ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Συγκριτική Αξιολόγηση Μοντέλων Βαθιάς Μάθησης για Προβλήματα Όρασης Υπολογιστών σε Εφαρμογές Κινητών Συσκευών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Δερέμπεης Στυλιανής-Αποστολίας

Επιβλέπων: Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Συγκριτική Αξιολόγηση Μοντέλων Βαθιάς Μάθησης για Προβλήματα Όρασης Υπολογιστών σε Εφαρμογές Κινητών Συσκευών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Δερέμπεης Στυλιανής-Αποστολίας

Επιβλέπων: Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Νοεμβρίου 2021.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
……	……	……
Ι. Βενιέρης	**Δ.-Θ. Κακλαμάνη**	**Γ. Ματσόπουλος**
Καθηγητής Ε.Μ.Π.	**Καθηγήτρια Ε.Μ.Π.**	**Καθηγητής Ε.Μ.Π.**

Αθήνα, Νοέμβριος 2021

(Υπογραφή)

……

Στυλιανή-Αποστολία Δερέμπεη
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Ευχαριστίες

Με την ολοκλήρωση των προπτυχιακών σπουδών μου, θα ήθελα να ευχαριστήσω κάποιους ανθρώπους, οι οποίοι υπήρξαν καθοριστικοί για την πορεία μου αυτά τα χρόνια.

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή του Εθνικού Μετσόβιου Πολυτεχνείου κ. Ιάκωβο Βενιέρη, για την επίβλεψη αυτής της διπλωματικής εργασίας και για την στήριξη σε όποια δυσκολία αντιμετώπισα στην εκπόνηση αυτής. Ακόμη ευχαριστώ την κ. Δήμητρα-Θεοδώρα Κακλαμάνη και τον κ. Γεώργιο Ματσόπουλο για τη συμμετοχή τους στην τριμελή εξεταστική επιτροπή.

Επίσης, θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα ΕΜΠ κ. Ιωάννη Πανόπουλο, χωρίς τη βοήθεια και τις ιδέες του οποίου δεν θα μπορούσα να ολοκληρώσω αυτή την εργασία.

Τελευταίοι αλλά πολύ σημαντικοί άνθρωποι που θέλω να ευχαριστήσω είναι οι γονείς μου και η αδερφή μου για όλη τη στήριξη που μου έχουν δείξει όλα αυτά τα χρόνια και την εμψύχωση που μου παρείχαν σε στιγμές που χρειαζόμουν για την ολοκλήρωση των σπουδών μου.

Αθήνα, Νοέμβριος 2021
Στυλιανή-Αποστολία Δερέμπεη

Περίληψη

Η ενσωμάτωση της Βαθιάς Μάθησης στις κινητές συσκευές αντιμετωπίζει μέχρι και σήμερα αρκετές προκλήσεις. Αν και τα τελευταία χρόνια έχουν γίνει τεράστια άλματα, δεν υπάρχει ακόμη ένα ενοποιημένο σύστημα - πλαίσιο που να επιτρέπει στους προγραμματιστές να εισάγουν εύκολα και γρήγορα βαθιά νευρωνικά δίκτυα στις εφαρμογές τους. Επομένως, κρίνεται αναγκαίο να γνωρίζουμε την απόδοση ευρέως γνωστών αρχιτεκτονικών βαθιών νευρωνικών δικτύων σε κοινές, ευρέως χρησιμοποιούμενες κινητές συσκευές.

Οι προκλήσεις της Βαθιάς Μάθησης στις κινητές συσκευές μπορούν να συνοψιστούν (α) στην ποικιλομορφία στις δυνατότητες επεξεργασίας, που οδηγεί σε ευρεία ετερογένεια και μη συνεπή απόδοση μεταξύ των κινητών συσκευών, (β) στην ποικιλία των μοντέλων Βαθιάς Μάθησης, όσον αφορά στο πρόβλημα, την αρχιτεκτονική και τις απαιτήσεις πόρων, και (γ) στην μεταβλητότητα των απαιτήσεων απόδοσης σχετικά με την ακρίβεια, την καθυστέρηση, τη μνήμη και την ενέργεια σε όλες τις εφαρμογές.

Στην παρούσα διπλωματική εργασία εκτελείται μια συγκριτική αξιολόγηση της απόδοσης μοντέλων Βαθιάς Μάθησης σε πραγματικά σενάρια εφαρμογών κινητών συσκευών. Γίνεται χρήση ενός πλήθους από δίκτυα, εκπαιδευμένα σε διαφορετικά προβλήματα Βαθιάς Μάθησης, ενώ παράλληλα εξετάζονται διάφοροι τρόποι κβαντοποίησης. Οι είσοδοι των μοντέλων είναι εικόνες και προκύπτουν περιοδικά από τον αισθητήρα της κάμερας της κινητής συσκευής, ενώ για την εκτέλεση της συμπερασματολογίας υπάρχει η δυνατότητα επιλογής όλων των διαθέσιμων επεξεργαστών.

Τα αποτελέσματα δείχνουν ότι είναι δύσκολο να βρεθεί ένα πλήρως αντιπροσωπευτικό δείγμα μοντέλων και συσκευών με βάση το οποίο να μπορούν να εξαχθούν ορθά συμπεράσματα και για άλλες περιπτώσεις. Επομένως, κατά την ενσωμάτωση της Βαθιάς Μάθησης σε δυναμικά περιβάλλοντα κινητού υπολογισμού κρίνεται απαραίτητο να υπάρχει ένα υποσύστημα, το οποίο θα ελέγχει συνεχώς τις δυναμικές παραμέτρους του συστήματος και θα προσαρμόζει κατάλληλα τις παραμέτρους της εφαρμογής, όπως είναι το μοντέλο, το σημείο εκτέλεσης, κ.α.

