτεκτονική του SqueezeNet 1.0

3.6.6 DenseNet 121

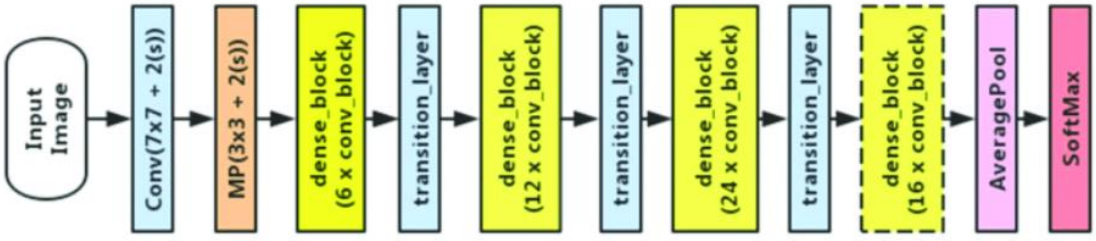
Η αρχιτεκτονική DenseNet [48] παρουσιάστηκε το 2018 από τους G.Huang et al. και βασίζεται στην παραδοχή ότι τα ΣΝΔ μπορούν να είναι ακόμα βαθύτερα, ακριβέστερα και αποτελεσματικότερα στην εκπαίδευσή τους αν έχουν κοντύτερες συνδέσεις μεταξύ των επιπέδων των κοντινών στην είσοδο και στην έξοδο. Έτσι, στην αρχιτεκτονική του DenseNet κάθε επίπεδο συνδέεται με κάθε

άλλο επίπεδο. Για κάθε L επίπεδα υπάρχουν $L(L+1)/2$ απευθείας συνδέσεις. Σε κάθε επίπεδο, τα χαρακτηριστικά όλων των προηγούμενων επιπέδων γίνονται η είσοδός του και τα δικά του χαρακτηριστικά γίνονται είσοδος στα επόμενα επίπεδα. Για να λειτουργήσει αυτή η συνένωση των χαρακτηριστικών θα πρέπει και οι αντίστοιχοι χάρτες χαρακτηριστικών (feature maps) να είναι των ίδιων διαστάσεων ή να υποστούν υποδειγματοληψία (downsampling) για να το πετύχουν. Έτσι, οι δημιουργοί του DenseNet δημιούργησαν και τοποθέτησαν μέσα στο δίκτυο τα λεγόμενα πυκνά μπλοκ (Dense blocks) μέσα στα οποία το μέγεθος του χάρτη χαρακτηριστικών παραμένει ίδιο. Κάθε dense block αποτελείται από μια σειρά συνελικτικών επιπέδων 1x1 και 3x3, τα conv block. Η συνέλιξη και η συγκέντρωση (convolution & pooling) γίνονται σε διαφορετικά επίπεδα που βρίσκονται μεταξύ των dense blocks ονομάζονται transition layers (επίπεδα μετάβασης) και περιλαμβάνουν ένα batch-norm, ένα 1x1 conv και ένα 2x2 average pooling επίπεδο. Στην ακόλουθη Εικόνα 29 φαίνονται τα dense block, conv block, transition layer.



Εικόνα 29: Dense block (A), conv block (B), transition layer (C), όλα δομικά στοιχεία του DenseNet

Το μοντέλο του DenseNet που χρησιμοποιήσαμε στα πειράματα αυτής της εργασίας είναι το DenseNet 121 που ο σκελετός του απεικονίζεται στην Εικόνα 30.



Εικόνα 30: Η αρχιτεκτονική του DenseNet 121

4 Παραγωγικά Ανταγωνιστικά Δίκτυα (GANs)

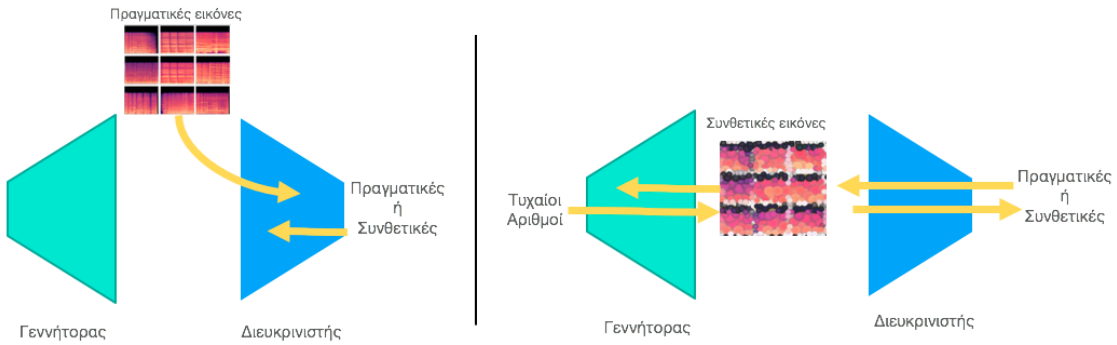
Το 2014 ο Ian Goodfellow δημοσιεύει ένα άρθρο [49] στο οποίο δύο ξεχωριστά νευρωνικά δίκτυα, ανταγωνιζόμενα μεταξύ τους, παράγουν συνθετικά δεδομένα όμοιων χαρακτηριστικών με αυτά των πρωτότυπων. Από τότε βρίσκονται στο προσκήνιο καθώς το ενδιαφέρον των ερευνητών για συνθετικά υπερ-ρεαλιστικά δεδομένα εικόνας, βίντεο, μουσικής, κειμένου κ.α. συνεχώς αυξάνεται.

4.1 Βασική ιδέα και εκπαίδευση των ΠΑΔ

Τα παραγωγικά δίκτυα προϋπήρχαν των ανταγωνιστικών και η βασική ιδέα πίσω από αυτά είναι ότι υπάρχει μια κατανομή δειγμάτων, έστω φασματογραφημάτων όπως στην παρούσα εργασία, που θέλουμε να την μοντελοποιήσουμε εκπαιδεύοντας έναν γεννήτορα (generator) πάνω στα δείγματά της. Αν η εκπαίδευση είναι επιτυχημένη τότε μπορούμε να παράγουμε νέες εικόνες που μοιάζουν με τις πρωτότυπες αλλά ταυτόχρονα είναι και διαφορετικές.

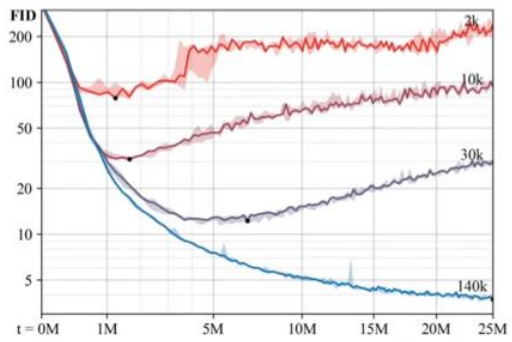
Κάθε ΠΑΔ αποτελείται από δύο δίκτυα: τον Γεννήτορα (Generator) και τον Διευκρινιστή (Discriminator) τα οποία συνεργάζονται (ανταγωνιστικά). Ο γεννήτορας παράγει τις εικόνες και ο διευκρινιστής συγκρίνει τις παραγόμενες εικόνες με τις πρωτότυπες (του σετ εκπαίδευσης). Η εκπαίδευση και των δύο ξεκινά τυχαία και… άσκοπα καθώς ο γεννήτορας ξεκινά την παραγωγή δειγματίζοντας από έναν χώρο τυχαίων αριθμών (συνήθως ονομάζεται latent space) και ο διευκρινιστής καταλαβαίνει τι πρέπει να κάνει όσο προχωρά η εκπαίδευση και την πληροφορία τη μεταφέρει στον γεννήτορα προκειμένου να βελτιώσει την ποιότητα των εικόνων. Η εκπαίδευση γίνεται εναλλάξ (Εικόνα 31): Ο γεννήτορας τρέχει πρώτος μερικές φορές την παραγωγή εικόνων από το latent space με τις οποίες τροφοδοτεί τον διευκρινιστή. Τότε ο διευκρινιστής τις βλέπει

μαζί με μερικές (πραγματικές) εικόνες από το σετ εκπαίδευσης και αρχίζει η εκπαίδευσή του προκειμένου να τις ταξινομήσει σωστά. Ο γενικός στόχος της εκπαίδευσης είναι η ταξινόμηση στον διευκρινιστή να γίνεται λανθασμένα δηλαδή να θεωρεί τις συνθετικές εικόνες ως αληθινές. Έτσι, με την οπισθοδιάδοση, το σφάλμα των εικόνων επιστρέφει στον γεννήτορα προκειμένου να βελτιώσει την ποιότητα των παραγόμενων εικόνων. Η επαναληπτική αυτή διαδικασία οδηγεί στη σταδιακή βελτίωση της ποιότητας των συνθετικών εικόνων ώστε στο τέλος να φαίνονται ως πραγματικές (αλλά όχι ίδιες με τις πρωτότυπες).



Εικόνα 31: Συνοπτική απεικόνιση εκπαίδευσης Διευκρινιστή (αρ), Γεννήτορα (δε) ενός ΠΑΔ

Δυστυχώς, τα ΠΑΔ είναι πολύ ιδιότροπα στην εκπαίδευσή τους. Ανάλογα με τον τύπο τους, άλλο απλώς καταρρέει κατά τη διάρκεια της εκπαίδευσης, άλλο δε συγκλίνει ποτέ ή συμβαίνουν διάφορα άλλα προβλήματα. Μια βασική πηγή των προβλημάτων εκπαίδευσης είναι η ελλιπής ποσότητα δεδομένων. Στην Εικόνα 32 απεικονίζεται η πορεία της εκπαίδευσης σετ δειγμάτων διαφόρων μεγεθών (2k, 10k, 30k, 140k) συναρτήσει της μετρικής FID [50] (Frechet Inception Distance) και των δειγμάτων (σε εκατομμύρια) που δείχνονται στον διευκρινιστή. Όσο μικρότερη είναι η τιμή της FID τόσο καλύτερη σύγκλιση επιτυγχάνεται και τόσο καλύτερη είναι η ποιότητα των συνθετικών εικόνων που παράγονται. Παρατηρούμε ότι η μικρότερη FID εμφανίζεται κατά την εκπαίδευση του μεγαλύτερου σετ της δοκιμής (140k).



Εικόνα 32: Διάγραμμα εκπαίδευσης ΠΑΔ σε σετ διαφόρων μεγεθών, συναρτήσει FID.

Το πρόβλημα που προαναφέραμε καλείται να λύσει το StyleGAN2-ADA που θα δούμε στη συνέχεια.

4.2 StyleGAN2-ADA (SG2A)

Το StyleGAN2-ADA (2020) [51] είναι το πιο πρόσφατο Παραγωγικό Ανταγωνιστικό Δίκτυο (ΠΑΔ) που δημιουργήθηκε από τον Teo Karras και την ομάδα ερευνητών της NVIDIA μετά τα Progressive GAN [52] (2017) , StyleGAN (2018), StyleGAN2(2019). Τα StyleGAN [53] είναι μια επέκταση της βασικής αρχιτεκτονικής των ΠΑΔ που έχουν όμως τη δυνατότητα διαχωρισμού επί των ξεχωριστών χαρακτηριστικών της παραγόμενης εικόνας. Είναι η εξέλιξη των Progressive GAN που ήταν ήδη ικανά να συνθέσουν εικόνες υψηλής ανάλυσης με τη σταδιακή ανάπτυξη των δικτύων του Διευκρινιστή και του Γεννήτορα κατά την εκπαιδευτική διαδικασία.

Το SG2A επιτυγχάνει την παραγωγή εικόνων πολύ υψηλής ανάλυσης από μικρά σετ εκπαίδευσης χωρίς προβλήματα υπερπροσαρμογής μέσω μιας νέας μεθόδου, της λεγόμενης Adaptive Discriminator Augmentation (Προσαρμοστική Επαύξηση Διευκρινιστή).

4.2.1 Υπερπροσαρμογή στα ΠΑΔ

Υπερπροσαρμογή στα ΠΑΔ γίνεται όταν ο Διευκρινιστής παύει να λειτουργεί αποδοτικά δηλαδή, όταν η πληροφορία που δίνει στον Γεννήτορα είναι άνευ ουσιαστικής σημασίας οπότε και οι παραγόμενες εικόνες από ένα σημείο και μετά συνεχώς χειροτερεύουν.

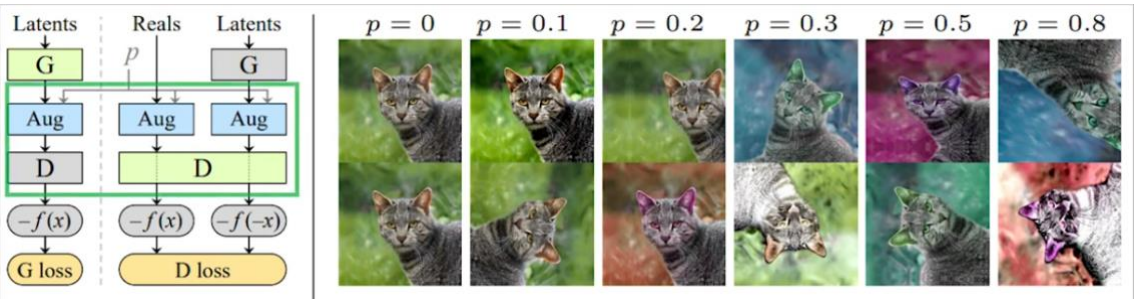
Στα ΣΝΔ ένας τρόπος αποφυγής της υπερπροσαρμογής κυρίως όταν έχουμε μικρό σετ εκπαίδευσης είναι να προβαίνουμε σε επαύξηση των δεδομένων, όπως

έγινε και στα πειράματα της παρούσας εργασίας. Λέγοντας επαύξηση εννοούμε διάφορους μετασχηματισμούς στις ήδη υπάρχουσες εικόνες όπως περιστροφή, προσθήκη ή και αλλαγή χρωμάτων, προσθήκη θορύβου και πολλούς άλλους. Το Pytorch για παράδειγμα διαθέτει πληθώρα τέτοιων μετασχηματισμών.

Όμως, αυτή η μέθοδος δεν μπορεί να χρησιμοποιηθεί αυτούσια στα ΠΑΔ καθώς ο Γεννήτορας θα παρήγαγε και τις μετασχηματισμένες εικόνες κάτι που δεν είναι επιθυμητό. Έπρεπε να βρεθεί μια μέθοδος που από τη μια να αποφεύγει την υπερπροσαρμογή κάνοντας χρήση των μετασχηματισμών των εικόνων και από την άλλη να διασφαλίζει ότι αυτοί δε θα παρεισφρήσουν στις παραγόμενες εικόνες.

4.2.2 Μέθοδος ADA

Στην Εικόνα 33 βλέπουμε το σκεπτικό της μεθόδου [51] όπου με μπλε χρώμα συμβολίζονται οι διεργασίες των επαυξήσεων, με πράσινο το δίκτυο που εκπαιδεύεται, με πορτοκαλί οι συναρτήσεις απωλειών και στα δεξιά η επίδραση της πιθανότητας (p) στους μετασχηματισμούς.



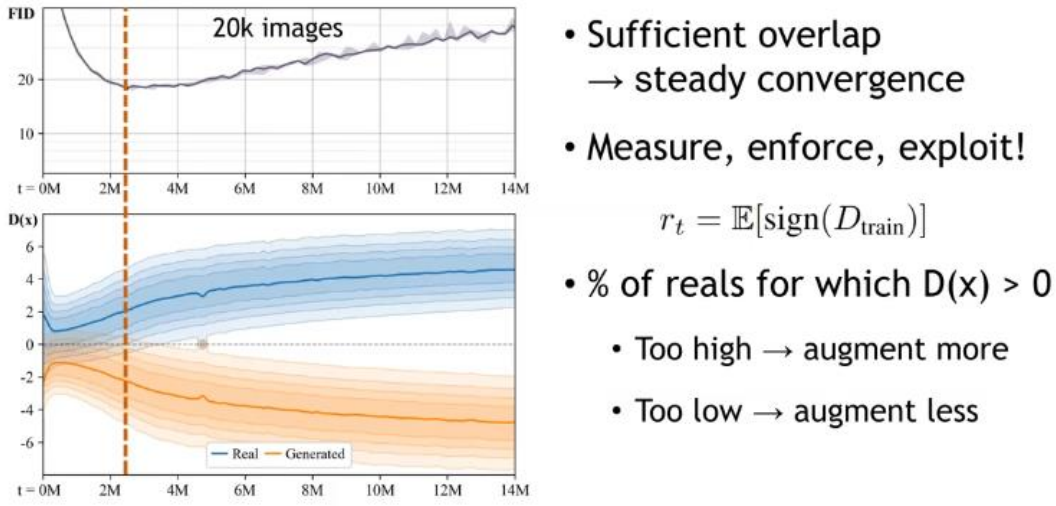
Εικόνα 33: Διάγραμμα ροής της ADA και δεξιά η πιθανότητα επαύξησης (p)

Όπως παρατηρούμε, όλες οι εικόνες που δέχεται ο Διευκρινιστής (D) είναι παραλλαγμένες με μια προεπιλεγμένη πιθανότητα (p) για κάθε παραλλαγή η οποία συμβαίνει τυχαία και η απόδοση του Διευκρινιστή εκτιμάται από αυτές τις εικόνες. Από την άλλη πλευρά, ο Γεννήτορας (G) εκπαιδεύεται στην παραγωγή μόνο καθαρών εικόνων εφόσον η πιθανότητα (p) παραμένει κάτω από το όριο ασφαλείας. Τα πειράματα των δημιουργών του StyleGAN2-ADA έδειξαν ότι οι διαρροές στον G αρχίζουν να εμφανίζονται όταν το p είναι πολύ κοντά στη μονάδα με το όριο ασφαλείας να είναι όταν $p<0.8$.

Δηλαδή, όσο μεγαλύτερη η πιθανότητα τόσο περισσότερες οι επαυξήσεις και τόσο περισσότερο παραλλαγμένο σετ θα πάρουμε. Ωστόσο, η χειροκίνητη

ρύθμιση της υπερπαράμετρου p είναι δύσκολη διαδικασία γι' αυτό οι δημιουργοί προχώρησαν στην αυτοματοποίηση της ιδανικής επιλογής της τιμής της.

Αυτή τη διαδικασία ελέγχου της έντασης της επαύξησης ονόμασαν ADA και είναι μια προσαρμοστική μέθοδος που βασίζεται στη διόρθωση της τιμής ενός ευριστικού r_t.



Εικόνα 34: Μέθοδος ADA

Όπως βλέπουμε στην Εικόνα 34 όσο υπάρχει ικανοποιητική επικάλυψη των κατανομών (real – generated) των εικόνων στην έξοδο του Διευκρινιστή (D) τόσο αυτός γίνεται καλύτερος και οι προβλέψεις του σωστότερες. Το ευριστικό βάσει αυτής της επικάλυψης (που εύκολα μετριέται καθ' όλη τη διάρκεια της εκπαίδευσης) υπολογίζει το ποσοστό των πραγματικών εικόνων για τις οποίες ισχύει D(x) > 0. Αν ο μέσος όρος είναι πολύ υψηλός (σημαίνει ότι οι κατανομές δεν επικαλύπτονται πια) τότε η τιμή της p ρυθμίζεται για περισσότερη επαύξηση εικόνων, αν είναι πολύ μικρός η p ρυθμίζεται για μικρότερη επαύξηση.

5 Πειραματική Διαδικασία

Σε αυτό το κεφάλαιο περιγράφεται λεπτομερειακά η πειραματική διαδικασία από την προετοιμασία των σετ δεδομένων έως την εκτέλεση των πειραμάτων Α, Β, Γ και την εξαγωγή συμπερασμάτων.

5.1 Προέλευση και προεπεξεργασία των σετ δεδομένων

Για τις ανάγκες των πειραμάτων της εργασίας χρησιμοποιήθηκαν δύο σετ μουσικών δεδομένων που εφεξής για διευκόλυνση θα ονομάζουμε big-set και 360-set.

Big-set

Το big-set περιλαμβάνει 17000 ηχητικά αποσπάσματα τραγουδιών σε ροκ και ποπ στυλ διάρκειας 10 δευτερολέπτων το καθένα, από τυχαίο σημείο του κάθε τραγουδιού. Η μορφή των αρχείων είναι wav, μονοφωνικά, και έχουν ρυθμό δειγματοληψίας 8kHz. Όλο το big-set ανήκει σε ιδιωτική συλλογή (όχι δημόσια διαθέσιμη) με τα μεταδεδομένα του (metadata) να προέρχονται από το Spotify.

Το Spotify είναι μια πολύ δημοφιλής διαδικτυακή πλατφόρμα αναπαραγωγής ψηφιακής μουσικής. Μέσω του API (Application Programming Interface) της επιτρέπει στους χρήστες να εξερευνούν και να εξάγουν μουσικά χαρακτηριστικά από μουσικές βάσεις δεδομένων επιλέγοντας από τα εκατομμύρια κομματιών της πλατφόρμας.

Οι κατηγορίες των χαρακτηριστικών είναι οι εξής και αφορούν κάθε μουσικό απόσπασμα ξεχωριστά: Acousticness (αν το κομμάτι είναι ακουστικό), Danceability (αν είναι χορευτικό), Energy (ένταση ενεργειακής δραστηριότητας), Instrumentalness (αν περιέχει φωνητικά), Liveness (αν υπάρχει ακροατήριο στην ηχογράφηση), Loudness (η μέση ένταση σε dB), Speechiness (ανίχνευση ομιλίας), Valence (μουσικό σθένος), Tempo (εκτίμηση ρυθμικού χαρακτήρα). Από αυτά τα χαρακτηριστικά χρήσιμα για εμάς ήταν τα Energy και Valence.

Κατά το Spotify API, τα Energy και Valence αναλύονται ως εξής:

Energy (Ενέργεια): Η ενέργεια μετριέται σε κλίμακα από 0.0 έως 1.0 και αναπαριστά το μέτρο της αντιληπτής έντασης και δραστηριότητας της μουσικής. Τα ενεργητικά κομμάτια ως επί το πλείστον, τα αντιλαμβανόμαστε ως γρήγορα, δυνατά και θορυβώδη. Για παράδειγμα ένα death metal κομμάτι έχει υψηλή ενέργεια αλλά ένα πρελούδιο του Bach πολύ χαμηλή. Τα αντιληπτά χαρακτηριστικά που χαρακτηρίζουν την ενέργεια περιλαμβάνουν το εύρος της δυναμικής περιοχής, την ένταση, το ηχόχρωμα, τον ρυθμό μουσικών συμβάντων ανά δευτερόλεπτο και τη γενική εντροπία.

Valence (Σθένος): Το σθένος μετριέται σε κλίμακα από 0.0 έως 1.0 και εκφράζει τη θετικότητα που απορρέει από το μουσικό κομμάτι. Αποσπάσματα με υψηλό σθένος ακούγονται θετικότερα (χαρά, ευφορία) ενώ κομμάτια με χαμηλό σθένος ακούγονται πιο αρνητικά (λύπη, θυμός, μελαγχολία).

Όλα τα μουσικά αποσπάσματα του big-set είναι πλήρως ετικετοποιημένα με τα προαναφερθέντα χαρακτηριστικά του Spotify API.

360-set

Το 360-set αποτελείται από 360 μουσικά αποσπάσματα σε μορφή mp3, στερεοφωνικά, με ρυθμό δειγματοληψίας 44.1kHz, διαρκούν από 15 έως 30 δευτερόλεπτα και είναι αυστηρά οργανικά δηλαδή δεν περιέχουν κάποιο τραγούδι ή απαγγελία στίχων ή ομιλία.

