ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Μοντέλων Βαθιάς Μάθησης και Όρασης για την Αναγνώριση Συσκευών Κλιματισμού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ραφαήλ Δάσκου

Επιβλέπων: Ευάγγελος Μαρινάκης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ
ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ
ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Μοντέλων Βαθιάς Μάθησης και Όρασης για την Αναγνώριση Συσκευών Κλιματισμού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ραφαήλ Δάσκου

Επιβλέπων: Ευάγγελος Μαρινάκης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3η Ιουλίου 2025.

.....	.....	.....
Ευάγγελος Μαρινάκης	Δημήτριος Ασκούνης	Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2025

……

Ραφαήλ Δάσκος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός
Ηλεκτρονικών Υπολογιστών Ε.Μ.Π.

Περίληψη

Η αναγνώριση και ταυτοποίηση οικιακών κλιματιστικών μονάδων με ακρίβεια αποτελεί σημαντικό βήμα για την ανάπτυξη ευφυών ενεργειακών συστημάτων και την αποδοτική διαχείριση της κατανάλωσης. Στην παρούσα εργασία προτείνεται μια προσέγγιση που αξιοποιεί μεθόδους βαθιάς μάθησης και τεχνικές υπολογιστικής όρασης, με έμφαση σε συνελικτικά νευρωνικά δίκτυα (CNNs), για την αυτόματη αναγνώριση τύπου και μοντέλου κλιματιστικών βάσει φωτογραφιών των συσκευών. Το πρόβλημα αντιμετωπίζεται ως λεπτομερής ταξινόμηση, δεδομένου ότι τα κλιματιστικά διαφορετικών κατασκευαστών παρουσιάζουν υψηλή οπτική ομοιότητα, καθιστώντας δύσκολη τη διάκρισή τους. Η προτεινόμενη μέθοδος εκπαιδεύεται σε κατάλληλο σύνολο δεδομένων και αξιολογείται ως προς την ικανότητά της να αναγνωρίζει με ακρίβεια το μοντέλο, επιτυγχάνοντας υψηλή απόδοση στην ταξινόμηση. Τα αποτελέσματα αναδεικνύουν τη δυνατότητα ενσωμάτωσης του συστήματος σε εφαρμογές κινητών συσκευών, διευκολύνοντας την άμεση ταυτοποίηση εξοπλισμού και τη δημιουργία καινοτόμων υπηρεσιών ενεργειακής διαχείρισης. Η εργασία θέτει τις βάσεις για περαιτέρω μελέτη και επέκταση της μεθόδου σε πιο ευρύ φάσμα οικιακών συσκευών.

Λέξεις κλειδιά: Βαθιά Μάθηση, Συνελικτικά Νευρωνικά Δίκτυα, Λεπτομερής Ταξινόμηση, Αναγνώριση Εικόνας, Κλιματιστικά, Ενεργειακή Διαχείριση, Υπολογιστική Όραση

Abstract

Accurate identification and recognition of home air conditioning units is a critical step towards developing intelligent energy systems and improving consumption management. This work proposes an approach leveraging deep learning methods and computer vision techniques, with an emphasis on convolutional neural networks (CNNs), for the automatic recognition of air conditioner types and models based on photographs of the devices. The problem is addressed as a fine-grained classification task, since air conditioners from different manufacturers often exhibit high visual similarity, making differentiation challenging. The proposed method is trained on a suitable dataset and evaluated for its ability to accurately recognize the model, achieving high classification performance. The results highlight the potential for deploying the system in mobile applications, enabling immediate equipment identification and facilitating innovative energy management services. This study lays the foundation for future research and the extension of the approach to a broader range of household appliances.

Keywords: Deep Learning, Convolutional Neural Networks, Fine-Grained Classification, Image Recognition, Air Conditioners, Energy Management, Computer Vision

Ευχαριστίες

Θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη προς τον επιβλέποντα καθηγητή κ. Ευάγγελο Μαρινάκη για την καθοδήγηση, την εμπιστοσύνη και τη στήριξή του καθ' όλη τη διάρκεια αυτής της διπλωματικής εργασίας. Ευχαριστώ ιδιαίτερα τον Νίκο Δημητρόπουλο για την πολύτιμη βοήθειά του, τις χρήσιμες συμβουλές και την άψογη συνεργασία μας. Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου για την αδιάκοπη στήριξη και την κατανόηση που μου έδειξαν όλα αυτά τα χρόνια, καθώς και τους φίλους μου που με ενθάρρυναν και με ενίσχυσαν κατά τη διάρκεια των σπουδών μου. Τέλος, ένα ξεχωριστό ευχαριστώ στη Σοφία, που στάθηκε δίπλα μου με αγάπη και υπομονή μέχρι το τέλος αυτής της διαδρομής.

Ραφαήλ Δάσκος
Αθήνα, 3 Ιουλίου 2025

Πίνακας Περιεχομένων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1 Εισαγωγή

1.1 Συστήματα Αναγνώρισης εικόνων

Τα συστήματα αναγνώρισης εικόνων αποτελούν έναν από τους πιο σημαντικούς κλάδους της υπολογιστικής όρασης ("computer vision") και της τεχνητής νοημοσύνης ("artificial intelligence"). Είναι το σύνολο των τεχνολογιών που επιτρέπουν στους υπολογιστές να "βλέπουν" και να "κατανοούν" τον κόσμο μέσω εικόνων και βίντεο.

Σκοπός είναι να μπορέσουν να αναλύσουν, να αναγνωρίσουν και να ερμηνεύσουν οπτικά δεδομένα προσομοιώνοντας έτσι σε μεγάλο βαθμό τον τρόπο με τον οποίο οι άνθρωποι αντιλαμβάνονται το περιβάλλον γύρω τους.

Η σημασία των συστημάτων αναγνώρισης εικόνων είναι τεράστια και αυξανόμενη καθώς επηρεάζουν πολλούς τομείς της καθημερινότητας και της βιομηχανίας. Οι βασικότεροι λόγοι για τους οποίους έχουν γίνει τόσο σημαντικά στη σύγχρονη εποχή περιλαμβάνουν:

- **Αυτοματοποίηση και Αποδοτικότητα:** επιτρέπουν την αυτοματοποίηση εργασιών που παραδοσιακά απαιτούσαν ανθρώπινη παρέμβαση με αποτέλεσμα την αύξηση ταχύτητας σε αυτές, όπως η επιθεώρηση ποιότητας σε βιομηχανίες ή ταξινόμηση εικόνων σε μεγάλες βάσεις δεδομένων.
- **Βελτίωση της ακρίβειας:** Σε πολλές περιπτώσεις οι αλγόριθμοι αναγνώρισης εικόνων μπορούν να ξεπεράσουν τα επίπεδα ακρίβειας των ανθρώπων, ιδίως σε επαναλαμβανόμενες πολύπλοκες διαδικασίες.
- **Εξαγωγή πολύτιμων πληροφοριών:** Ανίχνευση σημαντικών δεδομένων που μπορεί να μην είναι ορατά στον άνθρωπο. Τα πιο κλασικά παραδείγματα είναι η αναγνώριση παθολογιών σε ιατρικές εικόνες ή η ανάλυση δορυφορικών εικόνων για την παρακολούθηση αλλαγών του περιβάλλοντος.
- **Ενίσχυση της ανθρώπινης ικανότητας:** Συμπληρώνουν και ενισχύουν τις ανθρώπινες δεξιότητες προσφέροντας εργαλεία για ταχύτερη και πιο τεκμηριωμένη λήψη αποφάσεων.

Επομένως είναι εύκολα κατανοητό ότι τα συστήματα αναγνώρισης εικόνων έχουν πλέον εφαρμογή σε πάρα πολλούς τομείς όπως τους ακόλουθους:

- **Αναγνώριση προσώπων (Facial Recognition):** χρησιμοποιούνται σε συστήματα ασφαλείας, έξυπνες συσκευές και κοινωνικά δίκτυα για την ταυτοποίηση ατόμων.
- **Ταξινόμηση αντικειμένων (Object Classification):** αξιοποιείται σε αυτόνομα οχήματα, ρομποτικά συστήματα και βιομηχανικές διαδικασίες για την αναγνώριση και κατηγοριοποίηση αντικειμένων.
- **Ιατρική διάγνωση (Medical Imaging):** τα συστήματα αυτά επιτρέπουν την ανάλυση ακτινογραφιών, αξονικών και μαγνητικών τομογραφιών, βοηθώντας στην έγκαιρη ανίχνευση παθήσεων, όπως καρκινικοί όγκοι και νευροεκφυλιστικές ασθένειες.
- **Αναγνώριση κειμένου (Optical Character Recognition – OCR):** επιτρέπει τη μετατροπή έντυπου ή χειρόγραφου κειμένου σε ψηφιακή μορφή, διευκολύνοντας την αρχειοθέτηση, τη μετάφραση και τη δημιουργία προσβάσιμου περιεχομένου.

- **Αυτόνομη οδήγηση (self-driving cars):** οι κάμερες και οι αλγόριθμοι αναγνώρισης εικόνων βοηθούν τα αυτοοδηγούμενα οχήματα να αναγνωρίζουν σήματα κυκλοφορίας, πεζούς και άλλα οχήματα, συμβάλλοντας στην ασφαλή πλοήγηση.
- **Γεωργία:** τα συστήματα αυτά χρησιμοποιούνται για την παρακολούθηση της υγείας των καλλιεργειών, την ανίχνευση ασθενειών και τη βελτίωση της διαχείρισης της παραγωγής μέσω της ανάλυσης αγροτικών δεδομένων.
- **Ασφάλεια και επιτήρηση:** αξιοποιούνται σε συστήματα παρακολούθησης για την ανίχνευση ύποπτων δραστηριοτήτων και την αναγνώριση ατόμων σε δημόσιους και ιδιωτικούς χώρους.
- **Βιομηχανία,** οι τεχνολογίες αναγνώρισης εικόνων επιτρέπουν την ανίχνευση ελαττωμάτων στα προϊόντα, διασφαλίζοντας υψηλά πρότυπα ποιότητας στις γραμμές παραγωγής.

Οι εφαρμογές αυτές αποδεικνύουν τη σημαντική συμβολή της αναγνώρισης εικόνων στη βελτίωση της αποδοτικότητας, της ασφάλειας και της ποιότητας σε διάφορους κλάδους της κοινωνίας και της οικονομίας.

Η πρόοδος αυτή στην υπολογιστική ισχύ, σε συνδυασμό με τις βελτιώσεις στις τεχνικές μάθησης, έχει επιτρέψει τη δημιουργία αλγορίθμων αναγνώρισης εικόνων που επιτυγχάνουν επιδόσεις συγκρίσιμες με εκείνες του ανθρώπου. Παρόλα αυτά είναι σημαντικό να αναφερθεί ότι εξακολουθούν να αντιμετωπίζουν σημαντικές προκλήσεις. Μία από τις βασικότερες είναι η ανάγκη για μεγάλα σύνολα δεδομένων. Αυτό συμβαίνει καθώς η αύξηση της πολυπλοκότητας των αλγορίθμων απαιτεί και τεράστιες ποσότητες δεδομένων υψηλής ποιότητας, έτσι ώστε να επιτευχθεί ακρίβεια και αξιοπιστία.

1.2 Στόχος Διπλωματικής

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη και αξιολόγηση αλγορίθμων βαθιάς μάθησης για την αναγνώρισή τύπου και μοντέλων οικιακών συσκευών και συγκεκριμένα κλιματιστικών (air-condition) μέσω φωτογραφιών. Η αυτοματοποιημένη αναγνώριση των οικιακών συσκευών αποτελεί ένα σημαντικό πρόβλημα της υπολογιστικής όρασης, με εφαρμογές να επεκτείνονται από έξυπνα ενεργειακά συστήματα και σπίτια μέχρι της διαχείριση αποθεμάτων και ενέργειας και αυτοματοποιημένη ανάλυση εικόνων σε έξυπνες κατοικίες. Η σωστή ανάλυση και κατηγοριοποίηση διαφορετικών τύπων κλιματιστικών μπορεί να χρησιμοποιηθεί για ενεργειακή ανάλυση, έξυπνη παρακολούθηση κατανάλωσης και αυτοματοποιημένες συστάσεις χρήσεις, ενισχύοντας με αυτό τον τρόπο τη λειτουργικότητα συστημάτων που αφορούν τη βελτιστοποίηση της ενεργειακής απόδοσης.

Για την επίτευξη του στόχου αυτού, η εργασία επικεντρώνεται στη σύγκριση διαφορετικών αρχιτεκτονικών Συνελικτικών Νευρωνικών Δικτύων, αξιολογώντας την ακρίβεια αλλά και την απόδοσή τους στην ταξινόμηση των εικόνων κλιματιστικών. Οι αλγόριθμοι περιλαμβάνουν τόσο δημοφιλείς αρχιτεκτονικές CNN όπως το VGGNet όσο και πιο εξελιγμένα μοντέλα που είναι σχεδιασμένα για τέτοιου είδους ταξινομήσεις όπως τα MixDCNN, HBP-CNN και CN-CNN.

Ο λόγος χρήσης πιο εξελιγμένων μοντέλων ταξινόμησης είναι ότι το θέμα της εργασίας εντάσσεται στο τομέα λεπτομερούς ταξινόμησης (fine-grained classification), μιάς και ο στόχος είναι η αναγνώριση της μάρκας και του μοντέλου των κλιματιστικών μονάδων μέσω από φωτογραφίες. Επεκτείνεται, δηλαδή, πέρα της γενικής ταξινόμησης αντικειμένων, οπού ένα σύστημα καλείται να κατηγοριοποιήσει ένα αντικείμενο σε μια από πολλές διαφορετικές κατηγορίες (πχ. διαχωρισμός ψυγείου από κλιματιστικό).

2 Θεωρητικό Υπόβαθρο

2.1 Υπολογιστική Όραση και Επεξεργασία Εικόνας

2.1.1 Ορισμός Υπολογιστικής Όρασης

Η υπολογιστική όραση (computer vision) είναι ένας κλάδος της τεχνητής νοημοσύνης που ασχολείται με την ανάπτυξη μεθόδων για την κατανόηση και ερμηνεία οπτικών δεδομένων από τον πραγματικό κόσμο, όπως εικόνες και βίντεο. Ο στόχος είναι να αποκτήσουν οι υπολογιστές τη δυνατότητα να ”βλέπουν” και να ”αντιλαμβάνονται” το περιβάλλον τους με τρόπο παρόμοιο με την ανθρώπινη όραση, ώστε να λαμβάνουν αποφάσεις ή να εκτελούν ενέργειες με βάση τις οπτικές πληροφορίες.

Η υπολογιστική όραση ενσωματώνει γνώσεις από διάφορους τομείς, όπως την πληροφορική, τα μαθηματικά, τη στατιστική και τη μηχανική μάθηση, και χρησιμοποιείται σε ένα ευρύ φάσμα εφαρμογών, από την αυτόνομη οδήγηση και την αναγνώριση προσώπων έως την ιατρική απεικόνιση και την αυτοματοποίηση βιομηχανικών διεργασιών.

Για την επίτευξη των στόχων της, η υπολογιστική όραση βασίζεται σε μια σειρά από διαδοχικές διαδικασίες. Στην επόμενη ενότητα, παρουσιάζονται τα βασικά στάδια που συνθέτουν την τυπική ροή επεξεργασίας σε ένα σύστημα υπολογιστικής όρασης, από την απόκτηση εικόνας έως την αναγνώριση αντικειμένων. [1][2]

2.1.2 Κύριες Διαδικασίες Υπολογιστικής Όρασης

Σύμφωνα με τους Forsyth και Ponce [3], οι βασικές διαδικασίες της υπολογιστικής όρασης περιλαμβάνουν διάφορα βήματα, καθένα από τα οποία είναι κρίσιμο για την επίτευξη αποτελεσματικής και ακριβούς ανάλυσης εικόνας:

- **Απόκτηση εικόνας (Image Acquisition):**

  Η διαδικασία της απόκτησης εικόνας αποτελεί το πρώτο και θεμελιώδες βήμα στην υπολογιστική όραση και περιλαμβάνει τη λήψη οπτικών δεδομένων μέσω αισθητήρων ή καμερών. Τα δεδομένα αυτά συνήθως εκφράζονται ως τιμές εικονοστοιχείων (pixel values), και η ποιότητά τους επηρεάζει άμεσα την ακρίβεια των επόμενων σταδίων ανάλυσης. Ωστόσο, δεν αρκεί μόνο η ακρίβεια· εξίσου σημαντική είναι και η επάρκεια των εικόνων, τόσο ως προς το πλήθος όσο και ως προς την ποικιλία τους. Η επαρκής και υψηλής ποιότητας λήψη εικόνων είναι κρίσιμη για την αξιόπιστη εξαγωγή χαρακτηριστικών και την ικανότητα του συστήματος να γενικεύει αποτελεσματικά σε διαφορετικά οπτικά περιβάλλοντα.

- **Προεπεξεργασία εικόνας (Preprocessing):**

  Η προεπεξεργασία των εικόνων περιλαμβάνει ένα σύνολο τεχνικών που αποσκοπούν στη βελτίωση της ποιότητάς τους πριν από την εφαρμογή πιο σύνθετων μεθόδων ανάλυσης. Τυπικά βήματα σε αυτό το στάδιο είναι η εξομάλυνση της εικόνας, η απομάκρυνση του θορύβου, καθώς και η ενίσχυση της αντίθεσης ή της φωτεινότητας. Η προσεκτική εφαρμογή αυτών των τεχνικών είναι απαραίτητη, καθώς ο θόρυβος ή οι παραμορφώσεις στην αρχική εικόνα μπορούν να επηρεάσουν αρνητικά την ακρίβεια της εξαγωγής χαρακτηριστικών και, κατά συνέπεια, την αξιοπιστία των τελικών αποτελεσμάτων.

- **Ανάλυση χαρακτηριστικών (Feature Extraction):**

Στο στάδιο αυτό, η εικόνα αναλύεται με σκοπό τον εντοπισμό ουσιωδών χαρακτηριστικών που περιγράφουν τη δομή και το περιεχόμενό της. Τα χαρακτηριστικά αυτά μπορεί να περιλαμβάνουν ακμές, γωνίες, υφές, χρώματα ή σχήματα, και λειτουργούν ως αντιπροσωπευτικά στοιχεία για την αναγνώριση και την κατηγοριοποίηση των αντικειμένων εντός της εικόνας. Η εξαγωγή τέτοιων χαρακτηριστικών είναι κρίσιμη, καθώς αποτελούν τη βάση για πιο προχωρημένες διεργασίες, όπως η αναγνώριση αντικειμένων ή η παρακολούθηση της κίνησής τους. Για παράδειγμα, οι ακμές είναι συχνά καθοριστικές για την ανίχνευση των ορίων ανάμεσα σε διαφορετικές επιφάνειες ή αντικείμενα.

- **Αναγνώριση αντικειμένων (Object Recognition):**

  Η αναγνώριση αντικειμένων αποτελεί το στάδιο κατά το οποίο εντοπίζονται και ταυτοποιούνται αντικείμενα ή συγκεκριμένα γνωρίσματα μέσα σε μια εικόνα. Για την υλοποίηση αυτής της διαδικασίας, χρησιμοποιούνται σύγχρονοι αλγόριθμοι μηχανικής μάθησης και, κυρίως, βαθιά νευρωνικά δίκτυα (deep learning models), τα οποία εκπαιδεύονται σε μεγάλες ποσότητες δεδομένων ώστε να μπορούν να εντοπίζουν με ακρίβεια αντικείμενα διαφορετικών τύπων και σε ποικίλα περιβάλλοντα. Οι αλγόριθμοι αυτοί βασίζονται στα χαρακτηριστικά που έχουν εξαχθεί σε προηγούμενα στάδια, τα οποία και αξιοποιούν για να ταξινομήσουν ή να εντοπίσουν τα αντικείμενα στο οπτικό πεδίο.

Τα παραπάνω στάδια συνιστούν τον θεμέλιο πυρήνα κάθε εφαρμογής υπολογιστικής όρασης, καθώς καθιστούν δυνατή την αξιόπιστη κατανόηση και ερμηνεία των οπτικών δεδομένων. Χωρίς τον ορθό και συντονισμένο συνδυασμό αυτών των διαδικασιών, η αποτελεσματική ανάλυση εικόνας θα ήταν πρακτικά ανέφικτη.

2.1.3 Βασικές Αρχές Επεξεργασίας Εικόνας

Η επεξεργασία εικόνας αποτελεί υποσύνολο της υπολογιστικής όρασης και περιλαμβάνει ένα σύνολο τεχνικών που αποσκοπούν στη βελτίωση της ποιότητας των εικόνων και στην εξαγωγή χρήσιμων πληροφοριών από αυτές. Οι βασικές αρχές της επεξεργασίας εικόνας περιλαμβάνουν τα εξής:

- **Κανονικοποίηση (Normalization):**

  Η κανονικοποίηση αφορά την προσαρμογή των τιμών των pixel εντός ενός προκαθορισμένου εύρους — συνήθως $[0, 255]$ για εικόνες με 8-bit ένταση. Η διαδικασία αυτή διασφαλίζει την ομοιομορφία των δεδομένων εισόδου, γεγονός που διευκολύνει τη σύγκριση εικόνων και την εφαρμογή αλγορίθμων μηχανικής μάθησης. [4]

- **Εξομάλυνση και Απομάκρυνση Θορύβου (Smoothing and Noise Reduction):**

  Οι εικόνες συχνά περιέχουν θόρυβο, ο οποίος μπορεί να επηρεάσει την ακρίβεια της ανάλυσης. Η εξομάλυνση στοχεύει στη μείωση του θορύβου διατηρώντας παράλληλα τη βασική δομή της εικόνας. Δημοφιλή φίλτρα περιλαμβάνουν το ”Gaussian Filter”, που μειώνει τον θόρυβο με ήπια εξομάλυνση, και το ”Median Filter”, που αντικαθιστά κάθε εικονοστοιχείο με τη διάμεσο των γειτονικών του, εξαλείφοντας το θόρυβο χωρίς να αλλοιώνει έντονα τα όρια αντικειμένων. [5][6]

- **Ανίχνευση Ακμών (Edge Detection):**

  Η ανίχνευση ακμών είναι μία από τις πιο σημαντικές διαδικασίες στην επεξεργασία εικόνας, καθώς εντοπίζει τα όρια των αντικειμένων στην εικόνα. Αυτή η διαδικασία είναι

χρήσιμη για την κατανόηση της δομής της εικόνας και για την αναγνώριση των χαρακτηριστικών των αντικειμένων. Ο Sobel Filter και ο Canny Edge Detection είναι από τους πιο δημοφιλείς αλγόριθμους ανίχνευσης ακμών, οι οποίοι χρησιμοποιούνται για να εντοπίσουν μεταβολές στην ένταση της εικόνας σε διάφορες κατευθύνσεις, αποκαλύπτοντας τα όρια των αντικειμένων [7][8].

- **Ανάλυση και Εξαγωγή Χαρακτηριστικών (Feature Extraction):**

  Στο στάδιο αυτό, η εικόνα αναλύεται για να εντοπιστούν συγκεκριμένα χαρακτηριστικά που είναι χρήσιμα για την αναγνώριση και κατηγοριοποίηση αντικειμένων. Μεθόδοι όπως η SIFT (Scale-Invariant Feature Transform) [9] και η SURF (Speeded-Up Robust Features) [10] χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών που παραμένουν σταθερά παρά τις αλλαγές στην κλίμακα, τη γωνία ή την κίνηση της εικόνας.

2.2 Τεχνητή Νοημοσύνη και Μηχανική μάθηση

2.2.1 Ορισμός Τεχνητής Νοημοσύνης

Η Τεχνητή Νοημοσύνη (Artificial Intelligence - AI) είναι ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την ανάπτυξη υπολογιστικών συστημάτων ικανών να επιτελούν καθήκοντα που κανονικά απαιτούν ανθρώπινη νοημοσύνη, όπως η αντίληψη, η σκέψη, η λήψη αποφάσεων και η μάθηση.

Μπορεί να ταξινομηθεί στις εξής δύο βασικές κατηγορίες:

- **Ισχυρή:** που προσπαθεί να μιμηθεί πλήρως τις ανθρώπινες νοητικές ικανότητες
- **Ασθενής:** που επικεντρώνεται σε συγκεκριμένες εργασίες (π.χ., αναγνώριση εικόνας, μετάφραση γλώσσας)

Η σύγχρονη έρευνα επικεντρώνεται κυρίως στην ασθενή AI, αξιοποιώντας εξειδικευμένους αλγορίθμους που μπορούν να επιλύουν προβλήματα μέσα από τη μάθηση από δεδομένα. Στόχος είναι η ανάπτυξη ευφυών συστημάτων που λαμβάνουν αυτόνομες αποφάσεις και εκτελούν πολύπλοκες διαδικασίες με βάση την επεξεργασία πληροφοριών. [11]

2.2.2 Ορισμός Μηχανικής Μάθησης

Η Μηχανική Μάθηση (Machine Learning - ML) αποτελεί υποκατηγορία της τεχνητής νοημοσύνης και εστιάζεται στην ανάπτυξη αλγορίθμων οι οποίοι επιτρέπουν στους υπολογιστές να ”μαθαίνουν” από δεδομένα και να βελτιώνουν την απόδοσή τους χωρίς ρητά καθορισμένες εντολές μέσω ανθρώπινης παρέμβασης.

Ο πυρήνας της μηχανικής μάθησης βασίζεται στην ικανότητα των υπολογιστικών μοντέλων να αναγνωρίζουν μοτίβα μέσα από δεδομένα και να τροποποιούν δυναμικά τις εσωτερικές παραμέτρους τους, με σκοπό τη βελτίωση της ακρίβειας προβλέψεων ή αποφάσεων. [12]

Η διαδικασία μάθησης περιλαμβάνει την κατανόηση και αξιοποίηση υποκείμενων δομών στα δεδομένα, μέσω συνεχούς αναπροσαρμογής βάσει νέας πληροφορίας. Με την έκθεση σε μεγαλύτερο όγκο δεδομένων, τα μοντέλα μπορούν να ενισχύσουν την απόδοσή τους και να προσαρμοστούν σε νέες καταστάσεις αυτόνομα.

Στη διαδικασία της μηχανικής μάθησης αναλύονται τρία κύρια στάδια [12]:

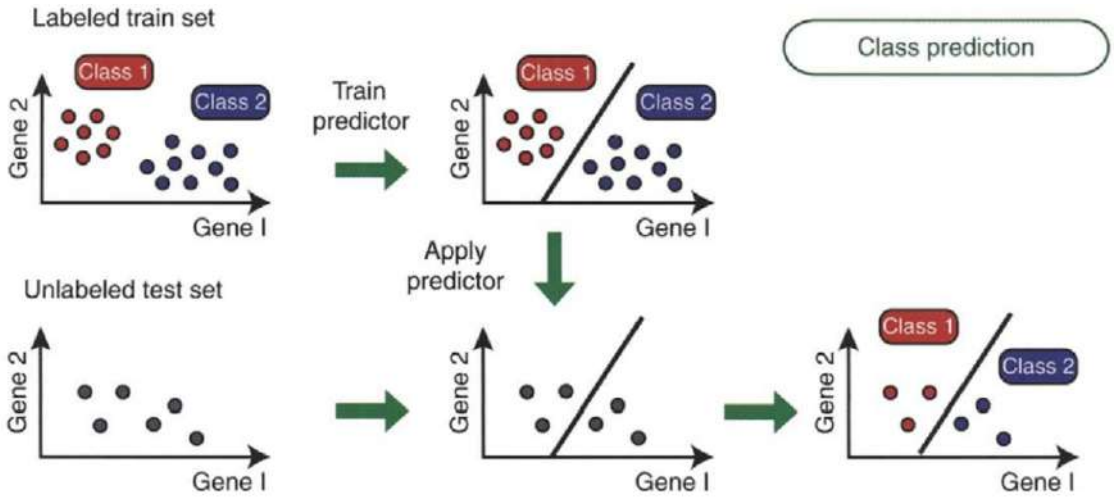
- **Ορισμός του προβλήματος μάθησης:** Προσδιορίζεται τι πρέπει να μάθει το σύστημα, ποια είναι τα διαθέσιμα δεδομένα και με ποιον τρόπο θα επιτευχθεί η μάθηση.
- **Κατασκευή του μοντέλου:** Σχεδιάζονται και εκπαιδεύονται μαθησιακοί αλγόριθμοι που αναγνωρίζουν μοτίβα στα δεδομένα. Οι μέθοδοι μάθησης διακρίνονται κυρίως σε επιβλεπόμενες (απαιτούν δεδομένα με ετικέτες) και μη επιβλεπόμενες (λειουργούν με μη ετικετοποιημένα δεδομένα) τεχνικές
- **Αξιολόγηση της απόδοσης:** Το μοντέλο αξιολογείται με νέα, ανεξάρτητα δεδομένα. Η την ακρίβεια (precision), την ανάκληση (recall) και τον συντελεστή F1 (F1-score) είναι μερικές από τις βασικές μετρικές που χρησιμοποιούνται για τη μέτρηση της απόδοσής του.

2.2.3 Υποκατηγορίες Μηχανικής Μάθησης

Η μηχανική μάθηση περιλαμβάνει ένα ευρύ φάσμα τεχνικών και αλγορίθμων, τα οποία μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες: Επιβλεπόμενη Μάθηση (Supervised Learning), Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) και Μάθηση Ενίσχυσης (Reinforcement Learning). Κάθε μία από αυτές διαφέρει ως προς τη φύση των δεδομένων που χρησιμοποιεί, τον τρόπο εκπαίδευσης και τον τύπο προβλημάτων που επιλύει. [13] [14]

2.2.3.1 Επιβλεπόμενη Μάθηση (Supervised Learning)

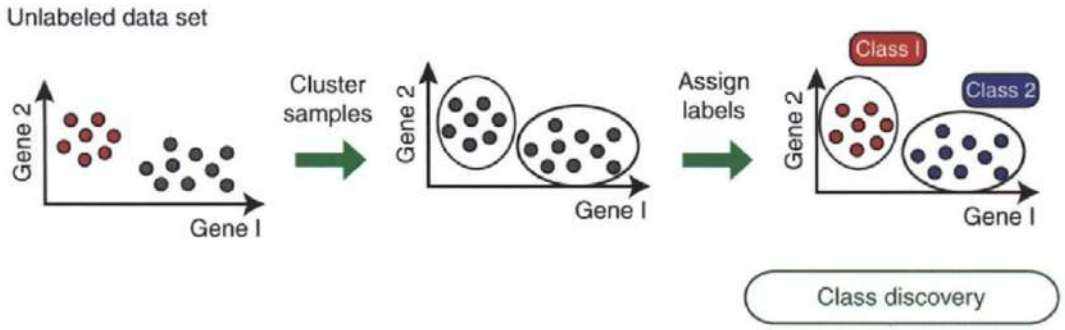
Η επιβλεπόμενη μάθηση βασίζεται στη χρήση ετικετοποιημένων δεδομένων για την εκπαίδευση του συστήματος. Δηλαδή, κάθε είσοδος συνοδεύεται από την αντίστοιχη σωστή έξοδο, και το μοντέλο μαθαίνει να συσχετίζει τις εισόδους με τις επιθυμητές εξόδους. Ο στόχος είναι η εκμάθηση μιας συνάρτησης αντιστοίχισης, η οποία μπορεί να χρησιμοποιηθεί για την πρόβλεψη αποτελεσμάτων σε νέα, άγνωστα δεδομένα. Οι βασικές κατηγορίες προβλημάτων είναι η κατηγοριοποίηση (classification) και η παλινδρόμηση (regression). Η απόδοση του μοντέλου αξιολογείται με μετρικές όπως η ακρίβεια, η ανάκληση και το F1-score. Παρόλο που η επιβλεπόμενη μάθηση μπορεί να προσφέρει υψηλή ακρίβεια, απαιτεί μεγάλο όγκο ποιοτικών, ετικετοποιημένων δεδομένων, ενώ ενδέχεται να οδηγήσει σε υπερεκπαίδευση (overfitting) αν δεν υπάρξει κατάλληρη ρύθμιση. [12][11]



Εικόνα 1: Έννοια Επιβλεπόμενης Μηχανικής Μάθησης [15]

2.2.3.2 Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)

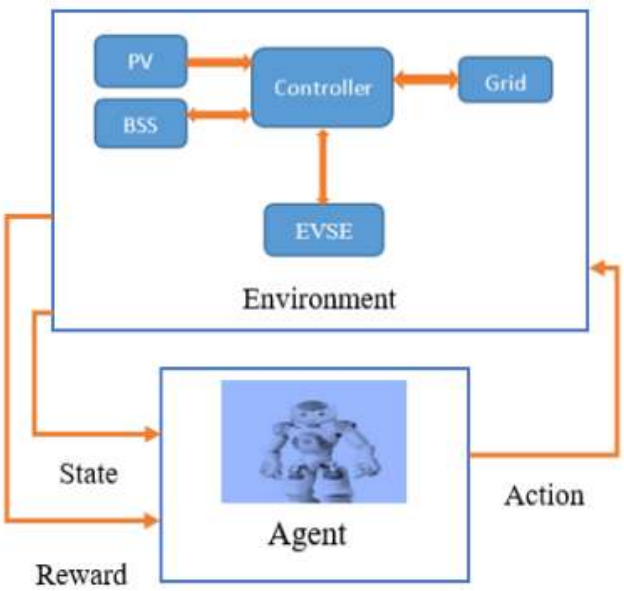
Η μη επιβλεπόμενη μάθηση εφαρμόζεται όταν τα διαθέσιμα δεδομένα δεν διαθέτουν ετικέτες. Ο αλγόριθμος επιχειρεί να ανακαλύψει μοτίβα, συσχετίσεις ή δομές στα δεδομένα χωρίς να γνωρίζει εκ των προτέρων τα σωστά αποτελέσματα. Οι πιο συνηθισμένες εφαρμογές περιλαμβάνουν την ομαδοποίηση (clustering), όπου τα δεδομένα ταξινομούνται σε ομάδες με βάση την ομοιότητα, και τη μείωση διαστάσεων (dimensionality reduction), με στόχο την απλοποίηση της αναπαράστασης χωρίς σημαντική απώλεια πληροφορίας. Αν και η μη επιβλεπόμενη μάθηση δεν απαιτεί ετικέτες, εντούτοις η αξιολόγηση των αποτελεσμάτων είναι πιο απαιτητική, καθώς δεν υπάρχει ”σωστή” απάντηση για σύγκριση. Ενδείκνυται ιδιαίτερα για εξερευνητική ανάλυση, αναγνώριση προτύπων και κατανόηση της εσωτερικής δομής των δεδομένων.[11][12]



Εικόνα 2: Έννοια μη επιβλεπόμενης μηχανικής Μάθησης [15]

2.2.3.3 Μάθηση Ενίσχυσης (Reinforcement Learning)

Η μάθηση ενίσχυσης είναι μια μέθοδος μηχανικής μάθησης στην οποία ο αλγόριθμος μαθαίνει μέσω αλληλεπίδρασης με το περιβάλλον του, με σκοπό τη μεγιστοποίηση κάποιας συνολικής ανταμοιβής. Ο πράκτορας (agent) εκτελεί ενέργειες στο περιβάλλον, παρατηρεί τα αποτελέσματα και λαμβάνει ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών. Με τον χρόνο, αναπτύσσει μια στρατηγική (policy) που μεγιστοποιεί τη μελλοντική αμοιβή. Η διαδικασία αυτή μοιάζει με το πώς τα ζώα ή οι άνθρωποι μαθαίνουν να δρουν ή να κάνουν διαδικασίες μέσω της εμπειρίας και της ανατροφοδότησης από το περιβάλλον τους. Αυτή η μορφή μάθησης εφαρμόζεται ευρέως σε περιβάλλοντα λήψης αποφάσεων σε διαδοχικά βήματα, όπως η ρομποτική, τα αυτόνομα οχήματα και τα παιχνίδια. Η βασική πρόκληση είναι η εξερεύνηση του χώρου των ενεργειών και η ισορροπία μεταξύ εξερεύνησης και εκμετάλλευσης, δηλαδή η εύρεση της βέλτιστης στρατηγικής για την επίτευξη του εκάστοτε σκοπού. [11][12]



Εικόνα 3: Μάθηση Ενίσχυσης [16]

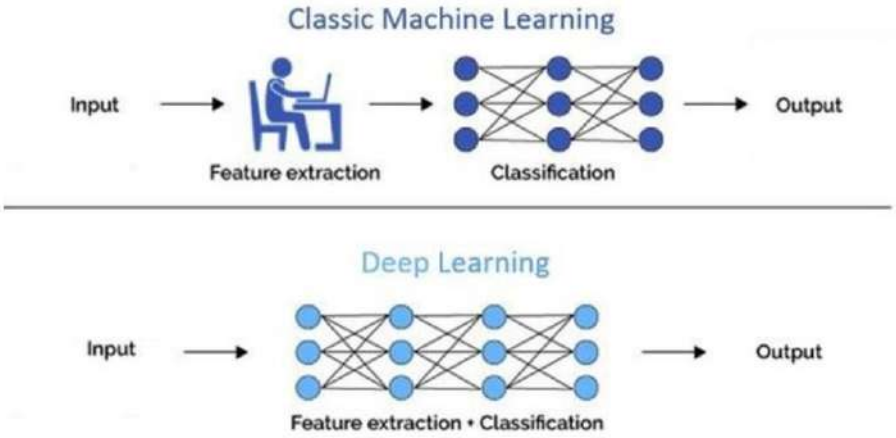
2.2.3.4 Βαθιά Μάθηση (Deep Learning)

Η Βαθιά Μάθηση (Deep Learning) αποτελεί εξειδικευμένο υποκλάδο της μηχανικής μάθησης και βασίζεται στη χρήση πολυεπίπεδων τεχνητών νευρωνικών δικτύων (deep neural networks) για την ανάλυση πολύπλοκων και μεγάλου όγκου δεδομένων. Σε αντίθεση με τις πιο απλές μορφές μηχανικής μάθησης, τα βαθιά δίκτυα είναι ικανά να εξάγουν αυτόματα αναπαραστάσεις υψηλού επιπέδου από τα ακατέργαστα δεδομένα, μαθαίνοντας χαρακτηριστικά χωρίς την ανάγκη ρητού σχεδιασμού.

Τα δίκτυα αυτά αποτελούνται από πολλαπλές διαδοχικές στρώσεις (layers), καθεμία από τις οποίες μετασχηματίζει τα δεδομένα σε διαφορετικό επίπεδο αφαιρετικότητας. Η εκπαιδευτική διαδικασία βασίζεται σε αλγόριθμους backpropagation και βελτιστοποίηση μέσω gradient descent, με τη βοήθεια μεγάλων ποσοτήτων δεδομένων και ισχυρών υπολογιστικών πόρων (π.χ. GPU).

Η βαθιά μάθηση έχει επιφέρει ριζικές εξελίξεις σε ποικίλους τομείς όπως η αναγνώριση εικόνας (π.χ. με συνελικτικά νευρωνικά δίκτυα – CNN), η ανάλυση ακολουθιών δεδομένων όπως κείμενο ή ήχος (π.χ. με αναδρομικά νευρωνικά δίκτυα – RNN), η επεξεργασία φυσικής γλώσσας (NLP) και η αυτόνομη λήψη αποφάσεων σε περιβάλλοντα πραγματικού χρόνου, όπως τα αυτόνομα οχήματα.

Παρά τις εντυπωσιακές της επιδόσεις, η βαθιά μάθηση παρουσιάζει και προκλήσεις όπως απαίτηση μεγάλων ποσοτήτων δεδομένων για εκπαίδευση, άμεση εξάρτηση από την ισχυρή υπολογιστική ισχύ και συχνά χαρακτηρίζεται από έλλειψη ερμηνευσιμότητας, γεγονός που δυσκολεύει την κατανόηση των αποφάσεων του μοντέλου. Παρόλα αυτά, η βαθιά μάθηση έχει αναδειχθεί σε βασικό πυλώνα της σύγχρονης τεχνητής νοημοσύνης και χρησιμοποιείται ήδη σε κρίσιμες εφαρμογές, όπως αναγνώριση προσώπων, αυτόματη μετάφραση, διάγνωση από ιατρικές εικόνες, και ανάλυση συναισθήματος σε κείμενο ή ομιλία.[14][17]



Εικόνα 4: Machine Learning vs Deep Learning [18]

2.3 Νευρωνικά Δίκτυα

2.3.1 Τεχνητός Νευρώνας

Οι τεχνητοί νευρώνες αποτελούν τα θεμελιώδη δομικά στοιχεία των τεχνητών νευρωνικών δικτύων (Artificial Neural Networks – ANNs). Εμπνέονται από τη βιολογική λειτουργία των νευρώνων στον ανθρώπινο εγκέφαλο και στοχεύουν στην προσομοίωση της διαδικασίας λήψης και επεξεργασίας πληροφορίας από τον ανθρώπινο εγκέφαλο[19].

Στο βιολογικό νευρικό σύστημα, οι νευρώνες λαμβάνουν ηλεκτροχημικά σήματα μέσω των δενδριτών, τα επεξεργάζονται στο σώμα του κυττάρου και, εφόσον το συνολικό σήμα υπερβεί ένα συγκεκριμένο κατώφλι ενεργοποίησης, αποστέλλουν νέο σήμα μέσω του νευράξονα προς άλλους νευρώνες. Η λειτουργία αυτή έχει αποτελέσει τη βάση για την ανάπτυξη των μαθηματικών υπολογιστικών μοντέλων των τεχνητών νευρώνων.