Τα αποσπάσματα του 360-σετ είναι κινηματογραφική μουσική από 110 ταινίες (γνωστές αλλά και όχι τόσο γνωστές) που επιλέχθηκαν και χρησιμοποιήθηκαν από τους Eerola & Vuoskoski [5] στη μελέτη τους πάνω στο διακριτό και διαστατικό μοντέλο κατάταξης των μουσικών συναισθημάτων. Η πρωτότυπη ονομασία του σετ είναι “Stimulus Set 1” και η πρόσβαση σε αυτό είναι ελεύθερη. Η επιλογή των αποσπασμάτων έγινε από ειδικούς (επαγγελματίες μουσικούς, καθηγητές και φοιτητές μουσικών Πανεπιστημιακών σχολών) οι οποίοι τα κατέταξαν-βαθμολόγησαν σε πέντε κατηγορίες του διακριτού (χαρά-happy, λύπη-sad, τρυφερότητα-tender, φόβος-fearful, θυμός-angry) και σε τρεις κατηγορίες του διαστατικού μοντέλου (σθένος-valence, ενέργεια-energy, ένταση-tension) κατάταξης του συναισθήματος στη μουσική.

Η επιλογή του 360-σετ ως σημείου αναφοράς (ground truth) της παρούσας εργασίας έγινε λόγω του ότι είναι πλήρως επισημειωμένο από ειδικούς στον τομέα της μουσικής, γεγονός σπάνιο για σετ που είναι δημόσια διαθέσιμο.

5.2 Προεπεξεργασία των σετ δεδομένων

Για την επίτευξη της ομοιομορφίας των πειραμάτων αλλά και επειδή όλα τα ηχητικά δείγματα μετατράπηκαν σε φασματογραφήματα της κλίμακας Mel (Mel-spectrograms) ώστε να γίνουν είσοδοι στα ΣΝΔ έγιναν τα εξής:

a) Το big-set παρέμεινε αναλλοίωτο δηλαδή (wav, mono, 8kHz, 10 sec)
b) Στο 360-set μειώθηκε η δειγματοληψία, η διάρκεια των δειγμάτων και ο ήχος έγινε μονοφωνικός (mp3, mono, 8kHz, 10sec)

Στη μελέτη των Eerola & Vouskoski το "Stimulus Set 1" (360-set) αποτελούσε τον προπομπό του κυρίως πειράματός τους που κατέληξε σε ένα ακόμα μικρότερο σετ 110 κομματιών το λεγόμενο "Stimulus Set 2" από το οποίο αφαίρεσαν μία κλάση, την έκπληξη (surprise) που υπήρχε στο "Stimulus Set 1". Αυτό έγινε καθώς οι ειδικοί δεν κατάφεραν να την αναγνωρίσουν ως αυτόνομο συναίσθημα γεγονός που φάνηκε από τις πολύ χαμηλές βαθμολογίες που έδωσαν στα αποσπάσματα που (υποτίθεται) ότι θα έπρεπε να υπερισχύει η έκπληξη έναντι των υπόλοιπων συναισθημάτων. Οι βαθμολογίες και η κατάταξη των αποσπασμάτων είναι δημοσιευμένα στο αρχείο "mean_ratings_set1". Έτσι, τα 30 κομμάτια της «έκπληξης» καταχωρίστηκαν από εμάς στις υπόλοιπες κατηγορίες συναισθημάτων αναλόγως βαθμολογίας που απέσπασαν, για παράδειγμα (*Εικόνα 35*) το απόσπασμα 153 ενώ είχε επισημειωθεί ως "surprise" το καταχωρίσαμε στο "tension" αφού αυτή η κλάση απέσπασε την υψηλότερη βαθμολογία των ειδικών στην ακουστική δοκιμασία.

Number	valence	energy	tension	anger	fear	happy	sad	tender	TARGET
153	3,83	3,67	5,00	1,50	1,83	1,67	2,67	1,00	SURPRISE

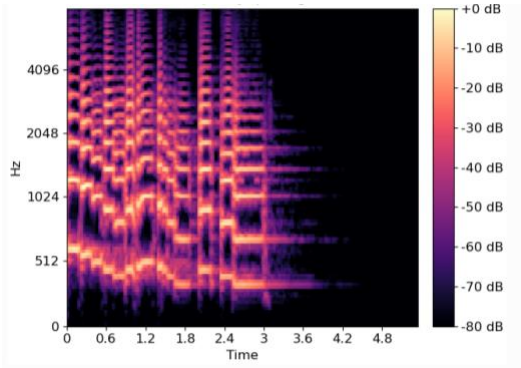
Εικόνα 35: Βαθμολογία του μουσικού αποσπάσματος Νο 153 στο πείραμα των Eerola & Vuoskoski

Ένα άλλο ζήτημα που έπρεπε να αντιμετωπιστεί ήταν ο περιορισμός της διάρκειας των αποσπασμάτων ώστε όλα να έχουν διάρκεια, όπως

προαναφέρθηκε, 10 δευτερόλεπτα. Κρατήσαμε και εδώ το πρώτο χρονικό παράθυρο δηλ. από 0 έως 10 sec γράφοντας κώδικα σε python (αρχείο `360_SliceAudio.py`).

Μετατροπή σε φασματογραφήματα της κλίμακας Mel

Προκειμένου τα δεδομένα των δύο σετ να γίνουν είσοδοι στα Βαθιά Νευρωνικά Δίκτυα των πειραμάτων ταξινόμησης, έπρεπε από ηχητικά να μετατραπούν σε αναπαραστάσεις εικόνας κατάλληλες για εξαγωγή χαρακτηριστικών του ήχου. Μια τέτοια απεικόνιση είναι το φασματογράφημα σε κλίμακα Mel (Mel-Spectrogram) [54] . Η κλίμακα Mel είναι μια λογαριθμική μετατροπή της συχνότητας του σήματος της οποίας η απεικόνιση είναι πιο κοντινή στον ανθρώπινο τρόπο αντίληψης του ήχου από ότι η γραμμική. Έτσι, το Mel-Spectrogram απεικονίζει το πεδίο των συχνοτήτων ενός σήματος συναρτήσει του χρόνου με τη διαφορά ότι οι συχνότητες εκφράζονται στην κλίμακα Mel, αντί της γραμμικής του «απλού» spectrogram, *Εικόνα 36*.



Εικόνα 36: Φασματογράφημα κλίμακας Mel

Η μετατροπή των ηχητικών δειγμάτων των σετ σε Mel-Spectrograms έγινε με τη βοήθεια της librosa. Η librosa [55] είναι ένα πακέτο Python για ανάλυση ηχητικού και μουσικού σήματος. Ο κώδικας της μετατροπής βρίσκεται το αρχείο `making mel-spectrograms.ipynb` και απόσπασμά του φαίνεται στην *Εικόνα 37*, όπου:

```
8   y, sr = librosa.load(songname, sr=8000, mono=True, duration=10)
9   print(y.shape, sr, songname)
10  S = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=128, fmax=8000)
11  S_dB = librosa.power_to_db(S, ref=np.max)
```

Εικόνα 37: Μετατροπή σε Mel-spectrogram μέσω της βιβλιοθήκης librosa

- Το κάθε μουσικό απόσπασμα (είτε wav, είτε mp3) θα φορτωθεί με sample rate = 8000, θα είναι μονοφωνικό και θα έχει διάρκεια `duration = 10` δευτερόλεπτα.
- Τα παράθυρα δειγματοληψίας θα είναι `n_fft = 2048`, με βήμα μεγέθους `hop_length=512` κάθε φορά για το επόμενο παράθυρο (οι παράμετροι δεν φαίνονται στον κώδικα καθώς το 2048, 512 είναι οι προκαθορισμένες τιμές τους και μπορεί να παραληφθούν).

- Όλο το εύρος συχνοτήτων του δείγματος χωρίζεται σε `n_mels=128` ομοιόμορφα κατανεμημένες συχνότητες σε αποστάσεις όπως θα ακούγονταν από το ανθρώπινο αυτί.
- Το `f_max=8000` ορίζει το μέγιστο όριο της συχνότητας που θα απεικονιστεί
- Το `S_dB` είναι το φασματογράφημα S που μετατρέπεται σε Mel-Spectrogram

Διαχωρισμός των σετ σε train, validation, test

Έχοντας μετατρέψει όλα τα ηχητικά δείγματα σε Mel-Spectrograms έγινε διαχωρισμός των big-set και 360-set σε:

- Σετ εκπαίδευσης (training set) όπου χρησιμοποιείται για την εκπαίδευση του μοντέλου
- Σετ επικύρωσης (validation set) όπου χρησιμοποιείται για την εκτίμηση της γενίκευσης του μοντέλου όπου αναλόγως των απωλειών προχωρούμε σε ρύθμιση (finetuning) του μοντέλου.
- Σετ δοκιμής (test set) όπου χρησιμοποιείται μετά το τέλος της εκπαίδευσης και περιέχει αποκλειστικά και μόνο δείγματα άγνωστα στο μοντέλο.

Ο διαχωρισμός σε train & val έγινε με τη βοήθεια της βιβλιοθήκης της Python που ονομάζεται Scikit-learn. Η Scikit-learn [56] προσφέρει πολλές δυνατότητες για ανάλυση δεδομένων προκειμένου να χρησιμοποιηθούν σε προβλήματα ταξινόμησης, συσταδοποίησης και επιλογής μοντέλων. Με τη συνάρτησή της `train_test_split` διαχωρίσαμε το κάθε σετ σε `train 0.8` και `val 0.2` όπως φαίνεται στην *Εικόνα 38*: (αρχείο κώδικα: `splitting datasets in train-val sets.ipynb`)

```
17    files = os.listdir(origin_label_dir)
18    file_idx = list(range(len(files)))
19    val_percentage = 0.2
20    train, val = train_test_split(file_idx, test_size=val_percentage, random_state=42)
```

Εικόνα 38: Τυχαίος διαχωρισμός του σετ σε train=0.8, val=0.2

Το 360-set μετά τον διαχωρισμό του έχει διαμορφωθεί ως εξής (*Πίνακας 2*):

Πίνακας 2: Διαχωρισμός 360-set, σε train, val, test σετ

Διαχωρισμός του 360-set σε train - validation - test															
Valence	*Train*	*Val*	*Test*	Tension	*Train*	*Val*	*Test*	Energy	*Train*	*Val*	*Test*	Emotions	*Train*	*Val*	*Test*
positive	62	26	50	high	64	28	50	high	61	27	50	anger	6	5	40
neutral	47	21	50	medium	33	15	50	medium	48	20	50	fear	32	21	40
negative	38	16	50	low	49	21	50	low	37	17	50	happy	15	10	40
												sad	21	14	40
												tender	22	14	40

Το big-set ήρθε στην κατοχή μας ήδη χωρισμένο σε train – test σετ οπότε σαν είσοδος στο Scikit-learn μπήκε το train set όπου χωρίστηκε σε train - val (*Πίνακας 3*):

Πίνακας 3: Διαχωρισμός big-set σε train, val, test σετ

Διαχωρισμός του big-set σε train - validation - test							
Valence	*Train*	*Val*	*Test*	Energy	*Train*	*Val*	*Test*
positive	3752	938	513	high	5739	1435	780
neutral	4762	1191	661	medium	4416	1104	628
negative	3882	971	541	low	2192	548	308

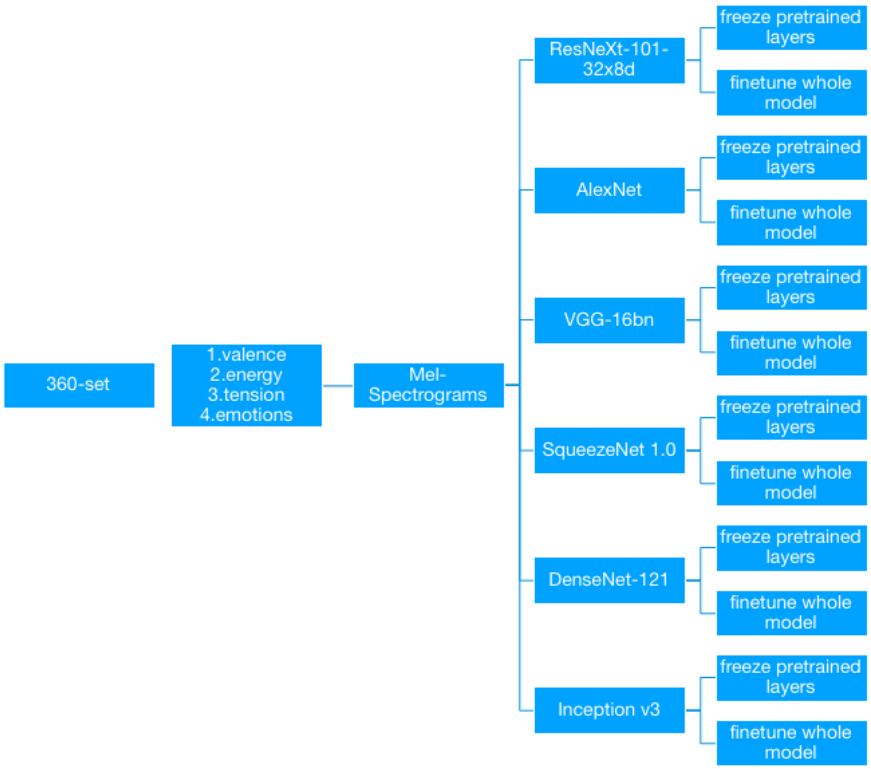
Τώρα, τα δύο σετ μας είναι έτοιμα να γίνουν είσοδοι στα βαθιά νευρωνικά δίκτυα ώστε να επιλύσουν τα προβλήματα ταξινόμησης.

5.3 Ταξινόμηση με Βαθιά Νευρωνικά Δίκτυα και Μεταφορά Μάθησης

5.3.1 Επισκόπηση πειραμάτων

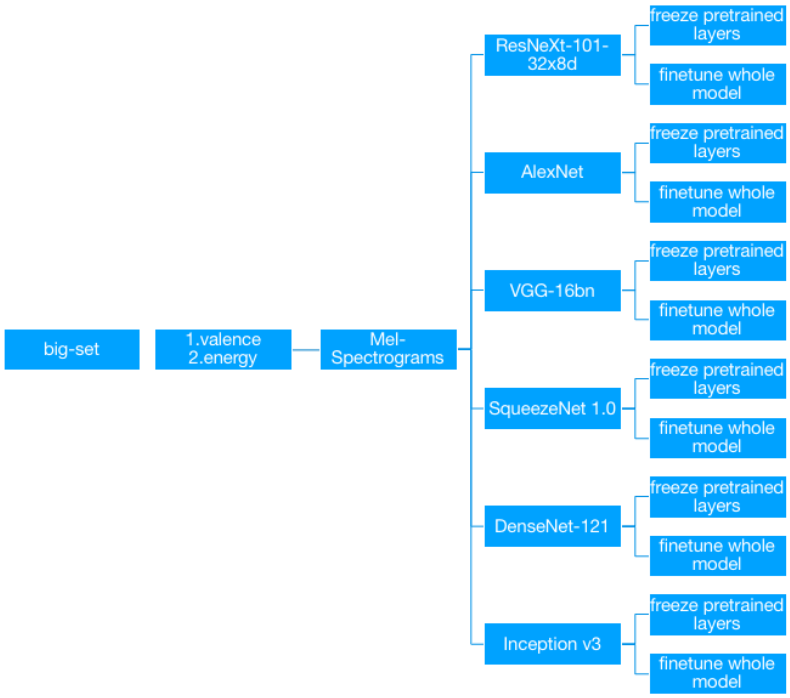
Τα δύο σετ, δηλαδή το big-set και το 360-set υποβλήθηκαν σε σειρά πειραμάτων ταξινόμησης με: χρήση Βαθιών Νευρωνικών Δικτύων, τεχνική Μεταφοράς Μάθησης (Transfer Learning), τεχνική Επαύξησης Δεδομένων (Data Augmentation) και παραγωγή τεχνητών δεδομένων με χρήση Παραγωγικού Ανταγωνιστικού Δικτύου και συγκεκριμένα του StyleGAN2-ADA. Τα δεδομένα που παρήγαγε το StyleGAN2 χρησιμοποιήθηκαν ως είσοδοι για τα πειράματα ταξινόμησης των συναισθημάτων (anger, fear, happy, sad, tender) του 360-set καθώς ο αρχικός αριθμός των δειγμάτων ήταν πολύ μικρός για Βαθιά Μηχανική Μάθηση και έγιναν συγκριτικές δοκιμές. Όλα τα μοντέλα που χρησιμοποιήθηκαν (ResNeXt101_32x8d, AlexNet, VGG16_bn, SqueezeNet1.0, DenseNet121, Inception_v3) είναι προ εκπαιδευμένα στο σύνολο δεδομένων ImageNet και προέρχονται από το πακέτο μοντέλων torchvision.models[57] της βιβλιοθήκης μηχανικής μάθησης Pytorch.

Συνοπτικά, τα πειράματα ταξινόμησης για το 360-set είναι: Valence, Energy, Tension, Emotion(anger, fear, happy, sad, tender) σε 6 διαφορετικές αρχιτεκτονικές και σε 2 σενάρια δηλαδή 4x6x2=48 πειράματα, όπως φαίνονται συνοπτικά στην ακόλουθη *Εικόνα 39*:



Εικόνα 39: Σύνοψη έργων ταξινόμησης για το 360-set, με μεταφορά μάθησης

Αντίστοιχα, τα πειράματα ταξινόμησης για το big-set είναι: Valence, Energy σε 6 διαφορετικές αρχιτεκτονικές και σε 2 σενάρια δηλαδή 2x6x2=24 πειράματα, όπως φαίνονται συνοπτικά στην ακόλουθη *Εικόνα 40*:



Εικόνα 40: Σύνοψη έργων ταξινόμησης για το big-set, με μεταφορά μάθησης

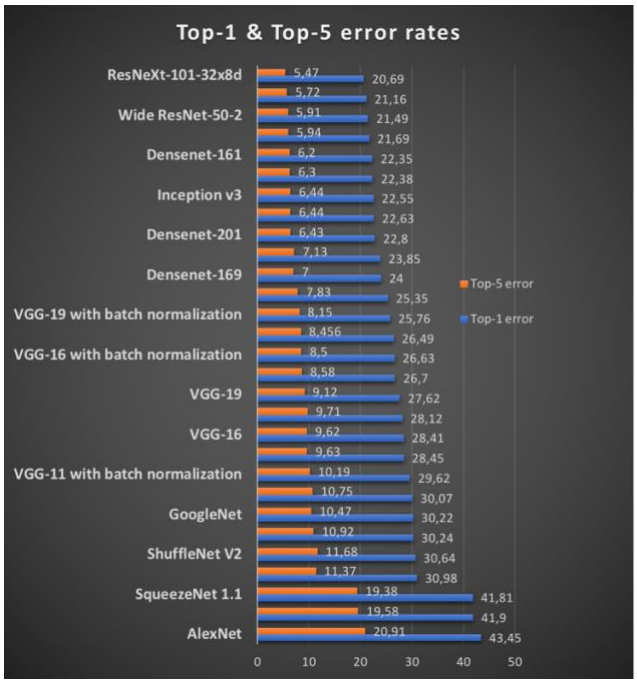
Στη συνέχεια επαναλήφθηκε η προαναφερθείσα σειρά πειραμάτων αλλά αυτή τη φορά τα μοντέλα τα είχαμε προεκπαιδεύσει στα φασματογραφήματα του big-set. Στο τέλος έγινε σύγκριση των αποτελεσμάτων τους.

Ο κώδικας των πειραμάτων είναι γραμμένος σε Python και εκτελέστηκε στο περιβάλλον της πλατφόρμας Google Colaboratory [58] (ή Colab) σε μορφή Jupyter Notebook καθώς η χρήση των GPU's που προσφέρει το Colab, έστω και με κάποιους περιορισμούς, κρίθηκε απαραίτητη για την εκπαίδευση των μοντέλων και την παραγωγή τεχνητών δειγμάτων με το StyleGAN2-ADA.

5.3.2 Μέθοδος επιλογής προεκπαιδευμένων μοντέλων

Υπάρχουν αρκετοί παράγοντες που μπορούμε να λάβουμε υπόψη μας πριν καταλήξουμε στην επιλογή ενός προεκπαιδευμένου μοντέλου έναντι κάποιου άλλου. Μερικοί από αυτούς είναι η πολυπλοκότητά του, η κατανάλωση μνήμης και ο χρόνος υπολογισμού συμπεράσματος (inference time) [59] . Επίσης, το μέγεθος του μοντέλου μπορεί να είναι ένας ακόμη παράγοντας όπως και η κατάταξη του στον πίνακα top-1 & top-5 error rates. Στην παρούσα εργασία επιλέξαμε 6 χαρακτηριστικές αρχιτεκτονικές βάσει αυτού του πίνακα χωρίς να λάβουμε υπόψη επιπλέον παράγοντες.

Για την ακρίβεια, επιλέξαμε μοντέλα που βρίσκονται τόσο στην κορυφή του πίνακα (ResNeXt-101) όσο και στη βάση του (AlexNet) καθώς και μερικά που καταλαμβάνουν τις ενδιάμεσες θέσεις. Το top-1 error είναι το ποσοστό των φορών όπου ο ταξινομητής δεν έδωσε στη σωστή κλάση το υψηλότερο σκορ και το top-5 error είναι το ποσοστό των φορών που ο ταξινομητής δεν συμπεριέλαβε τη σωστή κλάση μέσα στις πέντε κορυφαίες προβλέψεις του. Ιδανικά, θέλουμε το μοντέλο να προβλέπει τη σωστή κλάση στο top-1 error αλλά και η χαμηλή (χαμηλή = καλύτερη) βαθμολογία στην κατάταξη top-5 σημαίνει πιο ολοκληρωμένη εκτίμηση της απόδοσής του για τις κλάσεις που είχε αστοχία. Για παράδειγμα μπορεί να μην προβλέψει σωστά την κλάση στο top-1 αλλά στο top-5 σε ένα ικανοποιητικό ποσοστό να την έχει μέσα στις πρώτες επιλογές του. Ο πίνακας top-1 & top-5 error rates για τα μοντέλα του torchvision φαίνεται στην παρακάτω εικόνα (*Εικόνα 41*).



Εικόνα 41: Κατάταξη προεκπαιδευμένων μοντέλων του torchvision κατά τα top-1 & top-5 error rates

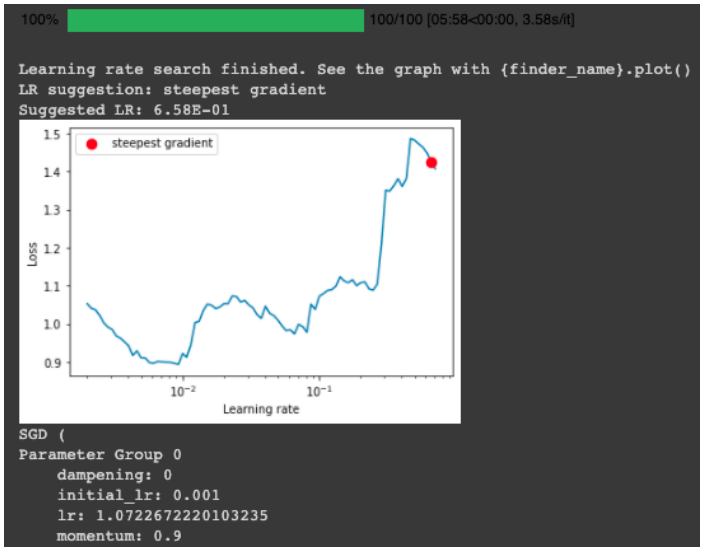
5.3.3 Ρύθμιση και επιλογή υπερπαραμέτρων

Ίσως η σημαντικότερη υπερπαράμετρος για την ικανοποιητική απόδοση ενός νευρωνικού δικτύου είναι ο ρυθμός εκπαίδευσης (learning rate) της συνάρτησης βελτιστοποίησης, όπως είχε αναφερθεί και σε προηγούμενο κεφάλαιο. Πολύ μικρός ρυθμός εκπαίδευσης σημαίνει καλύτερη, πιο αξιόπιστη εκπαίδευση αλλά η σύγκλιση επιτυγχάνεται μετά από υπερβολικά μεγάλο χρονικό διάστημα. Από την άλλη, μεγάλος ρυθμός μπορεί να σημαίνει γρήγορη εκπαίδευση αλλά συνήθως αποτυχημένη καθώς οι αλλαγές των βαρών είναι τόσο μεγάλες που ο αλγόριθμος αστοχεί μη μπορώντας να ανακαλύψει το ελάχιστο και ο αλγόριθμος τελικά αποκλίνει. Η εύρεση του ιδανικού ρυθμού εκπαίδευσης είναι δύσκολη υπόθεση που απαιτεί συνήθως πολλές δοκιμές, υπάρχουν φυσικά και κάποιες μέθοδοι (όπως annealing, scheduling) που βοηθούν προς αυτή την κατεύθυνση. Στην παρούσα εργασία χρησιμοποιήθηκαν δύο τέτοιες τεχνικές καθώς ο όγκος των πειραμάτων, οι περιορισμένοι υπολογιστικοί πόροι και ο χρόνος δεν επέτρεψαν ενδελεχή και ξεχωριστή ρύθμιση κάθε αρχιτεκτονικής που δοκιμάστηκε.

Ως συνάρτηση βελτιστοποίησης σε όλα τα πειράματα χρησιμοποιήθηκε η SGD (Στοχαστική Κάθοδος κλίσης) με momentum = 0.9 (ορμή), που γενικά έχει καλή απόδοση στα δίκτυα που επιλέξαμε.

PyTorch learning rate finder

Πρόκειται για μια υλοποίηση σε Pytorch [60] του "learning range test" όπως αυτό περιγράφεται στο άρθρο του Leslie N. Smith **Error! Reference source not found.**. Το learning rate range test είναι μια δοκιμασία που προτείνει έναν ιδανικό ρυθμό εκπαίδευσης. Κατά τη διάρκεια της δοκιμασίας αυτής ο ρυθμός εκπαίδευσης αυξάνεται γραμμικά ή εκθετικά μεταξύ δύο ορίων. Το χαμηλότερο όριο αφήνει το δίκτυο να ξεκινήσει τη σύγκλιση και τελικά καθώς ο ρυθμός αυξάνεται, κάποια στιγμή θα γίνει πολύ μεγάλος και τότε το δίκτυο θα αποκλίνει. Η εκπαίδευση με τους κυκλικούς ρυθμούς εκπαίδευσης αντί σταθερών τιμών πετυχαίνει βελτιωμένη ακρίβεια χωρίς την ανάγκη ρύθμισης και συχνά με λιγότερες επαναλήψεις. (*Εικόνα 42*)



Εικόνα 42: Το learning rate finder εδώ προτείνει ρυθμό εκπαίδευσης 6.58E-01

ReduceLROnPlateau

Το `torch.optim` είναι ένα πακέτο του Pytorch που περιλαμβάνει διάφορους αλγόριθμους βελτιστοποίησης. Ένας από αυτούς είναι και ο ReduceLROnPlateau [62] ο οποίος μειώνει τον ρυθμό εκπαίδευσης όταν μια μετρική σταματήσει να βελτιώνεται. Ο αλγόριθμος διαβάζει μια επιλεγμένη μετρική και αν δε σημειωθεί πρόοδος για συγκεκριμένο αριθμό εποχών τότε ο ρυθμός εκπαίδευσης μειώνεται., *Εικόνα 43*.

```
scheduler = ReduceLROnPlateau(optimizer_ft, patience=3,  mode='min')
```

Εικόνα 43: Συνάρτηση ReduceLROnPlateau

Στο παραπάνω παράδειγμα κώδικα του συγκεκριμένου αλγόριθμου η παράμετρος `optimizer_ft` είναι ο επιλεγμένος optimizer (π.χ. μια SGD), η παράμετρος patience είναι το όριο του αριθμού των εποχών κατά το οποίο επιτρέπεται η μη βελτίωση της επιλεγμένης μετρικής. Όταν αυτό το όριο ξεπεραστεί τότε θα μειωθεί και ο ρυθμός εκπαίδευσης.

5.3.4 Μετρικές εκτίμησης απόδοσης ταξινομητών

Για τη διερεύνηση της ποιότητας της εκπαίδευσης ενός μοντέλου χρησιμοποιούμε διάφορες μετρικές [63], [64] των οποίων οι τιμές μας βοηθούν να εντοπίσουμε τυχόν προβλήματα όπως υπερπροσαρμογή, προβληματικό σετ δεδομένων, σωστή επιλογή μοντέλου και διάφορα άλλα. Επίσης, μας βοηθάνε στη βελτίωση του δικτύου αφού, παρατηρώντας τις τιμές τους, μπορούμε να επέμβουμε στις υπερπαραμέτρους του μοντέλου και με επιπλέον δοκιμές να επιτύχουμε την καλύτερη απόδοσή του εφόσον επιδέχεται βελτίωσης. Στη συνέχεια θα παρουσιαστούν αυτές που χρησιμοποιήσαμε στα πειράματα.

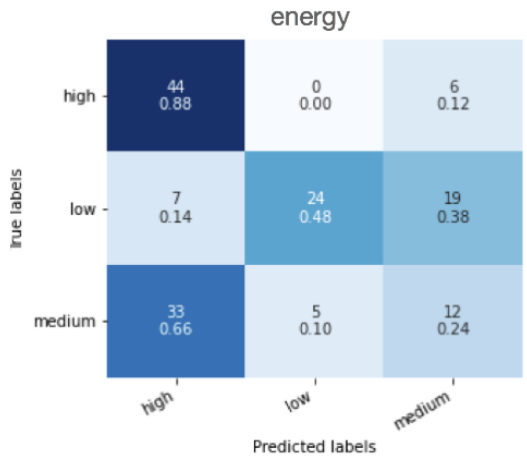
Πίνακας Σύγχυσης (Confusion Matrix)

Πρόκειται [63] για έναν πίνακα δύο διαστάσεων όπου οι στήλες και οι γραμμές του αποτυπώνουν από τη μία τα πραγματικά δείγματα και από την άλλη αυτά του πρόβλεψε το μοντέλο. Για παράδειγμα στην Εικόνα 44 βλέπουμε έναν τέτοιο πίνακα (από πείραμα της εργασίας) όπου στις γραμμές εμφανίζονται τα πραγματικά δείγματα των τριών κλάσεων και στις στήλες αυτά που προβλέφθηκαν κατά την ταξινόμηση. Το σετ περιέχει 50 δείγματα ανά κλάση και ο πίνακας σύγχυσης τα εμφανίζει ως:

- Αληθώς θετικά (True Positives, TP): Είναι τα δείγματα που ο ταξινομητής πρόβλεψε σωστά δηλαδή στο παράδειγμά μας για τις κλάσεις: $TP_{high} = 44$, $TP_{low} = 24$, $TP_{medium} = 12$ (τα στοιχεία της διαγωνίου)

- Αληθώς αρνητικά (True Negative, TN): Είναι τα δείγματα που ο ταξινομητής σωστά πρόβλεψε ότι δεν ανήκουν σε μία κλάση π.χ. για την FN_{high}=24 + 19 + 5 + 12
- Ψευδώς θετικά (False Positive, FP): Είναι τα δείγματα μιας κλάσης που ο ταξινομητής λανθασμένα πρόβλεψε ως θετικά π.χ. FP_{high}=0+6
- Ψευδώς αρνητικά (False Negative, FN): Είναι τα θετικά δείγματα μιας κλάσης που ο ταξινομητής λανθασμένα τα πρόβλεψε ως αρνητικά π.χ. FN_{high}=7+33

Οι παραπάνω εκφράσεις «θετικά», «αρνητικά» προέρχονται από τα προβλήματα ταξινόμησης δύο κλάσεων όπου τα σφάλματα που κάνει το μοντέλο είναι τα FP και τα FN.



Εικόνα 44: Παράδειγμα πίνακα σύγχυσης τριών κλάσεων σε σετ 150 δειγμάτων

Ορθότητα (Accuracy)

Η Ορθότητα ενός μοντέλου μας δείχνει πόσες φορές έκανε σωστή πρόβλεψη ο ταξινομητής στο σύνολο των δειγμάτων και υπολογίζεται από τη σχέση:

$$accuracy = \frac{TP_{all}}{all\ samples}$$

Ακρίβεια (Precision)

Δηλώνει τον λόγο των αληθώς θετικών δειγμάτων μιας κλάσης προς όλα τα δείγματα που ο ταξινομητής πρόβλεψε ως θετικά:

$$precision_{class} = \frac{TP_{class}}{TP_{class} + FP_{class}}$$

Ανάκληση (Recall)

Ονομάζεται και Ευαισθησία (Sensitivity) αναπαριστά το ποσοστό των δειγμάτων μιας κλάσης που ήταν αληθώς θετικά και ο ταξινομητής τα κατέταξε ως τέτοια:

$$recall_{class} = \frac{TP_{class}}{TP_{class} + FN_{class}}$$

F-1 Score

Η μετρική αυτή συνδυάζει την Ακρίβεια και την Ευαισθησία μιας κλάσης καθώς μας δίνει τον αρμονικό τους μέσο:

$$f1score = 2\frac{Precision_{class} * Recall_{class}}{Precision_{class} + Recal_{class}}$$

Αν θέλουμε να υπολογίσουμε το λεγόμενο *macro avg F1-score* [65] σε προβλήματα ταξινόμησης πολλών κλάσεων τότε υπολογίζουμε τα επιμέρους f1 ανά κλάση, τα αθροίζουμε και διαιρούμε με το πλήθος των κλάσεων.

5.3.5 Μεταφορά Μάθησης (Transfer Learning)

Προκειμένου ένα πολυεπίπεδο νευρωνικό δίκτυο να εκπαιδευτεί σωστά χρειάζεται μεγάλο όγκο δεδομένων, πολύ μεγαλύτερο από αυτόν που χρειάζονται οι αλγόριθμοι της μηχανικής μάθησης και μεγάλη υπολογιστική ισχύ. Όμως, πολύ συχνά, ούτε η εύρεση τεράστιων σετ δεδομένων είναι εφικτή αλλά ούτε και η διατιθέμενη υπολογιστική ισχύς είναι ανάλογη των απαιτήσεων της εκπαίδευσης ενός τέτοιου δικτύου. Η βασική ιδέα πίσω από την τεχνική της Μεταφοράς Μάθησης [66]–[69] είναι: «Αποθηκεύω γνώση που απέκτησα λύνοντας ένα πρόβλημα και τη χρησιμοποιώ για να λύσω ένα άλλο παρόμοιο πρόβλημα σε συντομότερο χρόνο και με μικρότερο κόστος». Πιο συγκεκριμένα, έστω για ένα πρόβλημα ταξινόμησης εικόνων, είναι κοινή πρακτική η χρήση ενός προεκπαιδευμένου μοντέλου που έχει προκύψει από την εκπαίδευση ενός βαθιού ΣΝΔ σε ένα πολύ μεγάλο σετ δεδομένων (π.χ. ImageNet των 1.2 εκατομμυρίων εικόνων, 1000 κλάσεων). Το προεκπαιδευμένο μοντέλο στη συνέχεια μπορεί να χρησιμοποιηθεί σε άλλα έργα ταξινόμησης προσαρμοσμένο κατάλληλα για τα συγκεκριμένα έργα. Αυτό που στην πραγματικότητα μεταφέρεται μέσω του προεκπαιδευμένου μοντέλου είναι τα βάρη του αρχικού δικτύου μετά την εκπαίδευσή του.

Κυριότερα σενάρια

- *εξαγωγέας χαρακτηριστικών:* από το προεκπαιδευμένο μοντέλο αφαιρείται το τελευταίο επίπεδο (του ταξινομητή), προσαρμόζεται στις ανάγκες της τρέχουσας ταξινόμησης και «παγώνουν» τα βάρη των υπόλοιπων επιπέδων. Για παράδειγμα, οι κλάσεις του ImageNet είναι 1000, ενώ το τρέχον έργο απαιτεί 5 κλάσεις, οπότε διορθώνουμε αναλόγως. Η τεχνική αυτή είναι χρησιμότερη όταν τα σετ δεδομένων (προεκπαίδευσης – εκπαίδευσης/δοκιμής) είναι παρόμοια. Μπορεί να χρησιμοποιηθεί ακόμα και κάποιος ταξινομητής μηχανικής μάθησης πχ SVM αντί του τελευταίου επιπέδου του μοντέλου.
- *μικρο-ρύθμιση:* σε αυτό το σενάριο επίσης γίνεται διόρθωση του επιπέδου του ταξινομητή αλλά παράλληλα γίνεται και μικρο-ρύθμιση των βαρών των προηγούμενων επιπέδων κατά την οπισθοδιάδοση. Μπορεί να ανανεωθούν είτε όλα είτε κάποια από τα επίπεδα και τα υπόλοιπα να μείνουν «παγωμένα» (με τα βάρη της προεκπαίδευσης). Αυτό λειτουργεί καθώς τα πρώτα επίπεδα ενός ΣΝΔ εντοπίζουν τα γενικότερα χαρακτηριστικά μια εικόνας (π.χ. εντοπισμός άκρων εικόνας) που είναι πιθανότερο να είναι κοινά ακόμα και σε εντελώς αντιδιαμετρικά σετ εικόνων.
- *ταξινομητής:* χωρίς κάποια επιπλέον εκπαίδευση ή αλλαγές σε κάποια από τα επίπεδά του, το μοντέλο χρησιμοποιείται αυτούσιο ως ταξινομητής. Το σενάριο αυτό προϋποθέτει αρκετή ομοιότητα των σετ τόσο της προεκπαίδευσης όσο και της δοκιμασίας στη νέα ταξινόμηση.

Επιλογή σεναρίου

Το ποιο από τα παραπάνω σενάρια θα υιοθετήσουμε ανά περίπτωση, εξαρτάται κυρίως από δύο παράγοντες: από το μέγεθος του σετ και από την ομοιότητά του με το σετ της προεκπαίδευσης. Γενικά ακολουθούμε τις εξής κατευθύνσεις:

- *το νέο σετ είναι μικρό και παρόμοιο του αρχικού:* Εάν γίνει μικρο-ρύθμιση ελλοχεύει ο κίνδυνος της υπερπροσαρμογής, οπότε προτιμάται το σενάριο του εξαγωγέα χαρακτηριστικών.
- *το νέο σετ είναι μεγάλο και παρόμοιο του αρχικού:* η παρουσία πολλών δειγμάτων εξαλείφει σχεδόν τον κίνδυνο της υπερπροσαρμογής οπότε το σενάριο της μικρο-ρύθμισης όλων (ή του μεγαλύτερου μέρους) των επιπέδων του μοντέλου επικρατεί.

- *το νέο σετ είναι μικρό και πολύ διαφορετικό του αρχικού:* από τη μία έχουμε μικρό σετ οπότε η υπερπροσαρμογή είναι αρκετά πιθανή σε περίπτωση μικρο-ρύθμισης όλου του δικτύου και από την άλλη, λόγω της διαφορετικότητας των σετ το «πάγωμα» των τελευταίων επιπέδων δεν είναι καλή ιδέα καθώς αυτά περιέχουν βάρη προσαρμοσμένα στις λεπτομέρειες του αρχικού σετ. Μια μέση λύση είναι να εκπαιδευτεί ο ταξινομητής (μπορεί και ένας svm) με τα βάρη μόνο κάποιων (με δοκιμή) αρχικών επιπέδων του μοντέλου.
- *το νέο σετ είναι μεγάλο και πολύ διαφορετικό του αρχικού:* Προκρίνεται το σενάριο της μικρο-ρύθμισης όλου του δικτύου πάνω στα αρχικά βάρη του προεκπαιδευμένου. Αν το σετ είναι πάρα πολύ μεγάλο τότε μπορεί να γίνει και εξαρχής εκπαίδευση των βαρών με τυχαίες αρχικές τιμές.

Μια χρήσιμη επισήμανση σε σχέση με την επιλογή ρυθμού εκπαίδευσης (learning rate) όταν πραγματοποιούμε μικρο-ρύθμιση βαρών, είναι να επιλέγουμε μια τιμή αρκετά μικρή ώστε να μην διαταραχθούν πολύ τα ήδη καλώς αρχικοποιημένα από την προεκπαίδευση) βάρη.

5.4 Πείραμα Α

Όπως ειπώθηκε και προηγουμένως στην παράγραφο της Μεταφοράς Μάθησης, η εκπαίδευση ενός Συνελικτικού Νευρωνικού Δικτύου από μηδενική βάση είναι μια ενέργεια εξαιρετικά κοστοβόρα τόσο υπολογιστικά όσο και χρονικά. Επιπλέον, μια τέτοια απόπειρα προϋποθέτει και ένα σετ εκπαίδευσης τόσο μεγάλο που είναι πολύ σπάνιο να βρεθεί. Η συνήθης διαδικασία είναι η προεκπαίδευση ενός ΣΝΔ σε ένα πραγματικά μεγάλο σετ (πχ ImageNet με τα 1.2 εκατομμύρια εικόνες και τις 1000 κλάσεις) και στη συνέχεια η χρήση του με τις τεχνικές της Μεταφοράς Μάθησης δηλαδή, είτε ως εξαγωγέα χαρακτηριστικών όπου όλα τα προηγούμενα μέρη του δικτύου «παγώνουν» τα βάρη τους και ενημερώνονται μόνο τα τελευταία επίπεδα του ταξινομητή, είτε με μικρο-ρύθμιση (fine-tuning) των βαρών όλων των επιπέδων του δικτύου, συμπεριλαμβανομένων αυτών του ταξινομητή. Στην περίπτωση αυτή χρειάζεται προσοχή καθώς μικρό σετ που εκπαιδεύεται σε πλήρους ανάπτυξης μεγάλο ΣΝΔ

μπορεί εύκολα να κάνει υπερπροσαρμογή. Στα πειράματα της εργασίας μας ακολουθήσαμε και τις δύο στρατηγικές και στα δύο σετ που χρησιμοποιήσαμε. Στη συνέχεια θα δούμε τις λεπτομέρειες της εκπαίδευσης για κάθε αρχιτεκτονική ξεχωριστά.

5.4.1 Μεταφορά Μάθησης με το AlexNet

Λεπτομέρειες για το AlexNet υπάρχουν στο 3.6.1.

Με τις ακόλουθες γραμμές εντολών και τη βοήθεια του torchsummary

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = models.alexnet().to(device)
summary(model, (3, 224, 224))
```

παίρνουμε τις πληροφορίες για το προεκπαιδευμένο μοντέλο AlexNet που παρέχει το torchvision, βλέπουμε ότι έχει 61.100.840 παράμετρους και μέγεθος 233MB.

```
================================================================
Total params: 61,100,840
Trainable params: 61,100,840
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 8.38
Params size (MB): 233.08
Estimated Total Size (MB): 242.03
----------------------------------------------------------------
```

Ο σκελετός του δικτύου έχει ως εξής:

```
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
```

```
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
  )
)
```

Προκειμένου να γίνει η εκπαίδευση φορτώνουμε το προεκπαιδευμένο μοντέλο και διαμορφώνουμε το επίπεδο του ταξινομητή ώστε από τις 1000 κλάσεις του ImageNet να προσαρμόσουμε είτε για τις 3 κλάσεις των valence (positive, neutral, negative), energy (high, medium, low), tension (high, medium, low), είτε για τις 5 κλάσεις των emotions (anger, fear, happy, sad, tender) με τις ακόλουθες γραμμές κώδικα:

```
model_ft = models.alexnet(pretrained=use_pretrained)
        set_parameter_requires_grad(model_ft, feature_extract)
        num_ftrs = model_ft.classifier[6].in_features
        model_ft.classifier[6] = nn.Linear(num_ftrs,num_classes)
        input_size = 224
```

Έτσι η τελευταία γραμμή του classifier του αρχικού μοντέλου (χρωματισμένη με γκρι χρώμα) διαμορφώνεται ως εξής:

```
    (6): Linear(in_features=4096, out_features=3, bias=True)
```

Το out_features αλλάζει αντίστοιχα για τις 5 κλάσεις σε `out_feature=5`

Τα αποτελέσματα των ταξινομήσεων στα δύο σετ αποτυπώνονται στους παρακάτω πίνακες:

Πίνακας 4: Αποτελέσματα ταξινόμησης για το AlexNet στο 360-set

Classification Results 360-set

	Energy				Valence				Tension				Emotions			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	AlexNet (freeze)															
val	0,43	0,57	0,49	0,61	0,55	0,56	0,51	0,59	0,75	0,74	0,74	0,77	0,38	0,36	0,36	0,39
test	0,49	0,45	0,38	0,45	0,56	0,58	0,53	0,58	0,58	0,57	0,57	0,57	0,45	0,43	0,42	0,43
	AlexNet (whole)															
val	0,64	0,65	0,62	0,67	60	0,59	0,58	0,62	0,74	0,72	0,73	0,78	0,28	0,34	0,27	0,36
test	0,56	0,53	0,51	0,53	0,64	0,6	0,61	0,6	0,56	0,58	0,55	0,58	0,34	0,4	0,35	0,4

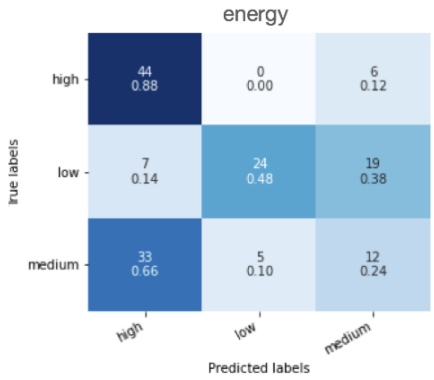
Πίνακας 5: Αποτελέσματα ταξινόμησης για το AlexNet στο big-set

Classification Results big-set

	Energy				Valence			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	AlexNet (freeze)							
val	0,7	0,63	0,65	0,66	0,58	0,52	0,52	0,54
test	0,69	0,61	0,63	0,65	0,57	0,51	0,51	0,52
	AlexNet (whole)							
val	0,68	0,7	0,69	0,7	0,6	0,61	0,59	0,59
test	0,68	0,69	0,68	0,69	0,59	0,6	0,59	0,59

Παρατηρούμε ότι σε όλα τα πειράματα εκτός αυτό των tension και emotions όταν η ταξινόμηση γίνεται με ανανέωση των βαρών όλων των επιπέδων του AlexNet τα αποτελέσματα είναι καλύτερα και στα δύο σετ δεδομένων. Στις ταξινομήσεις των tension και emotions το δίκτυο συμπεριφέρεται καλύτερα όταν ανανεώνονται τα βάρη μόνο του τελευταίου επιπέδου του ταξινομητή.

Ο πίνακας σύγχυσης (confusion matrix) για το Energy σε σύνολο 150 δειγμάτων του test set είναι ο εξής (*Εικόνα 45*) και παρατηρούμε ότι η πιο δύσκολα αναγνωρίσιμη κλάση είναι η medium με τα αληθώς θετικά (true positives) να είναι μόλις 12.

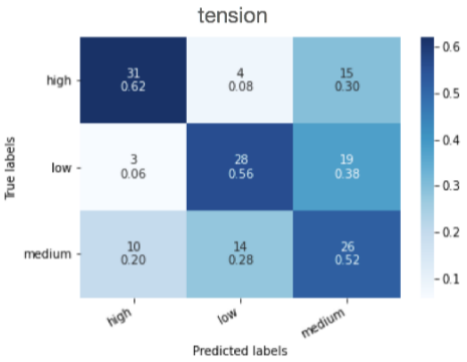


Εικόνα 45: Πίνακας σύγχυσης για το Energy στο 360-set, AlexNet

Ο πίνακας σύγχυσης για το Valence σε σύνολο 150 δειγμάτων του test set είναι ο ακόλουθος (*Εικόνα 46*) και παρατηρούμε ότι η κλάση που αναγνωρίστηκε σωστά περισσότερο από τις υπόλοιπες είναι η κλάση positive με 37 δείγματα.

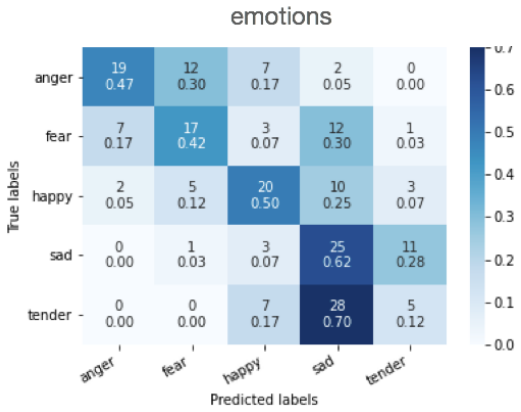
Εικόνα 46: Πίνακας σύγχυσης για το Valence στο 360-set, AlexNet

Ο πίνακας σύγχυσης για την ταξινόμηση Tension στα 150 δείγματα του test set μας έδειξε ότι και οι 3 κλάσεις (high, low, medium) αναγνωρίστηκαν σωστά σε περίπου ίδιο αριθμό δειγμάτων. (*Εικόνα 47*)



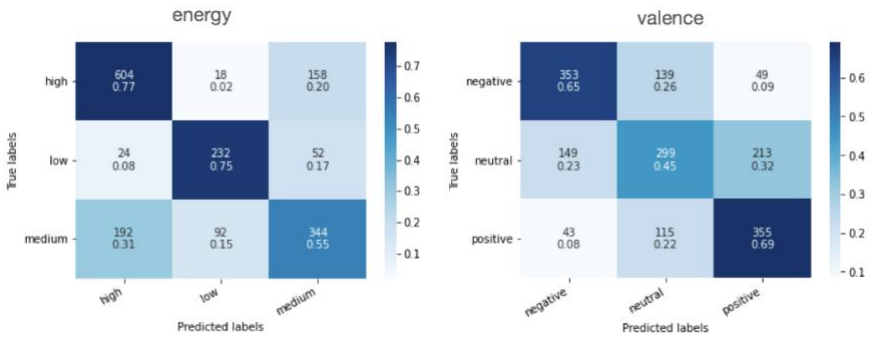
Εικόνα 47: Πίνακας σύγχυσης για το Tension στο 360-set, AlexNet (freeze)

Ο πίνακας σύγχυσης (*Εικόνα 48*) για το Emotions σε σύνολο 200 δειγμάτων του test-set μας έδειξε ότι η κλάση του συναισθήματος της τρυφερότητας (tender) σχεδόν δεν αναγνωρίστηκε καθόλου (5 δείγματα από τα 40) και είχε μια ισχυρή συσχέτιση με την κλάση sad καθώς 28 δείγματά της χαρακτηρίστηκαν ψευδώς ως λυπημένα (sad).



Εικόνα 48: Πίνακας σύγχυσης για το Emotions στο 360-set, AlexNet (freeze)

Οι πίνακες σύγχυσης για το big-set στις ταξινομήσεις Energy, Valence σε 1715 δείγματα του test-set μας έδειξαν ότι οι κλάσεις με τη μικρότερη ευαισθησία (sensitivity ή recall) ήταν αυτές που αντιπροσώπευαν τις μεσαίες διαβαθμίσεις της Ενέργειας και του Σθένους δηλαδή οι medium και neutral αντίστοιχα, όπως φαίνεται στην *Εικόνα 49*.



Εικόνα 49: Πίνακες σύγχυσης για Energy & Valence, big-set, AlexNet

5.4.2 Μεταφορά Μάθησης με το ResNeXt

Λεπτομέρειες για το ResNeXt υπάρχουν στο 3.6.4.

Με τις ακόλουθες γραμμές εντολών και τη βοήθεια του torchsummary που

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = models.resnext101_32x8d().to(device)
summary(model, (3, 224, 224))
```