Μοντέλο McCulloch-Pitts

Το πρώτο μαθηματικό μοντέλο τεχνητού νευρώνα προτάθηκε από τους McCulloch και Pitts το 1943. Πρόκειται για έναν απλοποιημένο υπολογιστικό κόμβο που λαμβάνει πολλαπλές εισόδους, υπολογίζει το ζυγισμένο άθροισμά τους και εφαρμόζει μία συνάρτηση ενεργοποίησης για την παραγωγή της εξόδου[20].

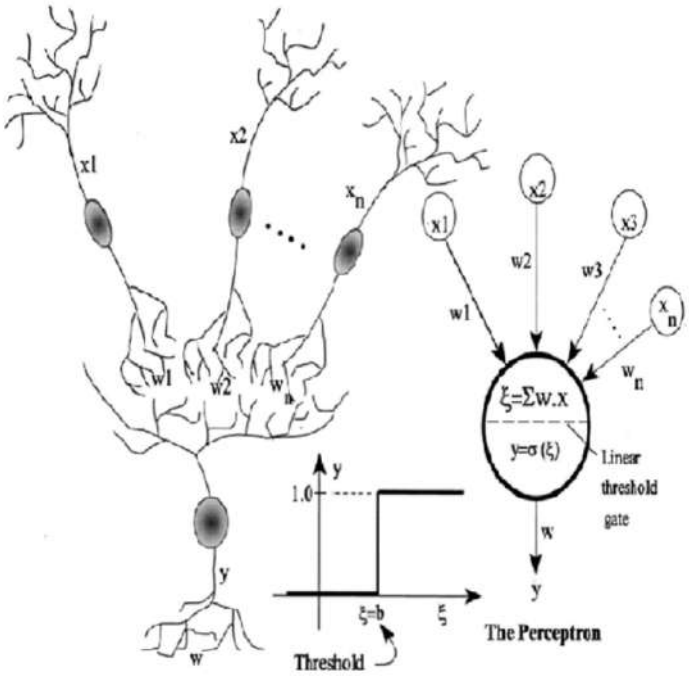
Η λειτουργία του περιγράφεται μαθηματικά ως:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.1}$$

όπου:

- x_i είναι οι είσοδοι του νευρώνα.
- w_i είναι τα βάρη σύνδεσης (weights) που καθορίζουν τη σημασία κάθε εισόδου.
- b είναι η προκατάληψη (bias), η οποία επιτρέπει στο μοντέλο να μάθει ακόμα και σε περιπτώσεις όπου όλες οι είσοδοι είναι μηδενικές μετατοπίζοντας το όριο ενεργοποίησης.

- $f(\cdot)$ είναι η συνάρτηση ενεργοποίησης που καθορίζει αν και πόσο ισχυρά ο νευρώνας θα ενεργοποιηθεί.



Εικόνα 5: Βιολογικός και Τεχνητός Νευρώνας [21]

Το μοντέλο McCulloch–Pitts μπορεί να θεωρηθεί ως ένας γραμμικός ταξινομητής, καθώς διαχωρίζει τις εισόδους βάσει γραμμικών ορίων. Ωστόσο, η απλότητά του περιορίζει τη χρήση του σε προβλήματα που περιλαμβάνουν μη-γραμμικές σχέσεις, γεγονός που οδήγησε στη μετέπειτα εξέλιξη πιο ισχυρών και πολύπλοκων μοντέλων τεχνητών νευρώνων.

2.3.2 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks – ANNs) αποτελούν έναν από τους πλέον ισχυρούς μηχανισμούς μηχανικής μάθησης, με δυνατότητα προσέγγισης πολύπλοκων, μη-γραμμικών συσχετίσεων μεταξύ εισόδων και εξόδων. Σε αντίθεση με τους παραδοσιακούς αλγορίθμους, τα ANNs δεν απαιτούν ρητούς κανόνες ή χειροκίνητα σχεδιασμένες συναρτήσεις και καθορισμένα χαρακτηριστικά, αλλά μαθαίνουν μέσω προσαρμογής των εσωτερικών τους παραμέτρων με βάση τα δεδομένα εισόδου. Αυτή η προσαρμοστικότητα τα καθιστά ιδιαίτερα αποτελεσματικά σε εφαρμογές όπως η αναγνώριση οπτικών προτύπων, η ανάλυση φυσικής γλώσσας και η πρόβλεψη χρονοσειρών [19][22].

Το βασικό πλεονέκτημα των νευρωνικών δικτύων είναι η ικανότητά τους να εξάγουν αυτόματα χρήσιμες αναπαραστάσεις από τα δεδομένα, χωρίς την ανάγκη χειροκίνητης προεπεξεργασίας. Μέσω ιεραρχικής μάθησης, κάθε επίπεδο του δικτύου επεξεργάζεται τις εισόδους σε διαφορετικό επίπεδο αφαιρετικότητας [23].

Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από διαδοχικά επίπεδα, καθένα από τα οποία περιλαμβάνει νευρώνες που εκτελούν υπολογισμούς βασισμένους σε σταθμισμένα αθροίσματα εισόδων και εφαρμογή μη-γραμμικής συνάρτησης ενεργοποίησης. Το δίκτυο μπορεί να θεωρηθεί ως μια

συνάρτηση που αντιστοιχίζει ένα διάνυσμα εισόδου $\mathbf{x}$ σε ένα διάνυσμα εξόδου y μέσω μιας σειράς μετασχηματισμών:

$$y = f_\theta(\mathbf{x}) \tag{2.2}$$

όπου f_θ είναι η σύνθετη συνάρτηση που υλοποιείται από το δίκτυο και θ το σύνολο των παραμέτρων του [19].

Τα νευρωνικά δίκτυα είναι δομημένα ιεραρχικά σε τρία βασικά επίπεδα με το καθένα να λαβάνει διαφορετικό ρόλο στην επεξεργασία δεδομένων:

- Επίπεδο εισόδου (input layer): Λαμβάνει τα αρχικά δεδομένα και τα μεταφέρει στο υπόλοιπο δίτυο, επομένως δεν εκτελούν κανέναν υπολογισμό και απλώς μεταφέρουν πληροφορία στα κρυφά επίπεδα. Σε πολυδιάστατα δεδομένα (π.χ. εικόνες, χρονοσειρές), κάθε νευρώνας εισόδου αντιστοιχεί σε μία διάσταση του διανύσματος εισόδου [23].
- Κρυφά επίπεδα (hidden layers): Πραγματοποιούν τη βασική επεξεργασία. Καθένα από τα επίπεδα τους αποτελείται από νευρώνες που λαμβάνουν εισόδους και υπολογίζουν νέες αναπαραστάσεις. Αν ένα επίπεδο έχει n νευρώνες και λαμβάνει m εισόδους, η έξοδός του δίνεται από τη σχέση:
$$H = \Phi(WX + b) \tag{2.3}$$
όπου:
  - X είναι το διάνυσμα εισόδου,
  - W είναι ένας πίνακας διαστάσεων $n \times m$ που περιέχει τα βάρη των συνδέσεων,
  - b είναι ένα διάνυσμα προκαταλήψεων μεγέθους n,
  - H είναι η έξοδος του επιπέδου,
  - $\Phi(\cdot)$ είναι μια μη-γραμμική συνάρτηση ενεργοποίησης [19].

  Τα κρυφά επίπεδα έχουν την ιδιότητα να δημιουργούν σταδιακά αφηρημένες αναπαραστάσεις των δεδομένων. Στην περίπτωση των εικόνων, τα πρώτα επίπεδα ανιχνεύουν απλά χαρακτηριστικά (όπως άκρα και γωνίες), ενώ τα βαθύτερα επίπεδα ανιχνεύουν πολύπλοκα σχήματα και αντικείμενα.
- Επίπεδο εξόδου (output layer): Παράγει την τελική έξοδο του δικτύου. Η μορφή του εξαρτάται από το είδος του προβλήματος (π.χ. ταξινόμηση, παλινδρόμηση).

Η επεξεργασία των δεδομένων μέσα στο δίκτυο ακολουθεί μια διαδοχική ροή, η οποία ονομάζεται *forward pass*. Κατά το *forward pass*, τα δεδομένα εισόδου μεταβιβάζονται από επίπεδο σε επίπεδο, μέχρι να παραχθεί η τελική έξοδος. Σε κάθε στάδιο, εκτελούνται πολλαπλασιασμοί πινάκων και εφαρμογές μη-γραμμικών συναρτήσεων.

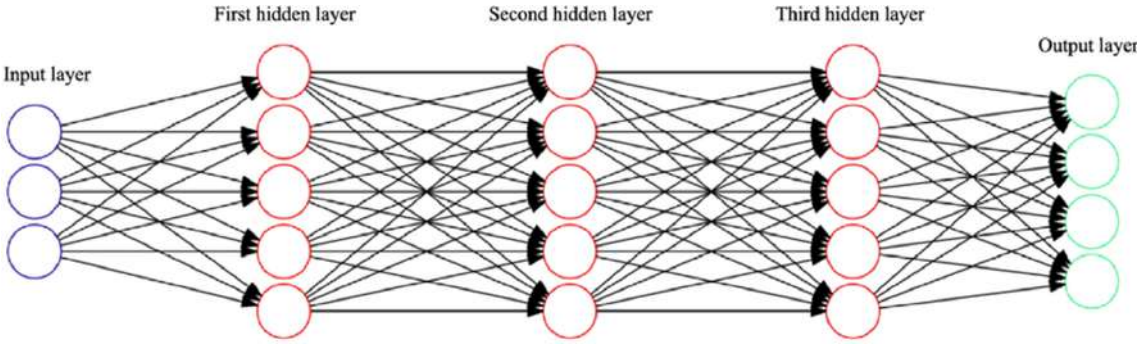
Αν συνοψίσουμε τη μαθηματική διαδικασία για ένα δίκτυο με L επίπεδα, η συνολική έξοδος προκύπτει από τη σχέση:

$$y = \Phi_L\left(W_L \Phi_{L-1}\left(W_{L-1} \cdots \Phi_1(W_1 x + b_1) \cdots + b_{L-1}\right) + b_L\right) \tag{2.4}$$

Η παραπάνω διαδικασία καθιστά το δίκτυο ένα επαναλαμβανόμενο σύνολο γραμμικών μετασχηματισμών και μη-γραμμικών συναρτήσεων ενεργοποίησης, επιτρέποντας τη μάθηση πολύπλοκων συναρτησιακών σχέσεων [19].

2.3.2.1 Πλήρως Συνδεδεμένα Νευρωνικά Δίκτυα

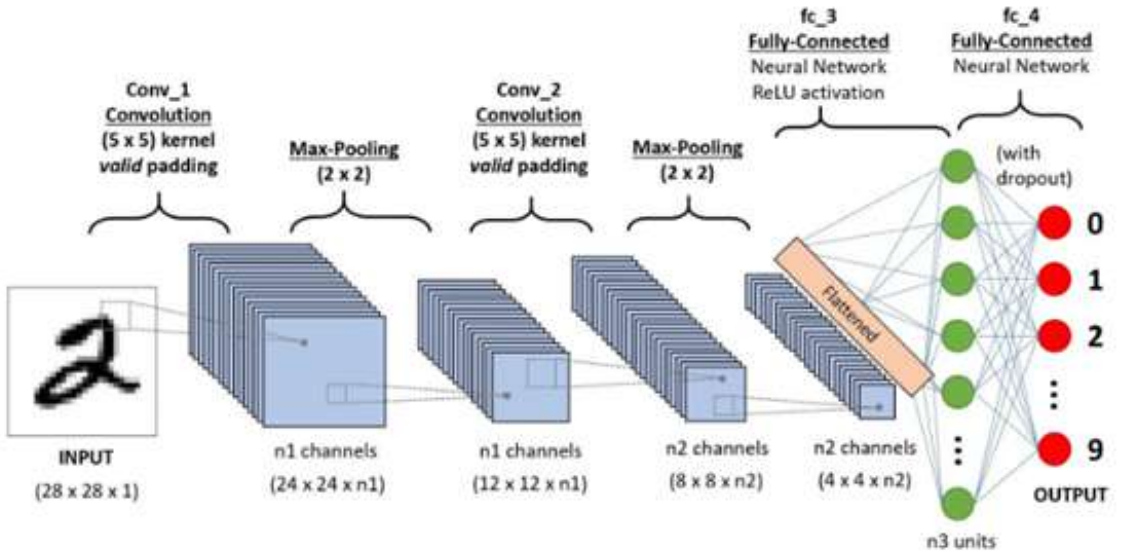
Τα Πλήρως Συνδεδεμένα Νευρωνικά Δίκτυα (Fully Connected Networks – FCNs) αποτελούν την απλούστερη μορφή ANN, στην οποία κάθε νευρώνας ενός επιπέδου συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου. Αυτό επιτρέπει στο δίκτυο να μαθαίνει πολύπλοκες σχέσεις, όμως παράλληλα δημιουργεί και πρόβλημα αποδοτικότητας, καθώς ο αριθμός των παραμέτρων αυξάνεται εκθετικά με τον αριθμό των νευρώνων.



Εικόνα 6: Fully Connected Network [24]

2.3.2.2 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks – CNNs) χρησιμοποιούνται ευρέως στην επεξεργασία εικόνων και θα αναλυθούν διεξοδικότερα στην επόμενη ενότητα. Σε αντίθεση με τα FCNs, τα CNNs χρησιμοποιούν συνελικτικά φίλτρα που μπορούν να ανιχνεύουν άκρα, υφές και σχήματα. Αυτό μειώνει δραστικά τον αριθμό των παραμέτρων και βελτιώνει την απόδοση στην αναγνώριση προτύπων [23]

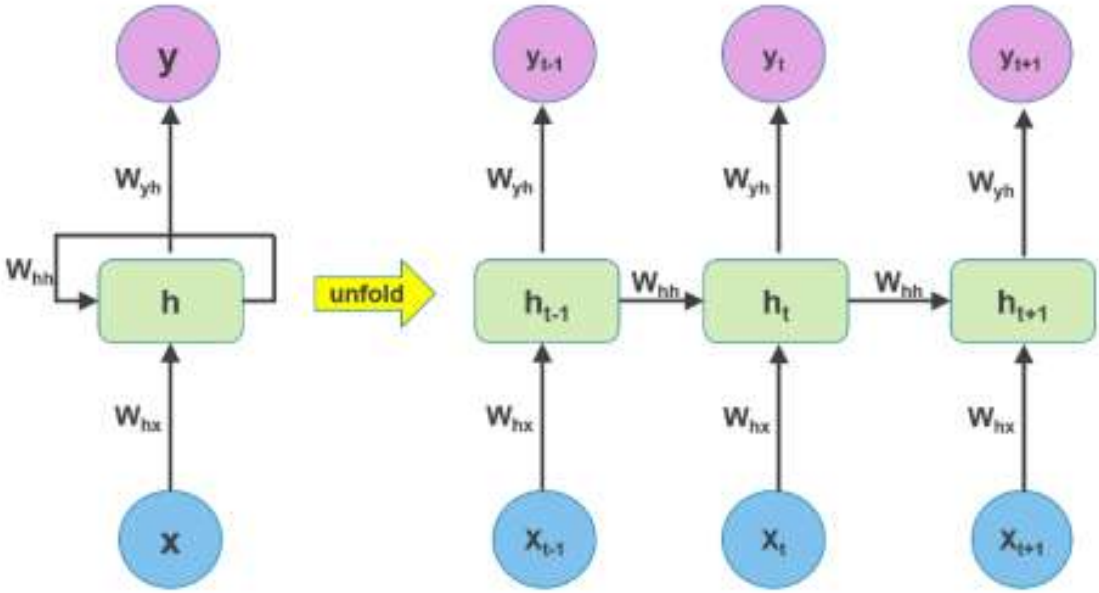


Εικόνα 7: Convolution Neural Network [23]

2.3.2.3 Αναδρομικά Νευρωνικά Δίκτυα

Τα Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks – RNNs) εξειδικεύονται στην επεξεργασία σειριακών δεδομένων, όπως κείμενα και ηχητικά σήματα. Σε αντίθεση με τα απλά

ANNs τα RNNs έχουν εσωτερική μνήμη, επιτρέποντας στο δίκτυο να διατηρεί πληροφορίες από προηγούμενα χρονικά βήματα. Αυτή η δυνατότητα τα καθιστά ιδανικά για εφαρμογές που βασίζονται σε μετάφραση φυσικής γλώσσας και αναγνώριση ομιλίας [25]



Εικόνα 8: Recurrent Neural Network [26]

2.4 Συνελικτικά Νευρωνικά Δίκτυα (CNNs)

2.4.1 Βασικές Αρχές

Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks – CNNs) αποτελούν μια εξειδικευμένη κατηγορία νευρωνικών δικτύων σχεδιασμένη για την ανάλυση δεδομένων με χωρική ή εντοπισμένη δομή, όπως εικόνες και βίντεο. Η αρχιτεκτονική τους είναι εμπνευσμένη από τη λειτουργία του ανθρώπινου οπτικού φλοιού στον εγκέφαλο, όπου εξειδικευμένοι νευρώνες ενεργοποιούνται ως απόκριση σε συγκεκριμένα οπτικά ερεθίσματα [27][28].

Σε αντίθεση με παραδοσιακές μεθόδους εξαγωγής χαρακτηριστικών, όπως οι SIFT και HOG, τα CNNs μπορούν να εντοπίζουν αυτόματα τα σημαντικότερα χαρακτηριστικά μέσα από την ίδια τη διαδικασία εκπαίδευσης [19]. Η δυνατότητα αυτή οφείλεται στην ικανότητά τους να μαθαίνουν ιεραρχικές αναπαραστάσεις των δεδομένων, από απλές γεωμετρικές δομές έως σύνθετα αντικείμενα.

Η αρχιτεκτονική των CNNs βασίζεται σε τρεις αρχές μετατρέποντάς τα ισχυρά για την ανάλυση εικόνων αλλά και χωρικών δεδομένων:

- Συνέλιξη (convolution)
- Ομαδοποίηση (pooling)
- Μη-γραμμικότητα (non-linearity)

Αυτές οι αρχές βοηθούν στη μείωση των παραμέτρων των μοντέλων που δημιουργούνται, την αποφυγή του overfitting και την αποτελεσματικότερη εξαγωγή χαρακτηριστικών.

Ένα βασικό χαρακτηριστικό τους είναι η ιεραρχική μάθηση των επιπέδων. Τα αρχικά ανιχνεύουν κυρίως βασικά μοτίβα, όπως άκρες και γωνίες, ενώ στα βαθύτερα επίπεδα του δικτύου συνδυάζονται αυτές οι πληροφορίες για την αναγνώριση πιο σύνθετων χαρακτηριστικών όπως σχήματα, επιφάνειες και αντικείμενα. Αυτή η ιεραρχική δομή είναι παρόμοια με τον τρόπο που το ανθρώπινο οπτικό σύστημα αναγνωρίζει εικόνες, ξεκινώντας από απλές δομές και σταδιακά συνθέτοντας μια πλήρη αντίληψη του περιβάλλοντος.[28][29]

Τέλος, ένα ακόμα κρίσιμο χαρακτηριστικό είναι ότι χρησιμοποιούν παραμέτρους που μοιράζονται (shared weights). Αυτό σημαίνει ότι το ίδιο φίλτρο μπορεί να χρησιμοποιείται σε διαφορετικές περιοχές των εικόνων μειώνονται έτσι τον αριθμό των παραμέτρων που απαιτούνται για να εκπαιδευτεί σωστά το CNN, κάνοντας το έτσι και υπολογιστικά πιο αποδοτικό. [28]

2.4.2 Βασική Αρχιτεκτονική

Η τυπική αρχιτεκτονική ενός CNN αποτελείται από τα εξής βασικά τμήματα:

- Επίπεδα Εξαγωγής Χαρακτηριστικών (Feature Extraction Layers): Περιλαμβάνουν τα συνελικτικά επίπεδα και τα επίπεδα ομαδοποίησης. Τα συνελικτικά φίλτρα εξάγουν χαρακτηριστικά από την είσοδο, ενώ τα επίπεδα ομαδοποίησης μειώνουν τη διαστασιμότητα, διατηρώντας τις πιο αντιπροσωπευτικές πληροφορίες.
- Πλήρως Συνδεδεμένα Επίπεδα (Fully Connected Layers): Επεξεργάζονται τα εξαγόμενα χαρακτηριστικά και τα μετασχηματίζουν σε ένα τελικό διάνυσμα εξόδου, κατάλληλο για προβλήματα ταξινόμησης ή παλινδρόμησης.
- Έξοδος (Output Layer): περιλαμβάνει μια συνάρτηση ενεργοποίησης για να μπορέσει να γίνει η τελική ταξινόμηση με βάση τις πληροφορίες εκπαίδευσης από τα προηγούμενα επίπεδα του δικτύου, μετατρέποντας τις εξόδους σε πιθανότητες για κάθε κατηγορία.

Η γενική ροή δεδομένων σε ένα CNN είναι η εξής:

Εικόνα Εισόδου → Συνέλιξη → Μη-γραμμικότητα → Ομαδοποίηση →

Πλήρως Συνδεδεμένα Επίπεδα → Έξοδος

Η αρχιτεκτονική αυτή επιτρέπει στα CNNs να επιτυγχάνουν υψηλή ακρίβεια σε προβλήματα υπολογιστικής όρασης, ακόμα και με περιορισμένα δεδομένα, χάρη στην εκμετάλλευση της εντοπισμένης δομής των εικόνων και στη χρήση κοινών παραμέτρων [28][30][31]

2.4.3 Συνέλιξη (Convolution)

Η συνέλιξη αποτελεί τη βασική υπολογιστική μονάδα των Συνελικτικών Νευρωνικών Δικτύων (CNNs), καθώς επιτρέπει την τοπική εξαγωγή χαρακτηριστικών από τα δεδομένα εισόδου μέσω της εφαρμογής συνελικτικών φίλτρων. Πρόκειται για μια πράξη φιλτραρίσματος κατά την οποία ένας μικρός πίνακας βαρών (kernel ή φίλτρο) σαρώνει την εικόνα, αναζητώντας τοπικά μοτίβα, αναλόγως με το επίπεδο του δικτύου στο οποίο χρησιμοποιείται [19].

Μαθηματικά, η πράξη της συνέλιξης μεταξύ μιας δισδιάστατης x και ενός πυρήνα k μπορεί να περιγραφεί ως:

$$y(i,j) = (x \star k)(i,j) = \sum_m \sum_n x(i+m, j+n) \cdot k(m,n) \tag{2.5}$$

Όπου:

- $x(i, j)$ είναι η τιμή της εισόδου στη θέση (i, j),
- $k(i, j)$ είναι η τιμή του πυρήνα στη θέση (i, j),
- $y(i, j)$ είναι η τιμή του παραγόμενου χάρτη χαρακτηριστικών (feature map) στη θέση (i, j).

Κατά την πράξη, ο πυρήνας μετακινείται (με ορισμένο βήμα) σε ολόκληρη την επιφάνεια της εικόνας, και για κάθε περιοχή εφαρμόζεται το στοιχειακό γινόμενο μεταξύ των στοιχείων του πυρήνα και των αντίστοιχων pixel. Το αποτέλεσμα αυτής της διαδικασίας είναι ένας νέος χάρτης χαρακτηριστικών, στον οποίο έχουν επισημανθεί οι περιοχές όπου το εκάστοτε φίλτρο ανιχνεύει την παρουσία συγκεκριμένων μοτίβων.

Φίλτρα

Τα φίλτρα (ή πυρήνες) είναι μικροί πίνακες (συνήθως διαστάσεων 3×3 ή 5×5) των οποίων οι τιμές μαθαίνονται κατά την εκπαίδευση του δικτύου. Σκοπός τους είναι να εντοπίζουν συγκεκριμένα τοπικά χαρακτηριστικά. Για παράδειγμα, τα φίλτρα Sobel χρησιμοποιούνται κλασικά για την ανίχνευση ακμών:

$$k_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad k_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.6}$$

όπου το φίλτρο k_x είναι για την ανίχνευση οριζοντίων και το k_y για την ανίχνευση κάθετων ακμών.

Παράμετροι Συνέλιξης

Η διαδικασία της συνέλιξης καθορίζεται από συγκεκριμένες παραμέτρους που επηρεάζουν τη συμπεριφορά και την απόδοση του επιπέδου:

- **Μέγεθος πυρήνα (kernel size):** καθορίζει το πεδίο αντίληψης του φίλτρου, το εύρο της περιοχής που λαμβάνει υπόψη η κάθε πράξη συνέλιξης. Μικρότεροι πυρήνες είναι πιο αποδοτικοί και επιτρέπουν βαθύτερες αρχιτεκτονικές με περισσότερη μη γραμμικότητα [32].
- **Το βήμα (stride):** Καθορίζει πόσα pixel μετακινείται ο πυρήνας κάθε φορά (π.χ. βήμα 1 σημαίνει ότι ο πυρήνας μετακινείται 1 pixel για κάθε συνέλιξη). Μεγαλύτερο βήμα μειώνει τις διαστάσεις του χάρτη εξόδου αλλά και την υπολογιστική επιβάρυνση, εις βάρος όμως της ακρίβειας.
- **Συμπλήρωση (padding):** Αναφέρεται στην προσθήκη τεχνητών pixel (συνήθως μηδενικών) γύρω από την είσοδο ώστε να διατηρηθούν οι διαστάσεις της εξόδου. Η συμπλήρωση ίδιας διάστασης (same padding) διατηρεί τις διαστάσεις τις εξόδου ίδες με την είσοδο, ενώ η έγκυρη συμπλήρωση (valid padding) αποφεύγει την προσθήκη νέων στοιχείων οδηγεί σε μικρότερη έξοδο, οδηγώντας σε έξοδο μικρότερης διάστασης [33][34].
- **Αριθμός φίλτρων:** Κάθε συνελικτικό επίπεδο χρησιμοποιεί πολλαπλά φίλτρα, καθένα από τα οποία ανιχνεύει διαφορετικά χαρακτηριστικά (π.χ. το Sobel που έχει 2 για οριζόντιες και κατακόρυφες ακμές). Το πλήθος αυτών των φίλτρων καθορίζει και το βάθος (depth) του παραγόμενου χάρτη χαρακτηριστικών.

Η πράξη της συνέλιξης είναι θεμελιώδης για την επιτυχία των CNNs, καθώς επιτρέπει την αναπαράσταση των δεδομένων με τρόπο που διατηρεί την τοπική συσχέτιση και εντοπίζει ουσιώδη μοτίβα ανεξάρτητα από τη θέση τους στην εικόνα.

2.4.4 Ομαδοποίηση

Η ομαδοποίηση (*pooling*) αποτελεί μια τεχνική μείωσης των χωρικών διαστάσεων των χαρτών χαρακτηριστικών (feature maps), συμβάλλοντας στη μείωση των παραμέτρων και της υπολογιστικής πολυπλοκότητας των συνελικτικών νευρωνικών δικτύων. Η βασική της λειτουργία έγκειται στην εφαρμογή μιας μη -γραμμικής συνάρτησης εντός ενός τοπικού παραθύρου εισόδου, όπως η μέγιστη ή μέση τιμή, με στόχο τη συμπύκνωση των πληροφοριών.

Η πιο συνηθισμένη τεχνική είναι η μέγιστη ομαδοποίηση (Max Pooling), κατά την οποία διατηρείται η μέγιστη τιμή εντός κάθε παραθύρου R του χάρτη χαρακτηριστικών. Με αυτόν τον τρόπο, το δίκτυο επικεντρώνεται στα εντονότερα και πιο σημαντικά σήματα του χώρου, αποκτώντας παράλληλα ανθεκτικότητα σε μικρές μετατοπίσεις ή παραμορφώσεις της εισόδου [2]:

$$y(i,j) = \max_{(m,n)\in R} x(m,n) \tag{2.7}$$

Εναλλακτικά, η μέση ομαδοποίηση (Average Pooling) υπολογίζει τη μέση τιμή των τιμών εντός κάθε παραθύρου R, διατηρώντας περισσότερη πληροφορία για τη συνολική ένταση των περιοχών. Συχνά χρησιμοποιείται σε περιβάλλοντα με υψηλό θόρυβο ή όταν δεν απαιτείται ενίσχυση των ακμών [19]:

$$y(i,j) = \frac{1}{|R|} \sum_{(m,n)\in R} x(m,n) \tag{2.8}$$

Επιπλέον, η παγκόσμια μέση ομαδοποίηση (Global Average Pooling) —όπως προτάθηκε στο Network in Network [35]— εφαρμόζεται σε ολόκληρο τον χάρτη χαρακτηριστικών, παράγοντας έναν μοναδικό μέσο όρο ανά κανάλι. Αυτή η τεχνική εξαλείφει την ανάγκη για πλήρως συνδεδεμένα επίπεδα πριν από την έξοδο, μειώνοντας περαιτέρω τον αριθμό των παραμέτρων και περιορίζοντας τον υπερεκπαίδευση (overfitting).

Η ομαδοποίηση συνήθως εφαρμόζεται με παράθυρα μεγέθους 2×2 και βήμα ίσο με το μέγεθος του παραθύρου, ώστε να αποφεύγεται η επικάλυψη μεταξύ των περιοχών. Αυτό έχει ως αποτέλεσμα τη μείωση των διαστάσεων στο μισό, επιτρέποντας στο CNN να επικεντρωθεί στις ουσιώδεις πληροφορίες. Σε σύγκριση με τη συνέλιξη, η ομαδοποίηση δεν έχει εκπαιδεύσιμες παραμέτρους, γεγονός που συμβάλλει στην αποδοτικότητα και την απλότητα της διαδικασίας.

2.4.5 Συνάρτηση Απώλειας (Loss Function)

Η συνάρτηση απώλειας (loss function) αποτελεί κρίσιμο στοιχείο στην εκπαίδευση αλγορίθμων μηχανικής και βαθιάς μάθησης, επιτρέποντας την ποσοτικοποίηση της απόκλισης μεταξύ της πρόβλεψης του μοντέλου και της πραγματικής τιμής. Η διαδικασία εκπαίδευσης βασίζεται στην ελαχιστοποίηση αυτής της συνάρτησης μέσω τεχνικών βελτιστοποίησης, όπως η gradient descent, με στόχο τη βελτίωση της γενίκευσης του μοντέλου [19].

Η επιλογή της κατάλληλης συνάρτησης εξαρτάται άμεσα από τη φύση του προβλήματος, όπως παλινδρόμηση, ταξινόμηση ή metric learning. Στη συνέχεια παρουσιάζονται ορισμένες από τις πιο διαδεδομένες συναρτήσεις απώλειας.

2.4.5.1 Mean Squared Error (MSE)

Η MSE χρησιμοποιείται κυρίως σε προβλήματα παλινδρόμησης [1] και ορίζεται ως:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.9}$$

όπου y_i είναι η πραγματική τιμή και $\hat{y}_i$ η πρόβλεψη του μοντέλου για το δείγμα i. Η MSE είναι ευαίσθητη σε μεγάλες αποκλίσεις (outliers) λόγω της τετραγωνικής μορφής της.

2.4.5.2 Cross-Entropy Loss

Η cross-entropy χρησιμοποιείται σε προβλήματα ταξινόμησης και ορίζεται ως:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{2.10}$$

όπου C το πλήθος των κατηγοριών, y_i η πραγματική πιθανότητα (συνήθως one-hot) και $\hat{y}_i$ η προβλεπόμενη πιθανότητα.

2.4.5.3 Focal Loss

Η Focal Loss [36] σχεδιάστηκε για να αντιμετωπίζει προβλήματα με ανισορροπία κλάσεων. Ορίζεται ως:

$$\mathcal{L}_{Focal} = -\alpha_t(1 - \hat{p}_t)^{\gamma} \log(\hat{p}_t) \tag{2.11}$$

όπου:

- $\hat{p}_t$ η προβλεπόμενη πιθανότητα για τη σωστή κατηγορία,
- α_t συντελεστής εξισορρόπησης,
- $\gamma > 0$ το *focusing parameter* που ελέγχει την έμφαση στα δύσκολα δείγματα.

2.4.5.4 Contrastive Loss

Η Contrastive Loss χρησιμοποιείται κυρίως σε Siamese Networks και γενικά σε metric learning[37]:

$$\mathcal{L}_{Contrastive} = \frac{1}{2}yD^2 + \frac{1}{2}(1 - y)\left[\max(0, m - D)\right]^2 \tag{2.12}$$

όπου D είναι η απόσταση μεταξύ δύο δειγμάτων, $y \in \{0, 1\}$ δείχνει αν τα δείγματα ανήκουν στην ίδια κατηγορία και m είναι ένα margin υπερπαράμετρος.

2.4.5.5 Triplet Loss

Η Triplet Loss [38] επεκτείνει τη λογική της Contrastive Loss χρησιμοποιώντας τριάδες (anchor, positive, negative):

$$\mathcal{L}_{Triplet} = \max\left\{0, D(a, p) - D(a, n) + m\right\} \tag{2.13}$$

όπου:

- $D(a, p)$ είναι η απόσταση anchor-positive,
- $D(a, n)$ η απόσταση anchor-negative,
- m είναι το margin.

Η Triplet Loss στοχεύει να επιβάλει το anchor να βρίσκεται πιο κοντά στο θετικό δείγμα απ' ότι στο αρνητικό κατά τουλάχιστον m μονάδες.

2.4.5.6 CosFace Loss

Η CosFace Loss [39] αποτελεί μια παραλλαγή της Cross-Entropy, κατάλληλη για προβλήματα fine-grained visual classification και metric learning. Εισάγει ένα angular margin στο embedding space, ενισχύοντας την διάκριση ανάμεσα σε κατηγορίες μέσω της γωνιακής απόστασης (angular distance) και όχι της ευκλείδειας.

Ορίζεται ως:

$$\mathcal{L}_{CosFace} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i})-m)}}{e^{s(\cos(\theta_{y_i})-m)}+\sum_{j\neq y_i}e^{s\cos(\theta_j)}} \tag{2.14}$$

όπου:

- s είναι το scaling factor,
- m είναι το margin,
- θ_i είναι η γωνία μεταξύ του i-οστού embedding και του κέντρου της σωστής κατηγορίας.

Η CosFace Loss επιτυγχάνει καλύτερο intra-class compactness και inter-class separability σε σχέση με την απλή Cross-Entropy, καθιστώντας την ιδιαίτερα αποτελεσματική σε FGVC προβλήματα.

2.4.5.7 Σύγκριση Συνήθων Συναρτήσεων Απώλειας

Συνάρτηση	**Εφαρμογή**	**Πλεονεκτήματα**	**Μειονεκτήματα**
MSE	Παλινδρόμηση	Απλή, εύκολη στη σύγκλιση	Ευαίσθητη σε outliers
Cross-Entropy	Κατηγοριοποίηση	Καλή απόδοση σε ισορροπημένα δεδομένα	Δεν διαχειρίζεται καλά την ανισορροπία
Focal Loss	Ανισορροπημένα σύνολα	Εστιάζει σε δύσκολα δείγματα	Απαιτεί προσεκτικό ορισμό παραμέτρων (γ)
Contrastive Loss	Metric Learning	Μαθαίνει ποιοτικές αποστάσεις μεταξύ embeddings	Ευαίσθητη στην επιλογή των ζευγών
Triplet Loss	Metric Learning	Αποτελεσματική για embedding learning	Απαιτεί επιλεγμένα triplets για καλή απόδοση
CosFace Loss	Fine-Grained Visual Classification	Βελτιώνει angular separability και αντιμετωπίζει class imbalance	Εξαρτάται από την επιλογή των παραμέτρων m, s

Πίνακας 1: Σύγκριση Συνήθων Συναρτήσεων Απώλειας

Η επιλογή της κατάλληλης συνάρτησης απώλειας εξαρτάται άμεσα από τη φύση και τις ιδιαιτερότητες του εκάστοτε προβλήματος. Οι συναρτήσεις που παρουσιάστηκαν έχουν διαφορετικά χαρακτηριστικά, πλεονεκτήματα και μειονεκτήματα, όπως συνοψίζεται στον Πίνακα 1.

Για προβλήματα παλινδρόμησης, η MSE αποτελεί την κλασική επιλογή καθώς εστιάζει στην ελαχιστοποίηση των αποστάσεων μεταξύ προβλεπόμενων και πραγματικών τιμών. Ωστόσο, η ευαισθησία της σε outliers μπορεί να οδηγήσει σε υποβάθμιση της γενίκευσης όταν υπάρχουν ακραίες τιμές στα δεδομένα.

Στην κατηγοριοποίηση, η Cross-Entropy Loss παραμένει η επικρατέστερη επιλογή καθώς βελτιστοποιεί απευθείας την ομοιότητα μεταξύ των κατανομών των προβλεπόμενων και των πραγματικών πιθανοτήτων. Παρ' όλα αυτά, σε προβλήματα με έντονη ανισορροπία κατηγοριών (class imbalance), η Cross-Entropy τείνει να παραμελεί τις σπάνιες κατηγορίες. Η Focal Loss λύνει αυτό το πρόβλημα δίνοντας μεγαλύτερη έμφαση στα δύσκολα ή σπάνια δείγματα, ωστόσο εισάγει επιπλέον υπερπαραμέτρους όπως το γ που απαιτούν προσεκτική ρύθμιση.

Σε προβλήματα metric learning, όπου στόχος είναι η μάθηση ενός χώρου χαρακτηριστικών (embedding space), οι Contrastive και Triplet Losses είναι ευρέως χρησιμοποιούμενες. Η Contrastive Loss βασίζεται στη σχέση ζευγών (pairs) θετικών και αρνητικών παραδειγμάτων, ενώ η Triplet Loss χρησιμοποιεί τριάδες (anchor, positive, negative) και επιβάλλει το anchor να είναι εγγύτερο στο positive παρά στο negative. Η χρήση της Triplet Loss προσφέρει πιο ισχυρές ιδιότητες διαφοροποίησης στο embedding space, αλλά εξαρτάται έντονα από την ποιότητα της επιλογής των τριάδων (triplet mining).

Τέλος, η CosFace Loss έχει αναδειχθεί ως μια ιδιαίτερα αποτελεσματική επιλογή σε εφαρμογές που απαιτούν την εκμάθηση διακριτικών embeddings, όπως το fine-grained visual classification και το face recognition. Χάρη στο scaling factor s και το angular margin m, το δίκτυο ενισχύει την intra-class compactness και την inter-class separability, οδηγώντας σε πιο σταθερό και αποτελεσματικό embedding space. Στην πράξη, η CosFace Loss αποτελεί σήμερα μια από τις πιο αξιόπιστες λύσεις για προβλήματα υψηλής δυσκολίας, όπως αυτά που σχετίζονται με το FGVC.

Συνοψίζοντας, η επιλογή της κατάλληλης συνάρτησης απώλειας αποτελεί κρίσιμο βήμα στον σχεδιασμό αλγορίθμων, επηρεάζοντας την ταχύτητα σύγκλισης, την απόδοση και την ικανότητα γενίκευσης του μοντέλου.

2.4.6 Συναρτήσεις Ενεργοποίησης (Activation Functions)

Οι συναρτήσεις ενεργοποίησης αποτελούν κρίσιμο στοιχείο της λειτουργίας των τεχνητών νευρωνικών δικτύων, καθώς καθορίζουν τον τρόπο με τον οποίο ένας νευρώνας μετασχηματίζει την είσοδό του σε έξοδο. Κάθε τεχνητός νευρώνας εκτελεί έναν γραμμικό μετασχηματισμό των εισόδων του, σε συνδυασμό με τα βάρη και την προκατάληψη, και στη συνέχεια εφαρμόζει μια συνάρτηση ενεργοποίησης $\Phi(\cdot)$ που εισάγει μη-γραμμικότητα. Αυτή η μη-γραμμικότητα είναι απαραίτητη ώστε το δίκτυο να μπορεί να προσεγγίζει πολύπλοκες συναρτήσεις και να επιλύει προβλήματα με μη-γραμμικές συσχετίσεις με αποδοτικό τρόπο [19].

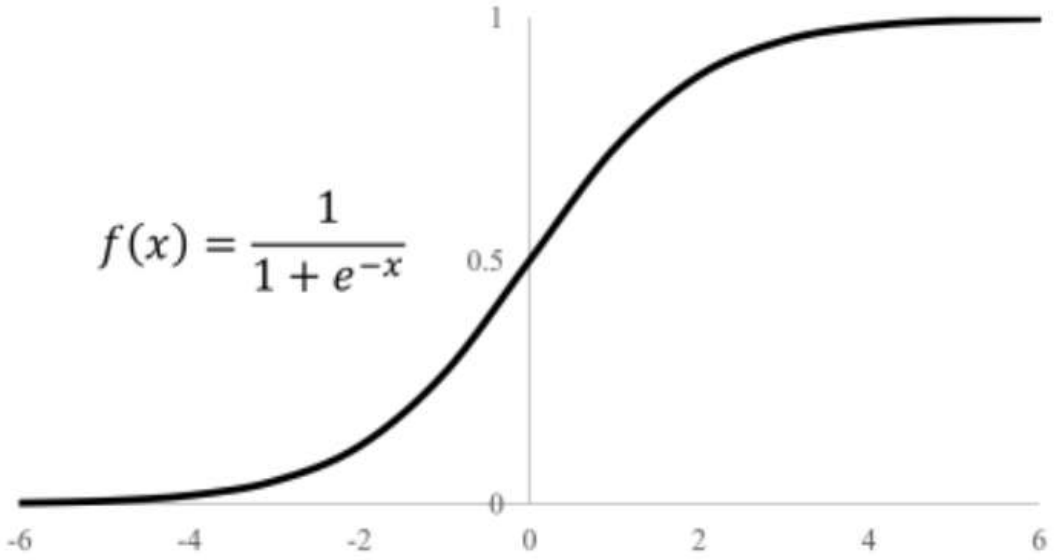
Αν δεν χρησιμοποιούνταν μη-γραμμικές συναρτήσεις ενεργοποίησης, το κάθε επίπεδο του δικτύου θα αποτελούσε έναν γραμμικό μετασχηματισμό, και τελικά το συνολικό δίκτυο θα ισοδυναμούσε με ένα μόνο γραμμικό επίπεδο – περιορίζοντας δραστικά την εκφραστική του δύναμη. Επομένως, εισάγουν μια ιδιότητα που επιτρέπει στο δίκτυο να αποδίδει σε πραγματικές συνθήκες, μαθαίνοντας αφηρημένες αναπαραστάσεις των δεδομένων.