παρέχει το torchvision, βλέπουμε ότι έχει 88.791.336 παράμετρους και μέγεθος 338.71MB.

```
==========================================================================
Total params: 88,791,336
Trainable params: 88,791,336
Non-trainable params: 0
--------------------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 772.54
Params size (MB): 338.71
Estimated Total Size (MB): 1111.83
--------------------------------------------------------------------------
```

Προκειμένου να γίνει η εκπαίδευση φορτώνουμε το προεκπαιδευμένο μοντέλο και διαμορφώνουμε το επίπεδο του ταξινομητή ώστε από τις 1000 κλάσεις του ImageNet να προσαρμόσουμε είτε για τις 3 κλάσεις των valence (positive, neutral, negative), energy (high, medium, low), tension (high, medium, low), είτε για τις 5 κλάσεις των emotions (anger, fear, happy, sad, tender) με τις ακόλουθες γραμμές κώδικα:

```
    model_ft = models.resnext101_32x8d(pretrained=use_pretrained)
        set_parameter_requires_grad(model_ft, feature_extract)
        num_ftrs = model_ft.fc.in_features
        model_ft.fc = nn.Linear(num_ftrs, num_classes)
```

```
        input_size = 224
```

Αλλάζει μόνο το τελευταίο Fully Connected επίπεδο (με γκρι χρώμα), μετά τα επίπεδα του τελευταίου bottleneck και του Average Pooling.

```
    (2): Bottleneck(
      (conv1): Conv2d(2048, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (conv2): Conv2d(2048, 2048, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
groups=32, bias=False)
      (bn2): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (conv3): Conv2d(2048, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=2048, out_features=1000, bias=True)
)
```

Όπου διαμορφώνεται ως εξής (για τις 5 κλάσεις το `out_features=5` αντίστοιχα)

```
  (fc): Linear(in_features=2048, out_features=3, bias=True)
```

Τα αποτελέσματα των ταξινομήσεων στα δύο σετ αποτυπώνονται στους παρακάτω πίνακες 5, 6:

Πίνακας 6: *Αποτελέσματα ταξινόμησης για το ResNext στο 360-set*

Classification Results 360-set

	Energy				Valence				Tension				Emotions			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	ResNeXt-101_32x8d (freeze)															
val	0,62	0,64	0,59	0,64	0,56	0,56	0,56	0,59	0,62	0,6	0,58	0,67	0,46	0,36	0,31	0,39
test	0,6	0,6	0,58	0,6	0,53	0,55	0,53	0,55	0,52	0,55	0,52	0,55	0,44	0,42	0,38	0,42
	ResNeXt-101_32x8d (whole)															
val	0,56	0,62	0,55	0,66	0,65	0,65	0,65	0,67	0,65	0,66	0,65	0,72	0,5	0,46	0,45	0,5
test	0,65	0,59	0,55	0,59	0,63	0,63	0,63	0,63	0,67	0,66	0,65	0,66	0,56	0,51	0,5	0,51

Πίνακας 7: Αποτελέσματα ταξινόμησης για το ResNext στο big-set

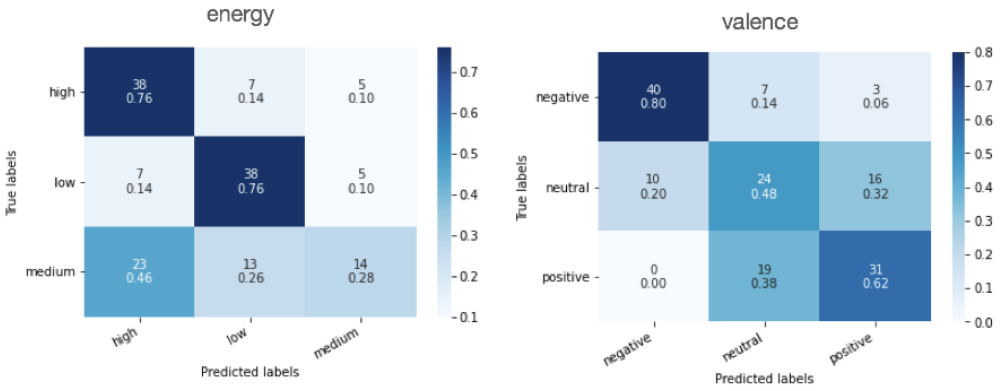
Classification Results big-set

	Energy				Valence			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	ResNeXt-101_32x8d (freeze)							
val	0,66	0,64	0,65	0,66	0,6	0,57	0,58	0,57
test	0,65	0,62	0,63	0,64	0,6	0,57	0,57	0,57
	ResNeXt-101_32x8d (whole)							
val	0,65	0,66	0,66	0,66	0,58	0,53	0,54	0,54
test	0,66	0,65	0,65	0,66	0,58	0,53	0,53	0,54

Παρατηρούμε ότι για τα πειράματα στο 360-set την καλύτερη τιμή στη μετρική f1-score την κέρδισε το ResNeXt μοντέλο που γίνεται finetuning σε όλα τα

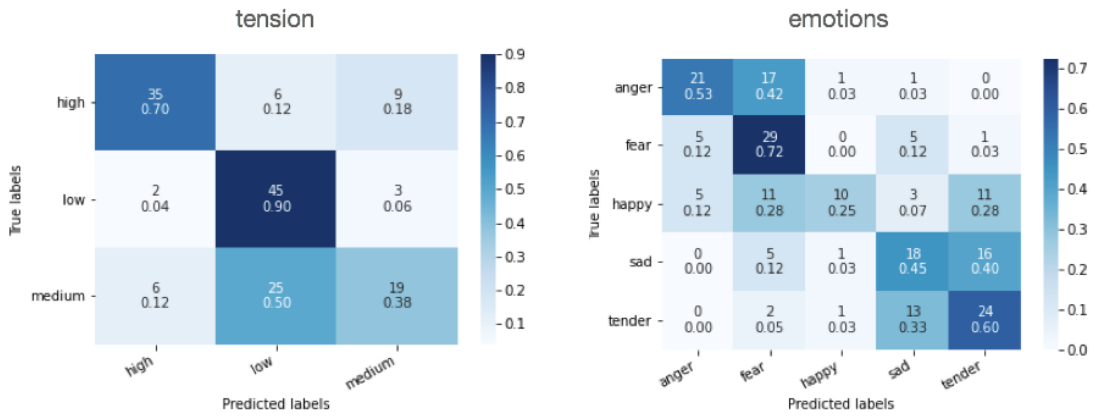
επίπεδά του εκτός από το πείραμα στο Energy όπου ελαφρώς καλύτερη επίδοση είχε το μοντέλο με τα «παγωμένα» βάρη σε όλα τα επίπεδα εκτός του τελευταίου. Παρόμοια, στο big-set για το Energy το μοντέλο με το finetuning όλων των επιπέδων ήταν ελαφρώς καλύτερο και στο Valence με 0.57 έναντι 0.53 στο f1-score προτιμήθηκε το μοντέλο με εκπαίδευση μόνο του ταξινομητή.

Ο πίνακας σύγχυσης για τα Energy και Valence (*Εικόνα 50*) μας δείχνει ότι οι λιγότερο αναγνωρίσιμες κλάσεις ήταν οι medium και neutral αντίστοιχα καθώς στο test set των 150 δειγμάτων είχαν τη μεγαλύτερη διασπορά δειγμάτων προς τις άλλες κλάσεις.



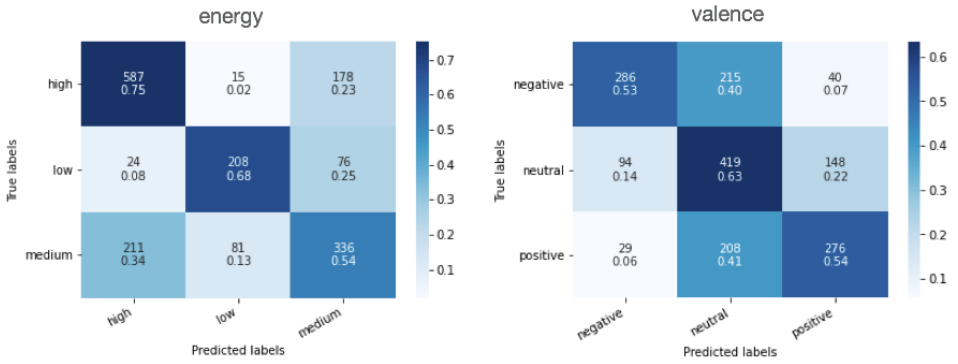
Εικόνα 50: Πίνακες σύγχυσης για Energy & Valence, 360-set, ResNeXt-101_32x8d

Στο πρόβλημα της ταξινόμησης του συναισθήματος της έντασης (tension) η κλάση medium είχε 19 δείγματα ορθώς αληθή (από τα 50 της κλάσης της) και είχε τη χαμηλότερη ευαισθησία. Από τον πίνακα σύγχυσης των emotions στην ταξινόμηση των 200 δειγμάτων του test-set η κλάση fear αναγνωρίστηκε σωστά περισσότερο από τις υπόλοιπες ενώ η κλάση happy λιγότερο με ευαισθησία μόλις 0.25 όπως φαίνεται και στην *Εικόνα 51*.



Εικόνα 51: Πίνακες σύγχυσης για Tension & Emotions, 360-set, ResNeXt-101_32x8d

Στα αποτελέσματα της ταξινόμησης του big-set με test-set 1716 δειγμάτων οι πίνακες σύγχυσης για τα Energy και Valence αποτυπώνονται στην *Εικόνα 52*:



Εικόνα 52: Πίνακες σύγχυσης για Energy & Valence, big-set, ResNeXt-101_32x8d

Στο Energy η μεγαλύτερη διαρροή δειγμάτων έγινε από την κλάση medium προς τις υπόλοιπες high και low. Στο Valence τα περισσότερα δείγματα ταξινομήθηκαν ως με ουδέτερο σθένος (neutral) και η κλάση είχε και τη μεγαλύτερη ευαισθησία με τιμή 0.63.

5.4.3 Μεταφορά Μάθησης με το VGG

Λεπτομέρειες για το VGG υπάρχουν στο 3.6.2.

Με τις ακόλουθες γραμμές εντολών και τη βοήθεια του torchsummary

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = models.vgg16_bn().to(device)
summary(model, (3, 224, 224))
```

παίρνουμε τις πληροφορίες για το προεκπαιδευμένο μοντέλο VGG16_bn που παρέχει το torchvision, βλέπουμε ότι έχει 138.365.992 παράμετρους και μέγεθος 527.82MB.

```
==========================================================================
Total params: 138,365,992
Trainable params: 138,365,992
Non-trainable params: 0
--------------------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 322.14
```

```
Params size (MB): 527.82
Estimated Total Size (MB): 850.54
----------------------------------------------------------------
```

Προκειμένου να γίνει η εκπαίδευση φορτώνουμε το προεκπαιδευμένο μοντέλο και διαμορφώνουμε το επίπεδο του ταξινομητή ώστε από τις 1000 κλάσεις του ImageNet να προσαρμόσουμε είτε για τις 3 κλάσεις των valence (positive, neutral, negative), energy (high, medium, low), tension (high, medium, low), είτε για τις 5 κλάσεις των emotions (anger, fear, happy, sad, tender) με τις ακόλουθες γραμμές κώδικα:

```
        model_ft = models.vgg16_bn(pretrained=use_pretrained)
        set_parameter_requires_grad(model_ft, feature_extract)
        num_ftrs = model_ft.classifier[6].in_features
        model_ft.classifier[6] = nn.Linear(num_ftrs,num_classes)
        input_size = 224
```

Διαμορφώνουμε κατάλληλα το τελευταίο επίπεδο του ταξινομητή (με γκρι χρώμα):

```
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
  )
)
```

Όπου γίνεται (για ταξινόμηση 5 κλάσεων `out_features=5` αντίστοιχα)

```
    (6): Linear(in_features=4096, out_features=3, bias=True)
```

Τα αποτελέσματα των ταξινομήσεων στα δύο σετ αποτυπώνονται στους παρακάτω πίνακες:

Πίνακας 8: Αποτελέσματα ταξινόμησης για το VGG στο 360-set

Classification Results 360-set

	Energy				Valence				Tension				Emotions			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	VGG16_bn (freeze)															
val	0,52	0,54	0,52	0,55	0,62	0,56	0,51	0,57	0,67	0,67	0,67	0,72	0,32	0,32	0,32	0,38
test	0,6	0,57	0,56	0,57	0,59	0,59	0,53	0,59	0,5	0,5	0,5	0,5	0,49	0,47	0,46	0,47
	VGG16_bn (whole)															
val	0,56	0,6	0,57	0,59	0,62	0,58	0,53	0,6	0,79	0,79	0,79	0,83	0,47	0,45	0,44	0,45
test	0,65	0,64	0,62	0,64	0,57	0,59	0,55	0,59	0,57	0,58	0,56	0,58	0,56	0,54	0,54	0,54

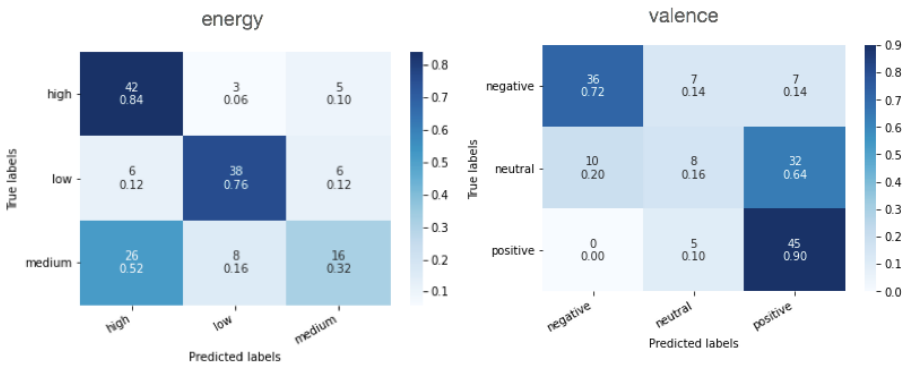
Πίνακας 9: Αποτελέσματα ταξινόμησης για το VGG στο big-set

Classification Results big-set

	Energy				Valence			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	VGG16_bn (freeze)							
val	0,66	0,66	0,66	0,66	0,56	0,54	0,54	0,54
test	0,67	0,66	0,67	0,67	0,56	0,52	0,53	0,53
	VGG16_bn (whole)							
val	0,69	0,7	0,7	0,7	0,62	0,63	0,62	0,62
test	0,7	0,71	0,7	0,7	0,64	0,65	0,64	0,64

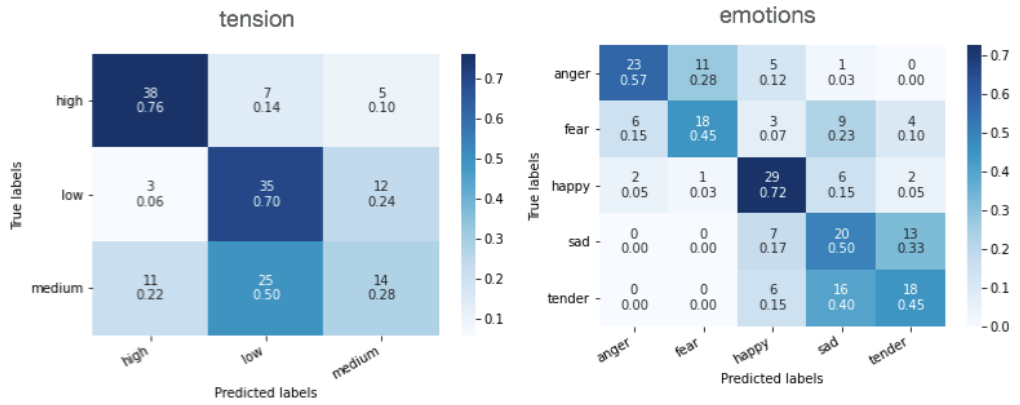
Και στα 2 σετ την καλύτερη επίδοση είχε το μοντέλο VGG16_bn(whole) δηλαδή αυτό του οποίου τα βάρη ενημερώθηκαν σε όλα τα επίπεδα.

Οι πίνακες σύγχυσης για το 360-set με test-set 150 δειγμάτων για Energy, Valence, Tension και 200 δειγμάτων για τα Emotions αποτυπώνονται στις Εικόνες *Εικόνα 53*, *Εικόνα 54*.



Εικόνα 53: Πίνακες σύγχυσης για Energy & Valence, 360-set, VGG16_bn

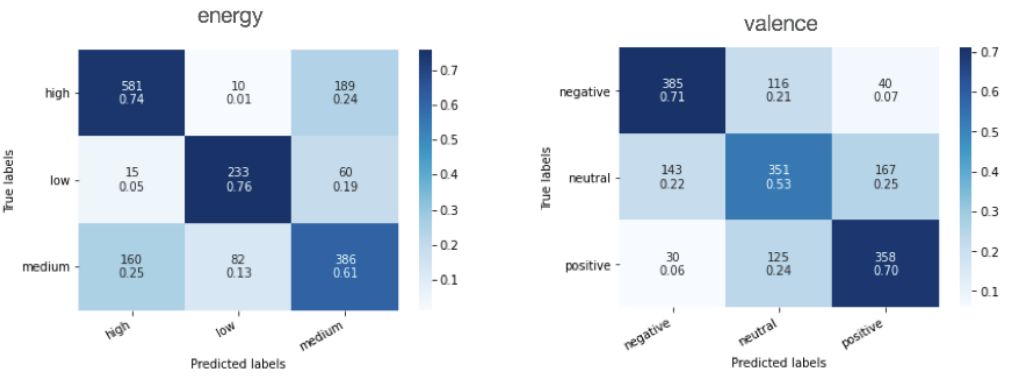
Στην ταξινόμηση της Ενέργειας (Energy) των μουσικών αποσπασμάτων η κλάση medium είχε την χαμηλότερη ευαισθησία (τιμή 0.32) και είχε και τα λιγότερα αναγνωρισμένα δείγματα. Στην ταξινόμηση του Σθένους (Valence) τα ουδέτερου σθένους μουσικά αποσπάσματα (neutral κλάση) είχαν τη μεγαλύτερη δυσκολία να αναγνωριστούν σωστά από το μοντέλο.



Εικόνα 54: Πίνακες σύγχυσης για Tension & Emotions, 360-set, VGG16_bn

Ο πίνακας σύγχυσης της Έντασης (Tension) δείχνει ότι οι μουσικές με μεσαίας κατάστασης συναισθηματική ένταση (medium tension) ήταν δύσκολο να αναγνωριστούν από το μοντέλο με μόλις 14 δείγματα αληθώς θετικά και με πολύ μικρή ευαισθησία (τιμή 0.28). Στην ταξινόμηση των συναισθημάτων (Emotions) το μοντέλο διέκρινε αρκετά καλά τα χαρούμενα δείγματα καθώς η κλάση happy είχε και τα περισσότερα (29) αληθώς θετικά και την υψηλότερη ευαισθησία (recall).

Στο big-set με τα 1715 δείγματα εκπαίδευσης, οι πίνακες σύγχυσης για τα Energy και Valence αποτυπώνονται στην *Εικόνα 55*, όπου οι μεσαίες κλάσεις medium και neutral αντίστοιχα εμφανίζουν αρκετή διαρροή δειγμάτων προς τις πιο ακραίες κλάσεις (high-low και negative-positive).



Εικόνα 55: Πίνακες σύγχυσης για Energy & Valence, big-set, VGG16_bn

5.4.4 Μεταφορά Μάθησης με το SqueezeNet

Λεπτομέρειες για το SqueezeNet υπάρχουν στο 3.6.5.

Με τις ακόλουθες γραμμές εντολών και τη βοήθεια του torchsummary

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = models.squeezenet1_0().to(device)
summary(model, (3, 224, 224))
```

παίρνουμε τις πληροφορίες για το προεκπαιδευμένο μοντέλο SqueezeNet-1.0 που παρέχει το torchvision, βλέπουμε ότι έχει 1.248.424 παράμετρους και μέγεθος 4.76MB.

```
================================================================
Total params: 1,248,424
Trainable params: 1,248,424
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.57
Forward/backward pass size (MB): 91.80
Params size (MB): 4.76
Estimated Total Size (MB): 97.14
----------------------------------------------------------------
```

Προκειμένου να γίνει η εκπαίδευση φορτώνουμε το προεκπαιδευμένο μοντέλο και διαμορφώνουμε το κατάλληλο επίπεδο του δικτύου ώστε από τις 1000 κλάσεις του ImageNet να προσαρμόσουμε είτε για τις 3 κλάσεις των valence (positive, neutral, negative), energy (high, medium, low), tension (high, medium, low), είτε για τις 5 κλάσεις των emotions (anger, fear, happy, sad, tender). Παρατηρούμε ότι η αρχιτεκτονική του SqueezeNet διαφοροποιείται σε σχέση με τις προηγούμενες καθώς ή έξοδος έρχεται από το 1ο συνελικτικό επίπεδο του ταξινομητή (σημειώνεται με γκρι χρώμα):

```
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Conv2d(512, 1000, kernel_size=(1, 1), stride=(1, 1))
    (2): ReLU(inplace=True)
    (3): AdaptiveAvgPool2d(output_size=(1, 1))
  )
)
```

Οπότε η αλλαγή γίνεται με τις ακόλουθες γραμμές κώδικα:

```
      model_ft = models.squeezenet1_0(pretrained=use_pretrained)
        set_parameter_requires_grad(model_ft, feature_extract)
        model_ft.classifier[1] = nn.Conv2d(512, num_classes,
kernel_size=(1,1), stride=(1,1))
        model_ft.num_classes = num_classes
        input_size = 224
```

και αλλάζει το επίμαχο επίπεδο σε:

```
    (1): Conv2d(512, 3, kernel_size=(1, 1), stride=(1, 1))
```

για την ταξινόμηση σε τρεις κλάσεις, ενώ για την ταξινόμηση σε πέντε κλάσεις διαμορφώνεται ως εξής:

```
(1): Conv2d(512, 5, kernel_size=(1, 1), stride=(1, 1))
```

Τα αποτελέσματα των ταξινομήσεων στα δύο σετ αποτυπώνονται στους παρακάτω πίνακες όπου στις δοκιμασίες Energy, Valence και Emotions στο 360-set ήταν καλύτερο το μοντέλο που εκπαιδεύτηκε με ενημέρωση όλων των βαρών ενώ, στη δοκιμασία του Tension το μοντέλο με ανανέωση βαρών μόνο στο επίπεδο του ταξινομητή ήταν ικανότερο, Πίνακας 10.

Πίνακας 10: Αποτελέσματα ταξινόμησης για το SqueezeNet στο 360-set

Classification Results 360-set

	Energy				**Valence**				**Tension**				**Emotions**			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	SqueezeNet 1.0 (freeze)															
val	0,58	0,54	0,52	0,58	0,34	0,48	0,39	0,52	0,52	0,52	0,52	0,58	0,31	0,36	0,29	0,3
test	0,52	0,47	0,41	0,47	0,39	0,56	0,45	0,56	0,6	0,61	0,61	0,61	0,45	0,38	0,36	0,38
	SqueezeNet 1.0 (whole)															
val	0,43	0,6	0,5	0,62	0,61	0,6	0,55	0,6	0,84	0,68	0,64	0,77	0,37	0,39	0,38	0,48
test	0,51	0,51	0,42	0,51	0,52	0,58	0,51	0,58	0,59	0,6	0,53	0,6	0,5	0,48	0,47	0,48

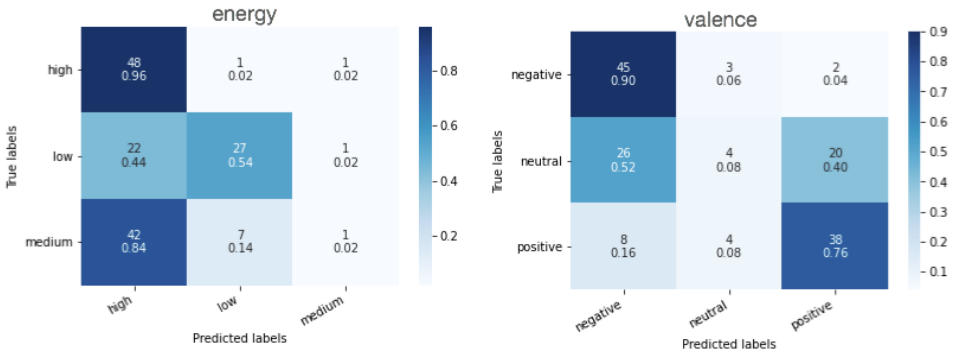
Στο big-set η ταξινόμηση ήταν περισσότερο επιτυχημένη όταν το μοντέλο έκανε ενημέρωση βαρών σε όλα τα επίπεδά του, Πίνακας 11.

Πίνακας 11: Αποτελέσματα ταξινόμησης για το SqueezeNet στο big-set

Classification Results big-set

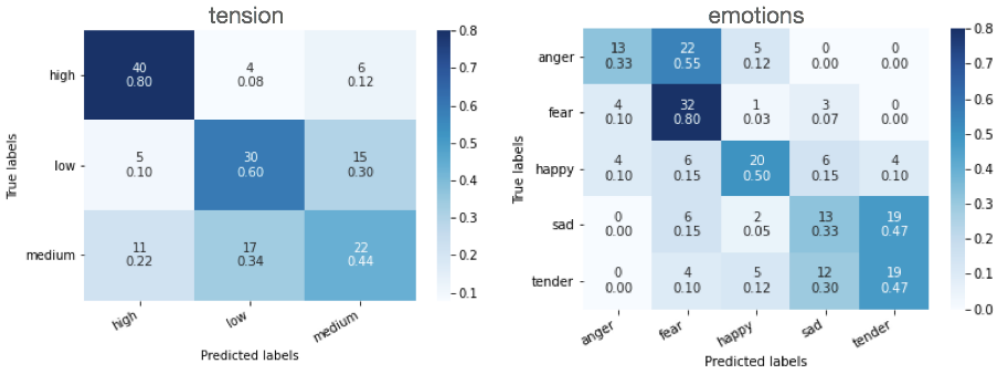
	Energy				**Valence**			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	SqueezeNet 1.0 (freeze)							
val	0,63	0,64	0,64	0,64	0,49	0,52	0,42	0,48
test	0,64	0,64	0,64	0,65	0,46	0,5	0,4	0,46
	SqueezeNet 1.0 (whole)							
val	0,71	0,68	0,69	0,7	0,61	0,61	0,61	0,6
test	0,71	0,66	0,68	0,69	0,59	0,59	0,59	0,58

Σε test-set 150 δειγμάτων (360-set) ο πίνακας σύγχυσης (*Εικόνα 56*) δείχνει αδυναμία ταξινόμησης των κλάσεων energy-medium(1 δείγμα) και valence-neutral (4 δείγματα).



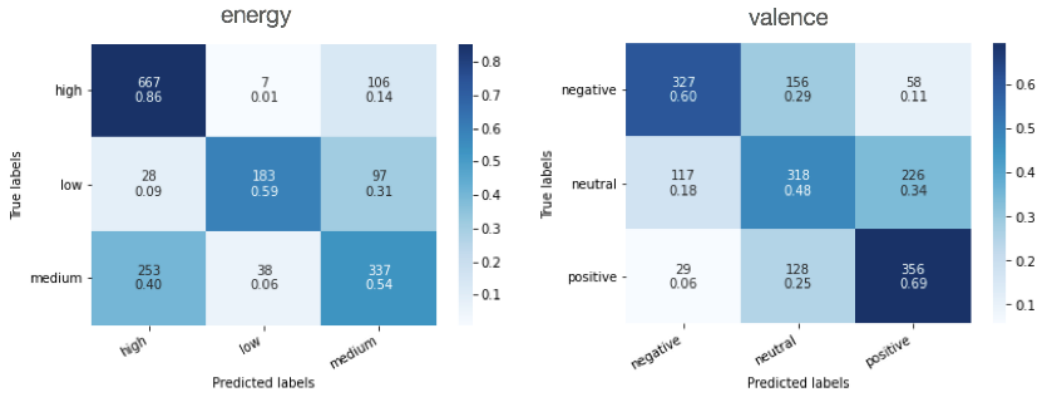
Εικόνα 56: Πίνακες σύγχυσης για Energy & Valence, 360-set, SqueezeNet 1.0 (whole)

Στη δοκιμασία ταξινόμησης Tension, ο διαχωρισμός των δειγμάτων των τριών κλάσεων ήταν περισσότερο ομαλός με την κλάση medium να υπολείπεται των άλλων δύο στην ευαισθησία. Τέλος, στην ταξινόμηση Emotions το μοντέλο έδειξε περισσότερη σιγουριά στην κλάση fear και τη λιγότερη στην κλάση sad, *Εικόνα 57*.



Εικόνα 57: Πίνακες σύγχυσης για Tension & Emotions, 360-set, SqueezeNet 1.0

Οι πίνακες σύγχυσης στο big-set (σε test-set 1715 δειγμάτων) που παρουσιάζονται στην *Εικόνα 58*, δείχνουν ότι υπάρχει σχετικά μοιρασμένη αντιπροσώπευση δειγμάτων για τις τρεις κλάσεις τόσο του Energy όσο και του Valence με τις «μεσαίες» κλάσεις να υστερούν και πάλι ως προς την ευαισθησία.



Εικόνα 58: Πίνακες σύγχυσης για Energy & Valence, big-set, SqueezeNet 1.0(whole)

5.4.5 Μεταφορά Μάθησης με το DenseNet

Λεπτομέρειες για το DenseNet υπάρχουν στο κεφάλαιο 3.6.6.

Με τις ακόλουθες γραμμές εντολών και τη βοήθεια του torchsummary

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = models.DenseNet121().to(device)
summary(model, (3, 224, 224))
```

παίρνουμε τις πληροφορίες για το προεκπαιδευμένο μοντέλο DenseNet-121 που παρέχει το torchvision, βλέπουμε ότι έχει 7.978.856 παράμετρους και μέγεθος 30.44MB.

```
==========================================================================================
==
Total params: 7,978,856
Trainable params: 7,978,856
Non-trainable params: 0
Total mult-adds (G): 2.85
==========================================================================================
==
Input size (MB): 0.57
Forward/backward pass size (MB): 172.18
Params size (MB): 30.44
Estimated Total Size (MB): 203.19
```

Προκειμένου να γίνει η εκπαίδευση φορτώνουμε το προεκπαιδευμένο μοντέλο και διαμορφώνουμε το επίπεδο του ταξινομητή ώστε από τις 1000 κλάσεις του ImageNet να προσαρμόσουμε είτε για τις 3 κλάσεις των valence (positive, neutral, negative), energy (high, medium, low), tension (high, medium, low), είτε για τις 5 κλάσεις των emotions (anger, fear, happy, sad, tender) με τις ακόλουθες γραμμές κώδικα:

```
        model_ft = models.DenseNet121(pretrained=use_pretrained)
```

```
        set_parameter_requires_grad(model_ft, feature_extract)
        num_ftrs = model_ft.classifier.in_features
        model_ft.classifier = nn.Linear(num_ftrs, num_classes)
        input_size = 224
```

Όπου το επίπεδο του ταξινομητή βλέπουμε ότι είναι το τελευταίο (γκρι χρώμα) μετά από το 16ο Dense Layer του δικτύου (απόσπασμα):

```
      (denselayer16): _DenseLayer(
        (norm1):     BatchNorm2d(992,     eps=1e-05,     momentum=0.1,     affine=True,
track_running_stats=True)
        (relu1): ReLU(inplace=True)
        (conv1): Conv2d(992, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (norm2):     BatchNorm2d(128,     eps=1e-05,     momentum=0.1,     affine=True,
track_running_stats=True)
        (relu2): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      )
    )
    (norm5):      BatchNorm2d(1024,      eps=1e-05,      momentum=0.1,      affine=True,
track_running_stats=True)
  )
  (classifier): Linear(in_features=1024, out_features=1000, bias=True)
)
```

Ο ταξινομητής προσαρμοσμένος για 3 κλάσεις:

```
  (classifier): Linear(in_features=1024, out_features=3, bias=True)
```

Και για ταξινόμηση 5 κλάσεων:

```
  (classifier): Linear(in_features=1024, out_features=5, bias=True)
```

Τα αποτελέσματα των ταξινομήσεων στα δύο σετ αποτυπώνονται στους παρακάτω πίνακες όπου παρατηρούμε ότι και για τα δύο σετ στα Energy και Valence οι δύο τεχνικές μεταφοράς μάθησης είχαν σχεδόν πανομοιότυπα αποτελέσματα ενώ, στα Tension και Emotions (του 360-set) η ενημέρωση των βαρών σε όλα τα επίπεδα του μοντέλου είχε καλύτερα αποτελέσματα.

Πίνακας 12: Αποτελέσματα ταξινόμησης για το DenseNet στο 360-set

Classification Results 360-set

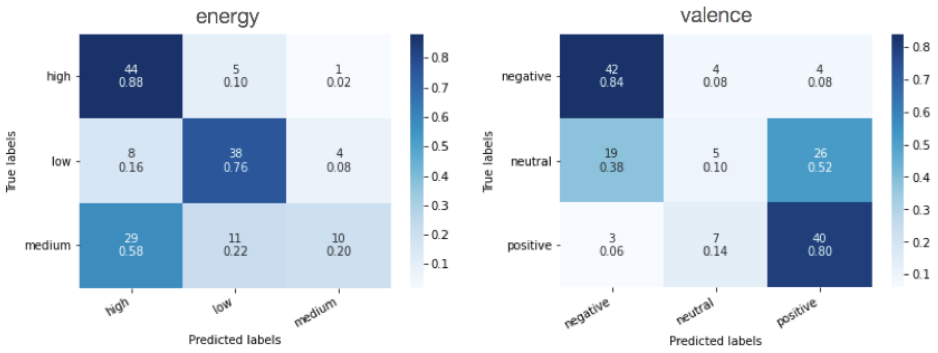
	Energy				**Valence**				**Tension**				**Emotions**			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	Densenet-121 (freeze)															
val	0,57	0,57	0,54	0,59	0,73	0,59	0,5	0,6	0,66	0,67	0,65	0,72	0,4	0,44	0,39	0,42
test	0,64	0,61	0,57	0,61	0,51	0,58	0,52	0,58	0,5	0,54	0,51	0,54	0,5	0,48	0,47	0,48
	Densenet-121 (whole)															
val	0,54	0,57	0,52	0,58	0,62	0,58	0,52	0,6	0,74	0,73	0,72	0,78	0,54	0,49	0,47	0,48
test	0,58	0,61	0,56	0,61	0,52	0,57	0,52	0,57	0,6	0,61	0,56	0,61	0,56	0,55	0,55	0,56

Πίνακας 13: Αποτελέσματα ταξινόμησης για το DenseNet στο big-set

Classification Results big-set

	Energy				Valence			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	Densenet-121 (freeze)							
val	0,66	0,68	0,67	0,67	0,6	0,56	0,57	0,56
test	0,66	0,67	0,66	0,67	0,61	0,56	0,57	0,57
	Densenet-121 (whole)							
val	0,67	0,68	0,67	0,68	0,58	0,56	0,57	0,56
test	0,67	0,68	0,67	0,68	0,58	0,56	0,56	0,56

Ο πίνακας σύγχυσης για τα Energy, Valence και Tension δείχνει την αδυναμία του δικτύου να διαχωρίσει σωστά τις ενδιάμεσες κλάσεις δηλαδή τις energy-medium, valence-neutral, tension-medium σε test set 150 δειγμάτων, όπως φαίνεται στις *Εικόνα 59Εικόνα 60*.

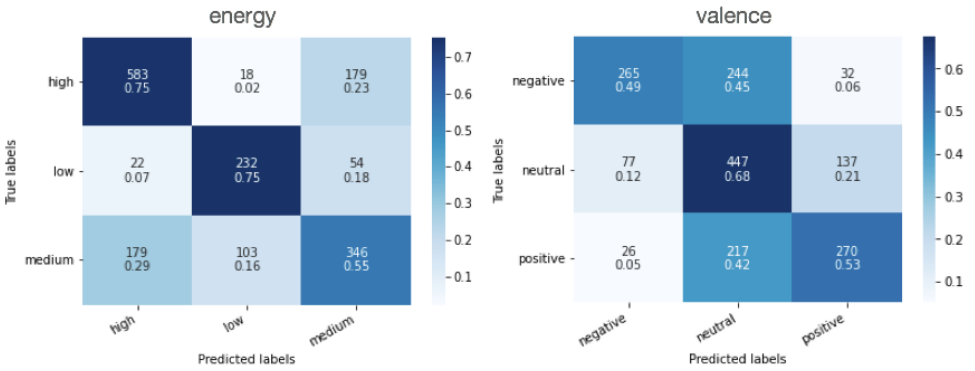


Εικόνα 59: Πίνακες σύγχυσης για Energy & Valence, 360-set, DenseNet-121(freeze)

Η ευαισθησία (recall) στις 5 κλάσεις του Emotions ήταν παρόμοια εκτός της κλάσης sad που υπολείπεται αρκετά έναντι των υπολοίπων. Έτσι , σε test-set των 200 δειγμάτων μόνο 12 καταχωρίστηκαν ως «θλιμμένα» με ευαισθησία στο 0.3, όπως παρατηρούμε στην *Εικόνα 60*.

Εικόνα 60: Πίνακες σύγχυσης για Tension & Emotions, 360-set, DenseNet-121(whole)

Οι πίνακες σύγχυσης που αφορούν στο big-set για τα έργα ταξινόμησης Energy και Valence σε test-set των 1715 δειγμάτων δείχνουν περισσότερο ισορροπημένα αποτελέσματα σε σχέση με τα αντίστοιχα του 360-set. Η κλάση energy-medium έχει ικανοποιητική παρουσία αληθώς θετικών δειγμάτων (με χαμηλότερη ευαισθησία όμως έναντι των άλλων 2 κλάσεων) και η κλάση valence-neutral έχει αυτή τη φορά την καλύτερη αντιπροσώπευση δειγμάτων, *Εικόνα 61*.



Εικόνα 61: Πίνακες σύγχυσης για Energy & Valence, big-set, DenseNet-121(whole)

5.4.6 Μεταφορά Μάθησης με το Inception

Λεπτομέρειες για το Inception υπάρχουν στο 3.6.3.

Με τις ακόλουθες γραμμές εντολών και τη βοήθεια του torchsummary

```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = models.inception_v3(init_weights=False).to(device)
summary(model, (3, 299, 299))
```

παίρνουμε τις πληροφορίες για το προεκπαιδευμένο μοντέλο Inception v3 που παρέχει το torchvision, βλέπουμε ότι έχει 23.834.568 παράμετρους και μέγεθος 90.92MB.

```
==========================================================================================
==
Total params: 23,834,568
Trainable params: 23,834,568
Non-trainable params: 0
Total mult-adds (G): 5.76
==========================================================================================
==
Input size (MB): 1.02
Forward/backward pass size (MB): 136.84
Params size (MB): 90.92
Estimated Total Size (MB): 228.79
==========================================================================================
==
```

Προκειμένου να γίνει η εκπαίδευση φορτώνουμε το προεκπαιδευμένο μοντέλο και διαμορφώνουμε τα κατάλληλα επίπεδα του δικτύου ώστε από τις 1000 κλάσεις του ImageNet να προσαρμόσουμε είτε για τις 3 κλάσεις ταξινόμησης των valence (positive, neutral, negative), energy (high, medium, low), tension (high, medium, low), είτε για τις 5 κλάσεις των emotions (anger, fear, happy, sad, tender). Το Inception v3 διαφοροποιείται από όλα τα προηγούμενα μοντέλα των πειραμάτων καθώς έχει δύο επίπεδα εξόδου, ένα fully connected που περιέχεται στο AuxLogits μέρος του δικτύου και ένα 2ο που είναι το τελευταίο fully connected απίπεδο του δικτύου (σημειώνονται με γκρι χρώμα). Κατά τη δοκιμή (inference phase), δηλαδή μετά την εκπαίδευση η έξοδος που λαμβάνεται υπόψη είναι αυτή του τελευταίου επιπέδου:

```
  (AuxLogits): InceptionAux(
    (conv0): BasicConv2d(
      (conv): Conv2d(768, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn):       BatchNorm2d(128,       eps=0.001,       momentum=0.1,       affine=True,
track_running_stats=True)
    )
    (conv1): BasicConv2d(
      (conv): Conv2d(128, 768, kernel_size=(5, 5), stride=(1, 1), bias=False)
      (bn):       BatchNorm2d(768,       eps=0.001,       momentum=0.1,       affine=True,
track_running_stats=True)
    )
    (fc): Linear(in_features=768, out_features=1000, bias=True)
  )
  (Mixed_7a): InceptionD(
  ...
  (Mixed_7b): InceptionE(
  ...
  (Mixed_7c): InceptionE(
...
   (branch_pool): BasicConv2d(
```

```
      (conv): Conv2d(2048, 192, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(192, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (dropout): Dropout(p=0.5, inplace=False)
  (fc): Linear(in_features=2048, out_features=1000, bias=True)
)
```

Με τις ακόλουθες γραμμές κώδικα κάνουμε τις απαραίτητες αλλαγές:

```
        model_ft = models.inception_v3(pretrained=use_pretrained)
         set_parameter_requires_grad(model_ft, feature_extract)
         num_ftrs = model_ft.AuxLogits.fc.in_features
         model_ft.AuxLogits.fc = nn.Linear(num_ftrs, num_classes)
         num_ftrs = model_ft.fc.in_features
         model_ft.fc = nn.Linear(num_ftrs,num_classes)
         input_size = 299
```

και τα επίμαχα επίπεδα διορθώνονται με τιμές στο out_features 3 ή 5 για ταξινόμηση τριών κλάσεων ή πέντε κλάσεων αντίστοιχα.