2.4.6.1 Sigmoid

Μία από τις πρώτες και πιο διαδεδομένες συναρτήσεις ενεργοποίησης είναι η Sigmoid, η οποία χρησιμοποιούνταν ήδη από τα πρώτα πολυεπίπεδα νευρωνικά δίκτυα για προβλήματα δυαδικής ταξινόμησης, καθώς επέτρεπε την εφαρμογή του αλγορίθμου backpropagation [40]. Ορίζεται από τη σχέση:

$$\Phi(x) = \frac{1}{1 + e^{-x}} \tag{2.15}$$

Παράγει τιμές στο διάστημα $(0, 1)$, γεγονός που επιτρέπει την ερμηνεία της εξόδου ως πιθανότητα. Παρόλο που η Sigmoid ήταν ιδιαίτερα δημοφιλής, πλέον χρησιμοποιείται σπάνια στα κρυφά επίπεδα των δικτύων, καθώς παρουσιάζει το πρόβλημα της εξασθένισης του βαθμωτού κατηφόρου (vanishing gradient problem). Όταν οι είσοδοι ενός νευρώνα παίρνουν πολύ μεγάλες ή πολύ μικρές τιμές, η παράγωγος της Sigmoid πλησιάζει το μηδέν, με αποτέλεσμα το δίκτυο να μαθαίνει εξαιρετικά αργά [41].



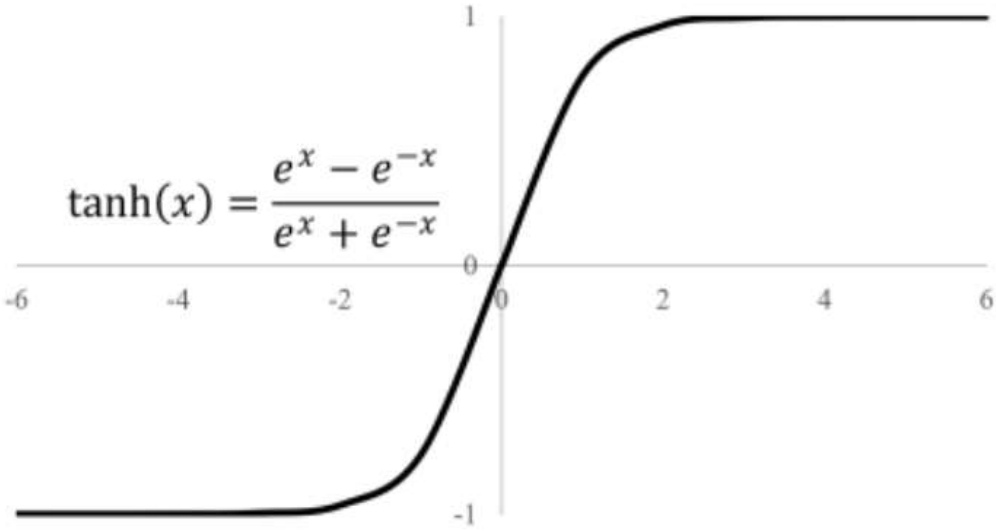
Εικόνα 9: Sigmoid Activation Function [42]

2.4.6.2 Hyperbolic Tangent

Μια βελτίωση της Sigmoid είναι η Hyperbolic Tangent (Tanh), η οποία δίνεται από τη σχέση:

$$\Phi(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.16}$$

και έχει τιμές στο διάστημα $(-1, 1)$. Η κύρια διαφορά της από τη Sigmoid είναι ότι η έξοδος είναι κεντραρισμένη στο μηδέν, γεγονός που επιτρέπει την καλύτερη διάδοση του σφάλματος κατά την εκπαίδευση. Ωστόσο, και η Tanh υποφέρει από το ίδιο πρόβλημα εξασθένισης του gradient, καθιστώντας τη λιγότερο αποτελεσματική σε βαθιά νευρωνικά δίκτυα [23].



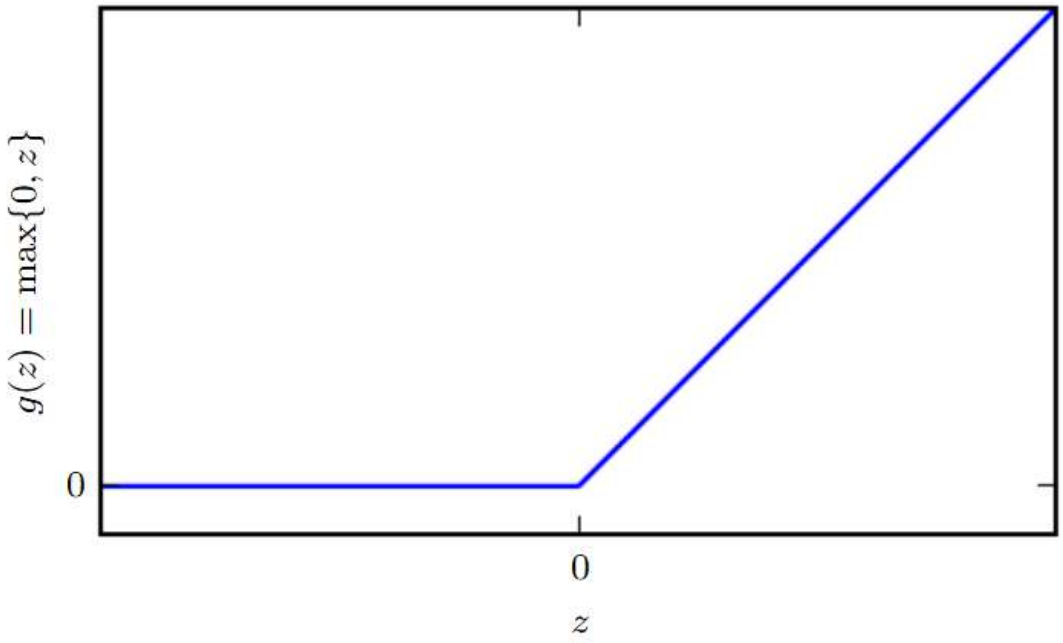
Εικόνα 10: Tanh Activation Function [42]

2.4.6.3 ReLu

Για την αντιμετώπιση αυτών των προβλημάτων, τα σύγχρονα νευρωνικά δίκτυα χρησιμοποιούν τη Rectified Linear Unit (ReLU), η οποία ορίζεται ως:

$$\Phi(x) = \max(0, x) \tag{2.17}$$

Η ReLU διαφέρει από τις προηγούμενες συναρτήσεις, καθώς δεν περιορίζει το εύρος των θετικών τιμών, επιτρέποντας στο δίκτυο να διατηρεί υψηλά gradients και να μαθαίνει γρήγορα. Ένα από τα βασικά πλεονεκτήματά της είναι η απλότητα του υπολογισμού της, αφού απαιτεί μόνο μια σύγκριση μεταξύ δύο αριθμών. Παρόλα αυτά, η ReLU παρουσιάζει το *dying ReLU problem*, δηλαδή το φαινόμενο κατά το οποίο ορισμένοι νευρώνες σταματούν να ενεργοποιούνται, όταν οι τιμές τους γίνονται μόνιμα αρνητικές. Αυτό σημαίνει ότι αυτοί οι νευρώνες δεν συμβάλλουν πλέον στη μάθηση, περιορίζοντας την αποδοτικότητα του δικτύου [43].



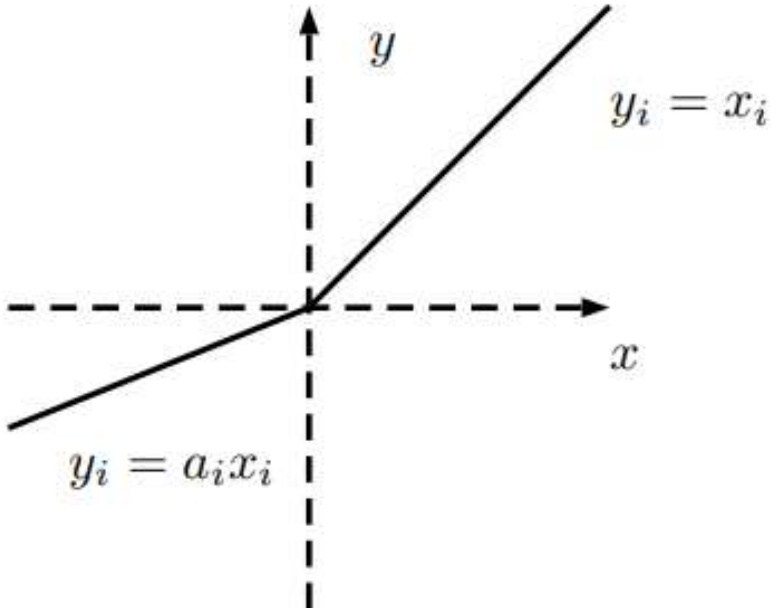
Εικόνα 11: ReLu Activation Function [19]

2.4.6.4 Leaky ReLu

Για να ξεπεραστεί το *dying ReLU problem*, χρησιμοποιείται μια παραλλαγή της, η Leaky ReLU, η οποία επιτρέπει σε αρνητικές τιμές να περάσουν, αλλά με έναν μικρό συντελεστή α:

$$\Phi(x) = \max(\alpha x, x), \text{ όπου } \alpha \approx 0.01 \tag{2.18}$$

Αυτή η προσαρμογή διασφαλίζει ότι κανένας νευρώνας δεν παραμένει μόνιμα απενεργοποιημένος, διατηρώντας τη ροή πληροφορίας σε όλο το δίκτυο. Παρόλο που η Leaky ReLU αποτελεί βελτίωση, η επιλογή της βέλτιστης τιμής για τον συντελεστή α απαιτεί προσοχή, καθώς διαφορετικές τιμές μπορεί να προκαλέσουν αστάθεια στην εκπαίδευση [44].



Εικόνα 12: Leaky ReLU Activation Function [44]

2.4.6.5 GELU

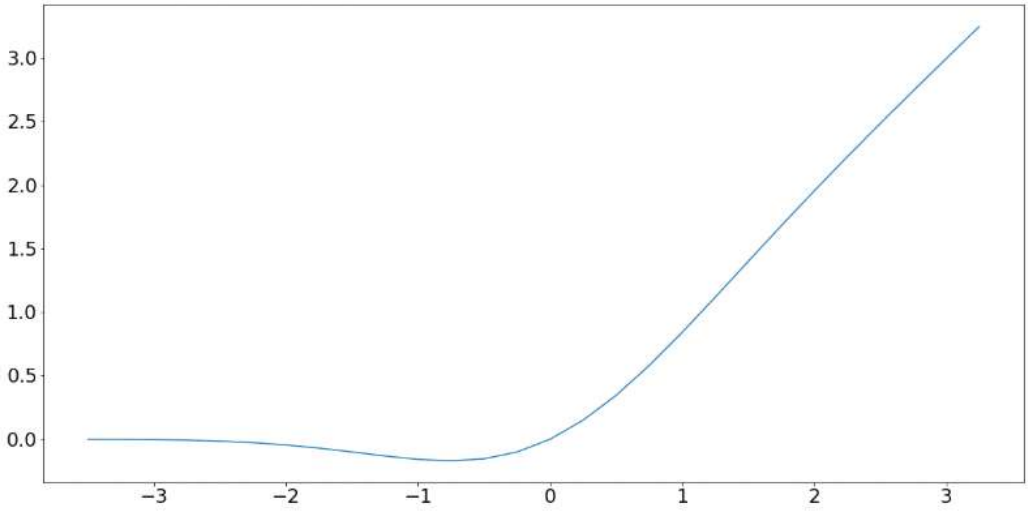
Η GELU (Gaussian Error Linear Unit) είναι μία σύγχρονη συνάρτηση ενεργοποίησης που συνδυάζει τις ιδιότητες της ReLU με στοχαστικά χαρακτηριστικά, βασισμένα στην κανονική κατανομή. Ορίζεται ως:

$$\Phi(x) = x \cdot \Phi_{\mathcal{N}}(x) \tag{2.19}$$

όπου $\Phi_{\mathcal{N}}(x)$ είναι η σωρευτική συνάρτηση κατανομής της κανονικής κατανομής. Στην πράξη χρησιμοποιείται συχνά η προσεγγιστική μορφή:

$$\Phi(x) \approx 0.5x \left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right]\right) \tag{2.20}$$

Η GELU επιτρέπει στις εισόδους να περνούν με ”μαλακό” τρόπο, αντί να κόβονται απότομα όπως στη ReLU. Αυτό έχει αποδειχθεί χρήσιμο σε μεγάλα δίκτυα καθώς οδηγεί σε ταχύτερη σύγκλιση και καλύτερη γενίκευση [45].



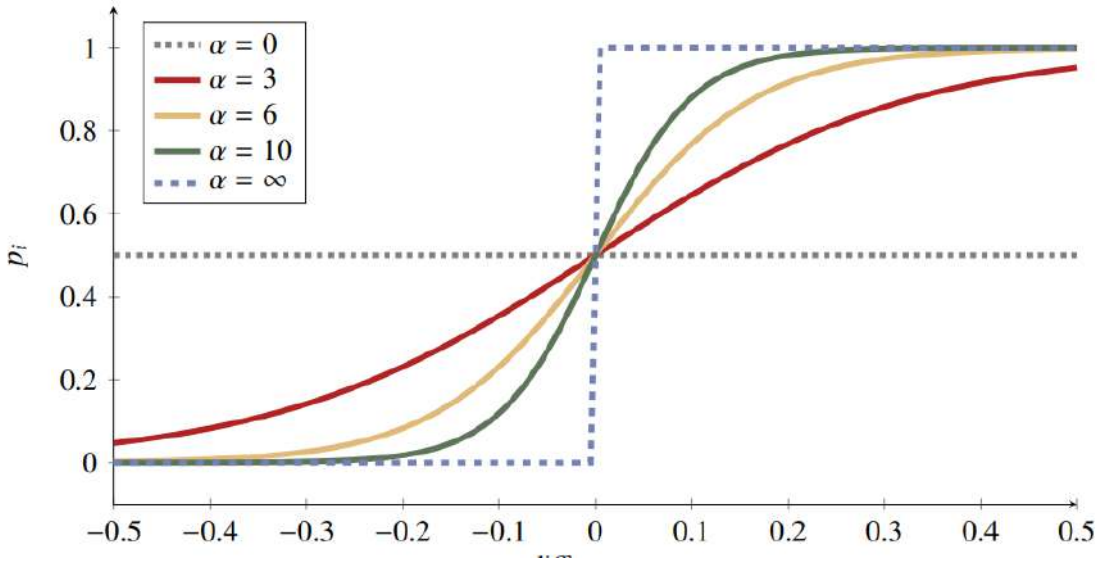
Εικόνα 13: GELU activation Function [45]

2.4.6.6 Softmax

Στα νευρωνικά δίκτυα που εκτελούν ταξινόμηση πολλαπλών κατηγοριών, η πιο συνηθισμένη συνάρτηση ενεργοποίησης στο τελικό επίπεδο είναι η Softmax. Η Softmax μετατρέπει ένα διάνυσμα πραγματικών αριθμών σε μια κατανομή πιθανοτήτων, εξασφαλίζοντας ότι το άθροισμα όλων των εξόδων είναι ίσο με 1. Δίνεται από τη σχέση:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.21}$$

και χρησιμοποιείται κυρίως σε ταξινομητές, όπου κάθε νευρώνας του επιπέδου εξόδου αντιστοιχεί σε μία κατηγορία. Παρότι η Softmax είναι ιδανική για την παραγωγή πιθανοτήτων, μπορεί να είναι ευαίσθητη σε μεγάλες τιμές εισόδου, καθώς η εκθετική φύση της μπορεί να οδηγήσει σε αριθμητική αστάθεια [44].



Εικόνα 14: Softmax Activation Function [46]

2.4.6.7 Mish

Η Mish είναι μία νεότερη συνάρτηση ενεργοποίησης, η οποία προτάθηκε με στόχο να συνδυάζει την απόδοση των προηγμένων μη γραμμικών συναρτήσεων με ομαλότητα στη μετάβαση και συνεχή παραγώγιση. Ορίζεται από τη σχέση:

$$\Phi(x) = x \cdot \tanh(\ln(1 + e^x)) \tag{2.22}$$

Η Mish επιτρέπει την είσοδο να περνάει με μη γραμμικό αλλά συνεχές τρόπο, χωρίς να την «κόβει» όπως η ReLU ή η Leaky ReLU. Παρουσιάζει εξαιρετικές ιδιότητες γενίκευσης και έχει αποδειχθεί αποτελεσματική σε εφαρμογές όπου απαιτείται λεπτομερής αναγνώριση μοτίβων. Η συνεχής φύση της και η ομαλή παραγώγισή της επιτρέπουν καλύτερη ροή πληροφορίας και σταθερότερη εκπαίδευση σε βαθιά δίκτυα.

Ωστόσο, η Mish είναι υπολογιστικά πιο απαιτητική λόγω των λογαριθμικών και υπερβολικών τριγωνομετρικών συναρτήσεων που περιλαμβάνει. Παρά αυτό το μειονέκτημα, η υψηλότερη απόδοσή της σε συγκεκριμένες εφαρμογές την καθιστά μια πολλά υποσχόμενη επιλογή [47].

2.4.6.8 SiLU (Swish)

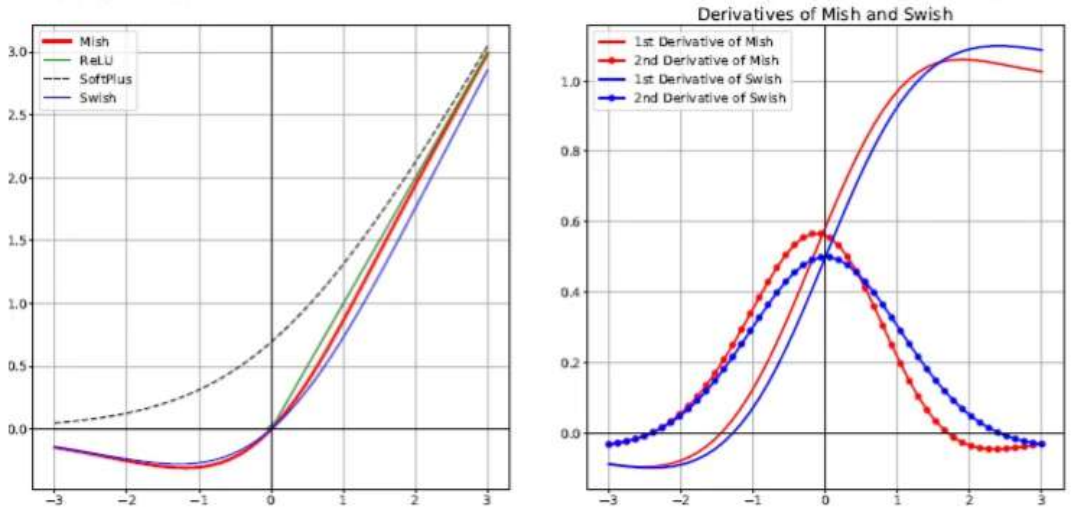
Η SiLU (Sigmoid Linear Unit), γνωστή και ως Swish, είναι μία ακόμη σύγχρονη συνάρτηση ενεργοποίησης που εισήχθη από την Google. Ορίζεται ως το γινόμενο της εισόδου με τη Sigmoid της ίδιας της εισόδου:

$$\Phi(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \tag{2.23}$$

Η SiLU εισάγει μία μη γραμμικότητα που είναι smooth και διαφοροποιήσιμη, επιτρέποντας στις αρνητικές τιμές να συνεισφέρουν στη μάθηση χωρίς να «μηδενίζονται», όπως συμβαίνει στη ReLU. Αυτό έχει ως αποτέλεσμα τη βελτίωση της ροής του σφάλματος και καλύτερη ευαισθησία σε μικρές διακυμάνσεις της εισόδου.

Η χρήση της SiLU έχει οδηγήσει σε καλύτερη απόδοση και ταχύτερη σύγκλιση σε μεγάλα δίκτυα βαθιάς μάθησης, όπως τα EfficientNet, και έχει αποδειχθεί ιδιαίτερα κατάλληλη για προβλήματα που απαιτούν ακρίβεια και γενίκευση, όπως η επεξεργασία εικόνων υψηλής ανάλυσης και η λεπτομερής ταξινόμηση αντικειμένων.

Παρότι η υπολογιστική της πολυπλοκότητα είναι μεγαλύτερη από της ReLU, θεωρείται μία από τις πιο αποτελεσματικές επιλογές για σύγχρονα μοντέλα, συνδυάζοντας απλότητα, ομαλότητα και ισχυρή απόδοση [48].



Εικόνα 15: Mish vs Swish Activation Functions [47]

2.4.6.9 Σύγκριση Συνήθων Συναρτήσεων Ενεργοποίησης

Συνάρτηση	Εφαρμογή	Πλεονεκτήματα	Μειονεκτήματα
Sigmoid	Δυαδική ταξινόμηση	Εξαγωγή πιθανοτήτων, smooth	Vanishing gradient σε μεγάλες τιμές
Tanh	Γενική χρήση	Κεντραρισμένη στο 0, καλύτερη διάδοση σφάλματος από Sigmoid	Vanishing gradient, ακατάλληλη για βαθιά δίκτυα
ReLU	Κρυφά επίπεδα CNNs	Απλή, υπολογιστικά αποδοτική, διατηρεί μεγάλα gradients	Dying ReLU (νευρώνες με 0 έξοδο)
Leaky ReLU	Κρυφά επίπεδα CNNs	Λύνει το Dying ReLU επιτρέποντας αρνητικές τιμές	Επιλογή κατάλληλου α
GELU	Μεγάλα μοντέλα, Transformers	Smooth, καλό για fine-grained μοτίβα, ταχύτερη σύγκλιση	Υπολογιστικά πιο απαιτητική
Softmax	Έξοδος για ταξινόμηση πολλαπλών κατηγοριών	Παράγει κατανομή πιθανοτήτων	Αριθμητική αστάθεια σε μεγάλες τιμές
Mish	Fine-grained recognition, ResNets, NLP	Ομαλότητα, συνεχής παράγωγος, ισχυρή γενίκευση	Πολύπλοκη, πιο αργή
SiLU (Swish)	Σύγχρονα CNNs (π.χ. EfficientNet)	Smooth, καλή διάδοση σφάλματος, ισχυρή απόδοση	Ελαφρώς πιο αργή από ReLU

Πίνακας 2: Σύγκριση Συνήθων Συναρτήσεων Ενεργοποίησης

2.5 Εκπαίδευση Νευρωνικών Δικτύων

Η εκπαίδευση ενός νευρωνικού δικτύου βασίζεται σε δύο βασικά στάδια: το Forward Pass και το Backpropagation. Μέσω αυτών, το δίκτυο μπορεί να μάθει τις κατάλληλες παραμέτρους ώστε να μοντελοποιεί μη γραμμικές σχέσεις μεταξύ εισόδων και εξόδων [19][40].

2.5.1 Forward Pass

Κατά το Forward Pass, τα δεδομένα εισόδου διαδίδονται διαδοχικά από το πρώτο έως το τελευταίο επίπεδο του δικτύου, ώστε να παραχθεί η τελική πρόβλεψη $\hat{y}$.

Δοθέντος ενός διανύσματος εισόδου $x \in R^{d_{in}}$, ο υπολογισμός της εξόδου γίνεται ως εξής:

- Κάθε επίπεδο l (layer) αποτελείται από ένα σύνολο νευρώνων.
- Κάθε νευρώνας εκτελεί έναν γραμμικό μετασχηματισμό ακολουθούμενο από μια μη γραμμική συνάρτηση ενεργοποίησης.

Η γενική μορφή του υπολογισμού στο επίπεδο l είναι:

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)} \tag{2.24}$$

$$a^{(l)} = f^{(l)}(z^{(l)}) \tag{2.25}$$

όπου:

- $W^{(l)} \in R^{n_l \times n_{l-1}}$ ο πίνακας βαρών του επιπέδου,
- $b^{(l)} \in R^{n_l}$ το διάνυσμα των bias,
- $a^{(l-1)}$ οι ενεργοποιήσεις από το προηγούμενο επίπεδο (με $a^{(0)} = x$),
- $f^{(l)}$ η συνάρτηση ενεργοποίησης (π.χ. ReLU, Sigmoid).

Η τελική έξοδος του δικτύου είναι:

$$\hat{y} = a^{(L)} \tag{2.26}$$

και συγκρίνεται με την πραγματική τιμή y μέσω της συνάρτησης απώλειας $\mathcal{L}(y, \hat{y})$.

2.5.2 Backpropagation

Το στάδιο του *Backpropagation* είναι υπεύθυνο για τον υπολογισμό των παραγώγων (gradients) της απώλειας ως προς τις παραμέτρους του δικτύου, δηλαδή τα βάρη και τα bias. Αυτό επιτυγχάνεται με εφαρμογή του Κανόνα της Αλυσίδας (Chain Rule) προς τα πίσω, από την έξοδο προς την είσοδο.

Ο στόχος είναι να υπολογίσουμε:

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} \text{ και } \frac{\partial \mathcal{L}}{\partial b^{(l)}} \text{ για κάθε } l \tag{2.27}$$

Ξεκινάμε από το τελικό επίπεδο:

$$\delta^{(L)} = \frac{\partial \mathcal{L}}{\partial z^{(L)}} = \nabla_a \mathcal{L} \odot f'^{(L)}(z^{(L)}) \tag{2.28}$$

και για κάθε προηγούμενο επίπεδο $l = L-1, L-2, ..., 1$:

$$\delta^{(l)} = \left(W^{(l+1)T} \delta^{(l+1)}\right) \odot f'^{(l)}(z^{(l)}) \tag{2.29}$$

όπου:

- $\delta^{(l)}$ ονομάζεται *error signal* ή *backpropagated error*,
- $f'^{(l)}$ είναι η παράγωγος της συνάρτησης ενεργοποίησης.

Οι παράγωγοι της απώλειας ως προς τα βάρη και τα bias προκύπτουν ως:

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \delta^{(l)} a^{(l-1)T} \tag{2.30}$$

$$\frac{\partial \mathcal{L}}{\partial b^{(l)}} = \delta^{(l)} \tag{2.31}$$

Αυτές οι τιμές χρησιμοποιούνται από αλγορίθμους βελτιστοποίησης όπως ο Stochastic Gradient Descent (SGD) ή ο Adam για την ενημέρωση των παραμέτρων του δικτύου με σκοπό την ελαχιστοποίηση της απώλειας.

2.6 Βελτιστοποίηση Νευρωνικών Δικτύων

Μετά τον υπολογισμό των παραγώγων μέσω του αλγορίθμου backpropagation, το επόμενο κρίσιμο βήμα είναι η βελτιστοποίηση των παραμέτρων του νευρωνικού δικτύου. Σκοπός της διαδικασίας βελτιστοποίησης είναι η ελαχιστοποίηση της συνάρτησης απώλειας $\mathcal{L}$, ώστε το δίκτυο να βελτιώσει την ακρίβειά του στα δεδομένα εκπαίδευσης και να γενικεύει αποτελεσματικά σε άγνωστα δείγματα.

2.6.1 Gradient Descent (GD)

Ο αλγόριθμος φθήνουσας πορείας κλίσης (Gradient Descent) αποτελεί τη βασικότερη τεχνική βελτιστοποίησης ευρέως χρησιμοποιούμενη στη μηχανική μάθηση [19]. Η βασική του ιδέα είναι η ενημέρωση των παραμέτρων (βάρη W και bias b) βάρη προς την κατεύθυνση του αρνητικού gradient της απώλειας:

$$W^{(l)} \leftarrow W^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial W^{(l)}} \tag{2.32}$$

$$b^{(l)} \leftarrow b^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial b^{(l)}} \tag{2.33}$$

όπου:

- η είναι ο ρυθμός εκμάθησης (*learning rate*),
- $\frac{\partial \mathcal{L}}{\partial W^{(l)}}$, $\frac{\partial \mathcal{L}}{\partial b^{(l)}}$ οι παράγωγοι της απώλειας στο επίπεδο .

Ο ρόλος του ρυθμού εκμάθησης είναι καθοριστικός. Πολύ μικρές τιμές οδηγούν σε αργή σύγκλιση, ενώ πολύ μεγάλες μπορούν να προκαλέσουν αστάθεια ή απόκλιση του αλγορίθμου.

Ο κύκλος εκπαίδευσης που επαναλαμβάνεται για όλα τα δεδομένα του συνόλου περιλαμβάνει τα εξής στάδια:

1. Εκτέλεση Forward Pass: Υπολογισμός της πρόβλεψης $\hat{y}$.
2. Υπολογισμός της απώλειας $\mathcal{L}(y, \hat{y})$, σύγκριση με την πραγματική τιμή y .
3. Εκτέλεση Backpropagation: Υπολογισμός των παραγώγων.
4. Ενημέρωση παραμέτρων.

Η διαδικασία επαναλαμβάνεται για πολλές εποχές (epochs) έως ότου το μοντέλο συγκλίνει. Αν και το Gradient Descent είναι θεωρητικά απλό, έχει ορισμένους περιορισμούς σε πρακτικά σενάρια:

- Υψηλό υπολογιστικό κόστος σε μεγάλα σύνολα δεδομένων, καθώς απαιτεί τον υπολογισμό των παραγώγων για ολόκληρο το dataset
- Ευαισθησία στην επιλογή του η.
- Δυσκολία σε μη κυρτές συναρτήσεις απώλειας.

Για την αντιμετώπιση αυτών των προβλημάτων έχουν προταθεί βελτιώσεις του βασικού αλγορίθμου, όπως το Stochastic Gradient Descent, Momentum, και Adam.

2.6.2 Stochastic Gradient Descent (SGD)

Το Stochastic Gradient Descent (SGD) είναι μια πιο αποδοτική παραλλαγή του κλασικού Gradient Descent και αποτελεί την κυρίαρχη επιλογή στην εκπαίδευση βαθιών δικτύων [49]. Αντί να χρησιμοποιεί το πλήρες σύνολο δεδομένων για κάθε ενημέρωση, το SGD εκτιμά το gradient από ένα δείγμα ή ένα μικρό τυχαίο υποσύνολο (mini-batch):

$$W^{(l)} \leftarrow W^{(l)} - \eta \frac{\partial \mathcal{L}_i}{\partial W^{(l)}} \tag{2.34}$$

$$b^{(l)} \leftarrow b^{(l)} - \eta \frac{\partial \mathcal{L}_i}{\partial b^{(l)}} \tag{2.35}$$

Όπου:

- η είναι ο ρυθμός εκμάθησης (*learning rate*),
- $\mathcal{L}_i$ είναι η απώλεια για το i-οστό δείγμα ή mini-batch,

- $W^{(l)}$, $b^{(l)}$ είναι τα βάρη και τα bias του επιπέδου l.

Τα πλεονεκτήματά του είναι:

- Γρήγορες ενημερώσεις και άρα μεγαλύτερη ταχύτητα ανά iteration.
- Καταλληλότητα για μεγάλα datasets.
- Δυνατότητα διαφυγής από τοπικά ελάχιστα λόγω του θορύβου.

Ενώ στα μειονεκτήματά του είναι:

- Δημιουργεί τυχαίες ταλαντώσεις στην τροχιά εκπαίδευση.
- Πιθανή ασταθής σύγκλιση χωρίς κατάλληλες τεχνικές (π.χ. momentum).

2.6.3 Momentum

Η μέθοδος *Momentum* [50] εισάγει έναν όρο "δυναμικής" που λειτουργεί σαν μνήμη προηγούμενων gradients, επιτρέποντας πιο ομαλή και επιταχυνόμενη πορεία προς το ελάχιστο:

$$v^{(l)} \leftarrow \gamma v^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial W^{(l)}} \tag{2.36}$$

$$W^{(l)} \leftarrow W^{(l)} + v^{(l)} \tag{2.37}$$

όπου:

- $v^{(l)}$ είναι η μεταβλητή ορμής (momentum) για το επίπεδο l,
- $\gamma \in [0, 1)$ είναι ο συντελεστής ορμής (momentum coefficient), συνήθως 0.9 ή 0.99.

Η χρήση του Momentum επιταχύνει την εκπαίδευση και βελτιώνει τη σύγκλιση σε κυρτές και μη κυρτές συναρτήσεις απώλειας.

2.6.4 Adam Optimizer

Ο Adam (Adaptive Moment Estimation) [51] είναι από τους πιο δημοφιλείς αλγορίθμους βελτιστοποίησης, συνδυάζοντας τα πλεονεκτήματα των Momentum και Adaptive Gradient Algorithm (AdaGrad).

Υπολογίζει δύο κινούμερους μέσους όρους:

$$m_t^{(l)} = \beta_1 m_{t-1}^{(l)} + (1 - \beta_1) \frac{\partial \mathcal{L}}{\partial W^{(l)}} \tag{2.38}$$

$$v_t^{(l)} = \beta_2 v_{t-1}^{(l)} + (1 - \beta_2) \left(\frac{\partial \mathcal{L}}{\partial W^{(l)}} \right)^2 \tag{2.39}$$

Τα m_t και v_t διορθώνονται ως:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2.40}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{2.41}$$

Η ενημέρωση των παραμέτρων γίνεται ως:

$$W^{(l)} \leftarrow W^{(l)} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{2.42}$$

όπου:

- β_1, β_2 είναι παράμετροι που ελέγχουν το momentum,
- ϵ είναι σταθερά για αριθμητική σταθερότητα.

Τα πλεονεκτήματα του Adam είναι τα εξής:

- Προσαρμόσιμο learning rate.
- Ταχεία και σταθερή σύγκλιση.
- Λειτουργεί καλά χωρίς πολλή ρύθμιση υπερπαραμέτρων.

2.6.5 Σύγκριση Τεχνικών Βελτιστοποίησης

Μέθοδος	**Πλεονεκτήματα**	**Μειονεκτήματα**
Gradient Descent (GD)	Απλό στην υλοποίηση, θεωρητικά σταθερό	Αργό σε μεγάλα datasets, υψηλό υπολογιστικό κόστος ανά εποχή
Stochastic Gradient Descent (SGD)	Πολύ πιο γρήγορο ανά βήμα, αποδοτικό σε μεγάλα datasets	Ευαίσθητο σε τυχαίες ταλαντώσεις, πιθανή καθυστερημένη σύγκλιση
Momentum	Επιταχύνει τη σύγκλιση, μειώνει ταλαντώσεις	Προσθέτει μία επιπλέον υπερπαράμετρο (γ) που χρειάζεται ρύθμιση
Adam	Αυτόματη προσαρμογή learning rate, σταθερή και γρήγορη σύγκλιση	Ευαισθησία στην επιλογή των β_1, β_2, πιθανή υπερπροσαρμογή σε μικρά datasets

Πίνακας 3: Σύγκριση Τεχνικών Βελτιστοποίησης

Κάθε αλγόριθμος βελτιστοποίησης έχει τα δικά του χαρακτηριστικά και συμπεριφορά ανάλογα με το πρόβλημα. Το απλό Gradient Descent αποτελεί τη θεωρητική βάση, όμως το SGD με τεχνικές όπως το Momentum και κυρίως ο Adam, υπερέχουν στην πράξη — ειδικά σε μεγάλα, βαθιά και μη κυρτά προβλήματα. Η επιλογή της κατάλληλης τεχνικής εξαρτάται από τη φύση του προβλήματος, το μέγεθος του dataset και τις ιδιαιτερότητες του μοντέλου που χρησιμοποιείται.

2.7 Αρχιτεκτονικές CNN

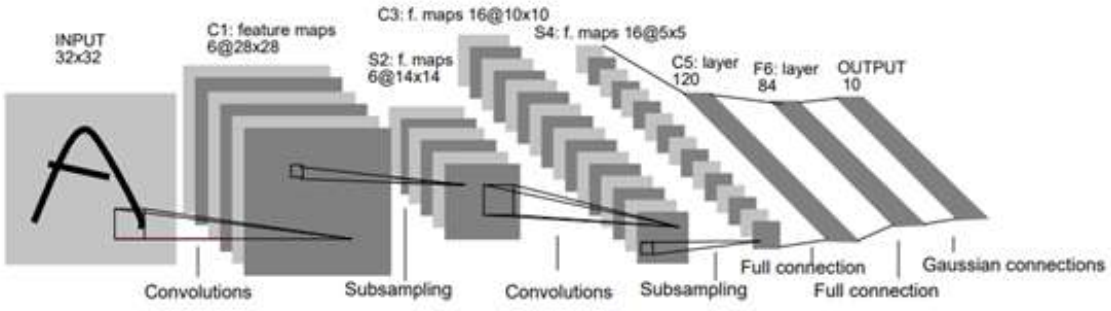
Η εξέλιξη των συνελικτικών νευρωνικών δικτύων (CNNs) έχει οδηγήσει στην ανάπτυξη πληθώρας αρχιτεκτονικών, καθεμία εκ των οποίων εισήγαγε καινοτομίες που στόχευαν στη βελτίωση της απόδοσης, τη μείωση της υπολογιστικής πολυπλοκότητας και την επίλυση προβλημάτων εκπαίδευσης βαθιών δικτύων. Στη συνέχεια παρουσιάζονται μερικές από τις πιο επιδραστικές αρχιτεκτονικές.

2.7.1 LeNet-5

Το LeNet-5, προτεινόμενο από τον Yann LeCun το 1998 [28], ήταν από τα πρώτα CNNs που εφαρμόστηκαν με επιτυχία στην ταξινόμηση εικόνων - στην αρχή σχεδιάστηκε για την αναγνώριση χειρόγραφων ψηφίων στο MNIST. Η αρχιτεκτονική του αποτελείται από τα εξής 7 επίπεδα:

- Δύο συνελικτικά επίπεδα (convolutional layers) που εξάγουν χαρακτηριστικά από τα δεδομένα εισόδου.
- Δύο επίπεδα ομαδοποίησης (subsampling/pooling) που μειώνουν τη διάσταση των χαρακτηριστικών.
- Δύο πλήρως συνδεδεμένα επίπεδα που λειτουργούν ως ταξινομητές.
- Ένα τελικό επίπεδο εξόδου που χρησιμοποιεί τη συνάρτηση ενεργοποίησης softmax για ταξινόμηση σε 10 κατηγορίες.

Χρησιμοποιούσε *tanh* ως συνάρτηση ενεργοποίησης και ήταν αρκετά συμπαγές μοντέλο σε σύγκριση με σύγχρονες αρχιτεκτονικές. αλλά αποτέλεσε θεμέλιο για τις επόμενες γενιές CNNs [28].



Εικόνα 16: LeNet-5 Arxhitecture

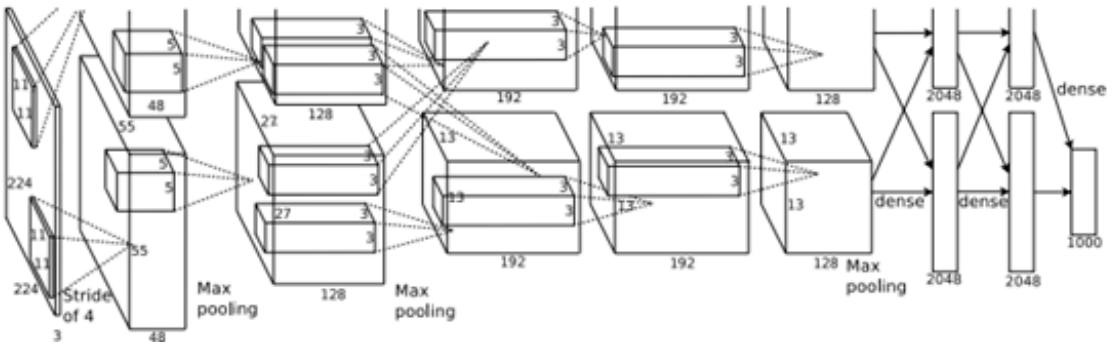
2.7.2 AlexNet

Το AlexNet αποτέλεσε ορόσημο στην ιστορία των CNNs, καθώς ήταν το πρώτο βαθύ δίκτυο που κέρδισε το ImageNet Challenge (ILSVRC) το 2012, μειώνοντας δραστικά το σφάλμα. Οι βασικές του καινοτομίες:

- Χρήση της συνάρτησης ReLU, επιταχύνοντας τη σύγκλιση στην εκπαίδευση.
- Dropout για την αποφυγή του overfitting.
- Χρήση δύο GPU για την παράλληλη εκπαίδευση του μοντέλου και αύξηση ταχύτητας.