```
  (AuxLogits): InceptionAux(
    (conv0): BasicConv2d(
      (conv): Conv2d(768, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(128, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (conv1): BasicConv2d(
      (conv): Conv2d(128, 768, kernel_size=(5, 5), stride=(1, 1), bias=False)
      (bn): BatchNorm2d(768, eps=0.001, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (fc): Linear(in_features=768, out_features=3, bias=True)
```

```
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (dropout): Dropout(p=0.5, inplace=False)
  (fc): Linear(in_features=2048, out_features=3, bias=True)
```

Τα αποτελέσματα των ταξινομήσεων στα δύο σετ αποτυπώνονται στους παρακάτω πίνακες Πίνακας 14Πίνακας 15.

Πίνακας 14: Αποτελέσματα ταξινόμησης για το Inception στο 360-set

Classification Results 360-set

	Energy				Valence				Tension				Emotions			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	Inception v3 (freeze)															
val	0,55	0,58	0,53	0,59	0,66	0,58	0,54	0,6	0,55	0,55	0,53	0,59	0,12	0,22	0,15	0,28
test	0,59	0,59	0,55	0,59	0,59	0,58	0,52	0,58	0,46	0,51	0,45	0,51	0,1	0,17	0,1	0,17
	Inception v3 (whole)															
val	0,42	0,58	0,49	0,59	0,62	0,6	0,57	0,62	0,68	0,68	0,67	0,7	0,19	0,23	0,2	0,3
test	0,56	0,58	0,5	0,58	0,5	0,56	0,51	0,56	0,58	0,59	0,58	0,59	0,46	0,22	0,17	0,22

Για την ταξινόμηση στο 360-set του Energy και Valence το καλύτερο f1-score είχε το μοντέλο που ανανεώνει τα βάρη μόνο του ταξινομητή. Βέβαια όπως είδαμε λίγο παραπάνω, το Inception έχει άλλο ένα επίπεδο εξόδου στο επίπεδο AuxLogits οπότε θα ανανεωθεί και αυτό. Στο έργο ταξινόμησης Tension το καλύτερο μοντέλο ήταν αυτό που ανανεώνει τα βάρη όλων των επιπέδων του. Όμως, στην ταξινόμηση του Emotions κανένα από τα 2 μοντέλα δεν κατάφερε καλή γενίκευση αφού στην καλύτερη περίπτωση το f1-score είναι μόλις 0.17. Ίσως να χρειάζεται περισσότερη διερεύνηση, σε επόμενη ευκαιρία, καθώς ήδη από την εκπαίδευση με το validation set φαίνεται η αρρυθμία. Πιθανολογούμε ως αίτιο τον συνδυασμό πολλών κλάσεων – ελάχιστων δειγμάτων (στο 360-σετ).

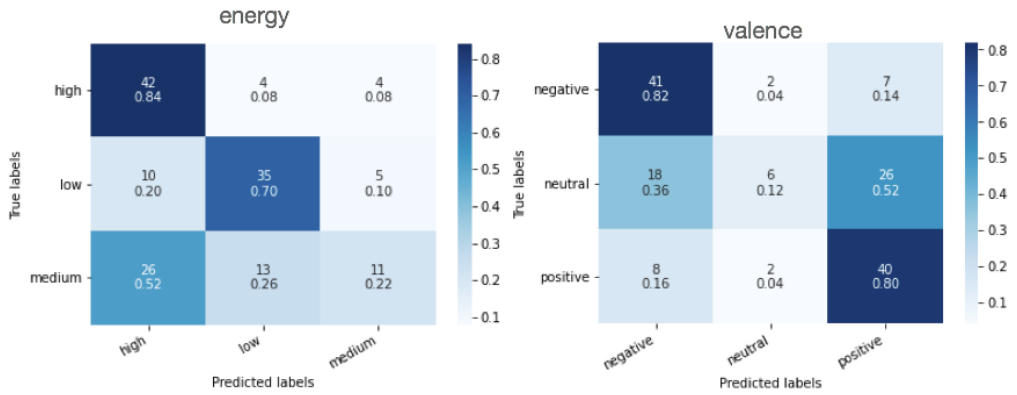
Στην ταξινόμηση με το big-set, όπως βλέπουμε στον Πίνακας 15, το επιλεγμένο δίκτυο είναι αυτό που ανανεώνει όλα τα βάρη του τόσο στο Energy όσο και στο Valence.

Πίνακας 15: Αποτελέσματα ταξινόμησης για το Inception στο big-set

Classification Results big-set

	Energy				Valence			
sets	*precision*	*recall*	*f1-score*	*accuracy*	*precision*	*recall*	*f1-score*	*accuracy*
	Inception v3 (freeze)							
val	0,64	0,66	0,65	0,65	0,55	0,54	0,54	0,54
test	0,63	0,64	0,63	0,64	0,5	0,51	0,49	0,49
	Inception v3 (whole)							
val	0,7	0,7	0,7	0,7	0,62	0,63	0,62	0,62
test	0,69	0,69	0,69	0,69	0,61	0,59	0,6	0,59

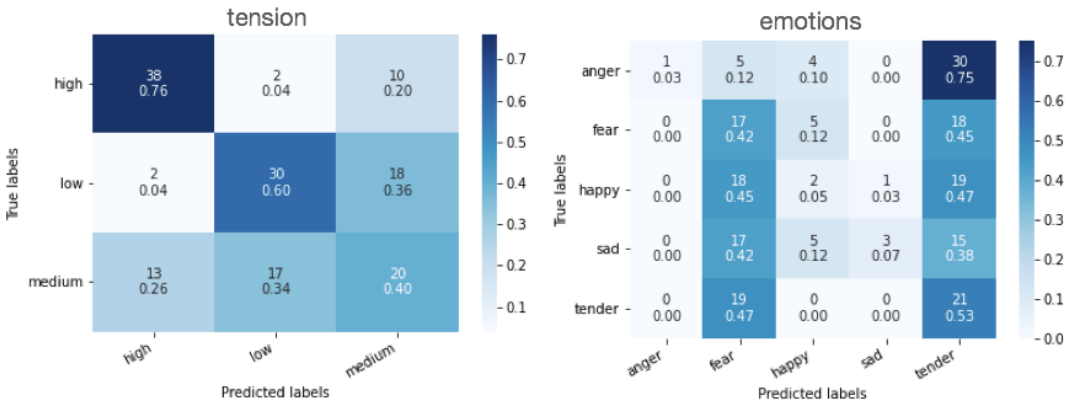
Οι πίνακες σύγχυσης για τα Energy και Valence στο 360-set, σε σετ δοκιμής 150 δειγμάτων αποτυπώνουν τα αποτελέσματα της εκπαίδευσης στην *Εικόνα 62*.



Εικόνα 62: Πίνακες σύγχυσης για Energy & Valence, 360-set, Inception v3 (freeze)

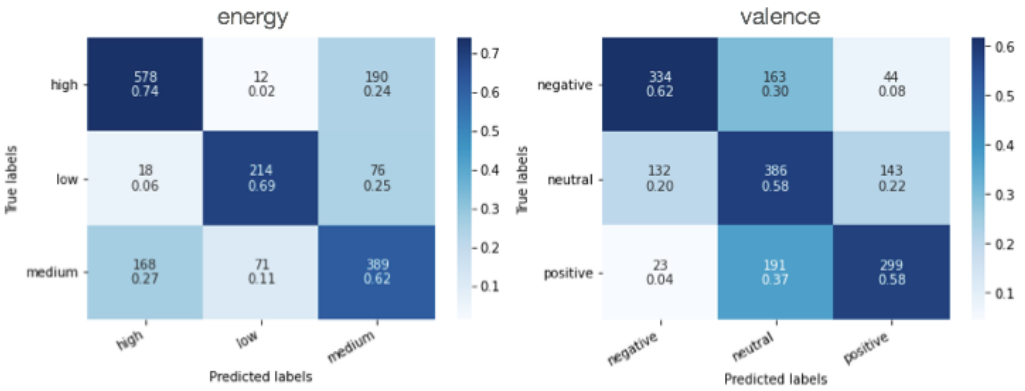
Παρατηρούμε ότι οι κλάσεις energy-medium και valence-neutral έχουν τη χαμηλότερη ευαισθησία οπότε και τα δείγματά τους διοχετεύτηκαν στις άλλες δύο κλάσεις που διατηρούνται σε πιο ισορροπημένα επίπεδα εκπροσώπησης δειγμάτων.

Παρόμοια αποτελέσματα παίρνουμε και στην ταξινόμηση του Tension, *Εικόνα 63*. Στην ταξινόμηση του Emotions και στα 200 δείγματα του σετ δοκιμής τα προβληματικά αποτελέσματα που είδαμε στον πίνακα 14 επιβεβαιώνονται και στον πίνακα σύγχυσης της *Εικόνα 63*. Για παράδειγμα τα αληθώς θετικά δείγματα της κλάσης anger είναι μόνο 1, αντίστοιχα της happy 2 και της κλάσης sad 3 από τα 50 που είχε καθεμία στο σετ δοκιμής.



Εικόνα 63: Πίνακες σύγχυσης για Tension & Emotions, 360-set, Inception v3 (whole)

Οι ταξινομήσεις στο big-set με τα 1715 δείγματα δοκιμής δείχνουν να ταξινομούν σχετικά καλά τις «δύσκολες» κλάσεις energy-medium και valence-neutral με την ευαισθησία τους να βρίσκεται σε παραπλήσιες τιμές, 0.62 και 0.58 αντίστοιχα.



Εικόνα 64: Πίνακες σύγχυσης για Energy & Valence, big-set, Inception v3 (whole)

5.4.7 Συγκεντρωτικά αποτελέσματα πειράματος Α

Από τις επί μέρους δοκιμές στα 6 προεκπαιδευμένα στο ImageNet μοντέλα που είδαμε προηγουμένως προκύπτουν τα ακόλουθα συγκεντρωτικά αποτελέσματα για τα 2 σετ των πειραμάτων.

Αποτελέσματα πειράματος Α για το 360-set

Η ταξινόμηση του χαρακτηριστικού Energy στις 3 κλάσεις (low, medium, high) έδειξε ως καλύτερο μοντέλο (βάσει macro avg f1-score) το VGG16_bn (whole) δηλαδή όταν γίνεται ρύθμιση των βαρών όλων των επιπέδων του, με σκορ 0.62.

Η ταξινόμηση του χαρακτηριστικού Valence στις 3 κλάσεις (negative, neutral, positive) έδειξε ως καλύτερο μοντέλο (βάσει macro avg f1-score) το ResNeXt-101_32x8d (whole) δηλαδή όταν γίνεται ρύθμιση των βαρών όλων των επιπέδων του, με σκορ 0.63.

Η ταξινόμηση του χαρακτηριστικού Tension στις 3 κλάσεις (low, medium, high) έδειξε ως καλύτερο μοντέλο (βάσει macro avg f1-score) το ResNeXt-101_32x8d (whole) δηλαδή όταν γίνεται ρύθμιση των βαρών όλων των επιπέδων του, με σκορ 0.65.

Η ταξινόμηση του χαρακτηριστικού Emotions στις 5 κλάσεις (anger, fear, happy, sad, tender) έδειξε ως καλύτερο μοντέλο (βάσει macro avg f1-score) το DenseNet-121 (whole) δηλαδή όταν γίνεται ρύθμιση των βαρών όλων των επιπέδων του, με σκορ 0.55.

Τα αποτελέσματα αποτυπώνονται στον Πίνακας 16

Πίνακας 16: Συγκεντρωτικά αποτελέσματα Πειράματος Α στο 360-set

Summary Results for 360-set

Network	*set*	*macro avg f1-score*			
		Energy	*Valence*	*Tension*	*Emotions*
ResNeXt-101_32x8d		0,58	0,53	0,52	0,38
ResNeXt-101_32x8d (whole)		0,55	**0,63**	**0,65**	0,5
AlexNet		0,38	0,53	0,57	0,42
AlexNet (whole)		0,51	0,61	0,55	0,35
VGG16_bn	*t*	0,56	0,53	0,5	0,46
VGG16_bn(whole)	*e*	**0,62**	0,55	0,56	0,54
SqueezeNet 1.0	*s*	0,41	0,45	0,61	0,36
SqueezeNet 1.0 (whole)	*t*	0,42	0,51	0,53	0,47
Densenet-121		0,57	0,52	0,51	0,47
Densenet-121(whole)		0,56	0,52	0,56	**0,55**
Inception v3		0,55	0,52	0,45	0,1
Inception v3(whole)		0,5	0,51	0,58	0,17

Παρατηρούμε ότι την καλύτερη επίδοση και στις 4 ταξινομήσεις έχουν τα μοντέλα στα οποία έγινε ανανέωση των βαρών όλων των επιπέδων του σε αντίθεση με εκείνα τα μοντέλα όπου τα επίπεδα του feature-extractor έμειναν παγωμένα και ανανεώθηκαν τα επίπεδα μόνο του ταξινομητή. Το αποτέλεσμα δικαιολογείται καθώς όλα τα μοντέλα ήταν ήδη εκπαιδευμένα στο ImageNet το οποίο (αν και τεράστιο σετ) δεν περιέχει ούτε στα δείγματα ούτε στις κλάσεις του φασματογραφήματα. Έτσι, αφού η εξαγωγή χαρακτηριστικών έγινε με «παγωμένα» τα αντίστοιχα επίπεδα δεν επέτρεψε στον ταξινομητή να ξεχωρίσει ικανοποιητικά τις κλάσεις των φασματογραφημάτων.

Αποτελέσματα πειράματος Α για το big-set

Η ταξινόμηση του χαρακτηριστικού Energy στις 3 κλάσεις (low, medium, high) έδειξε ως καλύτερο μοντέλο (βάσει macro avg f1-score) το VGG16_bn (whole) δηλαδή όταν γίνεται ρύθμιση των βαρών όλων των επιπέδων του, με σκορ 0.70.

Η ταξινόμηση του χαρακτηριστικού Valence στις 3 κλάσεις (negatine, neutral, positive) έδειξε ως καλύτερο μοντέλο (βάσει macro avg f1-score) πάλι το VGG16_bn (whole) δηλαδή όταν γίνεται ρύθμιση των βαρών όλων των επιπέδων του, με σκορ 0.64.

Τα αποτελέσματα αποτυπώνονται στον Πίνακας 17.

Πίνακας 17: Συγκεντρωτικά αποτελέσματα Πειράματος Α στο big-set

Summary Results for big-set

Network	*test*	*macro avg f1-score*	
		Energy	*Valence*
ResNeXt-101_32x8d		0,63	0,57
ResNeXt-101_32x8d (whole)		0,65	0,53
AlexNet		0,63	0,51
AlexNet (whole)		0,68	0,59
VGG16_bn	*t*	0,67	0,53
VGG16_bn(whole)	*e*	**0,7**	**0,64**
SqueezeNet 1.0	*s*	0,64	0,4
SqueezeNet 1.0 (whole)	*t*	0,68	0,59
Densenet-121		0,66	0,57
Densenet-121(whole)		0,67	0,56
Inception v3		0,63	0,49
Inception v3(whole)		0,69	0,6

Βλέπουμε δηλαδή παρόμοια συμπεριφορά των δικτύων και στο big-set παρόλο που είναι πολλές φορές μεγαλύτερο σε δείγματα από το 360-set. Έτσι, πάλι ένα προεκπαιδευμένο μοντέλο στο ImageNet αποδίδει καλύτερα όταν κατά την εκπαίδευσή του στο νέο γι' αυτό σετ (big-set) ενημερώνονται τα βάρη όλων των επιπέδων του.

5.5 Πείραμα Β

Με την ολοκλήρωση του πειράματος Α καταλήξαμε σε κάποιες αρχιτεκτονικές των οποίων τα μοντέλα είχαν τις καλύτερες επιδόσεις μεταξύ αυτών που είχαν προεκπαιδευτεί στο ImageNet, εκπαιδεύτηκαν στα big & 360-sets (train/val) δοκιμάστηκαν στα αντίστοιχα test-set τους και αποθηκεύτηκαν. Στο πείραμα Β και έχοντας ως σημείο αναφοράς (ground truth) το 360-set θα χρησιμοποιήσουμε πάλι την τεχνική της μεταφοράς μάθησης αλλά αυτή τη φορά τα μοντέλα θα είναι τα προεκπαιδευμένα στο big-set (αντί στο ImageNet). Τα χαρακτηριστικά που θα δοκιμαστούν στις ταξινομήσεις θα είναι αυτά των Energy και Valence καθώς υπάρχουν και στα 2 σετ. Τέλος οι συνθήκες διεξαγωγής του πειράματος Β είναι ίδιες με αυτές του πειράματος Α σε ότι αφορά αρχιτεκτονικές, υπερπαραμέτρους, αριθμό εποχών εκπαίδευσης, επομένως οι λεπτομέρειες των αρχιτεκτονικών δεν επαναλαμβάνονται στο κείμενο του πειράματος Β.

5.5.1 Αποτελέσματα Πειράματος Β

Στον Πίνακας 18 αποτυπώνονται τα αποτελέσματα της ταξινόμησης των χαρακτηριστικών Energy και Valence τόσο στο σετ επικύρωσης (val set) όσο και στο σετ δοκιμής (test set) που είναι από το 360-set. Το test set περιέχει 150

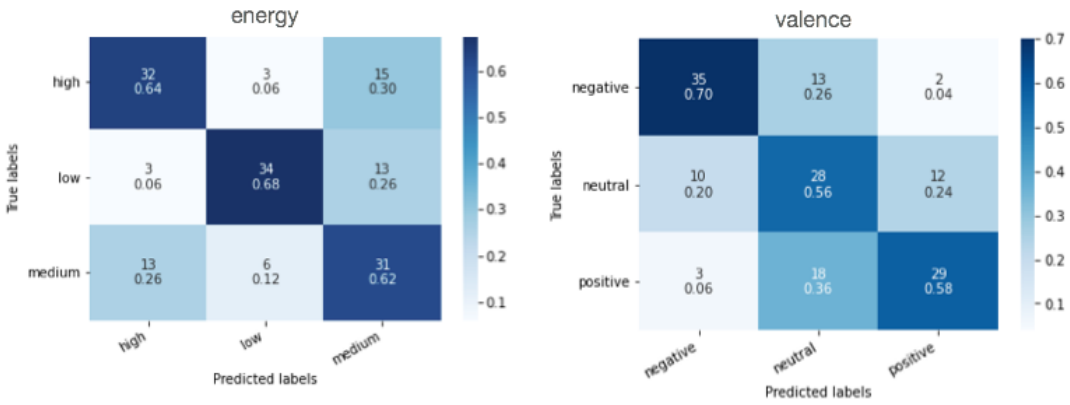
δείγματα, 50 από κάθε κλάση. Τα μοντέλα είναι τώρα προεκπαιδευμένα στο big-set και όχι στο ImageNet (Πείραμα Α). Μια πρώτη παρατήρηση είναι ότι η ταξινόμηση στο Valence είχε καλύτερα αποτελέσματα όταν τα μοντέλα ήταν στη μορφή "whole" δηλαδή όταν κατά την εκπαίδευση ενημέρωναν τα βάρη όλων των επιπέδων τους. Όμως, στο Energy βλέπουμε ότι στα ResNeXt, AlexNet, SqueezeNet η παραλλαγή του μοντέλου όπου ενημερώνει μόνο το επίπεδο του ταξινομητή ενώ τα υπόλοιπα (του μέρους του εξαγωγέα χαρακτηριστικών) παραμένουν «παγωμένα» κατά την εκπαίδευση είναι αυτό που υπερτερεί έναντι της "whole" μορφής.

Πίνακας 18: Αποτελέσματα ταξινόμησης, πείραμα Β, val-test στο 360-set, προεκπαίδευση στο big-set

Classification Results Experiment B

set	***Energy***				***Valence***			
	precision	*recall*	***f1-score***	*accuracy*	*precision*	*recall*	***f1-score***	*accuracy*
	macro avg				*macro avg*			
	ResNeXt-101_32x8d							
val	0,58	0,61	0,52	0,62	0,55	0,58	0,55	0,57
test	0,51	0,57	**0,47**	0,57	0,58	0,6	0,58	0,6
	ResNeXt-101_32x8d (whole)							
val	0,43	0,57	0,48	0,61	0,56	0,56	0,55	0,57
test	0,38	0,51	0,41	0,51	0,59	0,62	**0,59**	0,62
	AlexNet							
val	0,65	0,67	0,66	0,67	0,66	0,59	0,55	0,62
test	0,5	0,48	**0,48**	0,48	0,6	0,59	0,54	0,59
	AlexNet (whole)							
val	0,58	0,54	0,53	0,59	0,59	0,6	0,59	0,62
test	0,48	0,43	0,36	0,43	0,63	0,61	**0,62**	0,61
	VGG16_bn							
val	0,37	0,51	0,43	0,55	0,65	0,63	0,64	0,65
test	0,58	0,53	0,49	0,53	0,52	0,51	0,51	0,51
	VGG16_bn(whole)							
val	0,62	0,62	0,59	0,64	0,65	0,65	0,63	0,65
test	0,66	0,65	**0,65**	0,65	0,55	0,59	**0,54**	0,59
	SqueezeNet 1.0							
val	0,69	0,63	0,61	0,66	0,34	0,49	0,39	0,52
test	0,54	0,51	**0,48**	0,51	0,72	0,57	**0,47**	0,57
	SqueezeNet 1.0 (whole)							
val	0,61	0,62	0,54	0,66	0,47	0,35	0,23	0,43
test	0,49	0,52	0,46	0,52	0,45	0,34	0,18	0,34
	Densenet-121							
val	0,42	0,57	0,48	0,61	0,61	0,59	0,6	0,6
test	0,41	0,51	0,41	0,51	0,57	0,52	0,53	0,52
	Densenet-121(whole)							
val	0,66	0,61	0,55	0,64	0,57	0,56	0,51	0,59
test	0,51	0,51	**0,46**	0,51	0,6	0,61	**0,58**	0,61
	Inception v3							
val	0,43	0,58	0,49	0,61	0,44	0,45	0,43	0,51
test	0,45	0,39	0,3	0,39	0,37	0,35	0,35	0,35
	Inception v3(whole)							
val	0,64	0,59	0,6	0,62	0,57	0,55	0,55	0,57
test	0,58	0,5	**0,47**	0,5	0,55	0,55	**0,53**	0,55

Συνολικά, την καλύτερη επίδοση στο πείραμα Β είχε για το Energy το VGG16_bn (whole) με f1-score 0.65 και για το Valence το AlexNet (whole) με f1-score 0.62.



Εικόνα 65: Οι πίνακες σύγχυσης των καλύτερων μοντέλων (VGG16_bn-energy και AlexNet-valence) του πειράματος Β

Από την παρατήρηση της Εικόνα 65 στους πίνακες σύγχυσης φαίνεται ότι και τα δύο μοντέλα κατάφεραν να έχουν εκπροσώπηση των «δύσκολων» μεσαίων κλάσεων με κάποιον αριθμό δειγμάτων, με χαμηλότερη όμως ευαισθησία (recall) στην neutral-valence 0.56 έναντι της medium-energy 0.62.

5.6 Πείραμα Γ

Το μικρότερο σε μέγεθος σετ όλων των πειραμάτων μας ήταν το 360-set που όταν το υποβάλλαμε σε δοκιμασία ταξινόμησης πέντε κλάσεων (Emotions) στο Πείραμα Α, ο κατακερματισμός του σε train-validation-test σετ άφησε ελάχιστα δείγματα για εκπαίδευση. Στο Πείραμα Γ χρησιμοποιήθηκε η τεχνική της επαύξησης δεδομένων με σκοπό τη μεγέθυνση των train-val σετ. Τα δεδομένα αυτά προήλθαν από το Παραγωγικό Ανταγωνιστικό Δίκτυο StyleGAN2-ADA (SG2A) το οποίο εκπαιδεύσαμε με σετ το σύνολο των δειγμάτων train + val κάθε κλάσης (δηλ. anger, fear, happy, sad, tender) του 360-set. Το SG2A εκπαιδεύτηκε με τη μέθοδο της μεταφοράς μάθησης από προεκπαιδευμένο μοντέλο της NVIDIA.

5.6.1 Μεταφορά μάθησης με το StyleGAN2-ADA

Η τεχνική της μεταφοράς μάθησης έχει δοκιμαστεί στο SG2A όπως και σε άλλους τύπους ΠΑΔ με πολύ καλά αποτελέσματα, μειώνοντας έτσι ακόμα περισσότερο την ανάγκη για μεγάλα σετ εκπαίδευσης καθώς τα βάρη ενός ήδη εκπαιδευμένου μοντέλου χρησιμοποιούνται για την εκ του μηδενός εκπαίδευση ενός νέου σετ.

Το μοντέλο που χρησιμοποιήσαμε σε αυτό το πείραμα προέρχεται από τους δημιουργούς του SG2A και προέρχεται από το σετ FFHQ. Το FFHQ σετ αποτελείται από 70.000 εικόνες προσώπων υψηλής ανάλυσης και είναι αρκετά ποικιλόμορφο καθώς περιλαμβάνει πρόσωπα διαφόρων ηλικιών, εθνικοτήτων με διαφορετικά υπόβαθρα (backgrounds) και χρήση αξεσουάρ (όπως γυαλιά, καπέλα κλπ). Η μεγάλη ποικιλομορφία ενός σετ, έχει αποδειχθεί πειραματικά, ότι έχει μεγαλύτερη βαρύτητα από την θεματική ομοιότητα δύο σετ δεδομένων, σε ότι αφορά την επιτυχία της εκπαίδευσης.

Στο Πείραμα Γ ως σετ εκπαίδευσης χρησιμοποιήσαμε την ενοποιημένη μορφή των σετ εκπαίδευσης και επικύρωσης (train/val) του 360-set, όπως φαίνονται στον Πίνακας 19. Δηλαδή έτρεξαν 5 διαφορετικές εργασίες, μία για κάθε συναίσθημα με σκοπό την παραγωγή 5000 νέων φασματογραφημάτων (1000/κλάση). Στη συνέχεια αυτά τα δείγματα εκπαιδεύτηκαν με τις αρχιτεκτονικές και διαδικασίες του Πειράματος Α, εξάχθηκαν τα νέα μοντέλα και τέλος δοκιμάστηκαν στο αρχικό (και αμετάβλητο) test-set του 360-set.

Πίνακας 19: Τα 5 εξαιρετικά μικρού μεγέθους σετ εκπαίδευσης ως είσοδοι στο προεκπαιδευμένο StyleGAN2-ADA.

Emotions	*Train set*
anger	11
fear	53
happy	25
sad	35
tender	36

Η παραγωγή των τεχνητά δημιουργημένων φασματογραφημάτων έγινε εξ ολοκλήρου στο περιβάλλον του Colab. Συναντήσαμε αρκετά προβλήματα κατά την εκπαίδευση διότι οι πόροι που διαθέτει το Colab είναι πολύ περιοριστικοί σε σχέση με τις προδιαγραφές που έχουν θέσει οι δημιουργοί του SG2A. Έτσι, η εκπαίδευση του δικτύου είχε διάρκεια αρκετών ημερών για την κάθε κλάση και ήταν διακοπτόμενη λόγω των χρονικών περιορισμών που θέτει η πλατφόρμα αλλά και λόγω της (αυστηρά) μίας GPU που διαθέτει ανά συνεδρία. Επίσης παρατηρήσαμε ότι η εναλλαγή των μοντέλων GPU που ανά ημέρα μας όριζε το Colab είχε κάποια χρονική επίπτωση στην εκπαίδευση άλλοτε προς το ταχύτερο και άλλοτε όχι. Έτσι, κάθε φορά ανά τακτά χρονικά διαστήματα αποθηκεύαμε τα στιγμιότυπα (snapshots) της εκπαίδευσης ώστε την επόμενη φορά να συνεχίσουμε την εκπαίδευση από το προσωρινό μοντέλο του snapshot.

Ενδεικτικά, παρακάτω δείχνουμε τις πρώτες γραμμές από την εκκίνηση της εκτέλεσης της διαδικασίας παραγωγής τεχνητών δειγμάτων για την κλάση anger:

```
   Output directory:   /content/drive/MyDrive/train_output_anger/00000-
   stylegan2_anger-mirror-auto1-gamma10-kimg1000-ada-target0.7-resumecustom
   Training data:      /content/drive/MyDrive/stylegan2 anger
1  Training duration:  1000 kimg
2  Number of GPUs:     1
3  Number of images:   11
4  Image resolution:   256
   Conditional model:  False
5  Dataset x-flips:    True
   
   Creating output directory...
   Launching processes...
   Loading training set...
   