- Πέντε συνελικτικά και τρία πλήρως συνδεδεμένα επίπεδα.

Η εν λόγω αρχιτεκτονική απέδειξε ότι η αύξηση του βάθους του μοντέλου, σε συνδυασμό με ισχυρό υπολογιστικό υλικό, μπορεί να επιφέρει ουσιαστικές βελτιώσεις στην απόδοση, οδηγώντας σε ευρεία εφαρμογή στον τομέα της ταξινόμησης εικόνων [31].



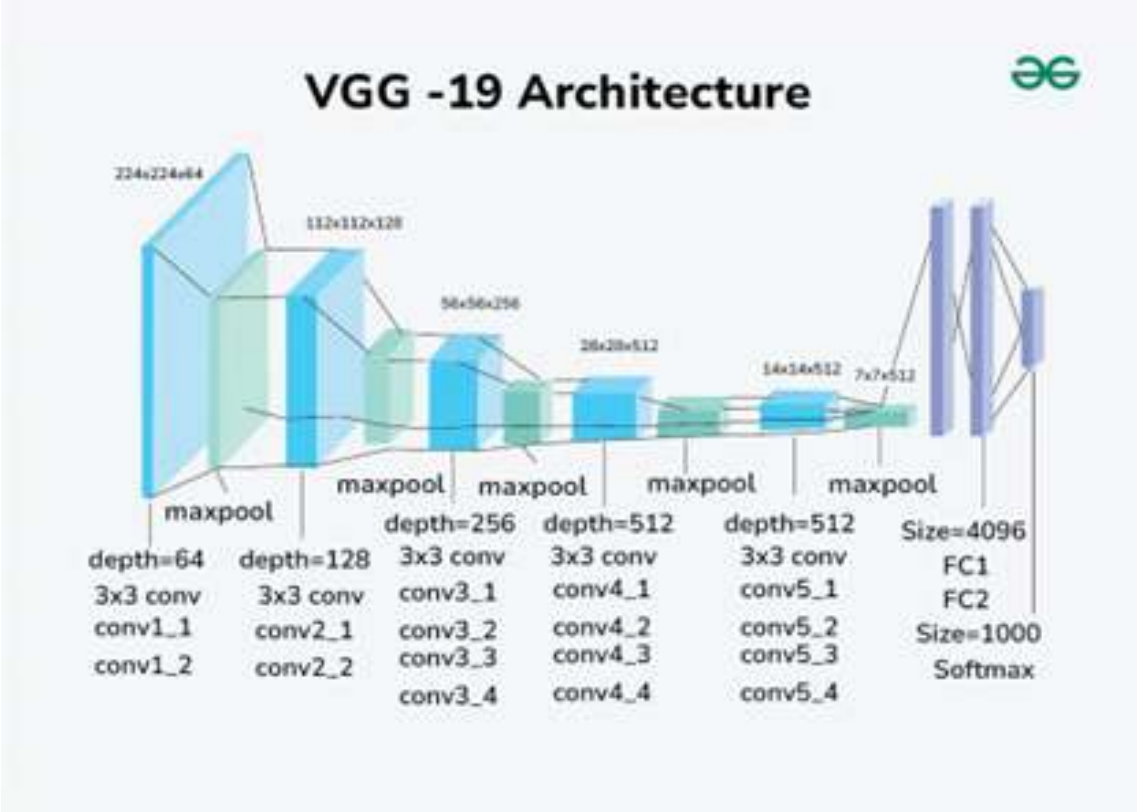
Εικόνα 17: AlexNet Arxhitecture

2.7.3 VGGNet

Το VGGNet, προτεινόμενο από το Visual Geometry Group (VGG) στο Πανεπιστήμιο της Οξφόρδης, εισήγαγε την ιδέα της ομοιομορφίας στην αρχιτεκτονική. Χρησιμοποίησε μικρά φίλτρα 3×3 σε όλα τα συνελικτικά επίπεδα, αυξάνοντας σταδιακά το βάθος για βελτίωση της απόδοσης. Οι δύο πιο γνωστές παραλλαγές:

- VGG-16: περιέχει 16 επίπεδα (13 συνελικτικά + 3 fully connected επίπεδα).
- VGG-19, περιέχει 19 επίπεδα (16 συνελικτικά + 3 fully connected επίπεδα).

Το βασικό του πλεονέκτημα ήταν η απλότητα και η αποτελεσματικότητα μέσω της αύξησης του βάθους με ελεγχόμενο αριθμό παραμέτρων [32].



Εικόνα 18: VGG-19 Arxhitecture

2.7.4 GoogLeNet / Inception

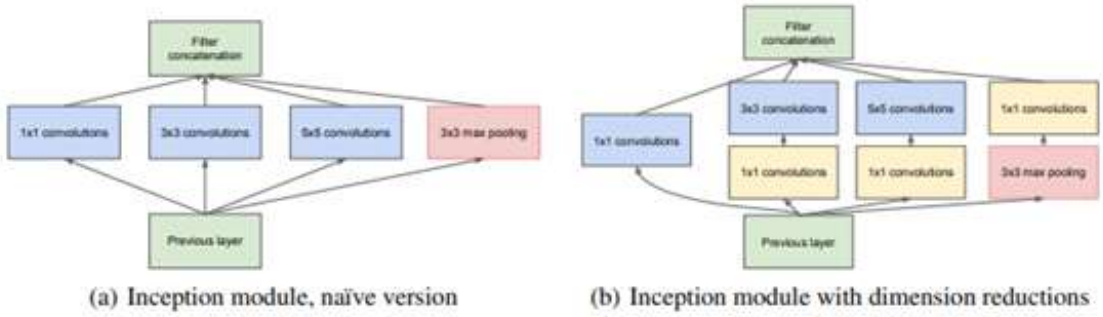
Το GoogLeNet (ή Inception v1) εισήγαγε μια νέα προσέγγιση στον σχεδιασμό CNNs, μέσω των Inception modules [52], τα οποία επέτρεπαν στο δίκτυο να εξάγει χαρακτηριστικά σε πολλαπλές κλίμακες ταυτόχρονα. Σε αντίθεση με τα προηγούμενα δίκτυα (π.χ. AlexNet, VGGNet), όπου τα συνελικτικά επίπεδα ακολουθούσαν απλή σειριακή δομή με σταθερό μέγεθος φίλτρου, τα Inception modules συνδύαζαν παράλληλες συνελίξεις διαφορετικών μεγεθών (1×1, 3×3, 5×5), καθώς και pooling.

Οι βασικές καινοτομίες που εισήγαγε το GoogLeNet ήταν:

- Παράλληλες συνελίξεις διαφορετικών μεγεθών: Η χρήση πολλαπλών διαδρομών (paths) μέσα σε κάθε Inception module επέτρεπε την ταυτόχρονη εκμάθηση χαρακτηριστικών σε διαφορετικές κλίμακες — προσέγγιση που δεν είχε χρησιμοποιηθεί σε προηγούμενες αρχιτεκτονικές.
- Συνελίξεις 1×1 (bottleneck layers): Η εισαγωγή 1×1 συνελίξεων ως μέθοδος μείωσης της διάστασης των feature maps, μειώνοντας δραστικά τον αριθμό των παραμέτρων και το υπολογιστικό κόστος. Αυτή η τεχνική επέτρεψε την κατασκευή βαθύτερων και πιο αποδοτικών δικτύων χωρίς υπερβολική αύξηση της μνήμης ή του χρόνου εκπαίδευσης.

Η συνολική σχεδίαση πέτυχε σημαντική μείωση του αριθμού παραμέτρων (περίπου 12 φορές λιγότερες από το AlexNet), κάνοντάς το πιο κατάλληλο για πρακτική χρήση σε μεγάλα datasets. Ακολούθησαν βελτιωμένες εκδόσεις (Inception v2, v3, v4) με επιπλέον τεχνικές όπως factorized

convolutions και batch normalization ενισχύοντας ακόμα περισσότερο την αποδοτικότητα και ακρίβεια των μοντέλων [53].



(a) Inception module, naïve version (b) Inception module with dimension reductions

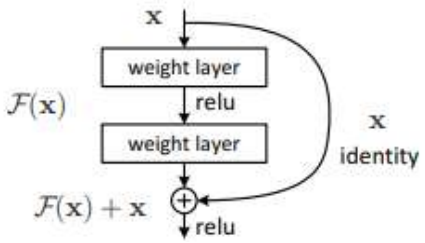
Εικόνα 19: Inception Module

2.7.5 ResNet (Residual Networks)

Το ResNet εισήγαγε τις υπολειμματικές συνδέσεις (residual connections) που έλυσαν το πρόβλημα του vanishing gradient. Ο πυρήνας της φιλοσοφίας ήταν το εξής:

$$y = F(x) + x \tag{2.43}$$

Δηλαδή, η έξοδος ενός μπλοκ περιλαμβάνει την είσοδο αυτού (identity mapping), ενισχύοντας τη ροή πληροφορίας. Το ResNet επέτρεψε την εκπαίδευση πολύ βαθιών δικτύων (έως και 152 επιπέδων) με μεγάλη επιτυχία στο ImageNet [30].



Εικόνα 20: ResNet Block

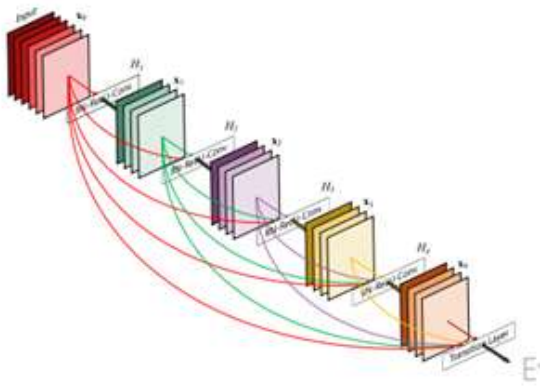
Αυτή η παράκαμψη (*skip connection*) βοηθάει στη διατήρηση της πληροφορίας και επιτρέπει την εκπαίδευση πολύ βαθιών δικτύων χωρίς να εξαφανίζεται το gradient. Το ResNet-152 πέτυχε νέα ρεκόρ στο ImageNet, αποδεικνύοντας ότι η αύξηση του βάθους μπορεί να οδηγήσει σε σημαντικές βελτιώσεις όταν χρησιμοποιούνται οι σωστές τεχνικές [30].

2.7.6 DenseNet (Densely Connected Networks)

Το DenseNet επέκτεινε την ιδέα των residual connections, εισάγοντας πυκνές συνδέσεις (dense connections) μεταξύ όλων των επιπέδων που έχουν ίδιο μέγεθος χάρτη χαρακτηριστικών. Κύρια χαρακτηριστικά του:

- Κάθε επίπεδο λαμβάνει ως είσοδο την έξοδο όλων των προηγούμενων επίπεδων.
- Ενισχύεται η επαναχρησιμοποίηση χαρακτηριστικών.

- Μειώνεται ο αριθμός των παραμέτρων, καθώς τα χαρακτηριστικά διατηρούνται και δεν επανυπολογίζονται.



Εικόνα 21: DenseNet Block

Το DenseNet απέδειξε ότι η σύνδεση όλων των επιπέδων μεταξύ τους μπορεί να βελτιώσει την εκπαίδευση και να μειώσει τις ανάγκες σε υπολογιστική ισχύ ιδίως σε μικρά datasets [29].

2.7.7 CoAtNet

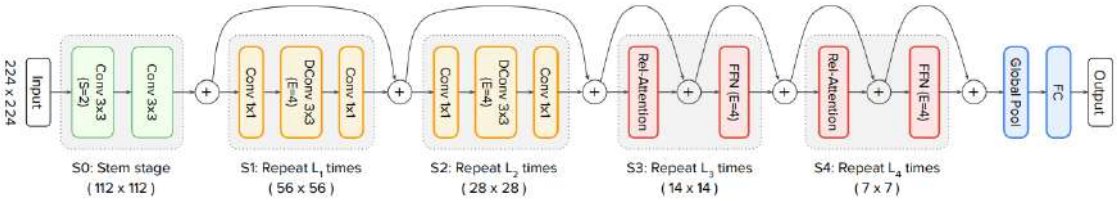
Το CoAtNet (Convolution and Attention Network) [54] αποτελεί μια σύγχρονη υβριδική αρχιτεκτονική, η οποία συνδυάζει τα βασικά πλεονεκτήματα των συνελικτικών νευρωνικών δικτύων (CNNs) με τα οφέλη των Vision Transformers (ViTs). Δεν πρόκειται για ένα τυπικό CNN, αλλά για μια ενοποιημένη προσέγγιση που αξιοποιεί τόσο convolution όσο και self-attention.

Τα Vision Transformers (ViTs) [55] εισήχθησαν ως εναλλακτική προσέγγιση στην επεξεργασία εικόνας, βασισμένοι στην επιτυχία των Transformer μοντέλων στη φυσική γλώσσα [56]. Σε αντίθεση με τα CNNs, τα οποία ενσωματώνουν τοπικά φίλτρα και inductive bias για spatial locality, οι ViTs αναλύουν την εικόνα ως ακολουθία από patches, εφαρμόζοντας self-attention για τη μοντελοποίηση συσχετίσεων σε μεγάλη χωρική απόσταση. Αυτή η δυνατότητα αποδεικνύεται ιδιαίτερα χρήσιμη σε προβλήματα fine-grained ταξινόμησης, όπου μικρές λεπτομέρειες μπορούν να κάνουν τη διαφορά.

Ωστόσο, η έλλειψη εγγενών περιορισμών στα ViTs συχνά απαιτεί εκπαίδευση σε μεγάλα datasets ώστε να επιτευχθεί υψηλή απόδοση, κάτι που περιορίζει τη χρηστικότητα τους σε σενάρια με περιορισμένα δεδομένα.

Επομένως, η ανάγκη για ένα ενδιάμεσο μοντέλο οδήγησε στην ανάπτυξη του CoAtNet. Τα πρώτα του στάδια βασίζονται σε MBConv blocks, παρόμοια με εκείνα του EfficientNet, προσφέροντας την ικανότητα εξαγωγής τοπικών χαρακτηριστικών με αποδοτικότητα. Στα υψηλότερα επίπεδα, η αρχιτεκτονική μεταβαίνει σε relative self-attention blocks, αξιοποιώντας global πληροφορία και επιτυγχάνοντας καλύτερη γενίκευση.

Αυτός ο συνδυασμός καθιστά το CoAtNet ιδιαίτερα κατάλληλο για προβλήματα Fine-Grained Visual Classification (FGVC), καθώς προσφέρει ισχυρή εκφραστικότητα και ταυτόχρονα διατηρεί την αποδοτικότητα των CNNs. Επιπλέον, έχει αποδειχθεί ότι το μοντέλο επιτυγχάνει υψηλές επιδόσεις ακόμα και σε περιορισμένα datasets, διευρύνοντας τις δυνατότητές του πέρα από τα μεγάλα benchmark σύνολα.



Εικόνα 22: CoAtNet exaample

2.7.8 Συγκριτική ανάλυση αρχιτεκτονικών CNN

Αρχιτεκτονική	Έτος	Βάθος (Επίπεδα)	Πλήθος Παραμέτρων	Top-5 Σφάλμα (ImageNet)
LeNet-5	1998	7	~60K	-
AlexNet	2012	8	~60M	15.3%
VGG-16	2014	16	~138M	9.3%
GoogLeNet (Inception v1)	2014	22	~6.8M	6.7%
ResNet-152	2015	152	~60M	4.5%
DenseNet-201	2017	201	~20M	4.8%
CoAtNet-0	2021	~50+	~25M	3.6%

Πίνακας 4: Συγκριτική ανάλυση αρχιτεκτονικών CNN

Οι παραπάνω αρχιτεκτονικές αντιπροσωπεύουν τη σταδιακή πρόοδο στον σχεδιασμό συνελικτικών νευρωνικών δικτύων, από τα πρώτα πρότυπα όπως το LeNet-5 έως τις σύγχρονες υβριδικές προσεγγίσεις όπως το CoAtNet. Κάθε μοντέλο εισήγαγε καινοτομίες που βελτίωσαν την απόδοση, τη δυνατότητα γενίκευσης ή την αποδοτικότητα. Η εξέλιξη αυτή υπογραμμίζει τη σημασία της σωστής αρχιτεκτονικής επιλογής ανάλογα με τη φύση του προβλήματος, τα διαθέσιμα δεδομένα και τους υπολογιστικούς περιορισμούς.

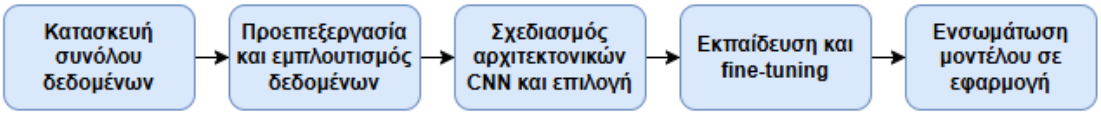
3 Μεθοδολογία και Υλοποίηση

Η ανάπτυξη ενός αποτελεσματικού συστήματος αναγνώρισης τύπων και μοντέλων κλιματιστικών βασισμένου σε εικόνες απαιτεί την ενιαία αξιοποίηση τόσο θεωρητικών γνώσεων από τον χώρο της υπολογιστικής όρασης και της βαθιάς μάθησης, όσο και πρακτικών εργαλείων υλοποίησης και αξιολόγησης. Στο παρόν κεφάλαιο περιγράφεται η μεθοδολογική πορεία που ακολουθήθηκε, η οποία καλύπτει όλα τα στάδια από την κατασκευή του συνόλου δεδομένων έως την εφαρμογή και βελτιστοποίηση του τελικού μοντέλου.

Η πρόκληση έγκειται στην επίλυση ενός προβλήματος fine-grained classification, όπου οι κατηγορίες διαφοροποιούνται οπτικά με πολύ μικρές λεπτομέρειες, γεγονός που καθιστά τις συμβατικές μεθόδους λιγότερο αποτελεσματικές. Για την αντιμετώπιση αυτής της δυσκολίας, ακολουθήθηκε πολυεπίπεδη προσέγγιση που περιλαμβάνει:

- **Κατασκευή κατάλληλου συνόλου δεδομένων**: συλλογή εικόνων μέσω mobile εφαρμογής και συμπλήρωση με εικόνες διαθέσιμες στο διαδίκτυο.

- **Προεπεξεργασία και εμπλουτισμός δεδομένων**: εφαρμογή τεχνικών normalization, κανονικοποίησης μεγέθους, και στοχευμένων μετασχηματισμών (augmentation).
- **Εκπαίδευση διαφορετικών CNN αρχιτεκτονικών**: όπως MixDCNN, HBP-CNN, CN-CNN και υβριδικές δομές με ViTs, με στόχο τη συγκριτική αξιολόγηση.
- **Στρατηγική *fine-tuning***: αξιοποίηση προεκπαιδευμένων μοντέλων με σταδιακή προσαρμογή (progressive unfreezing) στις απαιτήσεις του dataset.
- **Ανάλυση απόδοσης και επιλογή βέλτιστου μοντέλου**: βασισμένη σε μετρικές ακρίβειας (Top-1, Top-3), F1-score, και οπτικοποίηση embeddings.
- **Ενσωμάτωση του μοντέλου σε εφαρμογή**: ανάπτυξη backend και mobile περιβάλλοντος επίδειξης, για πειραματική χρήση του ταξινομητή σε πραγματικό σενάριο.



Εικόνα 23: Βήματα Υλοποίησης

Οι επόμενες ενότητες παρουσιάζουν διαδοχικά τις παραπάνω φάσεις, ξεκινώντας από τη συλλογή και επιμέλεια των δεδομένων, συνεχίζοντας με την ανάλυση των αρχιτεκτονικών και τελικά καταλήγοντας στην πειραματική αξιολόγηση και εφαρμογή.

3.1 Σύνολο δεδομένων εκπαίδευσης (dataset)

3.1.1 Επιλογή και Δημιουργία Dataset

Η επιλογή του κατάλληλου συνόλου δεδομένων αποτελεί κρίσιμο παράγοντα για την επιτυχή εκπαίδευση ενός CNN. Παρότι τα CNNs έχουν τη δυνατότητα να μαθαίνουν πολύπλοκα μοτίβα απευθείας από τα δεδομένα εισόδου, η ποιότητα, η ποικιλία και η ποσότητα των παραδειγμάτων παίζουν καθοριστικό ρόλο στη γενίκευση και την ακρίβεια του μοντέλου [57].

Ένα καλά επιλεγμένο dataset λειτουργεί ως θεμέλιο για τη διαδικασία μάθησης, επιτρέποντας στο δίκτυο να εντοπίζει συσχετίσεις και δομές που είναι απαραίτητες για το εκάστοτε πρόβλημα. Αντιθέτως, ένα ανεπαρκές ή μη αντιπροσωπευτικό dataset μπορεί να οδηγήσει σε overfitting, προκαταλήψεις στα αποτελέσματα (bias) και χαμηλή απόδοση σε νέα, άγνωστα δεδομένα. Αντίθετα, ένα καλό σύνολο δεδομένων εκπαίδευσης επιτρέπει στο μοντέλο να βελτιώσει την απόδοσή του, ώστε να λειτουργεί αξιόπιστα και σε νέα, αόρατα δείγματα.

Για να θεωρείται κατάλληλο ένα dataset, θα πρέπει να πληροί τις παρακάτω προδιαγραφές:

1. **Ποικιλία και Αντιπροσωπευτικότητα:** Τα δεδομένα θα πρέπει να καλύπτουν όλες τις πιθανές περιπτώσεις που μπορεί να συναντήσει το μοντέλο στην πράξη ώστε να είναι ανθεκτικό σε πραγματικές συνθήκες. Για παράδειγμα, σε ένα dataset ταξινόμησης αντικειμένων, οι εικόνες θα πρέπει να περιλαμβάνουν διαφορετικούς φωτισμούς, γωνίες και περιβάλλοντα.

2. **Ισορροπία μεταξύ των κατηγοριών (Class Balance):** Η ομοιόμορφη κατανομή ανάμεσα στις κατηγορίες αποτρέπει το φαινόμενο προκατάληψης προς πιο συχνές τάξεις. Για να διασφαλιστεί η σωστή εκπαίδευση, πρέπει να υπάρχει ισορροπία ή να εφαρμοστούν τεχνικές όπως data augmentation ή oversampling.

3. **Καθαρότητα και Ορθότητα:** Οι εικόνες πρέπει να είναι καθαρές, ευκρινείς και σωστά επισημασμένες. Εσφαλμένες ετικέτες μπορούν να αποπροσανατολίσουν το μοντέλο.

4. **Επαρκής Όγκος (Data Volume):** Δεδομένου ότι τα CNNs απαιτούν μεγάλο αριθμό παραδειγμάτων, η ύπαρξη επαρκών δεδομένων είναι κρίσιμη. Σε αντίθετη περίπτωση, αξιοποιούνται τεχνικές όπως transfer learning ή data augmentation.

5. **Κατάλληλος Διαχωρισμός (Data Splitting):** Το dataset πρέπει να διαχωρίζεται σε:
    - **Training set:** Για την εκπαίδευση του μοντέλου.
    - **Validation set:** Για την αξιολόγηση σε μη εκπαιδευμένα δείγματα ώστε να αποφευχθεί overfitting και τη ρύθμιση υπερπαραμέτρων.
    - **Test set:** Για την τελική αξιολόγηση απόδοσης. Δεν είναι απαραίτητο στη φάση των εκπαιδεύσεων και μπορεί να χρησιμοποιηθεί στο τέλος για έλεγχο ορθότητας του μοντέλου

Η σωστή επιλογή και προεπεξεργασία των δεδομένων είναι καθοριστική για την επιτυχία ενός CNN. Στο επόμενο κεφάλαιο, θα παρουσιαστεί το dataset που χρησιμοποιήθηκε για την παρούσα μελέτη.

3.1.2 Δημιουργία του Dataset

Δεδομένου ότι το πρόβλημα υπάγεται στην κατηγορία του fine-grained classification, η ύπαρξη ενός κατάλληλου dataset αποτελεί απαραίτητη προϋπόθεση για την αποτελεσματική εκπαίδευση των αρχιτεκτονικών.

Ωστόσο, δεν υπάρχουν διαθέσιμα δημόσια σύνολα δεδομένων που να περιλαμβάνουν εικόνες μοντέλων κλιματιστικών με αναλυτική επισήμανση κατηγορίας. Οι περισσότερες διαθέσιμες βάσεις δεδομένων επικεντρώνονται σε γενικές κατηγορίες συσκευών (π.χ., “air conditioner”), χωρίς λεπτομέρειες μοντέλου ή κατασκευαστή.

Το μεγαλύτερο πρόβλημα που προέκυψε είναι η έλλειψη μεγάλων, δημοσίων datasets που περιλαμβάνουν εικόνες κλιματιστικών με ετικέτες μοντέλου και άρα να μπορούν να βοηθήσουν στη συγκεκριμένη ταξινόμηση. Οι περισσότερες διαθέσιμες βάσεις δεδομένων επικεντρώνονται σε γενικές κατηγορίες συσκευών (π.χ. διαχωρισμός ψηγείου από air conditioner), χωρίς λεπτομέρειες μοντέλου ή κατασκευαστή.

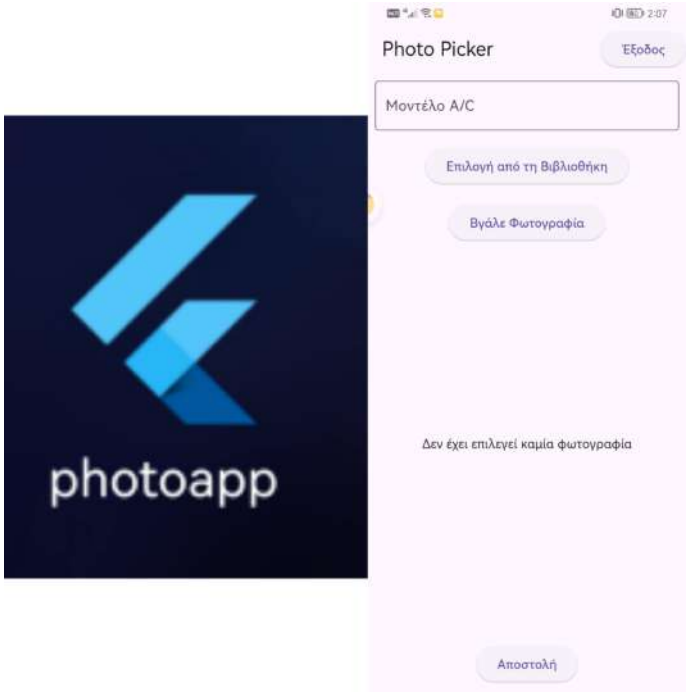
Για την αντιμετώπιση του προβλήματος, δημιουργήθηκε ένα νέο dataset μέσω δύο συμπληρωματικών στρατηγικών:

Εικόνα 24: Συνολικό Dataset

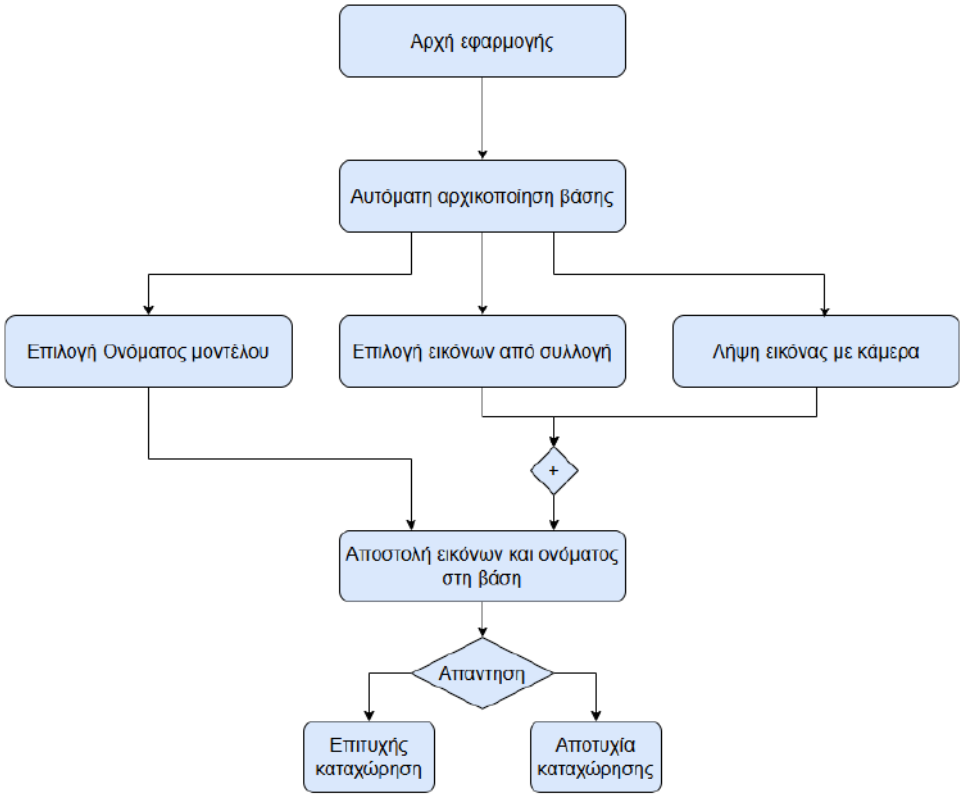
3.1.2.1 Συλλογή μέσω εφαρμογής κινητού

Αναπτύχθηκε προσωρινή εφαρμογή για κινητά που επέτρεπε σε χρήστες να ανεβάζουν φωτογραφίες των κλιματιστικών τους μαζί με την ετικέτα του μοντέλου. Η εφαρμογή διανεμήθηκε σε εθελοντές, οι οποίοι κλήθηκαν να καταγράψουν συσκευές από διαφορετικές γωνίες, αποστάσεις και φωτισμούς. Οι εικόνες αποθηκεύονταν με κατάλληλη ονομασία ώστε να διευκολυνθεί η κατηγοριοποίηση. Ακολούθησε διαδικασία φιλτραρίσματος, ώστε να παραμείνουν μόνο οι εικόνες που θα οδηγούσαν σε ενίσχυση της εκπαίδευσης. Απορρίφθηκαν θολές, σκοτεινές ή μη χρηστικές εικόνες, καθώς και επαληθεύτηκαν οι ετικέτες σε σχέση με τα ονόματά τους.

Το αποτέλεσμα ήταν ένα αρχικό σύνολο δεδομένων με περίπου 50 διαφορετικά μοντέλα, επαρκές για αρχικούς ελέγχους ακρίβειας και αξιολόγηση της καταλληλότητας του προβλήματος για CNN μοντέλα.



Εικόνα 25: Η εφαρμογή συλλογής δεδομένων για το dataset.



Εικόνα 26: Flowchart αρχικής εφαρμογής

3.1.2.2 Συμπλήρωση με Εικόνες από το διαδίκτυο

Για την επέκταση του dataset, συμπληρώθηκαν εικόνες από διαδικτυακές πηγές. Συλλέχθηκαν εικόνες κλιματιστικών από επίσημες ιστοσελίδες, e-shops και καταλόγους, με στόχο την κάλυψη ευρύτερης γκάμας μοντέλων και κατασκευαστών.

3.1.2.3 Τελικό Dataset

Το τελικό dataset υποβλήθηκε σε διαδικασία απο-διπλοποίησης και καθαρισμού, ώστε να αποφευχθεί η ύπαρξη πολλών διαφορετικών labels για την ίδια εικόνα ή η εισαγωγή πλεονάζοντων παραδειγμάτων.

Συνολικά δημιουργήθηκε ένα σύνολο δεδομένων από 216 διαφορετικά μοντέλα κλιματιστικών, αποτελούμενο από 720 καθαρές, επαληθευμένες εικόνες. Το συνολικό μέγεθος του dataset είναι περίπου 32.8 MB (34,443,704 bytes).

Κάθε εικόνα αντιστοιχίζεται σε ένα μοναδικό αριθμητικό label, το οποίο προέρχεται από λεξικό αντιστοίχισης (mapping dictionary) μεταξύ ονόματος μοντέλου και αριθμητικής ετικέτας.

Επιπλέον, παρέχεται αρχείο CSV, το οποίο περιέχει για κάθε εικόνα:

- Το **όνομα του αρχείου εικόνας**
- Την **αντιστοιχισμένη αριθμητική ετικέτα (label)**

Το συγκεκριμένο dataset αποτελεί τη βάση για την αρχική εκπαίδευση, αξιολόγηση και πειραματική σύγκριση των CNN αρχιτεκτονικών που παρουσιάζονται στην παρούσα εργασία. Επιπλέον,

είναι σημαντικό να αναφερθεί ότι εκτός από την τοπική αποθήκευση, το σύνολο των στοιχείων έχει καταχωριστεί και στη βάση Supabase για ευκολία πρόσβασης και μελλοντική επαναχρησιμοποίηση.

Η οργάνωση των δεδομένων περιλαμβάνει:

- Ένα bucket αποθήκευσης εικόνων, που περιέχει όλες τις εικόνες του dataset.
- Έναν πίνακα `class_labels`, ο οποίος περιλαμβάνει τα πεδία:
  - `id` (int4) — αριθμητικό label
  - `name` (text) — όνομα μοντέλου
  - `image_url` (text) — URL μιας εικόνας του μοντέλου στο bucket.

Οι διαδικασίες προεπεξεργασίας (ομογενοποίηση μεγέθους, κανονικοποίηση, data augmentation) περιγράφονται αναλυτικά στα επόμενα κεφάλαια.

3.2 Σχεδιασμός Εξειδικευμένων CNN

Στη συνέχεια αναλύονται οι εξειδικευμένοι αλγόριθμοι που έχουν επιλεγεί μετά από βιβλιογραφική έρευνα. Καθένας από αυτούς έχει διαφορετικά πλεονεκτήματα και τεχνικές που έχουν παρουσιάσει αρκετά καλά αποτελέσματα σε διάφορα FGVC προβλήματα.

3.2.1 MixDCNN

Η προσέγγιση Mixture of Deep Convolutional Neural Networks (MixDCNN) προτάθηκε για να αντιμετωπίσει το ζήτημα υψηλής ομοιότητας μεταξύ των κατηγοριών και των περιορισμένων διαφορών στις οπτικές λεπτομέρειες, προσφέροντας έναν δυναμικό μηχανισμό όπου πολλαπλά CNNs (experts) συνεργάζονται για την κατηγοριοποίηση ενός δείγματος [58].

Η βασική ιδέα του MixDCNN στηρίζεται στην έννοια των Mixture of Experts [59], όπου πολλαπλά υποδίκτυα αναλαμβάνουν από κοινού την επίλυση ενός προβλήματος. Αντί να βασίζεται σε ένα μόνο δίκτυο, το MixDCNN αξιοποιεί K ανεξάρτητα CNNs, το καθένα εκ των οποίων παράγει μια ανεξάρτητη εκτίμηση της κατηγορίας μιας εικόνας. Σε αντίθεση με τα παραδοσιακά MoE, το MixDCNN δε διαθέτει ξεχωριστό gating network αλλά υπολογίζει δυναμικά την «ευθύνη» κάθε expert μέσω των ίδιων των outputs του, κάτι που λειτουργεί ως μηχανισμός implicit attention.

Κάθε expert λαμβάνει ως είσοδο την εικόνα x και εξάγει μια προβλεπόμενη κατανομή πιθανότητας $f_k(x) \in R^C$, όπου C ο αριθμός των κλάσεων. Στη συνέχεια, για κάθε expert υπολογίζεται ένα *confidence score* το οποίο αναπαριστά το πόσο βέβαιος είναι ο εκάστοτε expert για την πρόβλεψή του. Ορίζεται ως:

$$c_k(x) = \max\left(\text{softmax}(f_k(x))\right) \tag{3.1}$$

Το confidence αυτός μετατρέπεται σε ένα κανονικοποιημένο βάρος μέσω της συνάρτησης softmax:

$$\alpha_k(x) = \frac{\exp(c_k(x))}{\sum_{j=1}^{K} \exp(c_j(x))} \tag{3.2}$$

Έτσι, οι experts δεν συνδυάζονται ισομερώς αλλά με δυναμικό τρόπο, ανάλογα με το confidence του καθενός. Η τελική εκτίμηση της αρχιτεκτονικής δίνεται από το σταθμισμένο άθροισμα των logits των experts:

$$f(x) = \sum_{k=1}^{K} \alpha_k(x) f_k(x) \tag{3.3}$$

Η τελική πρόβλεψη κατηγορίας προκύπτει ως:

$$\hat{y} = \arg\max_{c} f_c(x) \tag{3.4}$$

Η λειτουργία του μηχανισμού $\alpha_k(x)$ αποτελεί την ουσία του MixDCNN. Αν ένα input είναι πιο «καθαρό» ή εύκολο για κάποιον expert, τότε το αντίστοιχο confidence θα είναι υψηλό και το βάρος α_k θα κυριαρχήσει στο τελικό άθροισμα. Αντίθετα, σε περιπτώσεις αβεβαιότητας, τα βάρη θα είναι πιο ισομοιρασμένα, επιτρέποντας σε πολλαπλούς experts να συνεισφέρουν από κοινού στην πρόβλεψη.

Αξιοσημείωτο είναι ότι ο μηχανισμός αυτός δε χρειάζεται ρητή εκπαίδευση των experts για συγκεκριμένα υποσύνολα δεδομένων, αλλά οι ίδιοι οι experts εξειδικεύονται σταδιακά μέσα από το training λόγω των confidence weights.

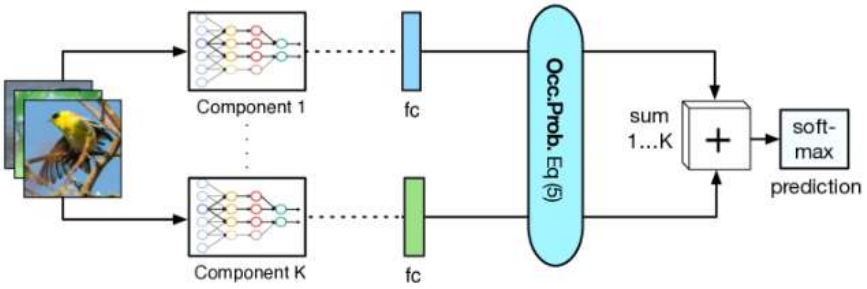
Στην εκπαίδευση του MixDCNN εφαρμόζεται συνήθως μια συνδυαστική συνάρτηση κόστους η οποία περιλαμβάνει την παραδοσιακή Cross-Entropy Loss, ενώ συχνά επιπλέον χρησιμοποιείται και η Focal Loss [36] για την ενίσχυση της εκπαίδευσης σε δύσκολα παραδείγματα, με την τελική συνάρτηση κόστους να διαμορφώνεται συνδυαστικά ως:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{Focal} \tag{3.5}$$

όπου λ ρυθμίζει τη σημασία της Focal Loss.

Το MixDCNN προσφέρει μια σειρά από πλεονεκτήματα στο FGVC. Πρώτον, επιτρέπει στους experts να εξειδικεύονται αυτόματα σε διαφορετικές περιοχές του feature space χωρίς ανάγκη ρητής εκπαίδευσης για συγκεκριμένες ομάδες δεδομένων. Δεύτερον, η τελική απόφαση προκύπτει ως δυναμικός συνδυασμός γνώσεων πολλαπλών experts και όχι ως απόφαση ενός μονολιθικού δικτύου. Επιπλέον, σε αντίθεση με τα παραδοσιακά MoE, το MixDCNN είναι πλήρως διαφορίσιμο, γεγονός που διευκολύνει την εκπαίδευση του μέσω backpropagation.

Για να βελτιωθεί η σταθερότητα της εκπαίδευσης, στην πράξη εφαρμόζεται συχνά τεχνική progressive fine-tuning, κατά την οποία αρχικά εκπαιδεύονται μόνο τα ανώτερα επίπεδα των CNNs και στη συνέχεια ξεπαγώνονται σταδιακά και τα χαμηλότερα layers. Αυτή η στρατηγική επιτρέπει στα experts να αξιοποιήσουν καλύτερα τις αρχικές τους προεκπαιδευμένες παραστάσεις, αποφεύγοντας φαινόμενα καταστροφικής λήθης.



Εικόνα 27: Διάγραμμα αρχιτεκτονικής MixDCNN[58]

3.2.2 CN-CNN

Η εργασία των Guo et al. [60] εισάγει το Cross-Layer Navigation Convolutional Neural Network (CN-CNN), μια αρχιτεκτονική σχεδιασμένη ειδικά για προβλήματα Fine-Grained Visual Classification. Η ιδιαίτερη δυσκολία αυτού του προβλήματος έγκειται στο γεγονός ότι οι κατηγορίες των εικόνων διαφέρουν οπτικά σε μικρό βαθμό, ενώ η ποικιλία εντός της ίδιας κατηγορίας μπορεί να είναι αρκετά μεγάλη. Παραδοσιακά CNNs, όπως τα ResNet ή VGG, αντιμετωπίζουν περιορισμούς σε τέτοια σενάρια, καθώς τα χαρακτηριστικά που παράγουν επικεντρώνονται περισσότερο σε γενικά global patterns και λιγότερο σε τοπικά λεπτομερή χαρακτηριστικά.

Η βασική συνεισφορά του CN-CNN είναι η εισαγωγή του Cross-Layer Navigation Module (CLNM). Ο ρόλος του CLNM είναι να συνδυάσει προσαρμοστικά χαρακτηριστικά από διαφορετικά επίπεδα του backbone δικτύου (ResNet-50), αξιοποιώντας πληροφορία τόσο από χαμηλόβαθμα όσο και από υψηλόβαθμα layers. Συγκεκριμένα, το CN-CNN λαμβάνει τα χαρακτηριστικά από τα layers 2, 3 και 4 του ResNet-50, τα οποία αντιστοιχούν σε διαφορετικά επίπεδα αφαίρεσης: τα χαμηλότερα layers περιέχουν λεπτομέρειες, τα μεσαία τοπικές μορφές και τα ανώτερα global context.

$$X_2 = f_2(x), \quad X_3 = f_3(x), \quad X_4 = f_4(x) \tag{3.6}$$

Όπου X_i είναι τα feature maps από τα αντίστοιχα επίπεδα. Η συγχώνευση αυτών των χαρακτηριστικών γίνεται μέσω ενός μηχανισμού ο οποίος πρώτα φέρνει όλα τα feature maps στο ίδιο πλήθος διαστάσεων χρησιμοποιώντας συναρτήσεις 1×1 convolution:

$$F_i = W_i * X_i, \quad i \in \{2, 3, 4\} \tag{3.7}$$

Τα προκύπτοντα feature maps F_2, F_3 και F_4 προσαρμόζονται χωρικά ώστε να έχουν κοινή ανάλυση μέσω bilinear interpolation. Στη συνέχεια, το CLNM εφαρμόζει learnable βάρη $\alpha_2, \alpha_3, \alpha_4$ ώστε το δίκτυο να αποφασίσει ποια επίπεδα θα έχουν μεγαλύτερη συμμετοχή στην τελική αναπαράσταση. Το σύνολο αυτό διαμορφώνει το fused feature:

$$F = \alpha_2 F_2 + \alpha_3 F_3 + \alpha_4 F_4 \tag{3.8}$$

Αξίζει να σημειωθεί ότι τα α_i είναι παράμετροι που μαθαίνονται κατά τη διάρκεια της εκπαίδευσης και προσαρμόζονται δυναμικά ανάλογα με το dataset. Για παράδειγμα, σε ένα πρόβλημα όπου οι κατηγορίες διαφέρουν κυρίως σε τοπικές υφές (π.χ., πτηνά ή προϊόντα με μικρά σχέδια), το δίκτυο θα δώσει μεγαλύτερη έμφαση στο F_2 που περιέχει fine-grained χαρακτηριστικά.

Μετά την συγχώνευση, εφαρμόζεται ένας ελαφρύς μηχανισμός προσοχής (attention) ο οποίος εκτιμά ένα channel-wise attention map:

$$A = \sigma(W_2 \cdot ReLU(W_1 * F)) \tag{3.9}$$

όπου σ είναι η σιγμοειδής συνάρτηση και W_1, W_2 είναι βάρη ενός μικρού convolutional block. Το attention map ενισχύει περαιτέρω τα σημαντικά κανάλια της πληροφορίας, επιτρέποντας στο δίκτυο να εστιάσει στα πιο διακριτά χαρακτηριστικά.

Το χαρακτηριστικό αυτό, αφού κανονικοποιηθεί μέσω Global Average Pooling και L2-normalization, εισέρχεται σε έναν Cosine Classifier [39], ο οποίος δεν βασίζεται στην ευκλείδεια απόσταση αλλά στην γωνιακή απόσταση του feature vector σε σχέση με τα κέντρα των κατηγοριών. Ο classifier παράγει τα τελικά logits με τον εξής τύπο:

$$s_i = \gamma \cdot \cos(\theta_i) = \gamma \cdot \frac{W_i^T f}{\|W_i\| \cdot \|f\|} \tag{3.10}$$

όπου γ είναι ένας scaling παράγοντας, f είναι το τελικό embedding του δείγματος και W_i το βάρος της κατηγορίας i. Αυτή η στρατηγική επιβάλλει στο μοντέλο να μάθει χαρακτηριστικά με ισχυρότερη angular διαχωρισιμότητα, κάτι ιδιαίτερα κρίσιμο για fine-grained tasks.

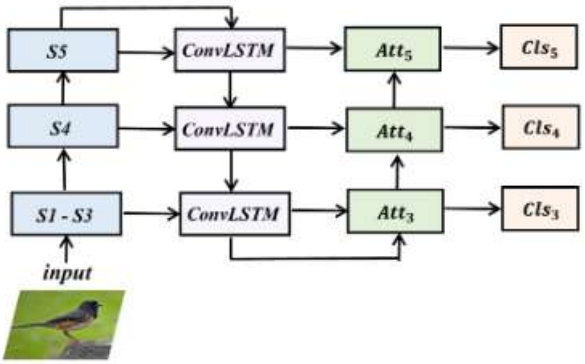
Η επιλογή του CosFace Loss ή άλλου margin-based loss ($L_{cosface}$) εξυπηρετεί στην αύξηση του intra-class compactness και inter-class separability, μειώνοντας την πιθανότητα τα embeddings διαφορετικών κατηγοριών να αλληλεπικαλύπτονται.

Καθ' όλη τη διάρκεια της εκπαίδευσης, το δίκτυο υιοθετεί την στρατηγική του progressive fine-tuning. Δηλαδή, αρχικά εκπαιδεύεται μόνο το CLNM και τα υψηλότερα επίπεδα του ResNet (layer4), ενώ σταδιακά (ανά 5-10 epochs) ξεκλειδώνονται και τα χαμηλότερα layers, επιτρέποντας στο δίκτυο να προσαρμοστεί σταδιακά στο νέο task χωρίς να χάσει το προϋπάρχον γενικό knowledge από το dataset.

Στην πλήρη εκδοχή του CN-CNN, οι συγγραφείς ενσωματώνουν ένα ConvLSTM block μετά το CLNM, με σκοπό την εκμετάλλευση χωρικών εξαρτήσεων στα τελικά feature maps, ιδιαίτερα σε περιπτώσεις video-based ή sequential δεδομένων. Αν και το στοιχείο αυτό δεν είναι απαραίτητο σε όλα τα σενάρια, προσφέρει σημαντική ενίσχυση σε περιπτώσεις όπου η πληροφορία παρουσιάζει χρονική ή εντοπισμένη αλληλουχία.

Η συγκεκριμένη αρχιτεκτονική αποδείχθηκε αποτελεσματική σε πολλά benchmarks του FGVC, όπως το CUB-200-2011 και το Stanford Dogs, παρουσιάζοντας σημαντική βελτίωση σε σύγκριση με τα baseline μοντέλα χωρίς cross-layer fusion ή attention.

Συνοψίζοντας, το CN-CNN βασίζεται στη θεμελιώδη παρατήρηση ότι η επιτυχής ταξινόμηση σε FGVC απαιτεί την ταυτόχρονη αξιοποίηση πληροφοριών σε πολλαπλές κλίμακες και επιπέδων αφαιρετικότητας. Μέσω του CLNM επιτυγχάνεται μια προσαρμοστική συγχώνευση που εστιάζει δυναμικά στα χαρακτηριστικά που έχουν πραγματική σημασία για το dataset. Η χρήση του Cosine Classifier επιβάλλει γεωμετρική πειθαρχία στο embedding space, βελτιώνοντας τη διαχωρισιμότητα μεταξύ παραπλήσιων κατηγοριών — κάτι που είναι κρίσιμο για FGVC.



Εικόνα 28: Διάγραμμα αρχιτεκτονικής CN-CNN με ConvLSTM και Attention[60]

3.2.3 HBP-CNN

Το Hierarchical Bilinear Pooling Convolutional Neural Network (HBP-CNN) εισήχθη από τους Yu et al. [61] ως μια ενισχυμένη εκδοχή του Bilinear CNN [62], ειδικά σχεδιασμένη για προβλήματα Fine-Grained Visual Classification (FGVC). Η βασική αρχή της αρχιτεκτονικής αυτής είναι ότι τα χαρακτηριστικά διαφορετικών επιπέδων ενός CNN περιέχουν συμπληρωματική πληροφορία: τα κατώτερα επίπεδα καταγράφουν λεπτομέρειες όπως υφές και άκρα, ενώ τα υψηλότερα εστιάζουν σε μορφές και αφηρημένες αναπαραστάσεις. Το HBP-CNN αξιοποιεί αυτήν τη διάρθρωση εφαρμόζοντας bilinear pooling σε πολλαπλά ενδιάμεσα επίπεδα, ενισχύοντας τη διακριτική ικανότητα του μοντέλου.

Η παραδοσιακή bilinear προσέγγιση εφαρμόζεται στο τελευταίο feature map και υπολογίζει εξωτερικό γινόμενο των τοπικών διανυσμάτων χαρακτηριστικών. Για είσοδο $F \in R^{h \times w \times d}$, ο bilinear operator δίνεται ως:

$$\phi(F) = \text{vec}\left(\frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w} F_{ij}F_{ij}^{\top}\right)$$

Αυτό το εξωτερικό γινόμενο καταγράφει τις συζευγμένες σχέσεις μεταξύ όλων των καναλιών και οδηγεί σε τελικό διάνυσμα διάστασης d^2, ιδιαίτερα εκφραστικό για τη σύλληψη fine-grained διαφορών. Όμως, εστιάζει αποκλειστικά στο τελευταίο επίπεδο, παραβλέποντας κρίσιμη πληροφορία που υπάρχει σε ενδιάμεσα στάδια.

Η HBP-CNN επεκτείνει αυτό το μοντέλο εφαρμόζοντας τον ίδιο μηχανισμό σε n διαφορετικά επίπεδα ενός CNN. Έστω ότι από κάθε επίπεδο i εξάγεται ένας πίνακας χαρακτηριστικών $F_i \in R^{h_i \times w_i \times d_i}$. Πριν από το bilinear pooling, εφαρμόζεται μια γραμμική απεικόνιση μέσω προβολής σε κοινό χώρο χαμηλότερης διάστασης r, ώστε να ελεγχθεί η έκρηξη παραμέτρων:

$$\tilde{F}_i = F_i W_i, \quad W_i \in R^{d_i \times r}$$

Έπειτα, υπολογίζεται η bilinear αναπαράσταση για κάθε επίπεδο:

$$B_i = \phi(\tilde{F}_i) = \text{vec}\left(\frac{1}{h_i w_i}\sum_{j=1}^{h_i}\sum_{k=1}^{w_i} \tilde{F}_{i,jk}\tilde{F}_{i,jk}^{\top}\right) \in R^{r^2}$$

Οι παραπάνω αναπαραστάσεις ενοποιούνται μέσω συνένωσης:

$$z = \text{Concat}(B_1, B_2, \ldots, B_n)$$

και κανονικοποιούνται με L2-norm:

$$\hat{z} = \frac{z}{\|z\|_2}$$

Η έξοδος $\hat{z}$ εισάγεται σε fully-connected layer και στη συνέχεια περνά από softmax για την τελική πρόβλεψη κατηγορίας.

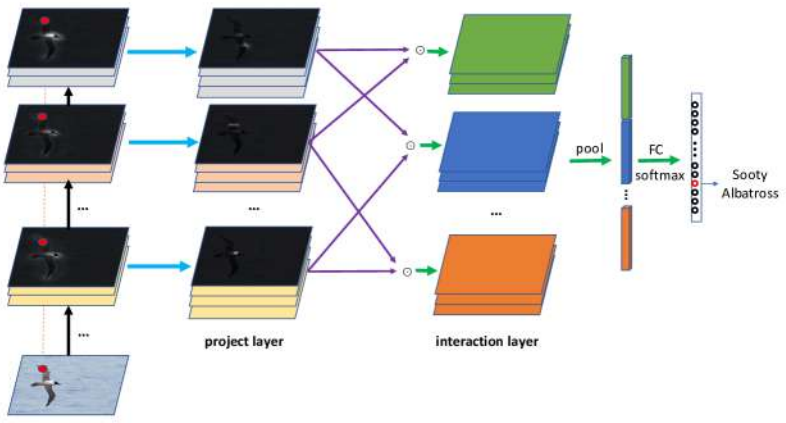
Η εκπαίδευση γίνεται end-to-end με Cross-Entropy Loss:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log \hat{y}_c$$

Συνήθως χρησιμοποιούνται προκαθορισμένα ενδιάμεσα επίπεδα, όπως τα `conv3_3`, `conv4_3` και `conv5_3` στην περίπτωση του VGG-16, ή τα output stages του ResNet. Η επιλογή αυτών των επιπέδων βασίζεται στην παρατήρηση ότι κάθε επίπεδο συνεισφέρει διαφορετικής ποιότητας πληροφορία στο τελικό representation. Επίσης, η χρήση πολλαπλών bilinear blocks προσφέρει πλουσιότερη αναπαράσταση χωρίς να βασίζεται σε μηχανισμούς προσοχής ή routing.

Η HBP-CNN ενδείκνυται για προβλήματα όπου η διαφοροποίηση μεταξύ κατηγοριών είναι εντοπισμένη σε τοπικά patterns — όπως χρώμα φτερού σε πτηνά ή λεπτομέρειες μπροστινού προφυλακτήρα σε οχήματα. Η πολυεπίπεδη σύζευξη δεύτερης τάξης βοηθά στο να «εντοπιστούν» αυτές οι διαφορές χωρίς ρητή εποπτεία.

Η κύρια πρόκληση του HBP είναι η υψηλή διαστατικότητα που προκύπτει λόγω των εξωτερικών γινομένων. Η τελική αναπαράσταση μπορεί να φτάσει εκατοντάδες χιλιάδες διαστάσεις, κάτι που απαιτεί επαρκή κανονικοποίηση και πιθανώς regularization ή dropout. Παρ' όλα αυτά, το πλεονέκτημα της μοντελοποίησης αλληλεπιδράσεων μεταξύ χαρακτηριστικών την καθιστά ιδιαίτερα αποτελεσματική — όπως φαίνεται από τα αποτελέσματά της σε datasets όπως CUB-200-2011, FGVC-Aircraft και Stanford Dogs.



Εικόνα 29: Διάγραμμα αρχιτεκτονικής HBP-CNN[61]

3.2.4 Θεωρητικές Διαφορές Μοντέλων

Τα μοντέλα MixDCNN, CN-CNN και HBP-CNN εκπροσωπούν τρεις διαφορετικές αρχιτεκτονικές στρατηγικές για την αντιμετώπιση του προβλήματος Fine-Grained Visual Classification. Παρότι ο τελικός στόχος είναι κοινός — η διακριτική ταξινόμηση παραπλήσιων κατηγοριών — κάθε προσέγγιση υλοποιεί ριζικά διαφορετική φιλοσοφία ως προς την εξαγωγή, συγχώνευση και αξιολόγηση χαρακτηριστικών.

Το MixDCNN στηρίζεται στην ιδέα της αρχιτεκτονικής πολλαπλών ανεξάρτητων experts, στους οποίους αποδίδεται δυναμικά «ευθύνη» βάσει confidence. Πρόκειται για ένα μοντέλο που βασίζεται στον καταμερισμό αρμοδιοτήτων, επιτρέποντας σε κάθε expert να ειδικευτεί σε συγκεκριμένες περιοχές του feature space χωρίς ρητή εποπτεία. Ο συνδυασμός των outputs είναι softmax-weighted, και η τελική αναπαράσταση προκύπτει από σταθμισμένη συνάθροιση logits.

Αντίθετα, το CN-CNN ακολουθεί μια ενδοδικτυακή προσέγγιση, ενοποιώντας πληροφορία από διαφορετικά επίπεδα αφαιρετικότητας εντός του ίδιου backbone. Ο συνδυασμός χαρακτηριστικών είναι προσαρμοστικός μέσω learnable συντελεστών και ενισχύεται περαιτέρω από channel-wise attention. Η τελική αναπαράσταση κανονικοποιείται γωνιακά, μέσω χρήσης Cosine Classifier, επιβάλλοντας ένα γεωμετρικά ευαίσθητο embedding space κατάλληλο για margin-based loss functions.

Το HBP-CNN, από την άλλη, προσεγγίζει το πρόβλημα από μαθηματική σκοπιά. Αντί να εφαρμόσει μηχανισμούς προσοχής ή gating, αξιοποιεί εξωτερικά γινόμενα για τη σύλληψη συζευγμένων συσχετίσεων μεταξύ χαρακτηριστικών. Το κυρίως στοιχείο διαφοροποίησης εδώ είναι η υιοθέτηση second-order pooling σε διαφορετικά ενδιάμεσα επίπεδα, κάτι που προσφέρει πολυεπίπεδη στατιστική ενίσχυση, χωρίς ρητή εξάρτηση από την κατηγορία ή το input. Η τελική αναπαράσταση έχει υψηλή διαστατικότητα, αλλά θεωρητικά ισχυρή εκφραστικότητα.

Ως προς τη γεωμετρία του χώρου προβολής, το CN-CNN υπερέχει χάρη στην χρήση angular-based projection, κάτι που του δίνει καλύτερο ενδοκατηγορικό συμπύκνωμα και διακατηγορική διάκριση. Το MixDCNN και το HBP-CNN λειτουργούν σε ευκλείδεια λογική, χωρίς να επιβάλλουν άμεση γεωμετρική δομή στο embedding space. Αυτό καθιστά το CN-CNN πιο κατάλληλο για περιβάλλοντα με ισχυρή κατηγορική εγγύτητα.

Στην πλευρά της εποπτείας, το MixDCNN πλησιάζει τη φιλοσοφία της ανεξάρτητης ειδίκευσης: δεν απαιτεί ρητή κατανομή των experts σε clusters ή tasks και βασίζεται στο confidence-based routing που αναδύεται κατά την εκπαίδευση. Το CN-CNN και το HBP-CNN εκπαιδεύονται end-to-end με shared loss και global supervision.

Ως προς την υπολογιστική πολυπλοκότητα, το MixDCNN είναι το πιο βαρύ, καθώς απαιτεί K πλήρεις αρχιτεκτονικές. Το HBP-CNN είναι επίσης απαιτητικό λόγω των εξωτερικών γινομένων και του μεγάλου μεγέθους των bilinear αναπαραστάσεων. Το CN-CNN εμφανίζεται πιο ελαφρύ και εύκολα υλοποιήσιμο, καθώς βασίζεται σε επεξεργασία υπάρχοντος backbone και απλές πράξεις συγχώνευσης.

Συνολικά, τα τρία μοντέλα αναδεικνύουν διαφορετικές όψεις του προβλήματος FGVC: εξειδίκευση μέσω ανεξάρτητων υποδικτύων (MixDCNN), αξιοποίηση πολυεπίπεδης πληροφορίας με metric-based προβολή (CN-CNN), και μαθηματικά δομημένη αναπαράσταση χαρακτηριστικών (HBP-CNN).

Κριτήριο	MixDCNN	CN-CNN	HBP-CNN
Αρχιτεκτονική Λογική	Πολλαπλοί ανεξάρτητοι experts με confidence-based ευθύνη	Συνδυασμός επιπέδων backbone με adaptive βάρη	Πολυεπίπεδη second-order αναπαράσταση μέσω εξωτερικών γινομένων
Συγχώνευση Πληροφορίας	Σταθμισμένο άθροισμα logits μέσω softmax βάσει confidence	Learnable fusion από ενδιάμεσα επίπεδα (CLNM)	Concatenation second-order αναπαραστάσεων
Projection Space	Ευκλείδειος (logit-level)	Γωνιακός (cosine embedding)	Ευκλείδειος second-order space
Εποπτεία / Εξειδίκευση	Implicit routing, χωρίς ρητή κατανομή των experts	Global supervision με attention	Global supervision χωρίς routing ή attention
Υπολογιστική Πολυπλοκότητα	Πολύ υψηλή (K πλήρη μοντέλα)	Μέτρια (μετατροπές εντός backbone)	Υψηλή (πολύ μεγάλα embeddings λόγω bilinear pooling)
Συμβατότητα με FGVC	Εξειδίκευση ανά expert σε δύσκολες υποκατηγορίες	Δυναμική εστίαση σε κλίμακες και γωνιακή διάκριση	Πλούσια εκφραστικότητα μέσω συζευγμένων χαρακτηριστικών

Πίνακας 5: Συγκριτική ανάλυση MixDCNN, CN-CNN και HBP-CNN

4 Πειραματική Αξιολόγηση και Αποτελέσματα

4.1 Διαμόρφωση Πειραμάτων

4.1.1 Προεπεξεργασία δεδομένων

Λόγω περιορισμένου αριθμού δειγμάτων ανά κατηγορία (ειδικά για μονάδες AC συγκεκριμένων τύπων), εφαρμόστηκε τεχνική εξισορρόπησης του dataset μέσω υπερδειγματοληψίας (oversampling). Για κάθε κλάση δημιουργήθηκε στόχος πλήθους δειγμάτων, και όπου κρίθηκε απαραίτητο, εφαρμόστηκε δειγματοληψία με επανάληψη για να φτάσουν όλες οι κατηγορίες σε παρόμοια ισχύ.

Επιπλέον, για τη βελτίωση της γενίκευσης των μοντέλων, εφαρμόστηκε ένα εκτεταμένο pipeline τεχνικών εμπλουτισμού (data augmentation), το οποίο περιλάμβανε:

- **Χρήση bounding boxes όπου ήταν διαθέσιμα:** με σκοπό την αποκοπή της άσχετης πληροφορίας φόντου και την εστίαση του μοντέλου στη σημαντική περιοχή του αντικειμένου (π.χ. ετικέτα, μονάδα AC).
- **Random Erasing:** εφαρμογή λευκής περιοχής σε τυχαίο σημείο της εικόνας ώστε να αυξηθεί η ανθεκτικότητα του μοντέλου σε τοπικά εμπόδια, εμπνευσμένο από regularization τεχνικές που μειώνουν την εξάρτηση από συγκεκριμένα pixel patterns.
- **Gaussian blur:** προσομοιώνει εικόνες με θόρυβο κίνησης ή εστίασης, κάτι που είναι συχνό σε datasets από κινητές κάμερες, ενισχύοντας τη γενίκευση σε υποβαθμισμένα δείγματα.

- **Παραμόρφωση φωτεινότητας και αντίθεσης:** προσαρμογές brightness και contrast μεταβάλλουν το οπτικό δυναμικό της εικόνας ώστε το δίκτυο να μη βασίζεται σε απόλυτες εντάσεις αλλά σε μορφολογικά μοτίβα.
- **Τυχαία παραμόρφωση προοπτικής (left/right, bottom):** επιτρέπει στο μοντέλο να γενικεύει σε εικόνες που δεν είναι ιδανικά ορθογωνισμένες, προσομοιώνοντας προβολές από μη-κανονικές γωνίες, όπως συμβαίνει σε πραγματικά σενάρια (π.χ. εικόνα από πλάγια, από κάτω).
- **Τυχαία περιστροφή:** προσφέρει ανοχή του μοντέλου σε μικρές γωνιακές αποκλίσεις ώστε να προσομοιάζει τις διαφορετικές γωνίες λήψης.

Όλες οι εικόνες επανακλιμακώθηκαν σε τετραγωνική μορφή 224×224 ή 448×448 pixels με padding (όπου απαιτείτο), ώστε να διατηρούνται οι αναλογίες χωρίς παραμόρφωση. Η επιλογή των μεγεθών έγινε ώστε να εξισορροπηθεί η απόδοση του μοντέλου με τους διαθέσιμους υπολογιστικούς πόρους, κυρίως σε σχέση με τον περιορισμό μνήμης GPU. Η κανονικοποίηση των καναλιών πραγματοποιήθηκε με βάση τις απαιτήσεις του εκάστοτε προεκπαιδευμένου backbone. Για παράδειγμα, σε ResNet και HBP-CNN χρησιμοποιήθηκαν οι τυπικές τιμές του ImageNet: μέσοι όροι (mean) $[0.485, 0.485, 0.485]$ και τυπικές αποκλίσεις (std) $[0.229, 0.229, 0.229]$, ενώ στο CoAtNet χρησιμοποιήθηκε απλοποιημένη κανονικοποίηση με $mean = std = 0.5$, σύμφωνα με τις οδηγίες του αρχικού implementation. Η επιλογή της κανονικοποίησης ήταν συνεπής με τις απαιτήσεις του αντίστοιχου pretrained weight space, ώστε να διατηρείται η μεταφερσιμότητα χαρακτηριστικών. Σημαντικό σημείο αποτελεί ότι οι εικόνες που χρησιμοποιήθηκαν μετατράπηκαν σε γκρίζες με σκοπό την απαλοιφή του χρώματος, κυρίως για να μην επηρεάζουν background ή φωτισμοί και γι αυτό το λόγο οι μέσοι όροι και τυπικές αποκλίσεις είναι ίδιες και στα τρία κανάλια.

Η κατανομή train/test δημιουργήθηκε με stratified sampling (80% - 20%), ώστε να διατηρείται αναλογία κατηγοριών και να διασφαλίζεται δίκαιη σύγκριση μεταξύ μοντέλων. Αυτό ίσχυε σε περιπτώσεις όπου είχαμε περισσότερες από μία εικόνα στην κλάση. Σε κλάσεις με μόνο μία εικόνα αυτή χρησιμοποιήθηκε με κατάλληλη προεπεξεργασία μόνο στο training set ώστε να μην υπάρξει overfitting από ίδιες εικόνες στην εκπαίδευση και επαλήθευση.

Συνολικά, οι τεχνικές εμπλουτισμού σχεδιάστηκαν ώστε να προσομοιώνουν ρεαλιστικές συνθήκες συλλογής εικόνων: ποικιλία φωτισμού, προοπτικής, θολούρας και μερικής απόκρυψης του αντικειμένου. Οι συγκεκριμένοι μετασχηματισμοί κρίθηκαν ιδιαίτερα σημαντικοί για fine-grained tasks, καθώς επιτρέπουν στο μοντέλο να εστιάζει σε δομικά χαρακτηριστικά του αντικειμένου και όχι σε τυχαίες μεταβολές, όπως προοπτικές αλλοιώσεις ή μετατοπίσεις υφής. Σε συνθήκες περιορισμένων δεδομένων, τέτοια augmentations συμβάλλουν στην καλύτερη μάθηση διακριτικών patterns χωρίς να εισάγουν θόρυβο στην κατηγορική πληροφορία.

4.1.2 Περιβάλλον Εκτέλεσης και Κοινές Ρυθμίσεις

Τα πειράματα υλοποιήθηκαν σε τοπικό σύστημα με χρήση της βιβλιοθήκης PyTorch και προεκπαιδευμένων μοντέλων μέσω της `timm`. Η εκπαίδευση των μοντέλων έγινε κατά περίπτωση είτε με σταθερό αριθμό εποχών είτε με διακοπή βάσει εμπειρικής αξιολόγησης της απόδοσης στο validation set. Δεν χρησιμοποιήθηκε αυστηρό early stopping, ωστόσο η επιλογή του τελικού checkpoint βασιζόταν στη βέλτιστη τιμή της `val_accuracy`.

Η αποθήκευση των μοντέλων γινόταν μέσω αρχείων τύπου `.pth`, τα οποία περιλάμβαναν το `state_dict` του μοντέλου, καθώς και τα εσωτερικά states του optimizer και scheduler, όπως και το τρέχον epoch και η καλύτερη επίδοση. Για κάθε epoch καταγραφόταν επίσης επιπλέον πληροφορία αξιολόγησης όπως ακρίβεια, f1-score, recall και χρόνος εκπαίδευσης/αξιολόγησης.

Η ανάπτυξη πραγματοποιήθηκε χωρίς χρήση εικονικού περιβάλλοντος ή Docker, με εγκατάσταση των βιβλιοθηκών απευθείας στο σύστημα μέσω `pip`.

Χαρακτηριστικό	**Τιμή**
Λειτουργικό σύστημα	Windows 10 (10.0.19045)
Κάρτα γραφικών	NVIDIA GeForce RTX 3060 (8GB VRAM)
Python version	3.12.3
PyTorch version	2.6.0 + cu118
Torchvision version	0.21.0 + cu118
Timm version	1.0.15
Pandas version	2.2.3
NumPy version	2.1.0
CUDA συσκευές	1 (CUDA διαθέσιμο)
Περιβάλλον εγκατάστασης	pip
Checkpointing	`model_state_dict`, optimizer, scheduler, epoch, validation accuracies
Logging ανά epoch	Loss, Accuracy, Precision, Recall, F1-score, Training και Validation time, χρήση RAM

Πίνακας 6: Περιβάλλον ανάπτυξης και κοινές ρυθμίσεις εκπαίδευσης

4.1.2.1 Κοινές Ρυθμίσεις

Όλα τα μοντέλα εκπαιδεύτηκαν end-to-end, με τυπικά βήματα κανονικοποίησης και χρήση augmentations κατά το training. Για τον βελτιστοποιητή χρησιμοποιήθηκε SGD με momentum 0.9 και weight decay. Ο ρυθμός μάθησης ρυθμιζόταν μέσω OneCycleLR scheduler, με warm-up στο 20% των εποχών και cosine annealing. Οι μετασχηματισμοί στο training περιλάμβαναν ColorJitter, Gaussian Blur, Random Rotation, Resize με Padding (στα 448×448 ή 224×224 ανάλογα το μοντέλο) και κανονικοποίηση. Στο test/validation set εφαρμοζόταν μόνο resize και normalization. Οι παρακάτω ρυθμίσεις αφορούν τον αρχικό έλεγχο των αλγορίθμων. Στη συνέχεια έγινε επέκταση και σε πιο εξεζητημένους ελέγχους στους καλύτερους αλγόριθμους όπως θα αναλυθούν στις επόμενες ενότητες.

4.1.2.2 CN-CNN

Το CN-CNN χρησιμοποίησε ResNet-50 ως backbone, με είσοδο 448×448 και ImageNet normalization. Το CLNM module εκπαιδεύεται ταυτόχρονα με το backbone, ενώ η τελική πρόβλεψη βασιζόταν σε Linear Classifier. Χρησιμοποιήθηκε συνδυασμός Cross-Entropy με label smoothing, Focal Loss και Triplet Loss με margin 0.3.

4.1.2.3 MixDCNN

Το MixDCNN σχεδιάστηκε με $K = 3$ ανεξάρτητους ResNet-based experts. Αρχικά εκπαιδεύονταν με shared warm-up, και στη συνέχεια ανεξάρτητα. Οι εξόδους συνδυάζονταν με softmax-weighted logits, με βάση confidence scores. Το loss function ήταν Focal Loss, χωρίς margin-based συνιστώσα.

4.1.2.4 HBP-CNN

Η αρχιτεκτονική HBP-CNN βασίστηκε σε ResNet-50. Εφαρμόστηκε bilinear pooling σε ενδιάμεσα επίπεδα του backbone, με προβολή σε χαμηλότερο embedding space. Η τελική έξοδος σχηματιζόταν μέσω concatenation των L2-normalized χαρακτηριστικών. Η εκπαίδευση έγινε με απλή Cross-Entropy Loss με label smoothing, Focal Loss ($\gamma = 2$, $\alpha = 0.25$) και Triplet Loss (margin $= 0.3$).

4.1.2.5 CN-CoAtNet

Το μοντέλο CN-CoAtNet βασίστηκε στο προεκπαιδευμένο coatnet_0_rw_224.sw_in1k, με είσοδο 224×224 και κανονικοποίηση mean $=$ std $= 0.5$. Το μοντέλο διαχωρίστηκε σε backbone, CLNM module και τελικό classifier. Χρησιμοποιήθηκαν διαφορετικά learning rates: 10^{-4} για το backbone, 10^{-3} για το CLNM και 10^{-3} για το classifier. Το loss function ήταν σύνθεση Cross-Entropy με label smoothing, Focal Loss ($\gamma = 2$, $\alpha = 0.25$) και Triplet Loss (margin $= 0.3$).

4.2 Πειραματική αξιολόγηση αρχιτεκτονικών μοντέλων

Αρχικά, εφαρμόστηκαν όλοι οι προαναφερθέντες αλγόριθμοι με στόχο την αξιολόγηση της απόδοσής τους. Μέσω αυτής της συγκριτικής ανάλυσης, εντοπίστηκε ο αλγόριθμος με την υψηλότερη απόδοση, ο οποίος και επιλέχθηκε ως βάση για περαιτέρω βελτιστοποίηση (fine-tuning) και πιο εξειδικευμένους πειραματικούς ελέγχους. Για την πλήρη αποτίμηση της απόδοσης των μοντέλων χρησιμοποιήθηκαν τέσσερις βασικές μετρικές, οι οποίες συνοψίζονται παρακάτω:

- **Accuracy**: Ποσοστό των συνολικά σωστών προβλέψεων (σωστά θετικά και σωστά αρνητικά) επί του συνόλου των δειγμάτων. Υπολογίζεται ως:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

  Όπου:

  - TP (True Positives): Σωστά θετικές προβλέψεις
  - TN (True Negatives): Σωστά αρνητικές προβλέψεις
  - FP (False Positives): Λανθασμένες θετικές προβλέψεις
  - FN (False Negatives): Λανθασμένες αρνητικές προβλέψεις

- **Precision**: Ποσοστό των σωστών προβλέψεων θετικής τάξης επί του συνόλου των προβλέψεων αυτής της τάξης:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

- **Recall**: Ποσοστό των σωστών προβλέψεων θετικής τάξης επί του συνόλου των πραγματικών δειγμάτων της τάξης:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

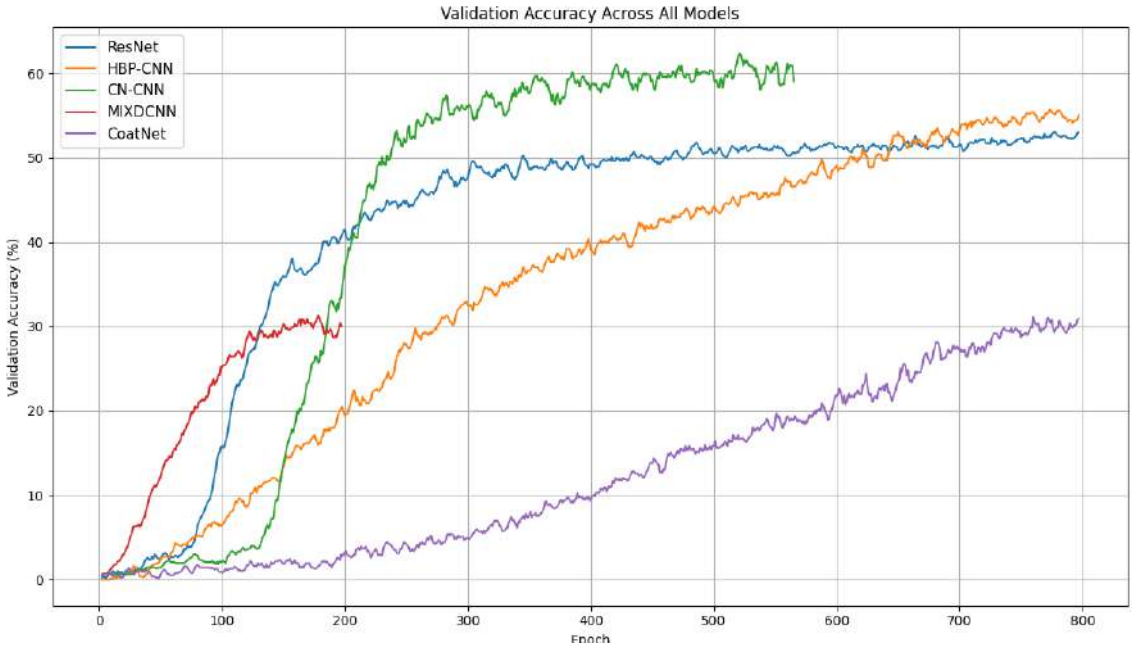
- **F1-score**: Ο αρμονικός μέσος του precision και του recall, που ισορροπεί ανάμεσα στις δύο μετρικές:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

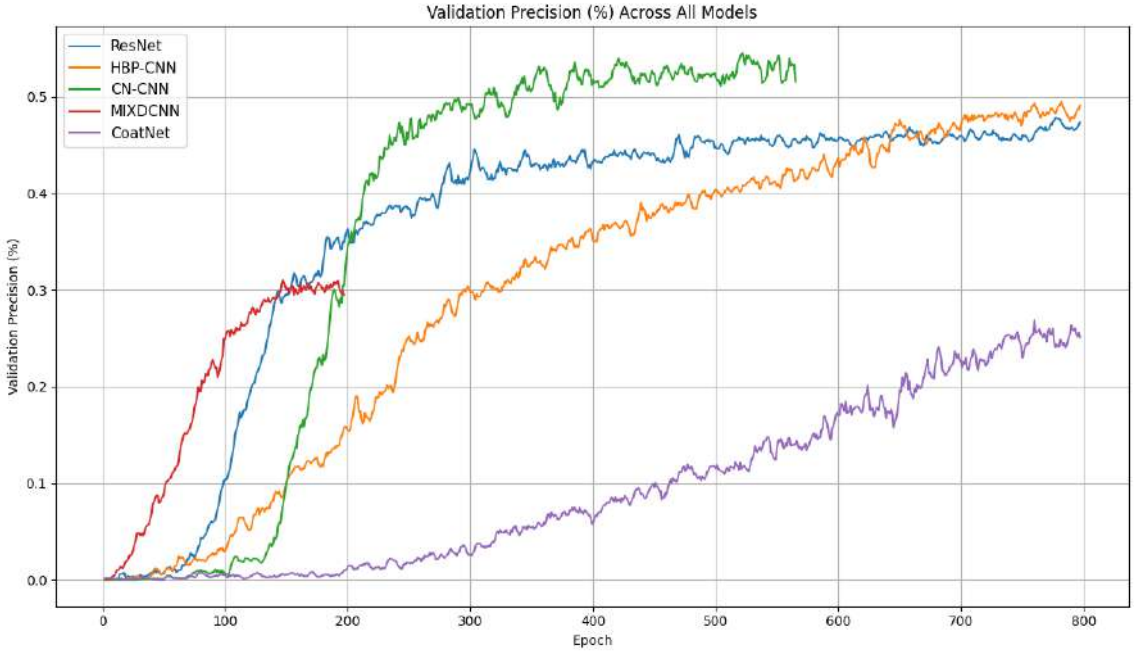
Σε προβλήματα δυαδικής ταξινόμησης με ισορροπημένες κατηγορίες και καλή συμπεριφορά σε όλες τις τάξεις, οι μετρικές accuracy και recall συχνά λαμβάνουν παρόμοιες τιμές. Ωστόσο, σε προβλήματα με πολλές ή άνισες κατηγορίες — όπως στην παρούσα περίπτωση — οι δύο μετρικές αποκλίνουν: η accuracy αποτυπώνει τη συνολική επίδοση, ενώ η recall εστιάζει στο ποσοστό σωστών εντοπισμών σε κάθε τάξη ξεχωριστά. Για το λόγο αυτό, το recall αποτελεί κρίσιμη μετρική σε προβλήματα fine-grained classification και ανισορροπίας κατηγοριών.

Όλα τα μοντέλα αξιολογήθηκαν στο ίδιο υποσύνολο δεδομένων επικύρωσης, διασφαλίζοντας συγκρισιμότητα αποτελεσμάτων.

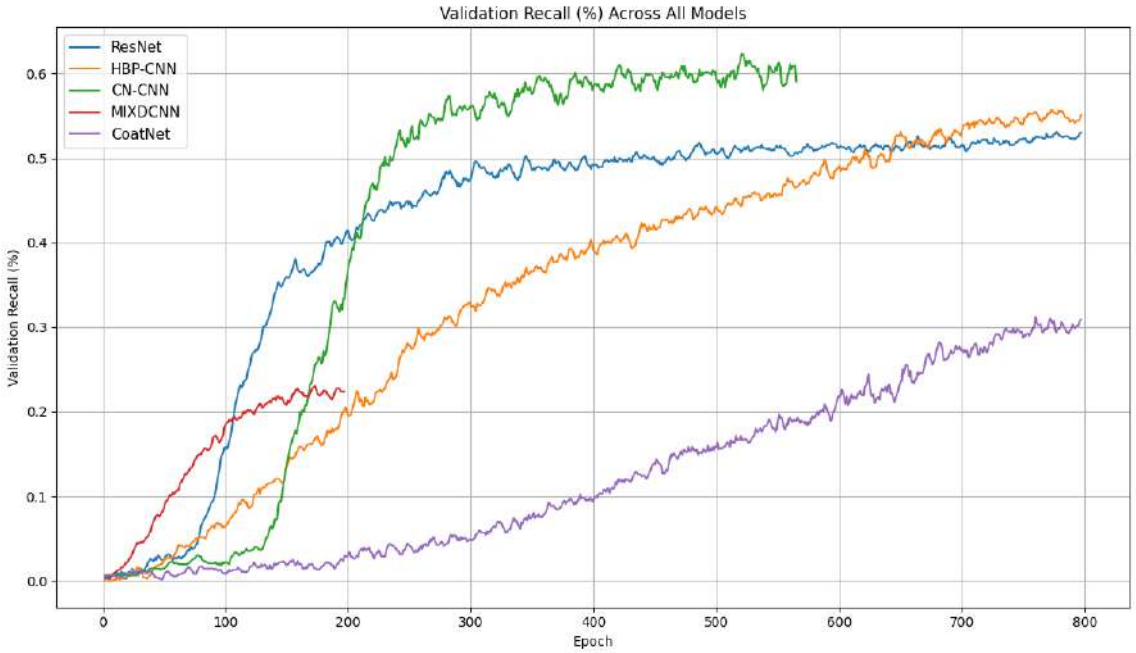
Τα αποτελέσματα της πειραματικής αξιολόγησης απεικονίζονται στα Σχήματα 30, 31 32 και 33. Στο πρώτο σχήμα καταγράφεται η πορεία της ακρίβειας επικύρωσης (validation accuracy) σε σχέση με τις εποχές εκπαίδευσης, ενώ στα επόμενα τρία παρουσιάζεται η μεταβολή του precision, recall και του F1-score αντίστοιχα. Οι τέσσερις αυτές καμπύλες προσφέρουν μία πληρέστερη εικόνα της απόδοσης κάθε αρχιτεκτονικής στο σύνολο των κατηγοριών, καθώς το dataset χαρακτηρίζεται από έντονη δυσμετρία και λεπτές διαφοροποιήσεις (fine-grained classification), όπου η ακρίβεια από μόνη της δεν επαρκεί για την αποτίμηση της ποιότητας του μοντέλου.