6  Num images:  22
7  Image shape: [3, 256, 256]
   Label shape: [0]
   
   Constructing networks...
8  Resuming from "/content/drive/MyDrive/pkl_files_transfer_learning/ffhq-res256-
   mirror-paper256-noaug.pkl"
   Setting up PyTorch plugin "bias_act_plugin"... Done.
   Setting up PyTorch plugin "upfirdn2d_plugin"... Done.
```

Όπου στο (1) διάρκεια της εκπαίδευσης, kimg σημαίνει «χιλιάδες πραγματικών εικόνων που βλέπει ο Διευκρινιστής», (3) είναι τα 11 φασματογραφήματα της

κλάσης anger, (5) χρησιμοποιούμε x-flips (οριζόντια αναστροφή) των εικόνων, στο (6) βλέπουμε ήδη την επαύξηση του σετ λόγω του x-flip (από 11 έγιναν 22), (7) διαστάσεις των εικόνων 256x256 RGB και στο (8) βλέπουμε το προεκπαιδευμένο μοντέλο στο FFHQ σετ από όπου θα ξεκινήσει η εκπαίδευση.

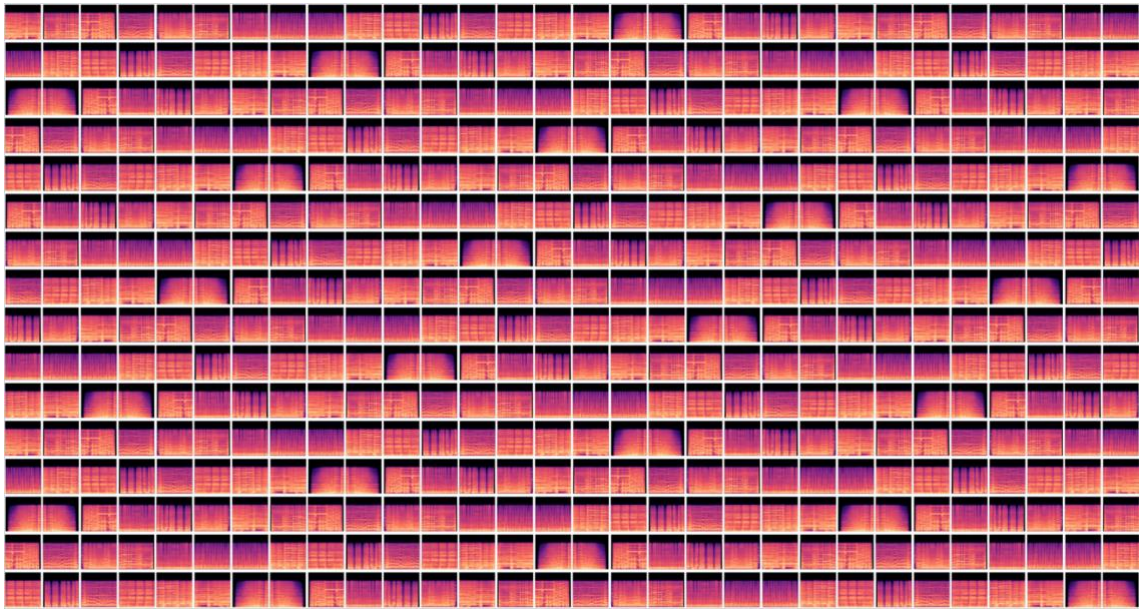
Στη συνέχεια παραθέτουμε την αρχιτεκτονική των δικτύων του Γεννήτορα (Generator) και του Διευκρινιστή (Discriminator) όπως σχηματίστηκαν:

Generator	Parameters	Buffers	Output shape	Datatype
mapping.fc0	262656	-	[16, 512]	float32
mapping.fc1	262656	-	[16, 512]	float32
mapping	-	512	[16, 14, 512]	float32
synthesis.b4.conv1	2622465	32	[16, 512, 4, 4]	float32
synthesis.b4.torgb	264195	-	[16, 3, 4, 4]	float32
synthesis.b4:0	8192	16	[16, 512, 4, 4]	float32
synthesis.b4:1	-	-	[16, 512, 4, 4]	float32
synthesis.b8.conv0	2622465	80	[16, 512, 8, 8]	float32
synthesis.b8.conv1	2622465	80	[16, 512, 8, 8]	float32
synthesis.b8.torgb	264195	-	[16, 3, 8, 8]	float32
synthesis.b8:0	-	16	[16, 512, 8, 8]	float32
synthesis.b8:1	-	-	[16, 512, 8, 8]	float32
synthesis.b16.conv0	2622465	272	[16, 512, 16, 16]	float32
synthesis.b16.conv1	2622465	272	[16, 512, 16, 16]	float32
synthesis.b16.torgb	264195	-	[16, 3, 16, 16]	float32
synthesis.b16:0	-	16	[16, 512, 16, 16]	float32
synthesis.b16:1	-	-	[16, 512, 16, 16]	float32
synthesis.b32.conv0	2622465	1040	[16, 512, 32, 32]	float16
synthesis.b32.conv1	2622465	1040	[16, 512, 32, 32]	float16
synthesis.b32.torgb	264195	-	[16, 3, 32, 32]	float16
synthesis.b32:0	-	16	[16, 512, 32, 32]	float16
synthesis.b32:1	-	-	[16, 512, 32, 32]	float32
synthesis.b64.conv0	1442561	4112	[16, 256, 64, 64]	float16
synthesis.b64.conv1	721409	4112	[16, 256, 64, 64]	float16
synthesis.b64.torgb	132099	-	[16, 3, 64, 64]	float16
synthesis.b64:0	-	16	[16, 256, 64, 64]	float16
synthesis.b64:1	-	-	[16, 256, 64, 64]	float32
synthesis.b128.conv0	426369	16400	[16, 128, 128, 128]	float16
synthesis.b128.conv1	213249	16400	[16, 128, 128, 128]	float16
synthesis.b128.torgb	66051	-	[16, 3, 128, 128]	float16
synthesis.b128:0	-	16	[16, 128, 128, 128]	float16
synthesis.b128:1	-	-	[16, 128, 128, 128]	float32
synthesis.b256.conv0	139457	65552	[16, 64, 256, 256]	float16
synthesis.b256.conv1	69761	65552	[16, 64, 256, 256]	float16
synthesis.b256.torgb	33027	-	[16, 3, 256, 256]	float16
synthesis.b256:0	-	16	[16, 64, 256, 256]	float16
synthesis.b256:1	-	-	[16, 64, 256, 256]	float32
---	---	---	---	---
Total	23191522	175568	-	-

Discriminator	Parameters	Buffers	Output shape	Datatype
b256.fromrgb	256	16	[16, 64, 256, 256]	float16
b256.skip	8192	16	[16, 128, 128, 128]	float16
b256.conv0	36928	16	[16, 64, 256, 256]	float16
b256.conv1	73856	16	[16, 128, 128, 128]	float16
b256	-	16	[16, 128, 128, 128]	float16
b128.skip	32768	16	[16, 256, 64, 64]	float16
b128.conv0	147584	16	[16, 128, 128, 128]	float16
b128.conv1	295168	16	[16, 256, 64, 64]	float16
b128	-	16	[16, 256, 64, 64]	float16
b64.skip	131072	16	[16, 512, 32, 32]	float16
b64.conv0	590080	16	[16, 256, 64, 64]	float16
b64.conv1	1180160	16	[16, 512, 32, 32]	float16
b64	-	16	[16, 512, 32, 32]	float16
b32.skip	262144	16	[16, 512, 16, 16]	float16
b32.conv0	2359808	16	[16, 512, 32, 32]	float16
b32.conv1	2359808	16	[16, 512, 16, 16]	float16