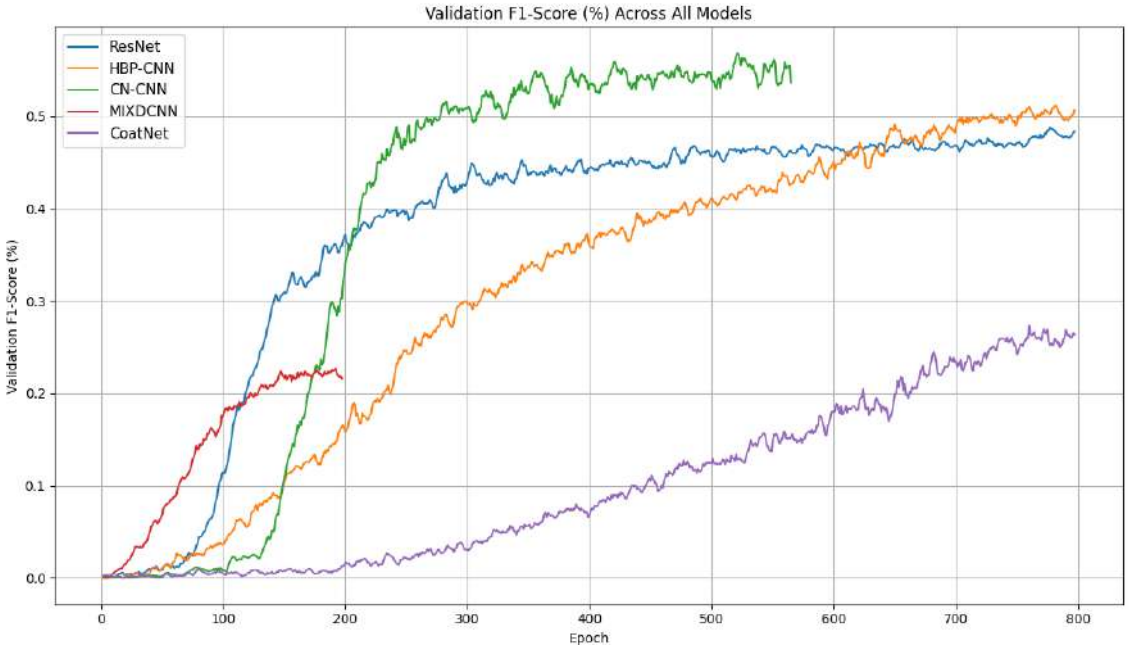
Εικόνα 30: Σύγκριση απόδοσης διαφορετικών CNN μοντέλων



Εικόνα 31: Σύγκριση precision διαφορετικών CNN μοντέλων



Εικόνα 32: Σύγκριση recall διαφορετικών CNN μοντέλων



Εικόνα 33: Σύγκριση f1-score διαφορετικών CNN μοντέλων

Για τη διευκόλυνση της σύγκρισης, ο Πίνακας 7 συνοψίζει τις καλύτερες επιδόσεις κάθε μοντέλου ανά μετρική, μαζί με την αντίστοιχη εποχή στην οποία επιτεύχθηκαν. Οι τιμές αυτές αποτελούν τη βάση για την ποιοτική ανάλυση που ακολουθεί ανά αρχιτεκτονική.

Μοντέλο	Val Accuracy (%)	Precision	Recall	F1-score
CN-CNN	**62.34**	**0.5455**	**0.6234**	**0.5683**
HBP-CNN	55.71	0.4950	0.5571	0.5120
ResNet-50	53.12	0.4783	0.5312	0.4878
CoAtNet	31.17	0.2690	0.3117	0.2732
MixDCNN	30.57	0.2872	0.3057	0.2230

Πίνακας 7: Συγκριτικά αποτελέσματα των αρχιτεκτονικών στο validation set

4.2.1 ResNet-50

Η αρχιτεκτονική ResNet χρησιμοποιήθηκε ως σημείο βάσης για τη σύγκριση των πιο εξειδικευμένων μοντέλων. Η καμπύλη ακρίβειας επικύρωσης σημειώνει ταχεία άνοδο κατά τις πρώτες 100–180 εποχές, φτάνοντας το 53.12%. Παράλληλα, κατέγραψε precision 0.4783, recall 0.5312 και F1-score 0.4878.

Το μοντέλο κατάφερε να εντοπίσει βασικά οπτικά μοτίβα και να αποκτήσει σταθερή γενικευτική ικανότητα χωρίς φαινόμενα υπερεκπαίδευσης. Το γεγονός αυτό αποδίδεται τόσο στις τεχνικές προεπεξεργασίας (όπως grayscale μετασχηματισμοί και aggressive augmentation), όσο και στη χρήση προεκπαιδευμένου backbone. Ωστόσο, η απόδοσή του σταθεροποιείται χωρίς περαιτέρω βελτίωση, κάτι που υποδεικνύει περιορισμένο εκφραστικό εύρος απέναντι σε πιο λεπτές ή αφαιρετικές διαφορές μεταξύ κατηγοριών.

Η μερική υπεροχή του recall έναντι της precision φανερώνει ότι το μοντέλο έχει την τάση να αναγνωρίζει περισσότερες πραγματικές περιπτώσεις (δηλαδή να ”βρίσκει” σωστά παραδείγματα), αλλά όχι πάντα με υψηλή εμπιστοσύνη. Το αποτέλεσμα είναι μια μέτρια F1-score, η οποία αποτυπώνει την έλλειψη ισχυρής διαχωρισιμότητας.

Η συνολική του συμπεριφορά καθιστά το ResNet-50 κατάλληλο ως baseline αρχιτεκτονική, χρήσιμη για τη συγκριτική αποτίμηση πιο εξειδικευμένων ή ενισχυμένων προσεγγίσεων. Παρά τη σταθερότητα και την ευρωστία του, δεν εμφανίζει τη δυναμική προσαρμογής που απαιτείται για προβλήματα fine-grained ταξινόμησης όπως το παρόν.

4.2.2 CN-CNN

Η αρχιτεκτονική CN-CNN κατέγραψε την υψηλότερη απόδοση από όλα τα μοντέλα που αξιολογήθηκαν, επιτυγχάνοντας 62.34% ακρίβεια επικύρωσης, precision 0.5455, recall 0.6234 και F1-score 0.5683. Η απόδοση αυτή υπερτερεί των υπολοίπων μοντέλων σε κάθε μετρική, γεγονός που αναδεικνύει την ικανότητα του μοντέλου να διαχειρίζεται τόσο εύκολες όσο και δύσκολες ταξινομήσεις.

Παρά την αρχικά βραδεία εκκίνηση — που σχετίζεται στην αρχική δυσκολία του προβλήματος — η καμπύλη ακρίβειας παρουσίασε έντονη και σταθερή αύξηση μεταξύ των 100 και 250 εποχών. Από την 500ή εποχή και έπειτα, το μοντέλο σταθεροποιήθηκε πάνω από το 60%, χωρίς ενδείξεις υπερεκπαίδευσης ή έντονων διακυμάνσεων, γεγονός που υποδηλώνει ισχυρή γενίκευση και ελεγχόμενη σύγκλιση.

Το υψηλό recall (0.6234) φανερώνει ότι το μοντέλο καταφέρνει να εντοπίζει την πλειονότητα των πραγματικών δειγμάτων όλων των κατηγοριών, ακόμα και σε δύσκολες ή λιγότερο αντιπροσωπευμένες κλάσεις. Ταυτόχρονα, η σημαντική τιμή precision (0.5455) δείχνει ότι οι προβλέψεις του είναι και αρκετά «σίγουρες», χωρίς πολλές ψευδώς θετικές ταξινομήσεις. Ο συνδυασμός αυτός αντανακλάται και στη σχετικά υψηλή F1-score, η οποία υποδεικνύει ισορροπία μεταξύ εύρους εντοπισμών και ακρίβειας.

Η συμπεριφορά αυτή είναι αποτέλεσμα της εσωτερικής δομής του μοντέλου. Συγκεκριμένα, το Cross-Layer Navigation Mechanism (CLNM)επιτρέπει τη δυναμική συγχώνευση πληροφορίας από πολλαπλά επίπεδα του backbone, αξιοποιώντας τόσο πρώιμα τοπικά χαρακτηριστικά όσο και αφηρημένες αναπαραστάσεις βαθύτερων στρωμάτων.

Η απουσία απότομων αστάθειων, η ομαλή σύγκλιση και η συνολικά υψηλή απόδοση σε όλες τις μετρικές καθιστούν το CN-CNN την πιο αξιόπιστη και ισορροπημένη επιλογή στο συγκεκριμένο πρόβλημα. Για τον λόγο αυτό επιλέγεται ως η κύρια αρχιτεκτονική προς περαιτέρω βελτιστοποίηση και fine-tuning στις επόμενες ενότητες.

4.2.3 MixDCNN

Η απόδοση του MixDCNN παρέμεινε περιορισμένη καθ’ όλη τη διάρκεια της εκπαίδευσης, με την ακρίβεια επικύρωσης να σταθεροποιείται σε 30.57%, την precision σε 0.2872, το recall σε 0.3057 και την F1-score σε μόλις 0.2230. Η F1-score, ως αυστηρή μετρική ισορροπίας μεταξύ precision και recall, υποδεικνύει ότι το μοντέλο έχει δυσκολία τόσο στην ακριβή όσο και στην πλήρη αναγνώριση των κατηγοριών.

Η πορεία του μοντέλου χαρακτηρίζεται από γρήγορη άνοδο μέχρι περίπου την 150ή εποχή, ακολουθούμενη από στασιμότητα χωρίς έντονα φαινόμενα υπερεκπαίδευσης. Το γεγονός αυτό δείχνει

ότι το μοντέλο δεν υπέστη κατάρρευση, αλλά απέτυχε να εξελίξει ένα συνεκτικό πλαίσιο ταξινόμησης ικανό να γενικεύσει πέρα από τις πρώτες εύκολες τάξεις. Η μορφή της καμπύλης δηλώνει ότι η μάθηση περιορίστηκε σε τοπικά πρότυπα χωρίς περαιτέρω κατανόηση της συνολικής δομής του dataset.

Η χαμηλή precision και recall υποδηλώνουν ότι το μοντέλο δεν είναι ούτε σίγουρο στις προβλέψεις του (χαμηλή precision), ούτε καταφέρνει να εντοπίσει αρκετές σωστές περιπτώσεις (χαμηλό recall), κάτι που αποτυπώνεται καθαρά στην ασθενή του F1-score. Το σχήμα εκπαίδευσης του MixDCNN, το οποίο στηρίζεται στη συνεργασία πολλών επιμέρους expert μοντέλων, προϋποθέτει την ύπαρξη διαχωρίσιμων clusters στο χαρακτηριστικό χώρο ώστε οι κλάδοι να αναπτύξουν εξειδίκευση.

Όπως θα αναλυθεί στην Ενότητα 4.3, ο χαρακτηριστικός χώρος του dataset παρουσιάζει σημαντική αλληλοεπικάλυψη μεταξύ των κατηγοριών, γεγονός που δεν επιτρέπει στο MixDCNN να εκμεταλλευτεί πλήρως την αρχιτεκτονική του λογική. Η αποτυχία αυτή δεν σχετίζεται με ελλιπή παραμετροποίηση ή ανεπαρκή εκπαίδευση, αλλά με τη δομική αναντιστοιχία του μοντέλου προς τη γεωμετρία των δεδομένων. Το αποτέλεσμα είναι μια σταθερή αλλά χαμηλής ποιότητας απόδοση, που δεν καθιστά το MixDCNN κατάλληλη επιλογή για το συγκεκριμένο πρόβλημα.

4.2.4 HBP-CNN

Η καμπύλη ακρίβειας επικύρωσης του HBP-CNN παρουσιάζει σταθερή και προοδευτική άνοδο καθ' όλη τη διάρκεια της εκπαίδευσης, φτάνοντας τελικά σε 55.71%. Το μοντέλο κατέγραψε επιπλέον precision 0.4950, recall 0.5571 και F1-score 0.5120, επιδόσεις σημαντικά υψηλότερες από το baseline και ενδεικτικές της εξελισσόμενης δυναμικής του.

Η αρχική φάση χαρακτηρίζεται από βραδεία πρόοδο με τιμές κάτω του 15% μέχρι και την 100ή εποχή. Ωστόσο, μετά τη φάση αυτή, το μοντέλο παρουσιάζει σχεδόν γραμμική βελτίωση χωρίς ενδείξεις κορεσμού ή υπερεκπαίδευσης, φτάνοντας σε υψηλές επιδόσεις στις τελευταίες εποχές της εκπαίδευσης.

Η αύξηση του recall σε σχέση με την precision υποδηλώνει ότι το HBP-CNN έχει την τάση να αναγνωρίζει περισσότερες πραγματικές περιπτώσεις από όσες προβλέπει με ακρίβεια — στοιχείο που φανερώνει αυξημένη «ευαισθησία» αλλά με κάποιο κόστος σε σιγουριά. Η F1-score κινείται σε αρκετά ισορροπημένα επίπεδα, αντανακλώντας τη σχετική επάρκεια του μοντέλου στην αντιμετώπιση τόσο της ανάκλησης όσο και της ακρίβειας των προβλέψεων.

Η μορφή της καμπύλης και η συνολική συμπεριφορά του μοντέλου δείχνουν ότι η αρχιτεκτονική αξιοποιεί σταδιακά τον πολυδιάστατο χώρο χαρακτηριστικών που δημιουργείται μέσω της τεχνικής bilinear pooling. Το γεγονός ότι η δυναμική της μάθησης δεν φθίνει στο πέρας των εποχών υποδηλώνει ότι υπάρχουν ακόμη περιθώρια βελτίωσης, είτε με μεγαλύτερο πλήθος εποχών είτε με πιο επιθετικό fine-tuning. Το HBP-CNN αποδεικνύεται συνεπώς ως μια σταθερή και αξιόπιστη προσέγγιση, με ισχυρή συμπεριφορά σε βάθος εκπαίδευσης.

4.2.5 CN-CoAtNet

Το CN-CoAtNet παρουσίασε αργή αλλά σταθερή βελτίωση κατά την εκπαίδευση, φτάνοντας τελικά 31.17% ακρίβεια επικύρωσης, με precision 0.2690, recall 0.3117 και F1-score 0.2732. Αν και οι απόλυτες επιδόσεις παραμένουν περιορισμένες, η συνολική καμπύλη δείχνει συνεχή πρόοδο χωρίς σημάδια υπερεκπαίδευσης ή ταλάντωσης, ακόμη και στις τελευταίες εποχές.

Το recall υπερβαίνει την precision, υποδηλώνοντας ότι το μοντέλο εντοπίζει ένα σημαντικό πο-

σοστό από τις πραγματικές περιπτώσεις, αλλά η χαμηλή precision δείχνει επίσης υψηλό ποσοστό λανθασμένων θετικών προβλέψεων. Η σχετικά χαμηλή F1-score επιβεβαιώνει τη δυσκολία του CN-CoAtNet να ισορροπήσει μεταξύ πληρότητας και ακρίβειας, κάτι που ερμηνεύεται ως αδυναμία στη διαχείριση λεπτών διαφορών μεταξύ κατηγοριών — θεμελιώδης πρόκληση σε fine-grained ταξινόμηση.

Η συμπεριφορά αυτή είναι αναμενόμενη για αρχιτεκτονικές που ενσωματώνουν attention μηχανισμούς, οι οποίοι τείνουν να απαιτούν μεγάλο πλήθος δεδομένων ώστε να μπορέσουν να μάθουν επαρκώς τα πρότυπα συσχέτισης. Η δομή του CoAtNet, η οποία συνδυάζει convolutional και transformer-βασισμένα blocks, έχει σημαντικό θεωρητικό δυναμικό, αλλά η αποτελεσματική αξιοποίησή του απαιτεί μεγαλύτερες βάσεις και αυξημένο εύρος παραδειγμάτων.

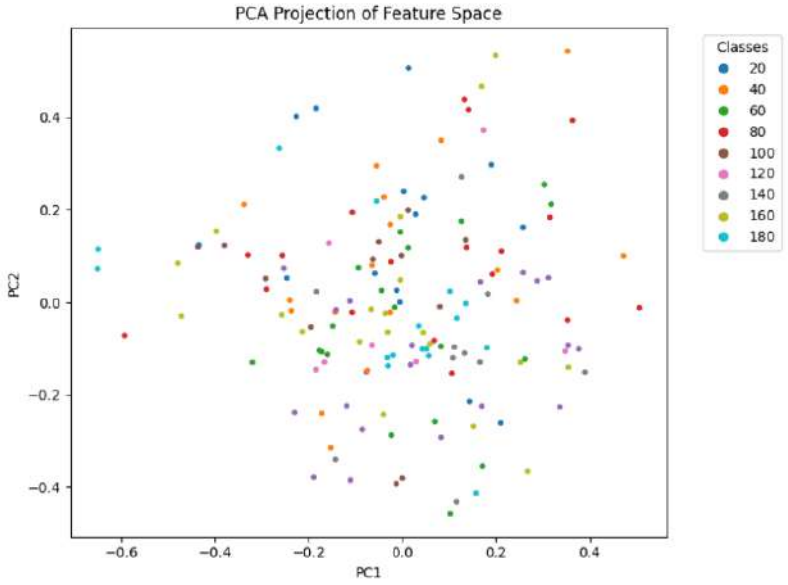
Παρά τους περιορισμούς, το CN-CoAtNet παρουσιάζει θετική κατεύθυνση και σταθερή ανοδική τροχιά. Η απουσία υπερεκπαίδευσης σε συνδυασμό με τη σταθερή πρόοδο υποδεικνύει ότι, με περαιτέρω ρύθμιση και αύξηση του όγκου δεδομένων, η αρχιτεκτονική αυτή θα μπορούσε να αποδώσει σημαντικά καλύτερα. Ως εκ τούτου, κρίνεται ενδιαφέρουσα ως εναλλακτική για μελλοντικά πειράματα, ιδίως σε σενάρια με πλουσιότερο training set.

4.3 Οπτική Ανάλυση του Χώρου Χαρακτηριστικών

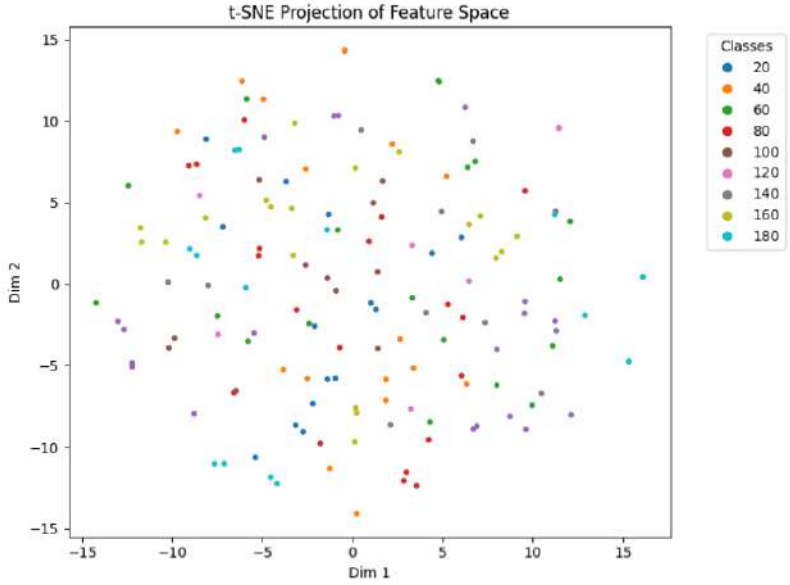
Για την περαιτέρω ερμηνεία της αποδοτικότητας των μοντέλων και ειδικότερα των ορίων στην απόδοση του MIXDCNN, πραγματοποιήθηκε οπτική ανάλυση του χώρου χαρακτηριστικών (feature space) που εξάγεται από το CN-CNN, λίγο πριν το τελικό επίπεδο ταξινόμησης. Οι προβολές των χαρακτηριστικών σε δύο διαστάσεις μέσω Principal Component Analysis (PCA) και t-SNE παρουσιάζονται στα Σχήματα 34 και 35 αντίστοιχα.

Από την επιθεώρηση των προβολών προκύπτει ότι τα δεδομένα δεν συγκροτούν πλήρως διαχωρίσιμα υποσύνολα (clusters) στο embedding space. Παρά την παρουσία ορισμένων τοπικών περιοχών συγκέντρωσης, παρατηρείται σημαντικός βαθμός αλληλοεπικάλυψης μεταξύ κατηγοριών. Το φαινόμενο αυτό είναι χαρακτηριστικό σε προβλήματα fine-grained ταξινόμησης, όπου οι διαφορές μεταξύ κατηγοριών εντοπίζονται σε λεπτομερή τοπικά χαρακτηριστικά.

Η μορφή αυτή του embedding space εξηγεί σε σημαντικό βαθμό την περιορισμένη απόδοση του MIXDCNN: οι expert κλάδοι του μοντέλου, οι οποίοι βασίζονται στην ύπαρξη καθαρών, διαχωρίσιμων περιοχών ώστε να αναπτύξουν εξειδίκευση, δεν μπορούν να εκμεταλλευτούν επαρκώς τη διαθέσιμη πληροφορία. Ως αποτέλεσμα, το μοντέλο επιτυγχάνει μεν σταθερή, αλλά περιορισμένη απόδοση (30%), χωρίς να μπορεί να ανταγωνιστεί τα κορυφαία μοντέλα της σύγκρισης. Η ανάλυση αυτή ενισχύει την ερμηνεία ότι η αρχιτεκτονική του MIXDCNN, παρά την εσωτερική της δυναμική, δεν ταιριάζει ιδανικά στη γεωμετρία του συγκεκριμένου dataset.



Εικόνα 34: Προβολή του χώρου χαρακτηριστικών μέσω PCA



Εικόνα 35: Προβολή του χώρου χαρακτηριστικών με t-SNE

4.4 Υπολογιστικό Κόστος Μοντέλων

Κατά την αξιολόγηση των μοντέλων, καταγράφηκαν ο μέσος χρόνος εκπαίδευσης και επικύρωσης καθώς και η χρήση μνήμης GPU για κάθε αρχιτεκτονική. Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

Μοντέλο	Μέσος Χρόνος Εκπαίδευσης (δευτ.)	Μέσος Χρόνος Επικύρωσης (δευτ.)	Μέση Χρήση GPU (MB)
ResNet	67.84	15.53	7665
HBPCNN	80.41	15.62	9324
CN-CNN	**150.71**	**17.05**	**10970**
MixDCNN	139.52	9.72	5539
CoAtNet	56.60	15.43	4802

Πίνακας 8: Σύγκριση Υπολογιστικού Κόστους Μοντέλων

Όπως φαίνεται, τα διαφορετικά μοντέλα παρουσιάζουν σημαντική διακύμανση ως προς το υπολογιστικό κόστος. Το CN-CNN και το MixDCNN εμφάνισαν τους υψηλότερους χρόνους εκπαίδευσης και χρήση GPU, αλλά για διαφορετικούς λόγους.

Το CN-CNN διαθέτει εγγενώς πιο σύνθετη αρχιτεκτονική, περιλαμβάνοντας cross-normalization και καναλιακά blocks, γεγονός που αυξάνει την υπολογιστική πολυπλοκότητα και το βάθος του μοντέλου. Αντίθετα, το MixDCNN βασίζεται στη συνδυαστική προσέγγιση τριών ανεξάρτητων CNN μοντέλων (experts), τα οποία εκπαιδεύονται και αξιολογούνται παράλληλα. Παρά το γεγονός ότι κάθε expert είναι σχετικά απλός, η ταυτόχρονη λειτουργία τους πολλαπλασιάζει το συνολικό υπολογιστικό κόστος.

Το CoAtNet εμφάνισε την πιο αποδοτική συμπεριφορά σε χρόνο και κατανάλωση GPU. Αυτό εξηγείται από τη στοχευμένη επιλογή ελαφριάς παραλλαγής του μοντέλου και τη χρήση μικρότερης ανάλυσης εικόνων. Η απόφαση αυτή ελήφθη με σκοπό την αποφυγή υπερπροσαρμογής, καθώς τα transformer-based μοντέλα, όπως το CoAtNet, είναι ιδιαίτερα ευαίσθητα σε μικρά datasets και απαιτούν ρύθμιση ώστε να διατηρείται η γενίκευση. Για αυτό τον λόγο περιορίστηκε η πολυπλοκότητα του συγκεκριμένου μοντέλου.

Συνολικά, οι διαφορές στους χρόνους επικύρωσης είναι μικρότερες σε σχέση με την εκπαίδευση, γεγονός που υποδηλώνει πως το επιπλέον βάρος αφορά κυρίως την αρχιτεκτονική και τον τρόπο μάθησης, και όχι την απλή αξιολόγηση.

4.5 Ανάλυση Υπεροχής του CN-CNN

Η αρχιτεκτονική CN-CNN κατέγραψε συνολικά την υψηλότερη απόδοση μεταξύ των μοντέλων που αξιολογήθηκαν. Η υπεροχή αυτή δεν περιορίζεται σε μια μόνο μετρική, αλλά παρατηρείται σταθερά στην ακρίβεια, την ανάκληση (recall) και τη F1-score, γεγονός που υποδεικνύει ισορροπημένη και αξιόπιστη συμπεριφορά κατά την εκπαίδευση και επικύρωση.

Παρά το γεγονός ότι το CN-CNN εμφανίζει το μεγαλύτερο υπολογιστικό κόστος, τόσο σε χρόνο εκπαίδευσης όσο και σε χρήση GPU, η σημαντική διαφορά στην απόδοση το καθιστά εύλογα προτιμητέο. Η αύξηση της ακρίβειας ξεπερνά τις 10 ποσοστιαίες μονάδες σε σχέση με το δεύτερο καλύτερο μοντέλο, ενώ η αντίστοιχη υπεροχή σε F1-score ενισχύει την εικόνα ενός μοντέλου με ισχυρή διαχωρισιμότητα και ικανότητα γενίκευσης, ιδίως σε συνθήκες περιορισμένων και λεπτομερώς διαφοροποιημένων κατηγοριών.

Η ακρίβεια αυτή φαίνεται να οφείλεται στην ενίσχυση της αναπαραστατικής ικανότητας του δικτύου, όπως επιτυγχάνεται μέσω του μηχανισμού CLN, ο οποίος επιτρέπει την ταυτόχρονη αξιοποίηση πληροφορίας από διαφορετικά βάθη, καθώς και της Attention Pyramid, που επικεντρώνει την προσοχή του μοντέλου σε τοπικά διακριτικά μοτίβα. Ο συνδυασμός των δύο οδηγεί σε πιο

εκλεπτυσμένες και πλήρεις αναπαραστάσεις, που εξηγούν την υπεροχή του μοντέλου ακόμη και σε περιπτώσεις οπτικά συγκεχυμένων κατηγοριών.

Η σταθερότητα της εκμάθησης, η απουσία φαινομένων υπερεκπαίδευσης, και η συνέπεια των αποτελεσμάτων μεταξύ εποχών και μετρικών, υποδεικνύουν ότι το CN-CNN αξιοποιεί αποτελεσματικά την πολυπλοκότητά του. Σε αντίθεση με άλλες αρχιτεκτονικές που παρουσίασαν περιορισμένη προσαρμοστικότητα ή ασυνεπή συμπεριφορά, το συγκεκριμένο μοντέλο κατέγραψε συνεχή πρόοδο χωρίς να εξαντλήσει τη δυναμική του.

Συνολικά, η επιλογή του CN-CNN δικαιολογείται όχι μόνο από την υψηλή του επίδοση, αλλά και από τη συνολική ισορροπία μεταξύ απόδοσης, σταθερότητας και προσαρμοστικότητας. Τα χαρακτηριστικά αυτά το καθιστούν ιδιαίτερα κατάλληλο για προβλήματα fine-grained ταξινόμησης όπως το παρόν.

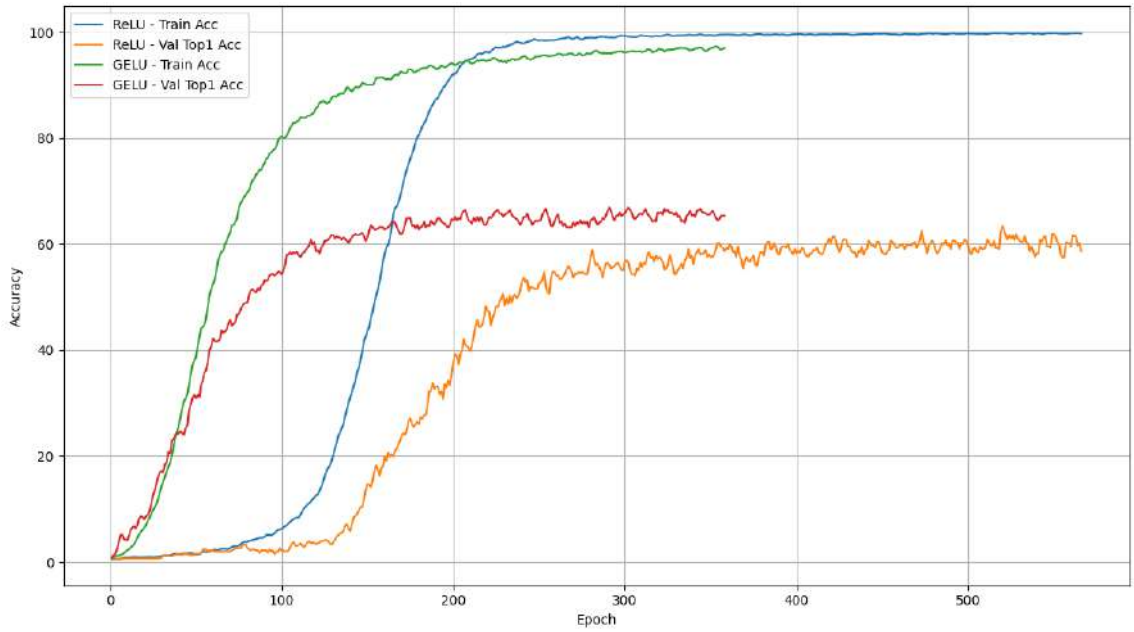
Λαμβάνοντας υπόψη τα παραπάνω, το CN-CNN αποτελεί την πλέον κατάλληλη βάση για περαιτέρω βελτιστοποίηση. Στην επόμενη ενότητα, διερευνώνται τεχνικές fine-tuning και αξιολογούνται επιπλέον μετρικές όπως η Top-3 Accuracy, με στόχο τη μεγιστοποίηση της πρακτικής αξιοπιστίας του μοντέλου.

4.6 Πειραματικές Διαμορφώσεις του CN-CNN

Αφετηρία των πειραματικών επεκτάσεων αποτέλεσε η βασική αρχιτεκτονική του CN-CNN, η οποία είχε ήδη επιδείξει υψηλή απόδοση στο πρόβλημα ταξινόμησης μοντέλων κλιματιστικών. Για τη διερεύνηση των δυνατοτήτων περαιτέρω βελτιστοποίησης, ακολουθήθηκε μια σειρά από παραλλαγές που κάλυπταν διαφορετικές όψεις του μοντέλου: αρχιτεκτονικές τροποποιήσεις, μεταβολές loss functions, προγραμματισμό learning rate και τεχνικές fine-tuning. Κάθε παραλλαγή σχεδιάστηκε με συγκεκριμένη υπόθεση ή προσδοκία, βασισμένη στη βιβλιογραφία ή στην εμπειρική συμπεριφορά του μοντέλου έως εκείνο το σημείο.

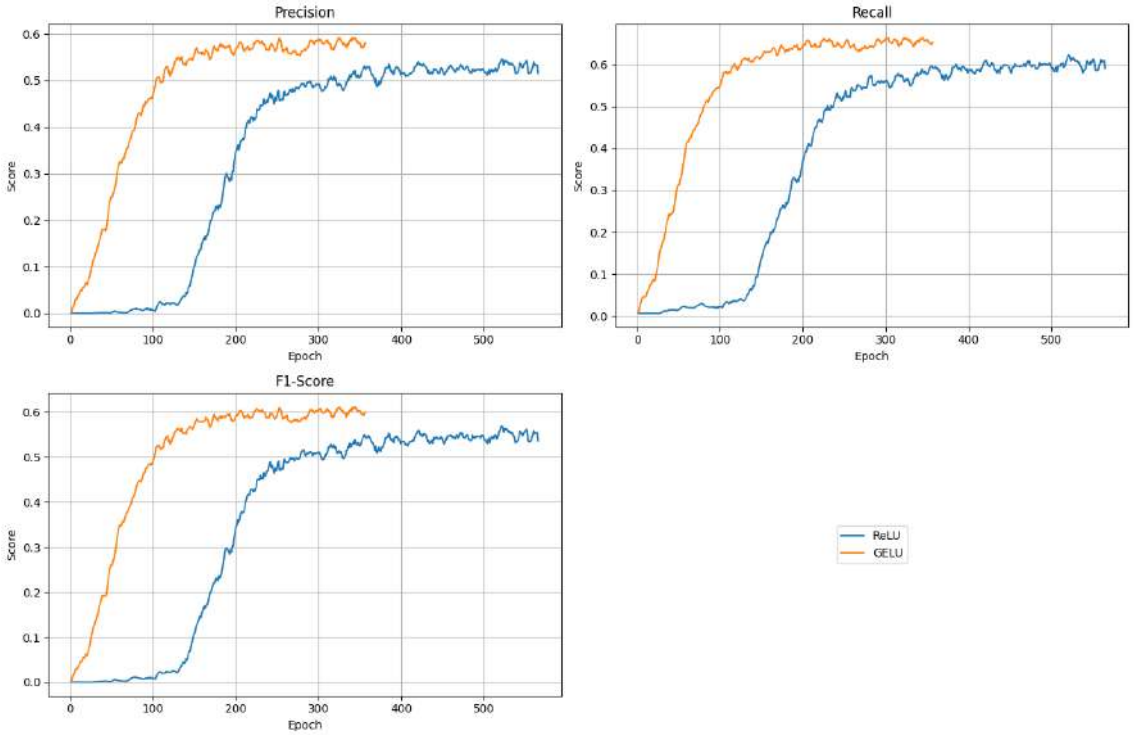
4.6.1 Ενεργοποιήσεις και Regularization

Αρχικά, η προεπιλεγμένη διαμόρφωση περιλάμβανε ReLU ως συνάρτηση ενεργοποίησης σε όλα τα επίπεδα του CLNM και στον τελικό ταξινομητή, χωρίς χρήση τεχνικών regularization όπως το dropout. Στόχος ήταν η αποτίμηση της βασικής αρχιτεκτονικής χωρίς την επίδραση επιπλέον παραμέτρων, ώστε να απομονωθεί η συμπεριφορά του μοντέλου με την ελάχιστη δυνατή επέμβαση. Στη συνέχεια, δοκιμάστηκε η αντικατάσταση της ReLU με τη συνάρτηση GELU, η οποία εισάγει πιο ήπια μη γραμμικότητα και αποφεύγει την απότομη αποκοπή τιμών, επιτρέποντας μεγαλύτερη ροή πληροφορίας και καλύτερη διαχείριση τιμών κοντά στο μηδέν. Η επιλογή αυτή βασίστηκε στη βιβλιογραφικά τεκμηριωμένη βελτίωση της απόδοσης που επιφέρει σε προβλήματα υψηλής δυσκολίας όπως το FGVC. Παράλληλα, εξετάστηκε η προσθήκη dropout στο τελικό επίπεδο ώστε να μειωθεί το φαινόμενο υπερεμπιστοσύνης του ταξινομητή.



Εικόνα 36: ReLU vs GELU – Σύγκριση Training και Validation Accuracy για CN-CNN

Όπως φαίνεται στο Σχήμα 37, το μοντέλο με GELU συγκλίνει σταθερότερα και εμφανίζει υψηλότερη ακρίβεια στο validation set, φτάνοντας μέγιστο 66.36% έναντι 62.34% για το ReLU.



Εικόνα 37: ReLu vs GELU μετρικές για το ResNet-50

Οι βασικές μετρικές ταξινόμησης επιβεβαιώνουν την υπεροχή της GELU σε σχέση με τη ReLU, τόσο ως προς την ποιότητα των προβλέψεων όσο και τη σταθερότητα της συμπεριφοράς κατά την εκπαίδευση. Συγκεκριμένα, η GELU εμφανίζει υψηλότερη τελική τιμή precision (0.5911 έναντι

0.5455), υποδεικνύοντας μεγαλύτερη ακρίβεια στις θετικές προβλέψεις, καθώς και βελτιωμένο recall (0.6636 έναντι 0.6234), γεγονός που μαρτυρά καλύτερη κάλυψη των πραγματικών θετικών περιπτώσεων. Η συνολική ισορροπία μεταξύ αυτών των δύο μεγεθών αποτυπώνεται στη βελτιωμένη F1-score (0.6113 έναντι 0.5683), ενώ επιπλέον παρατηρείται ταχύτερη σύγκλιση και μικρότερες διακυμάνσεις στις τιμές των μετρικών, καθ' όλη τη διάρκεια της εκπαίδευσης.

Ενεργοποίηση	Val Accuracy (%)	Precision	Recall	F1-score
ReLU	62.34	0.5455	0.6234	0.5683
GELU	**66.36**	**0.5911**	**0.6636**	**0.6113**

Πίνακας 9: Συγκριτική απόδοση μετρικών για ενεργοποιήσεις GELU και ReLU.

Αξιοσημείωτο είναι επίσης ότι το μοντέλο με GELU φτάνει ταχύτατα σε σταθερή συμπεριφορά και διατηρεί υψηλές τιμές μέχρι το τέλος της εκπαίδευσης, σε αντίθεση με τη ReLU όπου οι καμπύλες precision και F1-score παραμένουν χαμηλότερες καθ' όλη τη διάρκεια.

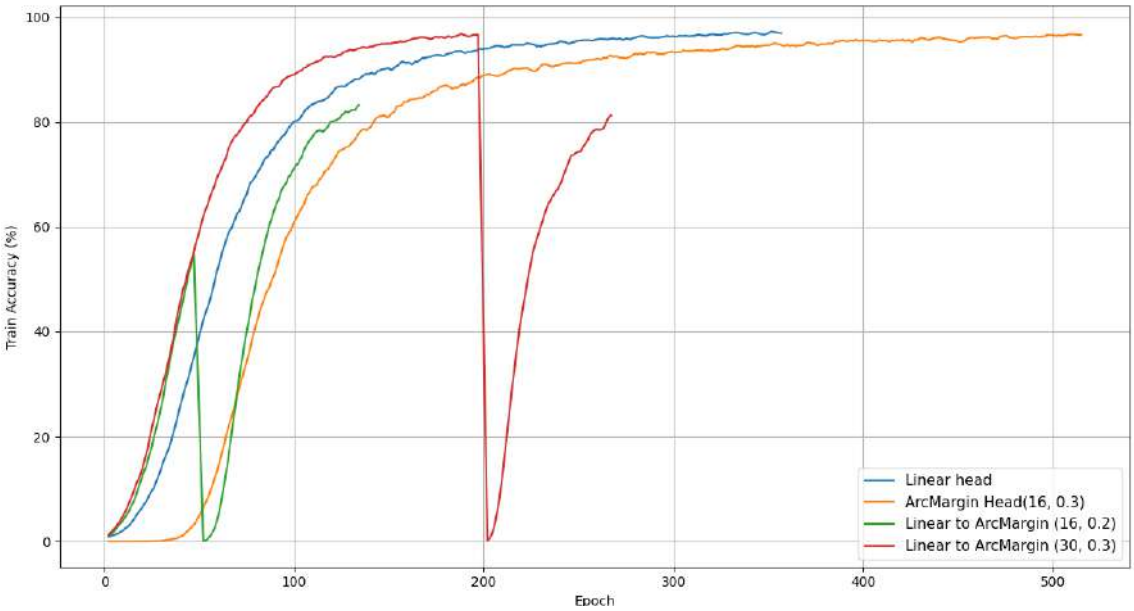
Συνολικά, η χρήση της GELU επέτρεψε καλύτερη αξιοποίηση των χαρακτηριστικών που εξάγονται από το CLNM, καθώς αποφεύγεται η απώλεια πληροφορίας σε σημεία μικρής ενεργοποίησης. Επιπλέον, η σταθερότερη συμπεριφορά των μετρικών και η μικρότερη απόκλιση μεταξύ training και validation καταδεικνύουν καλύτερη γενίκευση.

Με βάση τα παραπάνω αποτελέσματα, επιλέγεται η χρήση της GELU ως προεπιλεγμένης συνάρτησης ενεργοποίησης για όλα τα επόμενα πειράματα, καθώς προσφέρει συνολικά ανώτερη απόδοση και μεγαλύτερη αξιοπιστία στην ταξινόμηση των μονάδων κλιματιστικών.

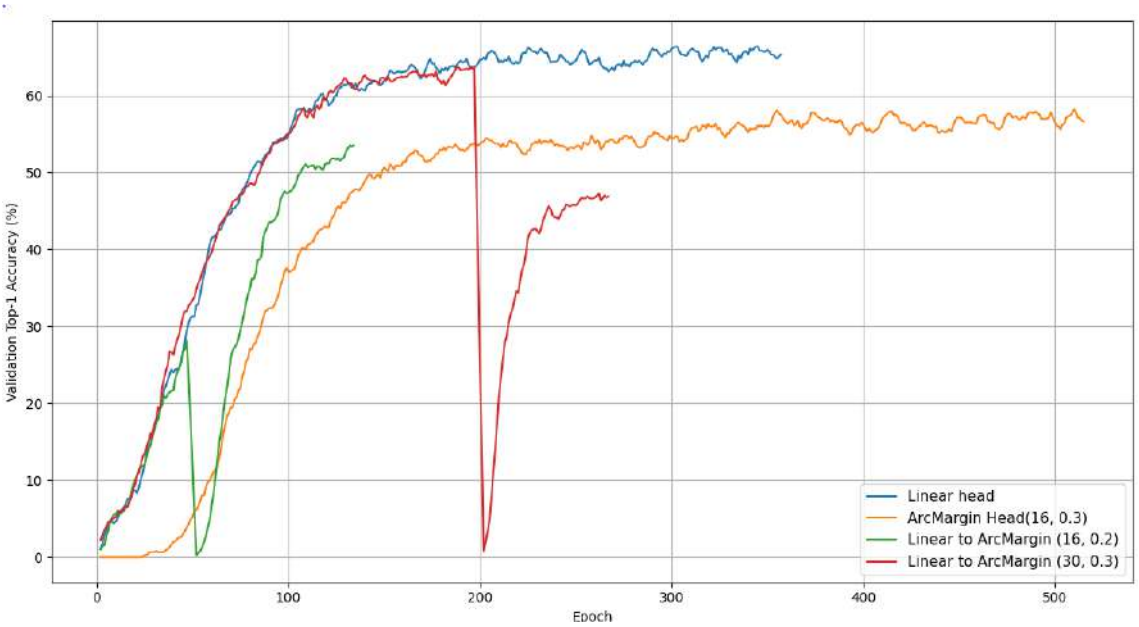
4.6.2 Angular Margin-Based Ταξινομητές

Ακολούθως, η κλασική γραμμική κεφαλή αντικαταστάθηκε με ArcMarginProduct ταξινομητή. Ο στόχος ήταν να επιβληθεί αυστηρότερη γεωμετρική διαφοροποίηση μεταξύ των προβολών των κλάσεων στο χαρακτηριστικό χώρο, μετατρέποντας το πρόβλημα από ευκλείδεια κατηγοριοποίηση σε γωνιακή (angular classification). Η χρήση margin-based κεφαλών, όπως το ArcFace ή CosFace, θεωρείται καίρια για προβλήματα fine-grained visual classification, καθώς ενισχύει τη διαχωρισιμότητα των embeddings χωρίς να βασίζεται αποκλειστικά στο μέγεθός τους.

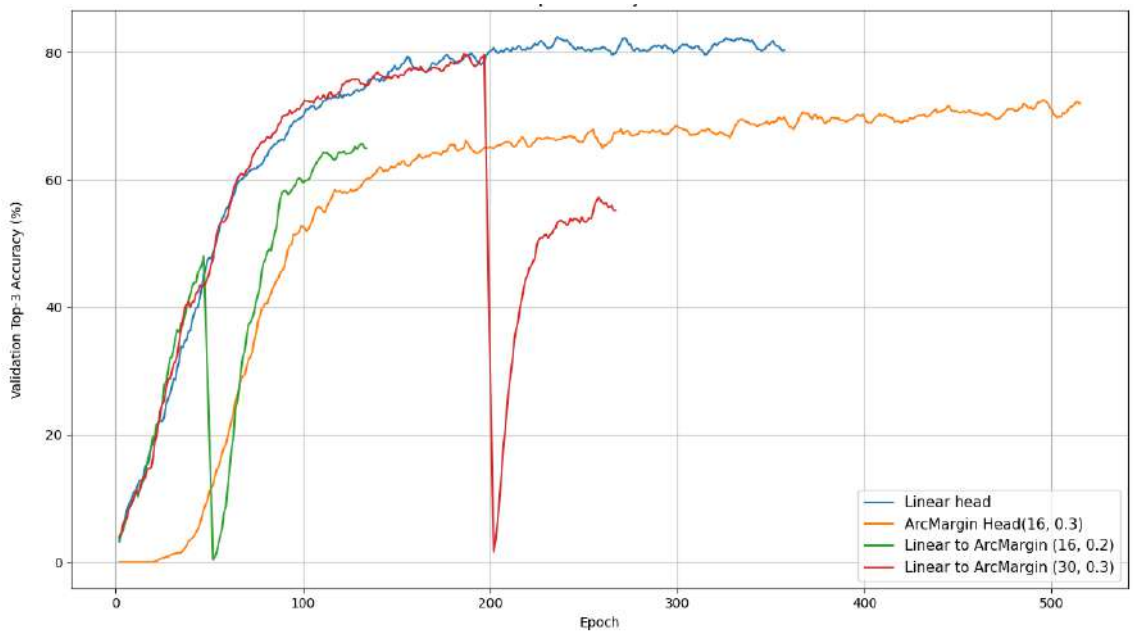
Πραγματοποιήθηκαν δοκιμές με παραλλαγές της ArcMargin κεφαλής: αρχικά με scale factor 16 και margin 0.3, στη συνέχεια με πιο ήπια διαμόρφωση Linear $\rightarrow$ ArcMargin (16, 0.2), ώστε να ενισχυθεί η σταθερότητα κατά τη μετάβαση, και τέλος με επιθετική παραμετροποίηση (30, 0.3), με σκοπό την αύξηση της γωνιακής πίεσης και την επίτευξη καθαρού χωρικού διαχωρισμού.



Εικόνα 38: Train Accuracy Linear vs ArcMargin Head για ResNet-50



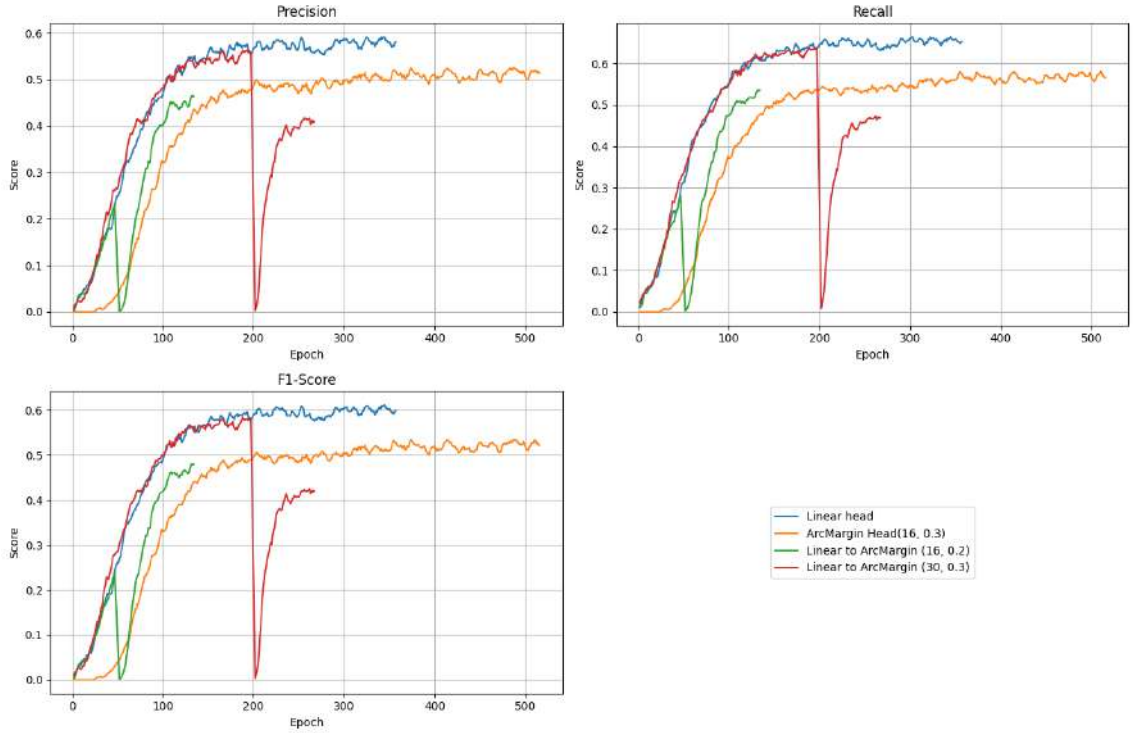
Εικόνα 39: Validation Accuracy Linear vs ArcMargin Head για ResNet-50



Εικόνα 40: Top3 Accuracy Linear vs ArcMargin Head για ResNet-50

Από τις παραπάνω γραφικές μπορεί να παρατηρηθεί ότι η χρήση του Linear head οδηγεί σε ταχύτερη σύγκλιση και υψηλότερη τελική ακρίβεια, τόσο στο σύνολο εκπαίδευσης όσο και στο σύνολο επικύρωσης (validation). Συγκεκριμένα, η τελική ακρίβεια Top-1 στο validation set φτάνει το 66.36%, ενώ η αντίστοιχη Top-3 ακρίβεια ανέρχεται σε 82.47%. Οι επιδόσεις αυτές καθιστούν τον γραμμικό ταξινομητή ένα ισχυρό baseline για το συγκεκριμένο πρόβλημα.

Αντίθετα, η απευθείας χρήση ArcMargin head ή η μετάβαση από Linear σε ArcMargin στη μέση της εκπαίδευσης οδηγούν σε πιο αργή σύγκλιση και υποδεέστερη τελική απόδοση. Στα σενάρια Linear $\rightarrow$ ArcMargin παρατηρείται χαρακτηριστική πτώση (reset) κατά τη στιγμή της αλλαγής κεφαλής, γεγονός που επιφέρει απώλεια γνώσης και απαιτεί σημαντικό αριθμό επιπλέον εποχών για αποκατάσταση. Παρότι η εκπαίδευση ανακάμπτει, οι τελικές τιμές ακρίβειας παραμένουν χαμηλότερες σε σχέση με την καθαρή γραμμική κεφαλή.



Εικόνα 41: Metrics Linear vs ArcMargin Head για ResNet-50

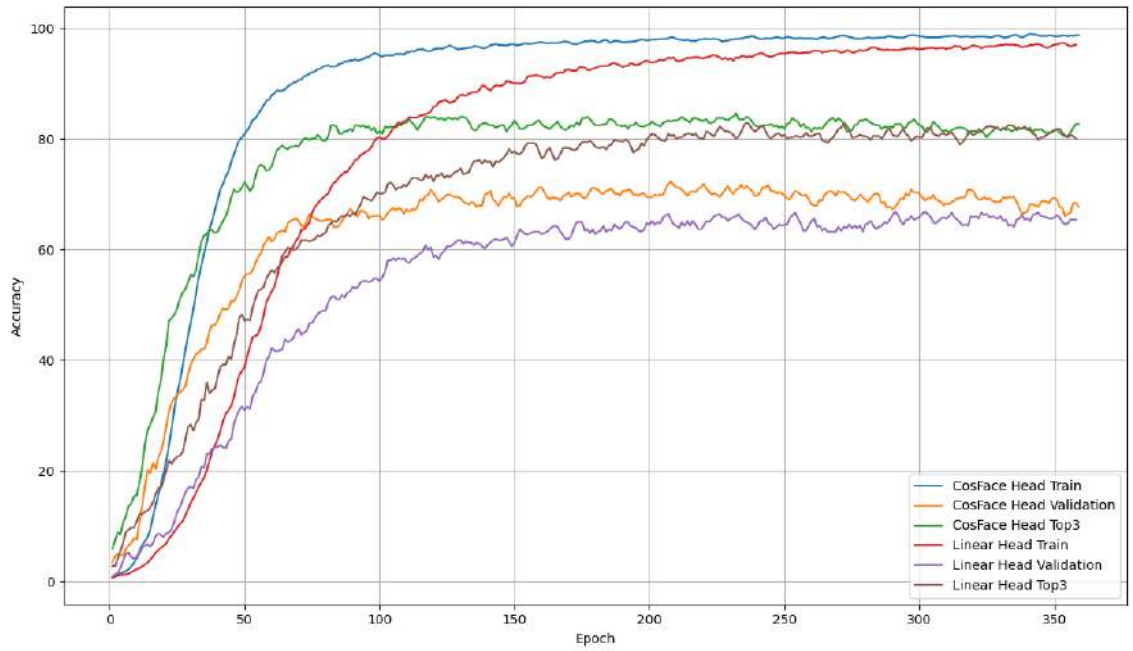
Η παραπάνω τάση ενισχύεται και από τις μετρικές precision, recall και F1-score, όπως παρουσιάζονται στο Σχήμα 41 και συνοψίζονται στον παρακάτω πίνακα. Παρατηρείται ότι η γραμμική κεφαλή διατηρεί τις υψηλότερες τελικές τιμές σε όλες τις μετρικές, με F1-score 0.6113 έναντι 0.5350 της ArcMargin και ακόμα χαμηλότερα σε μεταβατικά σενάρια. Η μόνη παραλλαγή που προσεγγίζει ικανοποιητικά την απόδοση του Linear είναι η Linear → ArcMargin (30, 0.3), χωρίς όμως να την υπερβαίνει.

Κεφαλή	**Val Top-1 (%)**	**Val Top-3 (%)**	**Precision**	**Recall**	**F1-score**
Linear head	**66.36**	**82.47**	**0.5911**	**0.6636**	**0.6113**
ArcMargin Head (16, 0.3)	58.18	72.47	0.5258	0.5818	0.5350
Linear → ArcMargin (16, 0.2)	53.51	65.71	0.4656	0.5351	0.4807
Linear → ArcMargin (30, 0.3)	63.77	79.74	0.5637	0.6377	0.5837

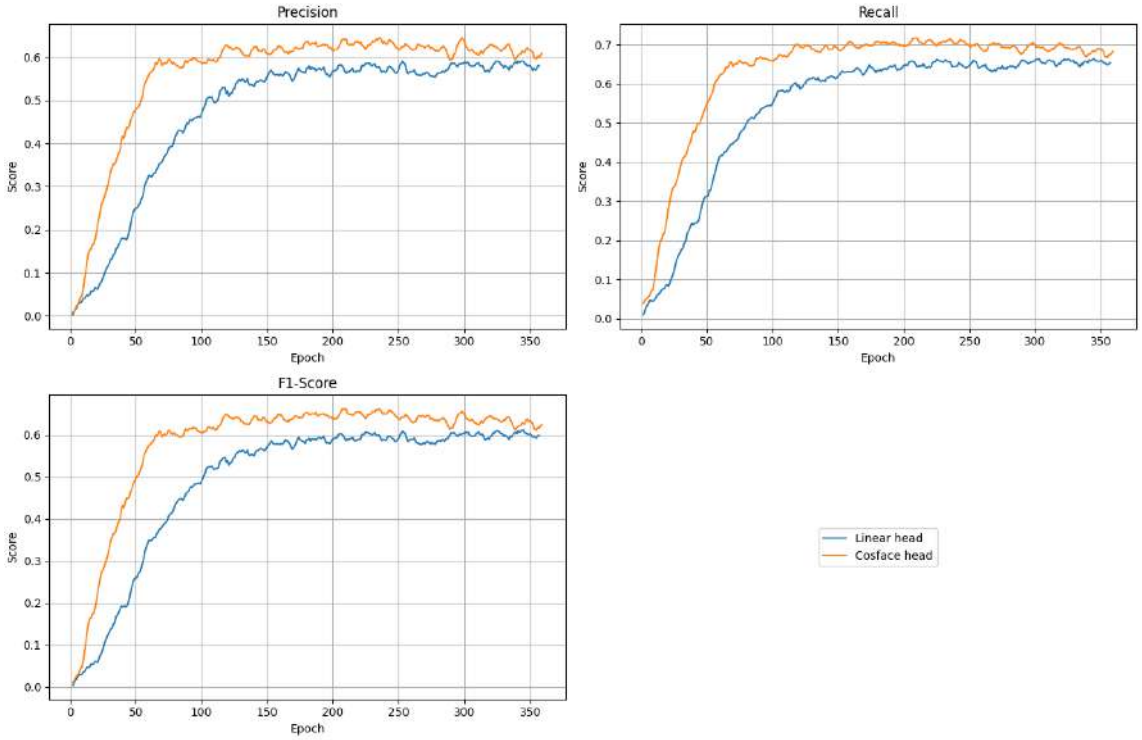
Πίνακας 10: Συγκριτική απόδοση μετρικών σε Linear και ArcMargin κεφαλές

Τα αποτελέσματα του πίνακα επιβεβαιώνουν ότι η κλασική γραμμική κεφαλή επιτυγχάνει την καλύτερη ισορροπία μεταξύ ταχύτητας σύγκλισης και τελικής απόδοσης. Παρότι οι margin-based κεφαλές προσφέρουν θεωρητικά πλεονεκτήματα στη γεωμετρική διαφοροποίηση των κατηγοριών, η απόδοσή τους στη συγκεκριμένη υλοποίηση φαίνεται να εξαρτάται έντονα από τις υπερπαραμέτρους και το σημείο εισαγωγής τους στην εκπαίδευση.

Επιπλέον, δοκιμάστηκε η πλήρης απομάκρυνση της cross-entropy loss και η αντικατάστασή της από CosFace Loss σε συνδυασμό με cosine κεφαλή. Το μοντέλο σε αυτή την παραλλαγή βασίζεται αποκλειστικά σε angular αποστάσεις, γεγονός που θεωρητικά του επιτρέπει να αγνοεί πληροφορία μη σχετική με τη μεταξύ-κλάσεων διάκριση (όπως φωτεινότητα, υφή ή μεγέθη), εστιάζοντας στη γωνιακή διαφοροποίηση μεταξύ των embeddings.



Εικόνα 42: Linear vs CosFace Head Accuracies για ResNet-50



Εικόνα 43: Linear vs CosFace Head metrics για ResNet-50

Από τα παραπάνω αποτελέσματα παρατηρείται ότι η χρήση CosFace head οδηγεί σε σαφώς βελτιωμένη απόδοση σε σχέση με την κλασική Linear head. Συγκεκριμένα, το CosFace επιτυγχάνει ταχύτερη σύγκλιση, υψηλότερη τελική ακρίβεια και σημαντικά βελτιωμένες τιμές στις βασικές μετρικές ταξινόμησης. Η τελική ακρίβεια Top-1 στο validation set φτάνει το 71.69% (έναντι 66.36% της Linear), ενώ η Top-3 ακρίβεια αγγίζει το 84.03% (έναντι 82.47%).

Αντίστοιχη βελτίωση καταγράφεται και στις υπόλοιπες μετρικές: το precision αυξάνεται σε 0.6444, το recall σε 0.7169 και η F1-score φτάνει το 0.6621, έναντι 0.6113 της Linear.

Τα αποτελέσματα συνοψίζονται στον ακόλουθο πίνακα:

Κεφαλή	**Val Top-1 (%)**	**Val Top-3 (%)**	**Precision**	**Recall**	**F1-score**
Linear	66.36	82.47	0.5911	0.6636	0.6113
CosFace	**71.69**	**84.03**	**0.6444**	**0.7169**	**0.6621**

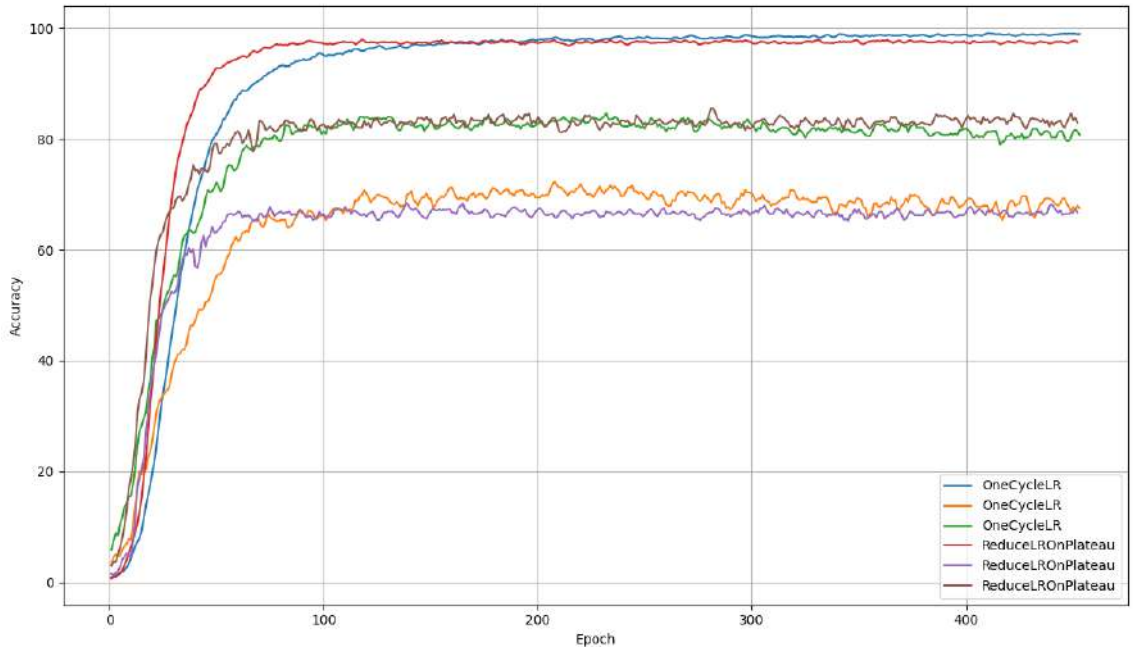
Πίνακας 11: Συγκριτική απόδοση μετρικών σε Linear και CosFace κεφαλές

Τα παραπάνω αποτελέσματα επιβεβαιώνουν τη θεωρητική υπεροχή margin-based κεφαλών όπως το CosFace στα προβλήματα fine-grained ταξινόμησης. Η ενίσχυση της διαχωρισιμότητας των embeddings μέσω angular margin loss αποδεικνύεται ιδιαίτερα αποτελεσματική για τη βελτίωση της γενίκευσης. Το CosFace head επιτρέπει στο δίκτυο να κατασκευάσει πιο "καθαρές" και διακριτές αναπαραστάσεις των κατηγοριών, περιορίζοντας την επικάλυψη στο χαρακτηριστικό χώρο και ενισχύοντας την ακρίβεια των προβλέψεων.

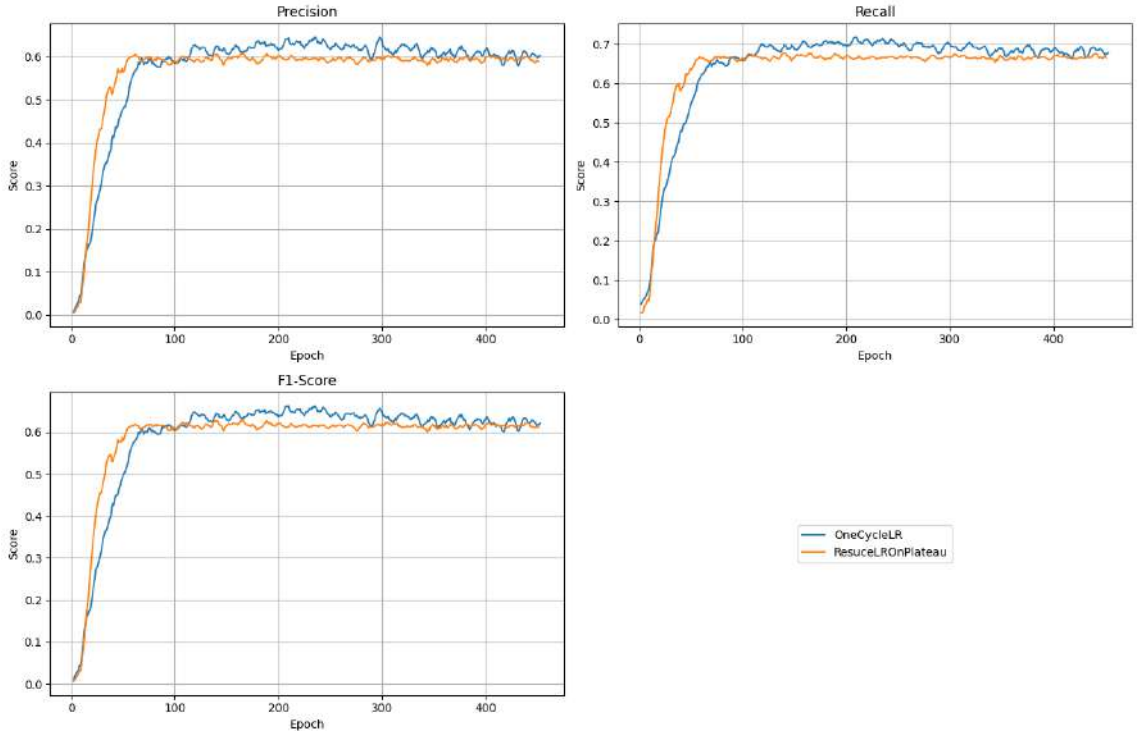
Με βάση τα ανωτέρω, οι επόμενοι πειραματισμοί βασίζονται στη χρήση CosFace head, καθώς αυτή η διαμόρφωση επιτυγχάνει την καλύτερη συνολική απόδοση στο παρόν fine-grained task ταξινόμησης μονάδων κλιματιστικών.

4.6.3 Προγραμματισμός Εκπαίδευσης

Σε επίπεδο ρύθμισης learning rate, αξιοποιήθηκε αρχικά ο OneCycleLR scheduler, ο οποίος θεωρείται κατάλληλος για γρήγορη σύγκλιση με μειωμένο κίνδυνο overfitting. Στη συνέχεια, αντικαταστάθηκε από τον ReduceLROnPlateau, έναν adaptive scheduler που μειώνει το learning rate μόνο όταν παρατηρείται στασιμότητα στην απόδοση στο validation set. Η αλλαγή αυτή βασίστηκε στην υπόθεση ότι ορισμένα μοντέλα (ιδίως εκείνα που κάνουν χρήση ευαίσθητων loss functions όπως το CosFace) ενδέχεται να ωφεληθούν από πιο «συντηρητική» προσέγγιση στη δυναμική του ρυθμού εκμάθησης.



Εικόνα 44: OneCycleLR vs ReduceLROnPlateau



Εικόνα 45: OneCycleLR vs ReduceLROnPlateau Metrics

Από τα αποτελέσματα των Σχημάτων 44 και 45, διαπιστώνεται ότι η επιλογή scheduler δεν επηρεάζει δραματικά τη συνολική απόδοση, ωστόσο υπάρχουν αξιοσημείωτες διαφορές ως προς τις επιμέρους μετρικές και τη δυναμική της εκπαίδευσης. Ο OneCycleLR οδηγεί σε υψηλότερη τελική ακρίβεια Top-1 (71.69%), ενώ ο ReduceLROnPlateau υπερέχει ελαφρώς στην Top-3 (84.81% έναντι 84.03%).

Όσον αφορά τις μετρικές αξιολόγησης, ο OneCycleLR παρουσιάζει σταθερά υψηλότερες τιμές σε precision (0.6444 έναντι 0.6097), recall (0.7169 έναντι 0.6779) και F1-score (0.6621 έναντι 0.6297), υποδεικνύοντας καλύτερη ισορροπία και ταξινομητική απόδοση. Το CosFace head φαίνεται να επωφελείται περισσότερο από τον πιο "επιθετικό" ρυθμό εκμάθησης του OneCycleLR, ιδιαίτερα στα πρώιμα στάδια της εκπαίδευσης.

Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

Scheduler	Val Top-1 (%)	Val Top-3 (%)	Precision	Recall	F1-score
OneCycleLR	**71.69**	84.03	**0.6444**	**0.7169**	**0.6621**
ReduceLROnPlateau	67.79	**84.81**	0.6097	0.6779	0.6297

Πίνακας 12: Συγκριτική απόδοση μετρικών σε OneCycleLR και ReduceLROnPlateau schedulers

Παρά τη σχετική ισοδυναμία, προκύπτουν σαφή πλεονεκτήματα ανάλογα με το πλαίσιο χρήσης κάθε scheduler. Ο OneCycleLR είναι πιο κατάλληλος όταν:

- ο αριθμός εποχών είναι εκ των προτέρων γνωστός·
- απαιτείται σταθερή, γρήγορη εκπαίδευση με περιορισμένο computational budget·
- το training pipeline είναι σταθερό και δεν απαιτείται online παρακολούθηση.

Αντίστοιχα, ο ReduceLROnPlateau είναι προτιμότερος όταν:

- δεν υπάρχει σαφής πρόβλεψη για τη διάρκεια εκπαίδευσης·
- απαιτείται adaptive προσέγγιση σε προβλήματα με plateaus·
- χρησιμοποιούνται ευαίσθητες loss functions που ωφελούνται από μικρότερα learning rates στα τελικά στάδια.

Με βάση τα παραπάνω, στις επόμενες δοκιμές θα γίνεται συνδυαστική χρήση των δύο schedulers. Ο ReduceLROnPlateau θα χρησιμοποιείται για διερεύνηση και ρύθμιση της διάρκειας εκπαίδευσης, ενώ το OneCycleLR θα εφαρμόζεται σε τελικές εκπαιδεύσεις, για σταθερή σύγκλιση και βέλτιστη απόδοση. Τα αποτελέσματα που θα παρουσιάζονται θα είναι από τη χρήση του OneCycleLR.

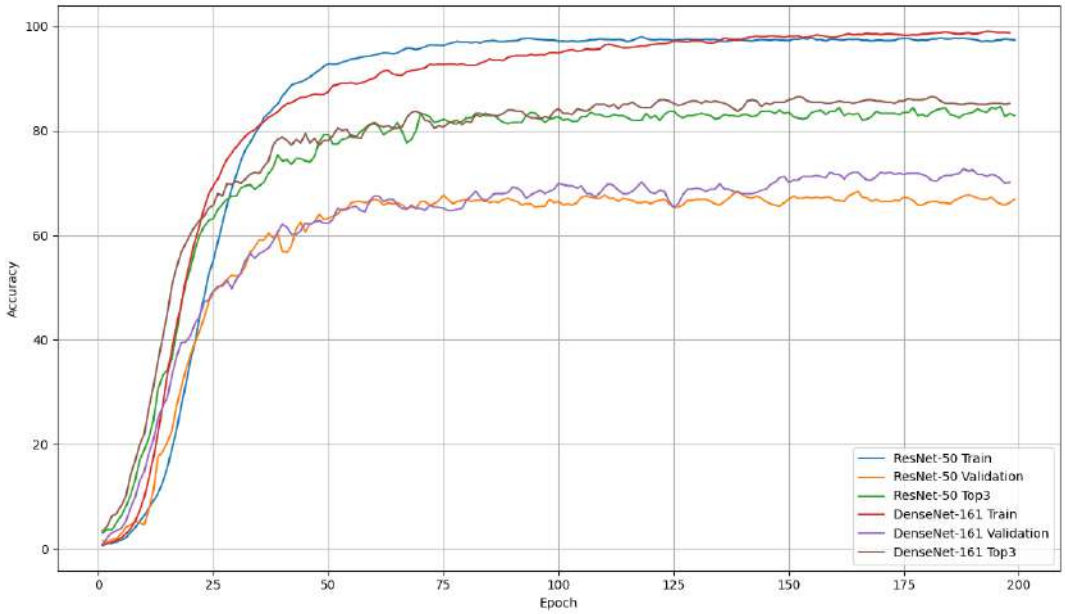
4.6.4 Αντικατάσταση Backbone

Στο τελικό στάδιο των πειραματικών επεκτάσεων, δοκιμάστηκε η αντικατάσταση του προεπιλεγμένου ResNet-50 backbone με το DenseNet-161, με στόχο την αξιολόγηση της επίδρασης ενός πιο ισχυρού και βαθύτερου feature extractor. Η επιλογή του συγκεκριμένου μοντέλου βασίστηκε στην ικανότητά του να επιτρέπει άμεση ροή πληροφορίας μεταξύ όλων των επιπέδων μέσω πυκνών συνδέσεων (dense connections), ενισχύοντας έτσι την επαναχρησιμοποίηση χαρακτηριστικών και τη βελτίωση της ροής των gradients κατά την εκπαίδευση.

Η υπόθεση ήταν ότι η χρήση ενός πιο εκφραστικού backbone θα ενίσχυε περαιτέρω την απόδοση του CN-CNN, ιδιαίτερα στις ρυθμίσεις όπου χρησιμοποιούνται margin-based loss functions και

ενεργοποιήσεις υψηλής ευαισθησίας (π.χ. GELU). Παράλληλα, εξετάστηκε κατά πόσο η αυξημένη υπολογιστική πολυπλοκότητα του DenseNet-161 μπορεί να δικαιολογηθεί από ενδεχόμενα οφέλη σε ακρίβεια.

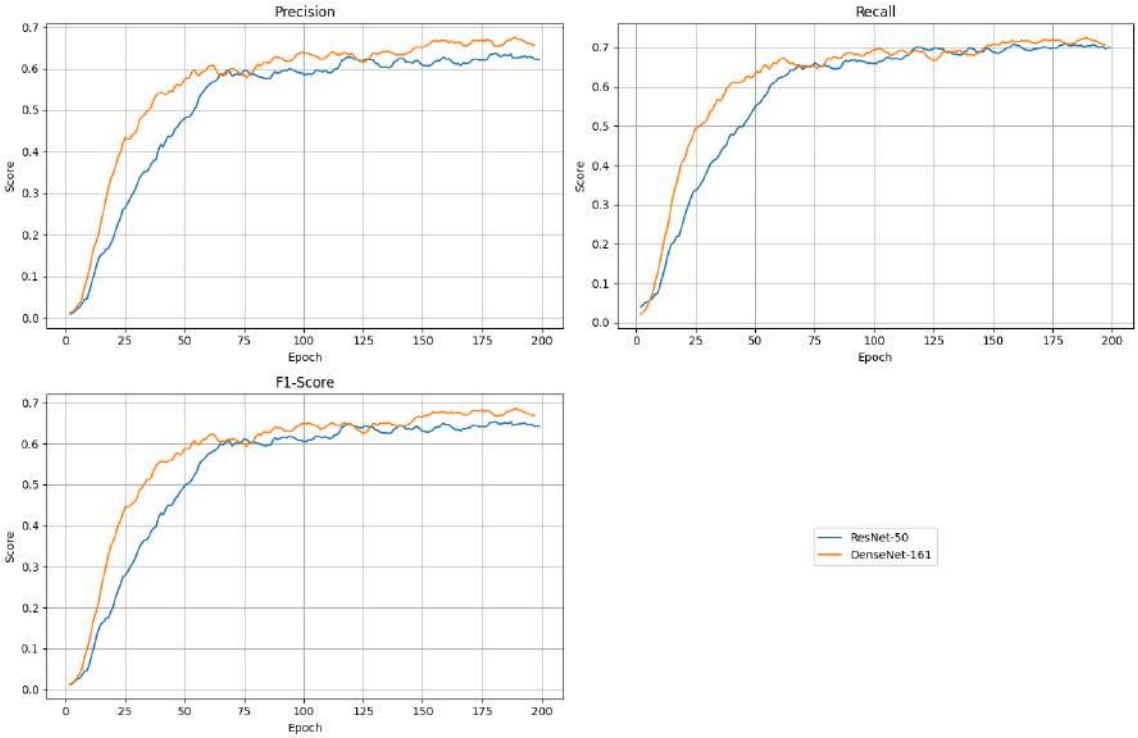
Τα συγκριτικά αποτελέσματα παρουσιάζονται στο Σχήμα 46, όπου απεικονίζεται η πορεία της ακρίβειας των μοντέλων ResNet-50 και DenseNet-161 κατά την εκπαίδευση (ανά epoch), τόσο για τα σύνολα εκπαίδευσης όσο και επικύρωσης.



Εικόνα 46: Σύγκριση απόδοσης CN-CNN με backbone ResNet-50 και DenseNet-161

Όπως παρατηρείται, το DenseNet-161 παρουσίασε συνολικά ανώτερη απόδοση έναντι του ResNet-50. Η καμπύλη ακρίβειας επικύρωσης (validation accuracy) για το DenseNet-161 παραμένει σταθερά υψηλότερη, ενώ και η Top-3 ακρίβεια παρουσιάζει αισθητή βελτίωση. Παράλληλα, το Train Accuracy φτάνει 98% χωρίς να παρατηρούνται έντονα σημάδια υπερεκπαίδευσης, υποδηλώνοντας ότι το μοντέλο γενικεύει αποτελεσματικά.

Η υπεροχή του DenseNet-161 επιβεβαιώνεται και από τις βασικές μετρικές ταξινόμησης, όπως παρουσιάζεται στο Σχήμα 47. Ο βαθύτερος αυτός backbone καταγράφει υψηλότερες τελικές τιμές σε precision, recall και F1-score, αποτυπώνοντας τη συνολικά καλύτερη ποιοτική συμπεριφορά του μοντέλου.



Εικόνα 47: ResNet-50 vs DenseNet-161 metrics

Τα συγκριτικά αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

Backbone	**Val Top-1 (%)**	**Val Top-3 (%)**	**Precision**	**Recall**	**F1-score**
ResNet-50	71.69	84.03	0.6444	0.7169	0.6621
DenseNet-161	**74.03**	**87.01**	**0.6959**	**0.7403**	**0.7054**

Πίνακας 13: Συγκριτική απόδοση μετρικών σε backbones ResNet-50 και DenseNet-161

Τα αποτελέσματα αυτά υποδεικνύουν ότι, στο πρόβλημα fine-grained ταξινόμησης μονάδων κλιματιστικών, ένα πιο βαθύ και εκφραστικό backbone όπως το DenseNet-161 μπορεί να προσφέρει ουσιαστικό πλεονέκτημα σε απόδοση. Ωστόσο, η χρήση του συνεπάγεται αυξημένες απαιτήσεις σε υπολογιστικούς πόρους και χρόνο εκπαίδευσης, παράγοντες που πρέπει να λαμβάνονται υπόψη ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής.

Για την υποστήριξη του πιο απαιτητικού DenseNet-161 backbone, απαιτήθηκε η εφαρμογή τεχνικών επιτάχυνσης και βελτιστοποίησης μνήμης, καθώς ήδη το ResNet-50 δούλευε στα όρια των δυνατοτήτων της GPU. Συγκεκριμένα, ενεργοποιήθηκε χρήση mixed-precision training και δυναμική κλιμάκωση loss (gradient scaling) μέσω:

```
from torch.cuda.amp import autocast, GradScaler
torch.backends.cudnn.benchmark = True
torch.backends.cuda.matmul.allow_tf32 = True
use_amp = torch.cuda.is_available()
scaler = GradScaler(enabled=use_amp)
```

Οι παραπάνω ρυθμίσεις χρησιμοποιούνται για τη βελτιστοποίηση της χρήσης GPU κατά την εκπαίδευση:

- `torch.backends.cudnn.benchmark = True`: επιτρέπει στο cuDNN να επιλέξει δυναμικά τον πιο αποδοτικό αλγόριθμο για το τρέχον μέγεθος batch, βελτιώνοντας την ταχύτητα.
- `torch.backends.cuda.matmul.allow_tf32 = True`: ενεργοποιεί την υποστήριξη TensorFloat-32 (TF32) σε GPUs αρχιτεκτονικής Ampere και άνω, επιτυγχάνοντας ταχύτερους υπολογισμούς σε πράξεις πολλαπλασιασμού πινάκων χωρίς σημαντική απώλεια ακρίβειας.
- `autocast, GradScaler`: ενεργοποιούν αυτόματα το mixed-precision training (FP16/FP32), μειώνοντας την κατανάλωση μνήμης και τον χρόνο εκπαίδευσης, ενώ η χρήση `GradScaler` διασφαλίζει την ορθή κλιμάκωση των gradients ώστε να αποφεύγονται αριθμητικά σφάλματα (underflows).

Οι παραπάνω ρυθμίσεις επέτρεψαν την αποδοτικότερη χρήση μνήμης και υπολογιστικής ισχύος, καθιστώντας εφικτή την εκπαίδευση του DenseNet-161 στο συγκεκριμένο hardware (NVIDIA RTX 3060, 12GB VRAM). Χωρίς τις τεχνικές αυτές, το δίκτυο δεν μπορούσε να ξεκινήσει την εκπαίδευση και παρουσίαζε OOM (Out Of Memory) σφάλματα.

Τα αποτελέσματα του υπολογιστικού κόστους για την εκπαίδευση του CN-CNN με backbone ResNet-50 και DenseNet-161 συνοψίζονται στον Πίνακα 14. Για να επιτευχθεί επιτυχής εκπαίδευση με το πιο απαιτητικό DenseNet-161, εφαρμόστηκαν επιπλέον τεχνικές βελτιστοποίησης χρήσης GPU, όπως mixed-precision training και ενεργοποίηση TF32, όπως παρουσιάστηκε παραπάνω.