```
b32             -           16       [16, 512, 16, 16]    float16
b16.skip        262144      16       [16, 512, 8, 8]      float32
b16.conv0       2359808     16       [16, 512, 16, 16]    float32
b16.conv1       2359808     16       [16, 512, 8, 8]      float32
b16             -           16       [16, 512, 8, 8]      float32
b8.skip         262144      16       [16, 512, 4, 4]      float32
b8.conv0        2359808     16       [16, 512, 8, 8]      float32
b8.conv1        2359808     16       [16, 512, 4, 4]      float32
b8              -           16       [16, 512, 4, 4]      float32
b4.mbstd        -           -        [16, 513, 4, 4]      float32
b4.conv         2364416     16       [16, 512, 4, 4]      float32
b4.fc           4194816     -        [16, 512]            float32
b4.out          513         -        [16, 1]              float32
---             ---         ---      ---                  ---
Total           24001089    416      -                    -
```

Σε κάθε χρονικό διάστημα που έχουμε επιλέξει να παίρνουμε το στιγμιότυπο της εκπαίδευσης, εκτός από το προσωρινό μοντέλο μας επιστρέφεται και μια προεπισκόπηση παρτίδας εικόνων τόσο της πρώτης πραγματικής (real) Εικόνα 66,

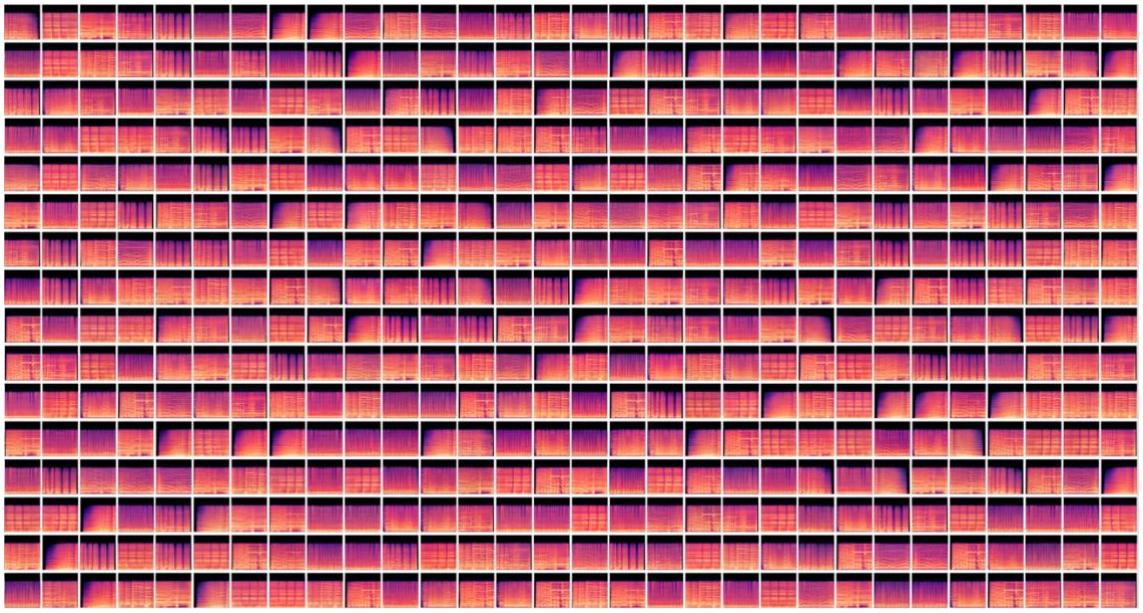


Εικόνα 66: Πραγματικά Φασματογραφήματα

που είδε ο διευκρινιστής όσο και της πρώτης τεχνητής (fake), που είναι αυτή από το μοντέλο που εκπαιδεύτηκε το SG2A, Εικόνα 67.

Εικόνα 67: Εικόνες από το προεκπαιδευμένο (fakes)

Στην Εικόνα 68 απεικονίζεται η προεπισκόπηση μιας παρτίδας τεχνητών φασματογραφημάτων και συγκεκριμένα αυτή του 4ου snapshot, δηλαδή μετά από την εμφάνιση 40kimg στον διευκρινιστή. Παρατηρούμε ότι αν και η εκπαίδευση είναι σε αρκετά πρώιμο στάδιο, καθώς έχουμε θέσει όριο ολοκλήρωσής της τα 1000kimg, τα δείγματα είναι ήδη πολύ πειστικά και η διαφοροποίησή τους από τα πρωτότυπα του σετ είναι εμφανής.



Εικόνα 68: Τεχνητά φασματογραφήματα μετά από 4 snapshots, 1 snap=10kimgs

Στο τέλος αυτής της διαδικασίας έχουμε εξάγει 1000 νέα τεχνητά δείγματα από κάθε κλάση (anger, fear, happy, sad, tender).

5.6.2 Ταξινόμηση των τεχνητών φασματογραφημάτων

Στη συνέχεια του Πειράματος Γ και με τα 5000 τεχνητά δείγματα από το StyleGAN2-ADA δημιουργούμε το νέο σετ για την ταξινόμηση του χαρακτηριστικού Emotions στις 5 κλάσεις του:

Πίνακας 20: Πείραμα Γ, το διαμορφωμένο σετ για την ταξινόμηση του Emotions

Emotions	*Train set*	*Val set*	*Test set*
	generated from SG2A		*from 360-set*
anger	800	200	40
fear	800	200	40
happy	800	200	40
sad	800	200	40
tender	800	200	40

Οι συνθήκες διεξαγωγής του πειράματος Γ είναι ίδιες με αυτές του πειράματος Α σε ότι αφορά αρχιτεκτονικές, υπερπαρμέτρους, αριθμό εποχών εκπαίδευσης, επομένως οι λεπτομέρειες των αρχιτεκτονικών δεν επαναλαμβάνονται στο κείμενο του πειράματος Γ. Στον πίνακα που ακολουθεί βλέπουμε τα αποτελέσματα της ταξινόμησης των Emotions με το νέο τεχνητό σετ, Πίνακας 21.

Πίνακας 21: Ταξινόμηση Emotions στο StyleGAN2-ADA set, Πείραμα Γ

Classification Results StyleGAN2-ADA set

set	***Emotions***			
	precision	*recall*	***f1-score***	*accuracy*
	macro avg			
	ResNeXt-101_32x8d			
val	0,88	0,88	0,88	0,88
test	0,43	0,44	0,41	0,44
	ResNeXt-101_32x8d (whole)			
val	1	1	1	1
test	0,64	0,55	0,53	0,56
	AlexNet			
val	0,76	0,75	0,76	0,76
test	0,49	0,45	0,44	0,45
	AlexNet (whole)			
val	0,98	0,98	0,98	0,98
test	0,5	0,49	0,48	0,49
	VGG16_bn			
val	0,84	0,84	0,84	0,84
test	0,49	0,47	0,43	0,47
	VGG16_bn(whole)			
val	1	1	1	1
test	0,44	0,47	0,37	0,47
	SqueezeNet 1.0			
val	0,78	0,78	0,78	0,78
test	0,52	0,53	0,51	0,53
	SqueezeNet 1.0 (whole)			
val	0,98	0,98	0,98	0,98
test	0,61	0,57	**0,58**	0,57
	Densenet-121			
val	0,84	0,84	0,84	0,84
test	0,49	0,44	0,41	0,44
	Densenet-121(whole)			
val	1	1	1	1
test	0,58	0,55	0,51	0,56
	Inception v3			
val	0,79	0,78	0,78	0,78
test	0,41	0,43	0,39	0,43
	Inception v3(whole)			
val	1	1	1	1
test	0,44	0,48	0,41	0,48

Παρατηρούμε ότι όλες οι αρχιτεκτονικές όταν κατά την εκπαίδευσή τους ενημέρωναν τα βάρη όλων των επιπέδων τους, στα train/val sets, έδωσαν μοντέλα με επίδοση (f1-score) από 0.98-1.0. Όταν τα επίπεδα του εξαγωγέα χαρακτηριστικών ήταν «παγωμένα» το ποσοστό έπεσε και κυμάνθηκε από 0.76-0.88. Στη συνέχεια όμως, οι δοκιμές των μοντέλων που εξάχθηκαν από την παραπάνω εκπαίδευση είχαν επιδόσεις που ήταν πολύ μακρινές σε σχέση με

αυτές της εκπαίδευσης που στην καλύτερη περίπτωση ήταν αυτές στο SqueezeNet1.0(whole) με f1-score 0.58. Η απόκλιση των αποτελεσμάτων μπορεί να εξηγηθεί από τη συμπεριφορά της εκπαίδευσης του SG2A. Αν και τροφοδοτήθηκε με ελάχιστα δεδομένα π.χ για το anger με 11 φασματογραφήματα, μας έδωσε 1000 τεχνητά δείγματα χωρίς να κάνει υπερπροσαρμογή. Αυτό διακρίνεται εύκολα κατά την οπτική παρατήρηση των παραγόμενων εικόνων όπου η ποιότητά τους είναι αντίστοιχη των αρχικών. Όμως, από το σετ έλειπε προφανώς η ποικιλομορφία καθώς και τα δείγματα ήταν ελάχιστα αλλά και ήταν από την ίδια κατανομή (όλα αντιστοιχούσαν στο συναίσθημα anger). Οι δημιουργοί του SG2A στο άρθρο τους αναφέρουν ότι η ποικιλομορφία στο σετ εκπαίδευσης παίζει σημαντικό ρόλο. Επίσης, αναφέρουν ότι η δοκιμή διαφορετικών τιμών στην πιθανότητα (p) της επαύξησης (που αναλύσαμε προηγουμένως) είναι σχεδόν επιβεβλημένη. Δυστυχώς τέτοιου είδους μικρορυθμίσεις (επιδέχονται τέτοιες και μερικές ακόμα υπερπαράμετροι) δεν ήταν δυνατόν να δοκιμαστούν στο πείραμα καθώς οι πόροι ήταν περιορισμένοι υπολογιστικά και χρονικά.

Παρ' όλα αυτά, στη δοκιμή με το test set του 360-set των 200 δειγμάτων (50 ανά κλάση) το SqueezeNet1.0(whole) με f1-score 0.58, έδωσε τον παρακάτω πίνακα σύγχυσης, Εικόνα 69.

Εικόνα 69: Πίνακας σύγχυσης, Emotions, Πείραμα Γ, SqueezeNet(whole)

Και οι 5 κλάσεις αντιπροσωπεύονται στα δείγματα, με αυτή του sad να έχει το χαμηλότερο 0.47 recall (ευαισθησία).

5.7 Σύνοψη και συγκεντρωτικά αποτελέσματα

Περιληπτικά θα αναφέρουμε τη διαδικασία των τριών πειραμάτων που περιλαμβάνουν τις τεχνικές της Μεταφοράς Μάθησης και της δημιουργίας τεχνητών δειγμάτων, για την κατανόηση των στοιχείων του Πίνακας 22.

Πείραμα Χ: Το πείραμα αυτό ήταν μια προεργασία που έτρεξε ολοκληρωτικά στη βιβλιοθήκη pyAudioAnalysis [27] η οποία διαθέτει, μεταξύ άλλων, αρκετούς ταξινομητές Μηχανικής Μάθησης (svm, svm_rbf, knn, extratrees, randomforest) όπου πραγματοποιήσαμε ταξινομήσεις του σετ αναφοράς (ground truth) δηλ. του 360-set, για λόγους σύγκρισης.

Πείραμα Α: Πραγματοποιήθηκαν 4 έργα ταξινόμησης χαρακτηριστικών που ανήκουν στα 2 κυριότερα μοντέλα κατάταξης του μουσικού συναισθήματος. Του διακριτού μοντέλου είναι μία ταξινόμηση 5 κλάσεων Emotions (anger, fear, happy, sad, tender) και του διαστατικού μοντέλου είναι 3 ταξινομήσεις τριών κλάσεων Energy (high, medium, low), Valence (positive, neutral, negative), Tension (high, medium, low). Οι αρχιτεκτονικές που χρησιμοποιήθηκαν ήταν 6 (ResNeXt101_32x8d, AlexNet, VGG16_bn, SqueezeNet1.0, DenseNet121, Inception_v3) και τα προεκπαιδευμένα μοντέλα τους πάρθηκαν από τη βιβλιοθήκη torchvision. Τα δεδομένα προέρχονται από α) Ιδιωτική συλλογή μέσω API Spotify, την ονομάσαμε big-set, β) δημόσια συλλογή 360 μουσικών αποσπασμάτων από πείραμα-μελέτη των Eerola & Vuoskoski, την ονομάσαμε 360-set. Οι ταξινομήσεις έγιναν και στα 2 σετ με 2 παραλλαγές κάθε μοντέλου: είτε με ενημέρωση των βαρών όλων των επιπέδων του (whole) είτε με «πάγωμα» όλων εκτός του επιπέδου του ταξινομητή (freeze). Χρησιμοποιήθηκαν αλγόριθμοι εύρεσης ιδανικού ρυθμού εκπαίδευσης και βελτιστοποίησης του αρχικά επιλεγμένου κατά τη διάρκεια της εκπαίδευσης. Όλες οι εκπαιδεύσεις σε όλα τα πειράματα είχαν διάρκεια 20 εποχές. Στον πίνακα εμφανίζονται όσα έδωσαν το καλύτερο αποτέλεσμα στο macro avg f1-score.

Πείραμα Β: Ίδιο σενάριο με το πείραμα Α αλλά αυτή τη φορά προεκπαιδεύσαμε τα μοντέλα στο big-set αντί του ImageNet και πραγματοποιήσαμε 2 ταξινομήσεις στα χαρακτηριστικά Energy και Valence αφού αυτά είναι κοινά και στα δύο σετ. Στον πίνακα εμφανίζονται όσα έδωσαν το καλύτερο αποτέλεσμα στο macro avg f1-score.

Πείραμα Γ: Το χαρακτηριστικό Emotions των 5 κλάσεων είχε την ιδιομορφία να έχει το μικρότερο αριθμό δειγμάτων ανά κλάση. Έτσι χρησιμοποιήσαμε το πολύ πρόσφατα δημοσιευμένο StyleGAN2-ADA για τη δημιουργία τεχνητών δειγμάτων (φασματογραφημάτων κλίμακας Μελ) προκειμένου στη συνέχεια να τα υποβάλλουμε στην ταξινόμηση και τη σύγκριση με το καλύτερο μοντέλο από το Πείραμα Α. Μεταφορά Μάθησης χρησιμοποιήθηκε και στο StyleGAN2-ADA και συγκεκριμένα προεκπαιδευμένο μοντέλο στο σετ εικόνων FFHQ.

Στον Πίνακας 22 εμφανίζεται η καλύτερη επίδοσή τους σε κάθε χαρακτηριστικό.

Πίνακας 22: Εμφάνιση των καλύτερων αποτελεσμάτων (f1-score) των πειραμάτων της εργασίας, τα χρώματα αντιστοιχούν στα σετ δοκιμών.

Συγκεντρωτικά αποτελέσματα πειραμάτων

ENERGY					
ΠΕΙΡΑΜΑ	*Είδος Μάθησης*	*Μοντέλο*	*Προεκπαίδευση στο*	*Δοκιμή στο*	*macro avg f1-score*
A	DL	VGG16_bn(whole)	ImageNet	test-set big-set	0,7
A	DL	VGG16_bn(whole)	ImageNet	test-set 360-set	0,62
B	DL	VGG16_bn(whole)	Big-set	test-set 360-set	0,65
X	ML	SVM		train+val 360-set	0,52
VALENCE					
ΠΕΙΡΑΜΑ	*Είδος Μάθησης*	*Μοντέλο*	*Προεκπαίδευση στο*	*Δοκιμή στο*	*macro avg f1-score*
A	DL	VGG16_bn(whole)	ImageNet	test-set big-set	0,64
A	DL	ResNeXt-101_32x8d (whole)	ImageNet	test-set 360-set	0,63
B	DL	AlexNet (whole)	Big-set	test-set 360-set	0,62
X	ML	SVM		train+val 360-set	0,56
TENSION					
ΠΕΙΡΑΜΑ	*Είδος Μάθησης*	*Μοντέλο*	*Προεκπαίδευση στο*	*Δοκιμή στο*	*macro avg f1-score*
A	DL	ResNeXt-101_32x8d (whole)	ImageNet	test-set 360-set	0,65
X	ML	RANDOMFOREST		train+val 360-set	0,53
EMOTIONS					
ΠΕΙΡΑΜΑ	*Είδος Μάθησης*	*Μοντέλο*	*Προεκπαίδευση στο*	*Δοκιμή στο*	*macro avg f1-score*
A	DL	Densenet-121(whole)	ImageNet	test-set 360-set	0,55
Γ	DL	SqueezeNet 1.0 (whole)	StyleGAN2-ADA	test-set 360-set	0,58
X	ML	SVM		train+val 360-set	0,4

Η επιλογή της μετρικής f1-score προτιμήθηκε (όπως είδαμε στο αντίστοιχο κεφάλαιο) διότι είναι μια μετρική που δίνει μεγαλύτερο βάρος στις μικρότερες κλάσεις και λόγω του υπολογισμού του αρμονικού μέσου (harmonic mean) επιβραβεύει τα μοντέλα που έχουν παρόμοιες τιμές στις μετρικές ακρίβειας (precision) και ευαισθησίας (recall or sensitivity). Ιδιαίτερα χρήσιμη ιδιότητα κυρίως στο χαρακτηριστικό Emotions που ήταν κατακερματισμένο σε 5 κλάσεις μόνο μερικών δεκάδων δειγμάτων.

Σε όλα τα πειράματα (DL) Α, Β, Γ χρησιμοποιήσαμε τεχνικές Μεταφοράς Μάθησης και βλέπουμε ότι τα επικρατέστερα μοντέλα ήταν αυτά που ονομάσαμε «whole» δηλαδή αυτά που έκαναν μικρο-ρύθμιση των βαρών όλων των επιπέδων

τους πάνω στα ήδη υπάρχοντα βάρη από το προεκπαιδευμένο δίκτυο που προήλθαν. Πιο συγκεκριμένα, με αυτή την τεχνική Μεταφοράς Μάθησης, οι παράμετροι ενός προεκπαιδευμένου μοντέλου προσαρμόζονται στα δεδομένα του νέου σετ που μπορεί να προέρχεται από άλλη κατανομή. Αυτή η περίπτωση συμβαίνει π.χ. στο Πείραμα Α όταν το VGG εκπαιδεύεται στο big-set που περιέχει μόνο εικόνες φασματογραφημάτων ενώ, το μοντέλο ήρθε προεκπαιδευμένο στο ImageNet των 1000 κλάσεων και των εκατομμυρίων εικόνων που όμως δεν περιλαμβάνει φασματογραφήματα. Βεβαίως, στο επίπεδο του ταξινομητή κάναμε αλλαγές ώστε να το προσαρμόσουμε στις 3 κλάσεις των Energy & Valence.

- ENERGY: Το προεκπαιδευμένο (στο ImageNet) μοντέλο VGG16_bn όταν εκπαιδεύτηκε στο 360-set και στη συνέχεια δοκιμάστηκε στο test-set του 360-set έδωσε f1-score 0.62 ενώ, όταν το προεκπαιδεύσαμε στο big-set, το εκπαιδεύσαμε στο 360-set και το δοκιμάσαμε στο test-set του 360-set έδωσε f1 0.65. Δηλαδή, η μεταφορά μάθησης λειτούργησε πιο αποδοτικά στη 2η περίπτωση όπου η προεκπαίδευση έχει γίνει σε σετ πολύ μικρότερου πλήθους δειγμάτων αλλά μεγαλύτερης σχετικότητας σε σχέση με το δοκιμαζόμενο σετ. Επιπλέον, ο χρόνος εκτέλεσης της εκπαίδευσης ήταν σημαντικά μικρότερος.
- VALENCE: Το ResNeXt-101_32x8d προεκπαιδευμένο στο ImageNet, όταν το εκπαιδεύσαμε στο 360-set και ταξινομήσαμε στο test-set του 360-set έδωσε f1-score 0.63. Με την ίδια εκπαίδευση και δοκιμή, το AlexNet που προεκπαιδεύσαμε στο big-set έδωσε τιμή στο f1 0.62, δηλ χαμηλότερη μόνο κατά 0.01. Βλέπουμε ότι μια αρχιτεκτονική των μόλις 8 επιπέδων απέδωσε σχεδόν παρόμοια με ένα δίκτυο 101 επιπέδων και πολύ πιο σύγχρονης αρχιτεκτονικής. Ο χρόνος εκπαίδευσης στο Colab για το AlexNet ήταν το 1/3 από αυτόν του ResNeXt.
- TENSION: Το χαρακτηριστικό Tension υπήρχε μόνο στο 360-set επομένως δεν πραγματοποιήθηκε συγκριτική δοκιμή με άλλο μοντέλο βαθιάς μηχανικής μάθησης (DL) πέραν του προεκπαιδευμένου (στο ImageNet) ResNeXt-101_32x8d το οποίο εκπαιδεύσαμε και δοκιμάσαμε στο 360-set. Πάντως, βάσει του f1-score 0.65 που πήραμε από το μοντέλο μπορούμε με σχετική ασφάλεια να υποθέσουμε ότι αν υπήρχε το

χαρακτηριστικό Tension στο big-set θα έδινε ανάλογο αποτέλεσμα, όπως έδωσε και στο Valence.

- EMOTIONS: Μια πρώτη παρατήρηση είναι ότι το Emotions είναι το μοναδικό χαρακτηριστικό των δοκιμών μας που έχει 5 κλάσεις και τα λιγότερα δείγματα ανά κλάση καθώς βρίσκεται μόνο στο 360-set (των 360 δειγμάτων). Από το Πείραμα Α, το μοντέλο με το υψηλότερο f1-score (0.55) ήταν το DenseNet-121 στη "whole" μορφή του. Με έναυσμα την έλλειψη δειγμάτων που σχολιάστηκε παραπάνω, προχωρήσαμε στην πραγματοποίηση του Πειράματος Γ δηλαδή στην παραγωγή τεχνητών δειγμάτων (φασματογραφημάτων) με τη βοήθεια του σύγχρονου ΠΑΔ της NVIDIA το StyleGAN2-ADA. Το μοντέλο του SG2A ήταν και αυτό προεκπαιδευμένο στο σετ εικόνων FFHQ. Οπότε, από τα 160 συνολικά δείγματα του train + val μέρους του 360-set, μετά από πολυήμερη εκπαίδευση, παράχθηκαν 5000 δείγματα (1000/κλάση). Στη συνέχεια το νέο σετ που δημιουργήθηκε εκπαιδεύτηκε και δοκιμάστηκε σε έργα ταξινόμησης όπως στο Πείραμα Α. Ως καλύτερο μοντέλο (προεκπαιδευμένο στο ImageNet) προκρίθηκε το SqueezeNet-1.0 στη "whole" εκδοχή του όπου απέδωσε f1-score = 0.58. Δηλαδή η απόδοσή του ήταν ελαφρώς καλύτερη (κατά 0.03) από το DenseNet που εκπαιδεύτηκε με τα πραγματικά δείγματα. Η διαφορά ίσως να μη φαίνεται μεγάλη ώστε να αξίζει τόσο χρόνο εκπαίδευσης (που είναι συντομότερος αν τρέξει σε σύγχρονη GPU ή σε συστοιχία). Όμως, αν αναλογιστούμε ότι η κλάση anger για παράδειγμα, τροφοδότησε το SG2A με μόνο 11 δείγματα ενώ το μικρότερο σετ εικόνων στο οποίο δοκιμάστηκε το SG2A από τους δημιουργούς του περιείχε 1336 εικόνες (MetFaces set), μπορούμε να δούμε τις προοπτικές αν εφαρμοστεί σε σετ των οποίων τα δείγματα είναι πραγματικά σπάνια να βρεθούν. Πιστεύουμε ότι αν υποβάλλαμε το μοντέλο (SqueezeNet) σε περαιτέρω ρύθμιση των υπερπαραμέτρων του, τα αποτελέσματα θα είναι ακόμα καλύτερα. Όμως, κάτι τέτοιο δεν ήταν εφικτό καθώς τα πειράματα ήταν πολλά και ο σκοπός τους συγκριτικός.

5.7.1 Συμπεράσματα

Σε αυτή την εργασία διερευνήσαμε τεχνικές βαθιάς μηχανικής μάθησης προκειμένου να επιτύχουμε την αναγνώριση και ταξινόμηση του μουσικού

συναισθήματος. Είχαμε στη διάθεσή μας 2 μουσικά σετ, ενός 17000 δειγμάτων ετικετοποιημένο σε κλάσεις από αλγόριθμους του Spotify (big-set) και ενός των 360 δειγμάτων ετικετοποιημένο από ειδικούς συμμετέχοντες σε μουσικολογικό-ψυχολογικό πείραμα (360-set), τα οποία και μετατρέψαμε σε φασματογραφήματα της κλίμακας Μελ. Χρησιμοποιήσαμε αρχικά, προεκπαιδευμένα στο ImageNet μοντέλα των βαθιών συνελικτικών νευρωνικών δικτύων ResNeXt101_32x8d, AlexNet, VGG16_bn, SqueezeNet1.0, DenseNet121, Inception_v3 και δείξαμε ότι αν και προεκπαιδευμένα σε «ξένο» από το ζητούμενο σετ εικόνων, μπορούν να αποδώσουν ικανοποιητικά στην ταξινόμηση αν γίνει ανανέωση των βαρών όλων των επιπέδων τους και όχι μόνο αυτών του ταξινομητή. Επίσης, κάνοντας δική μας προεκπαίδευση από τα δείγματα του big-set, δημιουργήσαμε μοντέλα και δείξαμε ότι έφεραν καλύτερα ή ισάξια αποτελέσματα στις ταξινομήσεις με αυτά των αρχικών με τη διαφορά ότι οι επιδόσεις επιτεύχθηκαν είτε με ελαφρύτερες αρχιτεκτονικές, π.χ. AlexNet αντί του ResNeXt είτε σε μικρότερο χρόνο εκπαίδευσης του ίδιου νευρωνικού.

Τέλος, στην ταξινόμηση 5 κλάσεων του 360-set που τα δείγματα για εκπαίδευση ήταν δραματικά μειωμένα κάναμε επαύξηση αυτών δημιουργώντας νέα τεχνητά φασματογραφήματα με χρήση προεκπαιδευμένου μοντέλου του StyleGAN2-ADA. Με τα τεχνητά δείγματα εκπαιδεύσαμε όλα τα ΣΝΔ των πειραμάτων και εξάγαμε τα αντίστοιχα μοντέλα. Στην ταξινόμηση που ακολούθησε με τα νέα μοντέλα, κρατήσαμε το καλύτερο και τα αποτελέσματα ήταν ανώτερα αυτών του αρχικού μοντέλου, δείχνοντας ότι σε αντίστοιχη περίπτωση ταξινόμησης σπάνιων δεδομένων, που μπορούν να μετατραπούν σε εικόνα, η μεταφορά μάθησης μαζί με την παραγωγή τεχνητών δειγμάτων δίνει ικανοποιητικά αποτελέσματα.

Σε μελλοντική επέκταση της παρούσας εργασίας θα θέλαμε να επιχειρήσουμε μια αναγωγή του προβλήματος της ταξινόμησης συναισθήματος σε πρόβλημα πολλών κλάσεων και πολλών ετικετών. Δηλαδή να φτιάξουμε μοντέλα βαθιάς μηχανικής μάθησης που να μπορούν να πουν ότι το τάδε μουσικό κομμάτι έχει π.χ. χαμηλή ενέργεια ΚΑΙ ουδέτερο σθένος ΚΑΙ μέτρια ένταση άρα έχει 70% πιθανότητες να είναι λυπητερό και 30% τρυφερό. Επίσης, ακόμα μια επέκταση θα μπορούσε να αποτελέσει και η προσπάθεια ενσωμάτωσης ακολουθιακών τεχνικών βαθιάς μηχανικής μάθησης.

Βιβλιογραφία

[1] P. R. Kleinginna and A. M. Kleinginna, “A categorized list of emotion definitions, with suggestions for a consensual definition,” *Motiv Emot*, vol. 5, no. 4, pp. 345–379, Dec. 1981, doi: 10.1007/BF00992553.

[2] J. K. Vuoskoski and T. Eerola, “The role of mood and personality in the perception of emotions represented by music,” *Cortex*, vol. 47, no. 9, pp. 1099–1106, Oct. 2011, doi: 10.1016/j.cortex.2011.04.011.

[3] I. Dufour and G. Tzanetakis, “Using Circular Models to Improve Music Emotion Recognition,” *IEEE Trans. Affective Comput.*, pp. 1–1, 2018, doi: 10.1109/TAFFC.2018.2885744.

[4] P. N. Juslin, *Musical emotions explained: unlocking the secrets of musical affect*, First edition. Oxford: Oxford University Press, 2019.

[5] T. Eerola and J. K. Vuoskoski, “A comparison of the discrete and dimensional models of emotion in music,” *Psychology of Music*, vol. 39, no. 1, pp. 18–49, Jan. 2011, doi: 10.1177/0305735610362821.

[6] T. Petri, “Exploring relationships between audio features and emotion in music,” *Front. Hum. Neurosci.*, vol. 3, 2009, doi: 10.3389/conf.neuro.09.2009.02.033.

[7] S. Chowdhury, A. Vall, V. Haunschmid, and G. Widmer, “Towards Explainable Music Emotion Recognition: The Route via Mid-level Features,” *arXiv:1907.03572 [cs, stat]*, Jul. 2019. Available: http://arxiv.org/abs/1907.03572

[8] M. B. Er and I. B. Aydilek, “Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features,” *International Journal of Computational Intelligence Systems*, p. 13.

[9] Y.-H. Yang and H. H. Chen, *Music emotion recognition*. Boca Raton, Fla. : London: CRC ; Taylor & Francis [distributor], 2011.

[10] P. Keelawat, N. Thammasan, M. Numao, and B. Kijsirikul, “Spatiotemporal Emotion Recognition using Deep CNN Based on EEG during Music Listening,” p. 11.

[11] T. Li and M. Ogihara, “Content-based music similarity search and emotion detection,” 2006.

[12] X. Liu, Q. Chen, X. Wu, Y. Liu, and Y. Liu, “CNN based music emotion classification,” *arXiv:1704.05665 [cs]*, Apr. 2017. Available: http://arxiv.org/abs/1704.05665

[13] T. Liu, L. Han, L. Ma, and D. Guo, “Audio-based deep music emotion recognition,” p. 5, doi: https://doi.org/10.1063/1.5039095.

[14] K. Palanisamy, D. Singhania, and A. Yao, “Rethinking CNN Models for Audio Classification,” *arXiv:2007.11154 [cs, eess]*, Nov. 2020. Available: http://arxiv.org/abs/2007.11154

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, “ImageNet: a Large-Scale Hierarchical Image Database,” Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[16] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Transfer learning for music classification and regression tasks,” *arXiv:1703.09179 [cs]*, Sep. 2017. Available: http://arxiv.org/abs/1703.09179
[17] N. Aneja and S. Aneja, “Transfer Learning using CNN for Handwritten Devanagari Character Recognition,” *arXiv:1909.08774 [cs, eess]*, Sep. 2019, doi: 10.1109/ICAIT47043.2019.8987286.
[18] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, “Deep Learning for Emotion Recognition on Small Datasets using Transfer Learning,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, New York, NY, USA, Nov. 2015, pp. 443–449. doi: 10.1145/2818346.2830593.
[19] S. Akcay, M. Kundegorski, M. Devereux, and T. Breckon, “Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery,” Sep. 2016, pp. 1057–1061. doi: 10.1109/ICIP.2016.7532519.
[20] A. Chatziagapi *et al.*, “Data Augmentation Using GANs for Speech Emotion Recognition,” p. 5.
[21] X. Zhu, Y. Liu, Z. Qin, and J. Li, “Data Augmentation in Emotion Classification Using Generative Adversarial Networks,” *arXiv:1711.00648 [cs]*, Dec. 2017. Available: http://arxiv.org/abs/1711.00648
[22] L. Bi, D. Feng, and J. Kim, “Improving Automatic Skin Lesion Segmentation using Adversarial Learning based Data Augmentation,” *arXiv:1807.08392 [cs]*, Jul. 2018. Available: http://arxiv.org/abs/1807.08392
[23] Z. W. Raś and A. A. Wieczorkowska, Eds., *Advances in Music Information Retrieval*, vol. 274. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-11674-2.
[24] Y.-H. Yang and H. H. Chen, “Machine Recognition of Music Emotion: A Review,” *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–30, May 2012, doi: 10.1145/2168752.2168754.
[25] T. Giannakopoulos, “Intro to Audio Analysis: Recognizing Sounds Using Machine Learning,” p. 4.
[26] T. Giannakopoulos and A. Pikrakis, *Introduction to audio analysis: a MATLAB approach*, First edition. Kidlington, Oxford: Academic Press is an imprint of Elsevier, 2014.
[27] T. Giannakopoulos, “pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis,” *PLoS ONE*, vol. 10, no. 12, p. e0144610, Dec. 2015, doi: 10.1371/journal.pone.0144610.
[28] D. Sharma and N. Kumar, “A Review on Machine Learning Algorithms, Tasks and Applications,” vol. 6, no. 10, p. 6, 2017.
[29] M. Khanam, T. Mahboob, W. Imtiaz, H. Ghafoor, and R. Sehar, “A Survey on Unsupervised Machine Learning Algorithms for Automation, Classification and Maintenance,” *International Journal of Computer Applications*, vol. 119, pp. 34–39, Jun. 2015, doi: 10.5120/21131-4058.
[30] R. Bansal, J. Singh, and R. Kaur, “Machine learning and its applications: A Review,” no. 1076, p. 8.
[31] K. Choi, G. Fazekas, K. Cho, and M. Sandler, “A Tutorial on Deep Learning for Music Information Retrieval,” *arXiv:1709.04396 [cs]*, May 2018. Available: http://arxiv.org/abs/1709.04396
[32] “Deep Learning.” https://www.deeplearningbook.org/

[33] A. Ng, “Deep Learning,” *Coursera*. https://www.coursera.org/specializations/deep-learning.
[34] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv:1207.0580 [cs]*, Jul. 2012. Available: http://arxiv.org/abs/1207.0580
[35] L. Datta, “A Survey on Activation Functions and their relation with Xavier and He Normal Initialization,” *arXiv:2004.06632 [cs]*, Mar. 2020. Available: http://arxiv.org/abs/2004.06632
[36] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” p. 6.
[37] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” *arXiv:1811.03378 [cs]*, Nov. 2018. Available: http://arxiv.org/abs/1811.03378
[38] L. Lu, Y. Shin, Y. Su, and G. Karniadakis, “Dying ReLU and Initialization: Theory and Numerical Examples,” *Communications in Computational Physics*, vol. 28, pp. 1671–1706, Nov. 2020, doi: 10.4208/cicp.OA-2020-0165.
[39] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” *arXiv:1710.05941 [cs]*, Oct. 2017,. Available: http://arxiv.org/abs/1710.05941
[40] Y. Lecun, “A Theoretical Framework for Back-Propagation,” Aug. 2001.
[41] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv:1609.04747 [cs]*, Jun. 2017. Available: http://arxiv.org/abs/1609.04747
[42] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980 [cs]*, Jan. 2017. Available: http://arxiv.org/abs/1412.6980
[43] 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, doi: 10.1145/3065386.
[44] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015. Available: http://arxiv.org/abs/1409.1556
[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *arXiv:1512.00567 [cs]*, Dec. 2015. Available: http://arxiv.org/abs/1512.00567
[46] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated Residual Transformations for Deep Neural Networks,” *arXiv:1611.05431 [cs]*, Apr. 2017. Available: http://arxiv.org/abs/1611.05431
[47] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” *arXiv:1602.07360 [cs]*, Nov. 2016. Available: http://arxiv.org/abs/1602.07360
[48] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *arXiv:1608.06993 [cs]*, Jan. 2018. Available: http://arxiv.org/abs/1608.06993
[49] I. J. Goodfellow *et al.*, “Generative Adversarial Networks,” *arXiv:1406.2661 [cs, stat]*, Jun. 2014. Available: http://arxiv.org/abs/1406.2661

[50] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, “Demystifying MMD GANs,” *arXiv:1801.01401 [cs, stat]*, Jan. 2021. Available: http://arxiv.org/abs/1801.01401
[51] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training Generative Adversarial Networks with Limited Data,” *arXiv:2006.06676 [cs, stat]*, Oct. 2020. Available: http://arxiv.org/abs/2006.06676
[52] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” *arXiv:1710.10196 [cs, stat]*, Feb. 2018. Available: http://arxiv.org/abs/1710.10196
[53] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” *arXiv:1812.04948 [cs, stat]*, Mar. 2019. Available: http://arxiv.org/abs/1812.04948
[54] D. Dalmazzo and R. Ramirez, “Mel-spectrogram Analysis to Identify Patterns in Musical Gestures: a Deep Learning Approach,” Nov. 2020.
[55] B. McFee *et al.*, “librosa: Audio and Music Signal Analysis in Python,” Austin, Texas, 2015, pp. 18–24. doi: 10.25080/Majora-7b98e3ed-003.
[56] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *arXiv:1201.0490 [cs]*, Jun. 2018. Available: http://arxiv.org/abs/1201.0490
[57] “torchvision.models — Torchvision 0.10.0 documentation.” https://pytorch.org/vision/stable/models.html.
[58] “Google Colaboratory.” https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index.
[59] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, “Benchmark Analysis of Representative Deep Neural Network Architectures,” *IEEE Access*, vol. 6, pp. 64270–64277, 2018, doi: 10.1109/ACCESS.2018.2877890.
[60] D. Silva, *davidtvs/pytorch-lr-finder*. 2021. Available: https://github.com/davidtvs/pytorch-lr-finder
[61] L. N. Smith, “Cyclical Learning Rates for Training Neural Networks,” *arXiv:1506.01186 [cs]*, Apr. 2017. Available: http://arxiv.org/abs/1506.01186
[62] “ReduceLROnPlateau — PyTorch master documentation.” https://pytorch.org/docs/master/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html.
[63] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” *arXiv:2008.05756 [cs, stat]*, Aug. 2020. Available: http://arxiv.org/abs/2008.05756
[64] P. Jadhav, “A Survey of Evaluation Metrics for MultiLabel Classi7cation,” p. 2.
[65] J. Opitz and S. Burst, “Macro F1 and Macro F1,” *arXiv:1911.03347 [cs, stat]*, Feb. 2021. Available: http://arxiv.org/abs/1911.03347
[66] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016, doi: 10.1186/s40537-016-0043-6.
[67] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A Survey on Deep Transfer Learning,” Aug. 2018. Available: https://arxiv.org/abs/1808.01974v1

[68] Y. Gao and K. Mosalam, “Deep Transfer Learning for Image-Based Structural Damage Recognition,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, Apr. 2018, doi: 10.1111/mice.12363.
[69] S. S. May, “Transfer Learning From Pre- Trained Model for Image Recognition,” p. 2.