Backbone	**Μέσος Χρόνος Εκπαίδευσης (δευτ.)**	**Μέσος Χρόνος Επικύρωσης (δευτ.)**	**Μέση Χρήση GPU (MB)**
ResNet-50	112.43	16.19	10970
DenseNet-161	151.45	11.83	11485

Πίνακας 14: Υπολογιστικό Κόστος για CN-CNN με διαφορετικά Backbone

Συνολικά, παρατηρείται ότι το DenseNet-161 προσφέρει ελαφρώς υψηλότερη απόδοση στο validation set έναντι του ResNet-50, με αύξηση στο υπολογιστικό κόστος (χρόνο εκπαίδευσης και κατανάλωση GPU). Επομένως, για εφαρμογές όπου η μέγιστη δυνατή ακρίβεια αποτελεί προτεραιότητα, η χρήση του DenseNet-161 μπορεί να δικαιολογείται. Αντίθετα, σε περιβάλλοντα με περιορισμένους υπολογιστικούς πόρους ή απαιτήσεις για ταχύτερη εκπαίδευση, το ResNet-50 παραμένει μια πολύ ανταγωνιστική επιλογή. Είναι συνεπώς σημαντικό να αξιολογείται, κατά περίπτωση, αν το όφελος σε ακρίβεια υπερτερεί του επιπλέον κόστους εκπαίδευσης και χρήσης.

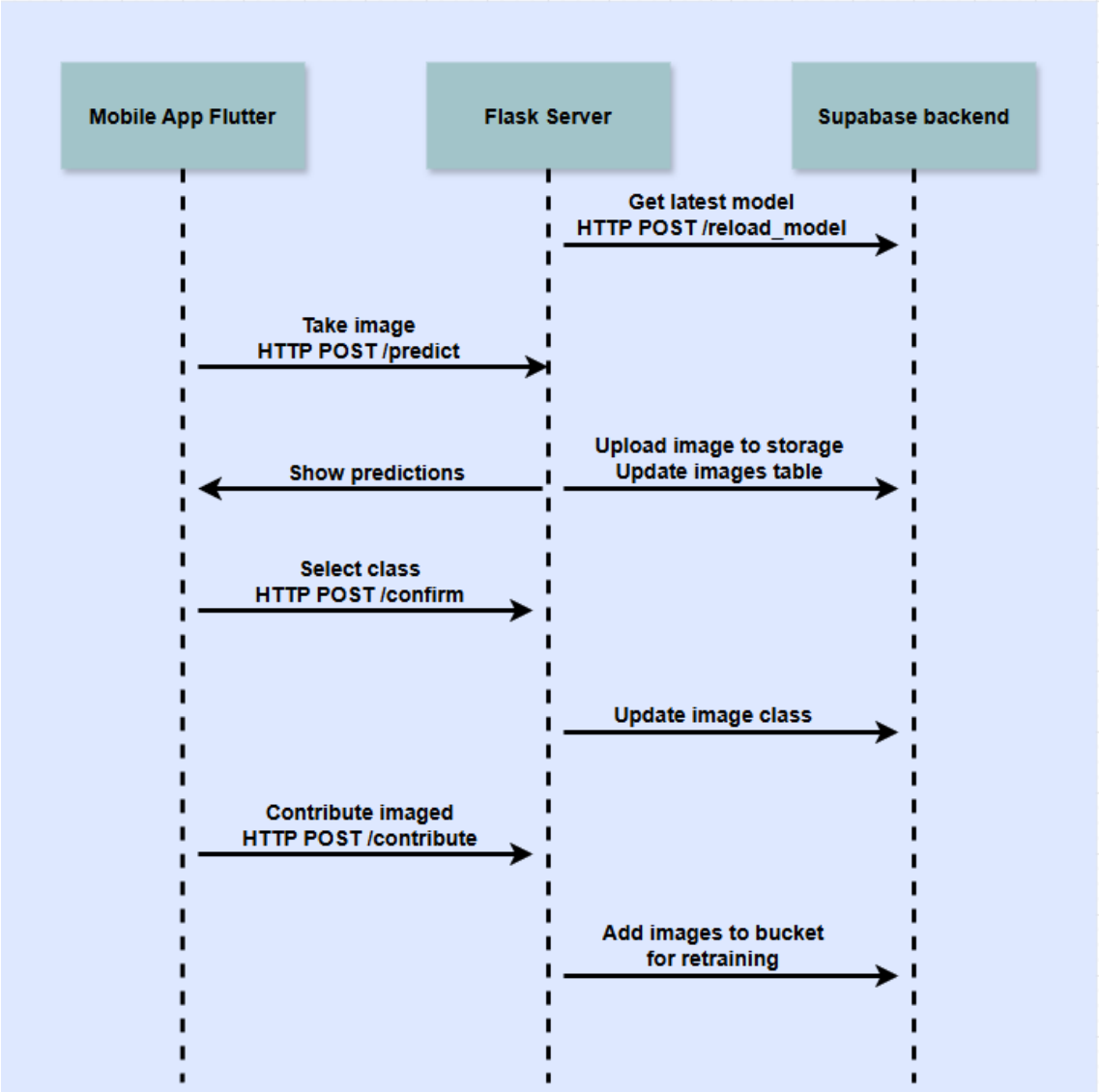
5 Εφαρμογή Αξιοποίησης Μοντέλου

Η αποτελεσματική αξιοποίηση του ανεπτυγμένου μοντέλου ταξινόμησης εικόνων CN-CNN πραγματοποιήθηκε μέσω της ανάπτυξης μιας ολοκληρωμένης εφαρμογής, η οποία βασίζεται σε τρεις κύριους άξονες: το backend (Supabase), τον server (Flask) και την κινητή εφαρμογή (Flutter). Η συνολική αρχιτεκτονική του συστήματος σχεδιάστηκε ώστε να παρέχει ένα απλό, ευέλικτο και

αποδοτικό πλαίσιο για την ταξινόμηση οικιακών συσκευών, με βασικό παράδειγμα εφαρμογής τα κλιματιστικά.

Ο τελικός στόχος της εφαρμογής είναι να επιτρέπει στον τελικό χρήστη να φωτογραφίζει μία συσκευή με το κινητό του τηλέφωνο και, μέσω της αξιοποίησης του ανεπτυγμένου μοντέλου μηχανικής μάθησης, να λαμβάνει προτεινόμενες ταξινομήσεις (κλάσεις) για τη συσκευή. Η διαδικασία ενσωματώνει τη δυνατότητα αλληλεπίδρασης του χρήστη, ο οποίος μπορεί να επιβεβαιώσει ή να τροποποιήσει την τελική κατηγορία, ενισχύοντας την αξιοπιστία και την προσαρμοστικότητα του συστήματος.

Η αρχιτεκτονική του συστήματος έχει σχεδιαστεί με στόχο την ευκολία ανάπτυξης και τη φορητότητα, ενώ παράλληλα υποστηρίζει επεκτασιμότητα (scalability) και μελλοντικές βελτιώσεις. Η συνολική δομή και ροή της εφαρμογής απεικονίζεται στο παρακάτω σχήμα.



Εικόνα 48: Αρχιτεκτονική επικοινωνίας εφαρμογής

Αξίζει να σημειωθεί ότι το πρώτο request για φόρτωση του μοντέλου (μοντέλο CN-CNN σε μορφή ONNX) είναι ανεξάρτητο από την υπόλοιπη ροή της εφαρμογής. Η ύπαρξη αυτής της λειτουργίας

είναι αναγκαία ώστε να μην απαιτείται επανεκκίνηση του server όταν αλλάζει το μοντέλο, καθώς επίσης επιτρέπει απομακρυσμένη διαχείριση και ενημέρωση χωρίς διακοπή λειτουργίας.

5.1 Backend

Η πλατφόρμα Supabase επιλέχθηκε ως backend λύση λόγω της πληρότητας, της ευκολίας ενσωμάτωσης που προσφέρει, καθώς και του ιδιαίτερα ανταγωνιστικού δωρεάν πακέτου σε επίπεδο αποθηκευτικού χώρου και αριθμού κλήσεων. Το Supabase παρέχει έναν πλήρη PostgreSQL database με ενσωματωμένο RESTful API, διαχείριση ταυτοποίησης χρηστών (auth) και δυνατότητα αποθήκευσης αντικειμένων (object storage). Επιπλέον, προσφέρει φιλικό περιβάλλον διαχείρισης, ισχυρό σύστημα ρόλων και δικαιωμάτων (row-level security) και μειώνει σημαντικά την ανάγκη ανάπτυξης και συντήρησης ξεχωριστού backend server. Η επιλογή αυτή επέτρεψε την ταχεία ανάπτυξη του συστήματος και την εύκολη διαχείριση τόσο των δεδομένων όσο και του ίδιου του μοντέλου.

- Αποθήκευση των φωτογραφιών που ανεβάζουν οι χρήστες σε ένα storage bucket
- Αποθήκευση των class images (storage bucket), τα οποία χρησιμεύουν ως αναφορές για τις κλάσεις ταξινόμησης
- Διαχείριση του μοντέλου ONNX, από όπου ο Flask server αντλεί το κατάλληλο μοντέλο
- Καταγραφή των φωτογραφιών (πίνακας SQL) και της επιλεγμένης τελικής κατηγορίας (selected_class_name)
- Διαχείριση των ετικετών (labels) για τις διαφορετικές κατηγορίες της ταξινόμησης (πίνακας SQL)

Η διασύνδεση γίνεται μέσω REST API του Supabase, ενώ δεν απαιτήθηκε ξεχωριστό backend code ή serverless functions.

5.2 Server

Η επιλογή του Flask ως framework για την υλοποίηση του server έγινε λόγω της απλότητας, ευελιξίας και ευρείας χρήσης του για την ανάπτυξη RESTful APIs. Το Flask επιτρέπει την ταχεία ανάπτυξη εφαρμογών μικρής έως μεσαίας κλίμακας με ελάχιστες εξαρτήσεις, ενώ παράλληλα παρέχει πλήρη έλεγχο στη διαχείριση των αιτημάτων και της ροής δεδομένων.

Για τη διασφάλιση φορητότητας, επεκτασιμότητας και ευκολίας στη διανομή, ο server συσκευάστηκε ως Docker container. Η ανάπτυξη σε περιβάλλον Kubernetes επιλέχθηκε ώστε να επιτρέπει την αυτοματοποιημένη διαχείριση των container instances (pods), την εύκολη κλιμάκωση (scalability), καθώς και την παρακολούθηση και ανάκαμψη από σφάλματα. Η παραπάνω προσέγγιση καθιστά το σύστημα εύκολα αναπτύξιμο σε οποιαδήποτε σύγχρονη υποδομή cloud ή on-premises.

Ο Flask server φορτώνει το εκπαιδευμένο μοντέλο (αυτή τη στιγμή CN-CNN) σε μορφή ONNX, το οποίο αποθηκεύεται στο Supabase και ανακτάται από τον πίνακα models, διασφαλίζοντας έτσι ότι χρησιμοποιείται πάντα η πιο πρόσφατη έκδοση του μοντέλου.

Προτιμήθηκε η χρήση του ONNX από πλήρες PTH μοντέλο για την επλότητα των βιβλιοθηκών που είναι απαραίτητες για την αξιοποίηση του. Insert needed libraries approximate download needed

Προτιμήθηκε η χρήση του ONNX αντί ενός πλήρους PyTorch (`.pth`) μοντέλου, προκειμένου να αποφευχθεί η ανάγκη εγκατάστασης του PyTorch runtime και των σχετικών εξαρτήσεων, όπως `torchvision`, `timm` και `pandas`, οι οποίες αυξάνουν σημαντικά το μέγεθος και την πολυπλοκότητα του container. Ένα πλήρες PyTorch-based container απαιτεί συνολικό αποτύπωμα της τάξεως των ∼1–3 GB, ενώ η χρήση ONNX runtime μειώνει το μέγεθος σε περίπου ∼100–150 MB. Αυτό καθιστά τη συσκευασία, τη διανομή και την εκκίνηση του server ταχύτερη και πιο αποδοτική, επιτρέποντας την εύκολη ανάπτυξη σε ελαφριά containerized περιβάλλοντα.

Ο server παρέχει δύο βασικά REST API endpoints για την επικοινωνία με την κινητή εφαρμογή:

- `POST /predict`: δέχεται φωτογραφία που αποστέλλει η κινητή εφαρμογή, πραγματοποιεί προεπεξεργασία (resize, normalization), εκτελεί inference μέσω του CN-CNN μοντέλου και επιστρέφει τις Top-N προβλέψεις (Top-5), κάθε μία από τις οποίες συνοδεύεται από την κλάση (class name) και την αντίστοιχη πιθανότητα (confidence score). Παράλληλα, κάθε φωτογραφία που υποβάλλεται αποθηκεύεται στο Supabase bucket user-uploads, με μοναδικό αναγνωριστικό (UUID), ώστε να διατηρείται αρχείο των προβλέψεων.
- `POST /confirm`: λαμβάνει από την κινητή εφαρμογή την τελική επιλογή του χρήστη (selected class name), είτε από τις προτεινόμενες κλάσεις είτε ως νέα custom κλάση, και ενημερώνει το σχετικό πεδίο στον πίνακα images του Supabase, καταγράφοντας την τελική ετικέτα που επιλέχθηκε από τον χρήστη.
- `POST /reload-model`: επιτρέπει την απομακρυσμένη ανανέωση του μοντέλου χωρίς διακοπή της υπηρεσίας. Το endpoint ενεργοποιεί λήψη του πιο πρόσφατου `.onnx` αρχείου από το Supabase (πίνακας `models`) και δημιουργεί εκ νέου το `InferenceSession` στο Flask server. Έτσι διασφαλίζεται ότι ο server χρησιμοποιεί πάντα την επικαιροποιημένη έκδοση του μοντέλου. Ο τυπικός τρόπος χρήσης αυτού του endpoint είναι μέσω εργαλείου τύπου cURL ή API client (π.χ., Postman), όταν ανανεωθεί το μοντέλο στο Supabase. Η επιτυχής απάντηση περιλαμβάνει τα στοιχεία του νέου μοντέλου: όνομα, timestamp, input size και αν πρόκειται για grayscale ή RGB input.
- `POST /contribute`: επιτρέπει στους χρήστες να ανεβάσουν επιπλέον φωτογραφίες για την κατηγορία που έχουν επιλέξει (είτε προτεινόμενη είτε custom), με στόχο να αξιοποιηθούν σε μελλοντικά στάδια επανεκπαίδευσης του μοντέλου. Οι φωτογραφίες αποθηκεύονται σε ειδικό Supabase bucket, οργανωμένες με βάση το `UUID` της αρχικής φωτογραφίας και το τελικό όνομα κατηγορίας που δήλωσε ο χρήστης, ώστε να διασφαλίζεται η ορθή συσχέτιση και να αποφεύγονται διπλοεγγραφές σε περιπτώσεις όπου χρήστες συνεισφέρουν πολλαπλές φορές για το ίδιο μοντέλο σε διαφορετικές χρονικές στιγμές.

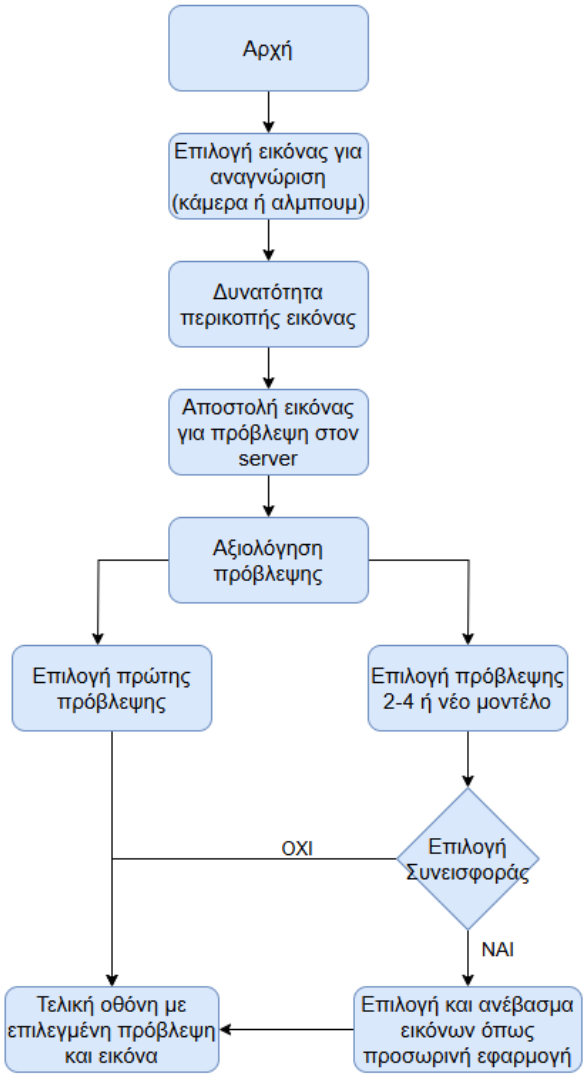
Η επικοινωνία με το Supabase υλοποιείται μέσω REST API, επιτρέποντας τη δυναμική αποθήκευση και ανάκτηση των δεδομένων χωρίς την ανάγκη επιπλέον backend εφαρμογής.

5.3 Εφαρμογή – Mobile App

Η κινητή εφαρμογή αναπτύχθηκε σε Flutter, αξιοποιώντας την πλατφόρμα για την υλοποίηση ενός σύγχρονου και εύχρηστου γραφικού περιβάλλοντος. Η επιλογή του Flutter έγινε λόγω της δυνατότητας ανάπτυξης εγγενών (native-like) εφαρμογών για πολλαπλές πλατφόρμες (Android, iOS) με ενιαίο κώδικα, προσφέροντας υψηλή απόδοση, ομαλά animations και συνεπή εμπειρία χρήστη. Επιπλέον, το ευρύ οικοσύστημα βιβλιοθηκών και η ταχεία ανάπτυξη μέσω `hot reload` συνέβαλαν στην επιτάχυνση του κύκλου ανάπτυξης και δοκιμών.

Ο στόχος ήταν η ανάπτυξη ενός εργαλείου που επιτρέπει στους τελικούς χρήστες να αλληλεπιδρούν εύκολα και γρήγορα με το μοντέλο ταξινόμησης, χωρίς να απαιτούνται τεχνικές γνώσεις.

Η βασική ροή λειτουργίας της εφαρμογής είναι η εξής:



Εικόνα 49: Flowchart Χρήσης Εφαρμογής Κινητού

Ο χρήστης μπορεί να τραβήξει φωτογραφία του κλιματιστικού του μέσω της κάμερας της κινητής συσκευής ή να επιλέξει μία υπάρχουσα εικόνα από τη συλλογή του. Η εικόνα προβάλλεται στον χρήστη, ο οποίος έχει τη δυνατότητα να καθορίσει αν επιθυμεί να περικόψει (crop) την περιοχή ενδιαφέροντος. Μετά την τελική επιλογή, η φωτογραφία αποστέλλεται μέσω HTTP POST αιτήματος στο endpoint `/predict` του Flask server.

Ο server επεξεργάζεται την εικόνα, εκτελεί το ONNX μοντέλο CN-CNN και επιστρέφει τις πέντε (Top-5) πιο πιθανές προβλέψεις. Στην οθόνη αποτελεσμάτων, η εφαρμογή εμφανίζει αρχικά την πρόβλεψη με τη μεγαλύτερη πιθανότητα (Top-1). Ο χρήστης μπορεί να αποδεχθεί την προτεινόμενη κατηγορία ή, αν αυτή δεν είναι σωστή, να επιλέξει να εμφανιστούν και οι υπόλοιπες τέσσερις

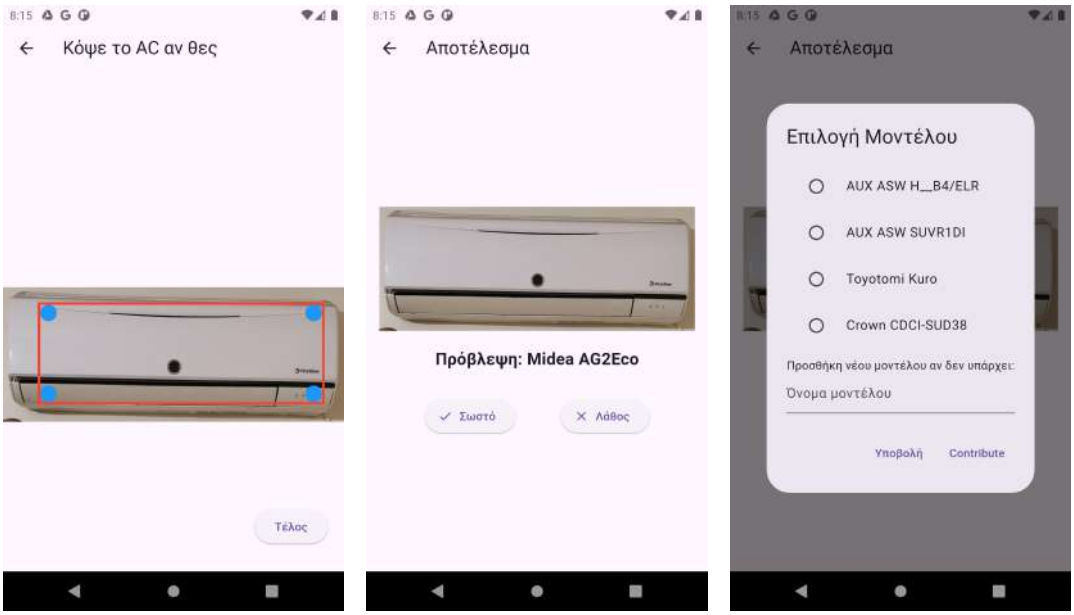
(Top-5). Για κάθε πρόβλεψη εμφανίζεται η αντίστοιχη κατηγορία (class name) και η πιθανότητα πρόβλεψης (confidence score).

Ο χρήστης μπορεί να επιλέξει την κατηγορία που θεωρεί σωστή από τις προτεινόμενες, ή εναλλακτικά να εισαγάγει νέα κατηγορία (custom label), σε περίπτωση που η συσκευή δεν αντιπροσωπεύεται επαρκώς από τις υπάρχουσες προβλέψεις. Μετά την επιλογή, η τελική απόφαση αποστέλλεται στο endpoint `/confirm`, και ο server ενημερώνει τον πίνακα `images` στο Supabase με το τελικό αποτέλεσμα (selected_class_name), διατηρώντας έτσι ιστορικό των αλληλεπιδράσεων του χρήστη με το σύστημα.

Τέλος, εάν η σωστή κατηγορία δεν περιλαμβάνεται στην πρώτη προτεινόμενη επιλογή, ο χρήστης έχει τη δυνατότητα να συμβάλει ενεργά στη βελτίωση του συστήματος. Πιο συγκεκριμένα, μπορεί να ανεβάσει επιπλέον φωτογραφίες της συσκευής του, οι οποίες αποθηκεύονται με σκοπό να χρησιμοποιηθούν σε μελλοντικά στάδια επανεκπαίδευσης του μοντέλου, ενισχύοντας έτσι την κάλυψη της βάσης δεδομένων και επεκτείνοντας τις δυνατότητες ταξινόμησης. Με αυτόν τον τρόπο, ενσωματώνεται ένας μηχανισμός συνεχούς εξέλιξης του συστήματος, στηριζόμενος στη συνεισφορά της κοινότητας χρηστών.

Η εφαρμογή επικοινωνεί αποκλειστικά με τον Flask server μέσω REST API, χωρίς απευθείας σύνδεση με το Supabase. Με αυτόν τον τρόπο διατηρείται σαφής διαχωρισμός ανάμεσα στο frontend και στο backend του συστήματος, ενισχύοντας την ασφάλεια και τη modular σχεδίαση της εφαρμογής.

Ενδεικτικά στιγμιότυπα της εφαρμογής παρατίθενται στις εικόνες που ακολουθούν, αναδεικνύοντας την εμπειρία χρήστη σε κάθε βήμα της διαδικασίας.

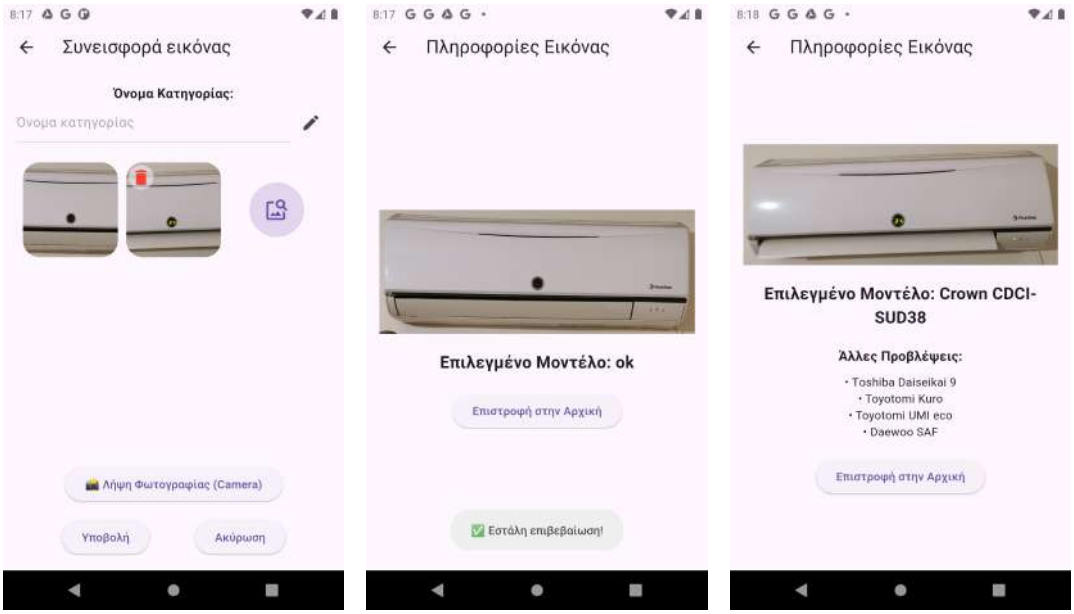


Επιλογή & περικοπή εικόνας

Εμφάνιση επικρατέστερης πρόβλεψης

Σε λάθος πρόβλεψη επιλογή σωστού μοντέλου

Εικόνα 50: Βήματα εφαρμογής από χρήση ταξινομητή



Επιλογή συνεισφοράς από συλλογή ή κάμερα

Αποτελέσματα από σωστή πρόβλεψη ή συνεισφορά

Αποτελέσματα από επιλογή εκτός top1 προβλέψεων

Εικόνα 51: Συνεισφορά και τελικά αποτελέσματα

6 Συμπεράσματα και Μελλοντική Έρευνα

6.1 Συνολική Απόδοση του Συστήματος

Η συνολική απόδοση του ανεπτυγμένου συστήματος ταξινόμησης αποδείχθηκε ιδιαίτερα ικανοποιητική, λαμβάνοντας υπόψη τη δυσκολία του προβλήματος και τους περιορισμούς του διαθέσιμου dataset. Γενικότερα η αρχιτεκτονική του CN-CNN πέτυχε σταθερά υψηλές τιμές ακρίβειας και παρουσίασε καλή ικανότητα γενίκευσης σε δεδομένα διαφορετικών συνθηκών λήψης.

Παρά τον περιορισμένο αριθμό εικόνων ανά κατηγορία και την απουσία μεγάλων δημόσιων βάσεων για το συγκεκριμένο πρόβλημα, το σύστημα κατόρθωσε να αναδείξει σαφή και χρήσιμα αποτελέσματα, επιβεβαιώνοντας τη βιωσιμότητα της προσέγγισης. Επιπλέον, η απόδοση διατηρήθηκε σταθερή ακόμη και μετά από παρατεταμένη εκπαίδευση και ποικιλία augmentations, χωρίς ενδείξεις υπερεκπαίδευσης.

Συνολικά, τα αποτελέσματα τεκμηριώνουν την καταλληλότητα της αρχιτεκτονικής και του pipeline που σχεδιάστηκε, ενώ καταγράφουν μία θετική βάση για μελλοντική επέκταση και βελτίωση του συστήματος.

6.2 Προτάσεις για Μελλοντική Έρευνα

Η ανάπτυξη ενός ευέλικτου και αποδοτικού συστήματος ταξινόμησης σε προβλήματα λεπτομερούς αναγνώρισης (FGVC) προϋποθέτει συνεχή βελτίωση σε πολλαπλά επίπεδα. Στην παρούσα ενότητα προτείνονται μελλοντικές κατευθύνσεις έρευνας που μπορούν να ενισχύσουν περαιτέρω τη λειτουργικότητα, την ακρίβεια και τη γενίκευση του συστήματος.

6.2.1 Επέκταση του dataset

Η ενίσχυση του dataset αποτελεί θεμελιώδη άξονα για την περαιτέρω βελτίωση της ακρίβειας και της γενίκευσης του μοντέλου. Πέρα από την αύξηση του πλήθους εικόνων ανά κλάση, εξίσου σημαντική είναι η ποικιλομορφία των δεδομένων όσον αφορά διαφορετικές γωνίες λήψης, φωτιστικές συνθήκες, περιβάλλοντα και φυσική κατάσταση των προϊόντων. Η εισαγωγή δεδομένων από ετερογενείς πηγές (όπως crowdsourcing, open datasets, φωτογραφίες από καταναλωτές) δύναται να ενισχύσει τη ρεαλιστικότητα του dataset.

Όλα τα υλοποιημένα μοντέλα υποστηρίζουν αρθρωτή (modular) επανεκπαίδευση, επιτρέποντας την ενσωμάτωση νέων δεδομένων χωρίς την ανάγκη πλήρους ανακατασκευής του μοντέλου. Η εκπαίδευση μπορεί να συνεχιστεί με εμπλουτισμένο σύνολο δεδομένων, π.χ. μέσω αύξησης εικόνων με κατάλληλη προεπεξεργασία.

Σε περίπτωση προσθήκης νέων κλάσεων, τα υπάρχοντα βάρη των επιπέδων εκτός του τελικού ταξινομητή μπορούν να επαναχρησιμοποιηθούν, επιτρέποντας την εκπαίδευση ενός νέου, πιο γενικευμένου μοντέλου με μειωμένο υπολογιστικό κόστος, με χρήση δηλαδή transfer learning.

6.2.2 Ενσωμάτωση χωρικών μεταβλητών και μέτρησης κλίμακας

Η διάκριση προϊόντων με παρόμοια μορφή αλλά διαφορετικές διαστάσεις (π.χ. μοντέλα κλιματιστικών με διαφορετική ισχύ) είναι δύσκολη με βάση μόνο τα texture-based χαρακτηριστικά. Η ενσωμάτωση χωρικών μεταβλητών ή πληροφορίας κλίμακας μπορεί να βελτιώσει αισθητά την ακρίβεια. Αυτό μπορεί να επιτευχθεί με:

- Χρήση auxiliary inputs (π.χ. μεταδεδομένα με αριθμητικά χαρακτηριστικά).
- Εξαγωγή αναλογιών bounding boxes.
- Εκτίμηση σχετικού μεγέθους με reference αντικείμενα (π.χ. τοίχοι, πρίζες).
- Χρήση spatial embeddings ή depth maps σε συνδυαστικά CNN+Geometry συστήματα.

6.2.3 Αυτόματη Βελτιστοποίηση Μετασχηματισμών (AutoAugment)

Αν και οι κλασικές τεχνικές data augmentation ενισχύουν την γενίκευση, η χρήση στατικών και μη βελτιστοποιημένων μετασχηματισμών ενδέχεται να περιορίσει την απόδοση. Μελλοντική εργασία θα μπορούσε να αξιοποιήσει τεχνικές όπως AutoAugment [63] και RandAugment [64], που χρησιμοποιούν αλγόριθμους policy search για την αυτόματη εύρεση των βέλτιστων μετασχηματισμών και παραμέτρων.

Μια τέτοια προσέγγιση θα επέτρεπε τη δημιουργία δυναμικών, dataset-specific augmentations που βελτιώνουν την ανθεκτικότητα σε θόρυβο, φωτεινότητα και άλλες παραμορφώσεις, μειώνοντας ταυτόχρονα τον κίνδυνο overfitting.

6.2.4 Αξιοποίηση Vision Transformers

Τα Vision Transformers (π.χ. ViT, Swin Transformer, CoaT, EfficientFormer)[55] ενσωματώνουν μηχανισμούς global self-attention, επιτρέποντας την κατανόηση μακρινών συσχετίσεων εντός της εικόνας. Σε αντίθεση με τα CNNs, τα οποία εστιάζουν κυρίως σε τοπικά χαρακτηριστικά, οι ViTs επιτρέπουν την ταυτόχρονη επεξεργασία πληροφορίας από διαφορετικές περιοχές, καθιστώντας τους ιδιαίτερα αποτελεσματικούς σε FGVC προβλήματα.

Η υβριδική ενοποίηση CNN- και Transformer-based δομών ή η αντικατάσταση των convolutional επιπέδων με attention blocks συνιστούν υποσχόμενες κατευθύνσεις για τη βελτίωση της εκφραστικότητας του μοντέλου.

6.2.5 Self-supervised και semi-supervised learning

Η απόκτηση πλήρως επισημασμένων δεδομένων αποτελεί ένα από τα μεγαλύτερα εμπόδια στη μαζική υιοθέτηση των συστημάτων FGVC. Η αξιοποίηση self-supervised, όπως SimCLR [65], MoCo v2, DINO καθώς και semi-supervised, όπως FixMatch [66], Mean Teacher τεχνικών μπορεί να συμβάλει στην εκπαίδευση ποιοτικών μοντέλων ακόμα και σε περιβάλλοντα με περιορισμένη διαθεσιμότητα ετικετών.

Μια τέτοια προσέγγιση είναι ιδιαίτερα χρήσιμη σε βιομηχανικά σενάρια όπου η επισημείωση απαιτεί ανθρώπινη εξειδίκευση και είναι χρονοβόρα.

6.2.6 Ανάλυση σφαλμάτων και στοχευμένη εκπαίδευση

Η ενδελεχής ανάλυση των σφαλμάτων του μοντέλου (π.χ. μέσω confusion matrix) μπορεί να αποκαλύψει περιοχές με υψηλή αλληλοεπικάλυψη κατηγοριών. Σε τέτοιες περιπτώσεις, η χρήση contrastive loss functions, hard triplet mining ή targeted oversampling μπορεί να οδηγήσει σε εντοπισμένη βελτίωση της διακριτικής ικανότητας.

Επιπλέον, η χρήση τεχνικών meta-learning ή few-shot learning μπορεί να βοηθήσει στην ενίσχυση ”αδύναμων” περιοχών του embedding space χωρίς ανάγκη για πλήρη αναεκπαίδευση του μοντέλου.

6.2.7 Συμπερασματικά

Οι παραπάνω ερευνητικές κατευθύνσεις μπορούν να οδηγήσουν στην ανάπτυξη ενός περισσότερο ευέλικτου, επεκτάσιμου και ερμηνεύσιμου συστήματος ταξινόμησης. Ιδίως σε σενάρια συνεχώς εξελισσόμενων δεδομένων και περιορισμένων πόρων για επισημείωση, οι εν λόγω τεχνικές μπορούν να διασφαλίσουν τη μακροχρόνια αξιοπιστία και προσαρμοστικότητα της λύσης.

Βιβλιογραφία

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[3] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[4] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 2008.

[5] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters", *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[6] W. K. Pratt, *Digital Image Processing: PIKS Inside*. Wiley-Interscience, 2007.

[7] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[8] I. Sobel, "An isotropic 3x3 image gradient operator", Stanford Research Institute, Tech. Rep. Report 69-1, 1970.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features", in *European Conference on Computer Vision (ECCV)*, 2006, pp. 404–417.

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson Education, 2010.

[12] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[13] T. Ayodele, "Types of machine learning algorithms", *IntechOpen*, 2010. DOI: 10.5772/9385.

[14] S. Sah, "Machine learning: A review of learning types", *Preprints*, 2020. DOI: 10.20944/preprints202007.0230.v1.

[15] H. Horlings and M. Vijver, "Clinical genomics in oncology", in *Molecular Genetic Pathology: Second Edition*, Springer, 2014, ISBN: 978-1-58829-974-1. DOI: 10.1007/978-1-59745-405-6_8.

[16] E. Arwa and K. Folly, "Reinforcement learning techniques for optimal power control in grid-connected microgrids: A comprehensive review", *IEEE Access*, vol. 8, pp. 1–16, Nov. 2020. DOI: 10.1109/ACCESS.2020.3038735.

[17] Y. Bengio, A. Courville, and P. Vincent, "Learning deep architectures for ai", *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2013.

[18] R. Khudeyer and N. Almoosawi, "Combination of machine learning algorithms and resnet50 for arabic handwritten classification", *Informatica*, vol. 46, Jan. 2023. DOI: 10.31449/inf.v46i9.4375.

[19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[20] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[21] S. Bhattacharjee, S. Bhattacharyya, and N. Mondal, "Quantum backpropagation neural network approach for modeling of phenol adsorption from aqueous solution by orange peel ash", in *Handbook of Research on Computational Intelligence for Engineering, Science, and Business*, 2013, pp. 649–671. DOI: 10.4018/978-1-46662-518-1.ch025.

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] N. Nedjah, I. Santos, and L. Mourelle, "Sentiment analysis using convolutional neural network via word embeddings", *Evolutionary Intelligence*, vol. 15, Apr. 2019. DOI: 10.1007/s12065-019-00227-4.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] P. Venugopal and V. T., "State-of-health estimation of li-ion batteries in electric vehicle using indrnn under variable load condition", *Energies*, vol. 12, p. 4338, Nov. 2019. DOI: 10.3390/en12224338.

[27] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *arXiv preprint arXiv:1409.1556*, 2014.

[33] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning", *arXiv preprint arXiv:1603.07285*, 2016.

[34] F. Chollet, *Deep Learning with Python*, 2nd ed. Manning Publications, 2021.

[35] M. Lin, Q. Chen, and S. Yan, "Network in network", in *arXiv preprint arXiv:1312.4400*, 2013.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[37] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1735–1742.

[38] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[39] H. Wang, Y. Wang, Z. Zhou, X. Ji, and D. Gong, "Cosface: Large margin cosine loss for deep face recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5265–5274.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[41] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of AISTATS 2010*, 2010.

[42] P. Purnawansyah, H. H., H. Darwis, H. Azis, and Y. Salim, "Backpropagation neural network with combination of activation functions for inbound traffic prediction", *Knowledge Engineering and Data Science*, vol. 4, p. 14, Aug. 2021. DOI: 10.17977/um018v4i12021p14-28.

[43] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models", in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[44] B. Xu, N. Wang, T. Chen, and M. Li, *Empirical evaluation of rectified activations in convolutional networks*, https://arxiv.org/abs/1505.00853, arXiv preprint arXiv:1505.00853, 2015.

[45] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)", *arXiv preprint arXiv:1606.08415*, 2016.

[46] G. Franke and H. Degen, "The softmax function: Properties, motivation, and interpretation", *arXiv preprint arXiv:2303.01541*, 2023. [Online]. Available: https://arxiv.org/abs/2303.01541.

[47] D. Misra, "Mish: A self regularized non-monotonic neural activation function", *arXiv preprint arXiv:1908.08681*, 2019. [Online]. Available: https://arxiv.org/abs/1908.08681.

[48] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions", *arXiv preprint arXiv:1710.05941*, 2017. [Online]. Available: https://arxiv.org/abs/1710.05941.

[49] L. Bottou, "Large-scale machine learning with stochastic gradient descent", in *Proceedings of COMPSTAT*, 2010, pp. 177–186.

[50] N. Qian, "On the momentum term in gradient descent learning algorithms", *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[52] C. Szegedy, W. Liu, Y. Jia, *et al.*, *Going deeper with convolutions*, 2014. arXiv: arXiv:1409.4842 [cs.CV].

[53] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning", in *arXiv preprint arXiv:1602.07261v2*, 2016.

[54] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes", *arXiv preprint arXiv:2106.04803*, 2021.

[55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", in *International Conference on Learning Representations (ICLR)*, 2021.

[56] A. Vaswani and et al., "Attention is all you need", in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[57] H.-C. Shin, H. R. Roth, M. Gao, *et al.*, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning", *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016. DOI: 10.1109/TMI.2016.2528162.

[58] Z. Ge, A. Bewley, and C. McCool, "Fine-grained classification via mixture of deep convolutional neural networks", in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2016, pp. 1–9.

[59] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts", *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[60] C. Guo, J. Xie, K. Liang, X. Sun, and Z. Ma, "Cross-layer navigation convolutional neural network for fine-grained visual classification", *arXiv preprint arXiv:2106.10920*, 2021.

[61] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, *Hierarchical bilinear pooling for fine-grained visual recognition*, https://arxiv.org/abs/1807.09915, 2018.

[62] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1449–1457.

[63] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[64] E. D. Cubuk, H. Touvron, B. Zoph, Q. V. Le, and F. Hutter, "Randaugment: Practical automated data augmentation with a reduced search space", in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 613–18 624.

[65] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations", in *International Conference on Machine Learning*, PMLR, 2020, pp. 1597–1607.

[66] K. Sohn, D. Berthelot, C. Li, *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence", in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 596–608.