Λέξεις Κλειδιά

Βαθιά Μάθηση, Κινητές Συσκευές, Συνελικτικά Νευρωνικά Δίκτυα, Αναγνώριση Αντικειμένων, Κατάτμηση Εικόνας, Εκτίμηση Ανθρώπινης Πόζας, Android, Εφαρμογή Κάμερας, TensorFlow Lite, Συγκριτική Αξιολόγηση

Abstract

The integration of Deep Learning into mobile devices still faces several challenges. Although huge leaps have been made in recent years, there is not yet a unified system or framework that allows developers to easily and quickly utilize deep neural networks into their applications. Therefore, it is necessary to have knowledge of the performance of well-known deep neural network architectures on common, widely used mobile devices.

The challenges of Deep Learning on mobile devices can be summarized in (a) the diversity of processing capabilities, leading to wide heterogeneity and inconsistent performance between mobile devices, (b) the variety of Deep Learning models, in terms of task, architecture and resource requirements, and (c) the variability of performance requirements regarding accuracy, latency, memory, and energy across applications.

In this thesis we benchmark Deep Learning models in real mobile application scenarios. A number of networks are used, trained in different Deep Learning tasks, while at the same time various quantization techniques are examined. The inputs of the models are images and periodically emerge from the sensor of the camera of the mobile device, while for the execution of the inference there is the possibility of selecting all available processors.

Results show that it is difficult to find a fully representative sample of models and devices on the basis of which correct conclusions can be drawn for other use cases. Therefore, when integrating Deep Learning in dynamic mobile computing environments, it is necessary to have a subsystem, which will constantly control the dynamic parameters of the wider system and will adapt the parameters of the application, such as the model, the execution point, etc.

Keywords

Deep Learning, Mobile Devices, Convolutional Neural Networks, Object Detection, Semantic Segmentation, Pose Estimation, Android, Camera Application, TensorFlow Lite, Benchmarking

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1: Εισαγωγή

Στο 1ο Κεφάλαιο θα παρουσιαστεί το αντικείμενο της διπλωματικής εργασίας και στη συνέχεια θα αναλυθεί η δομή της μαζί με το περιεχόμενο κάθε Κεφαλαίου.

1.1 Αντικείμενο Διπλωματικής Εργασίας

Τα τελευταία χρόνια, η ταχεία ανάπτυξη των κινητών συσκευών σε συνδυασμό με την εξαιρετική επίδοση των βαθιών νευρωνικών δικτύων στην επίλυση πολύπλοκων προβλημάτων (κατηγοριοποίηση εικόνας, εντοπισμός αντικειμένων, αναγνώριση φωνής, μοντελοποίηση κειμένου) έχουν οδηγήσει στην ανάπτυξη ευφυών εφαρμογών που σέβονται την ιδιωτικότητα του χρήστη και παρέχουν την απαιτούμενη ποιότητα υπηρεσίας.

Οι δύο βασικές προσεγγίσεις στην εκτέλεση συμπερασματολογίας βαθιών νευρωνικών δικτύων στα πλαίσια εφαρμογών κινητών συσκευών είναι: (α) η αποστολή των δειγμάτων εισόδου σε κάποιον απομακρυσμένο εξυπηρετητή και η επιστροφή των αποτελεσμάτων πίσω στην εφαρμογή και (β) η χρήση των τοπικών πόρων (υπολογιστική ισχύς, μνήμη, μπαταρία) της κινητής συσκευής. Η επιλογή του σημείου στο οποίο θα εκτελείται η συμπερασματολογία μπορεί να εξαρτηθεί από τον τύπο της εφαρμογής (π.χ. real-time) ή από τους στόχους επίδοσης (π.χ. ακρίβεια, FPS, διαπερατότητα). Με την τοπική εκτέλεση τα δεδομένα δεν αποχωρίζονται τη συσκευή του χρήστη, οπότε διασφαλίζεται η ιδιωτικότητα και ακόμη δεν απαιτείται συνεχής σύνδεση στο διαδίκτυο, αυξάνοντας την αξιοπιστία.

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η ενσωμάτωση διαφόρων αρχιτεκτονικών βαθιών συνελικτικών δικτύων σε μια εφαρμογή Android «έξυπνης» κάμερας και η συγκριτική αξιολόγηση της απόδοσης τους κάτω από διαφορετικές διαμορφώσεις του περιβάλλοντος εκτέλεσης. Κατά την περίοδο που γράφεται η εργασία, το 72-73% των συνολικών χρηστών κατέχει συσκευές με λειτουργικό σύστημα Android [1], οπότε τα συμπεράσματα που εξάγονται στη συνέχεια αντιπροσωπεύουν την πλειονότητα των χρηστών κινητών συσκευών.

Σε σύγκριση με την Κατηγοριοποίηση Εικόνας, πιο περίπλοκα προβλήματα Όρασης Υπολογιστών δεν έχουν διερευνηθεί επαρκώς στα πλαίσια εφαρμογών κινητών συσκευών. Για αυτό τον λόγο, η εφαρμογή έχει αναπτυχθεί ώστε να υποστηρίζει μοντέλα που έχουν εκπαιδευτεί για Κατάτμηση Εικόνας, Ανίχνευση Αντικειμένων και Εκτίμηση Ανθρώπινης Πόζας. Για την αξιολόγηση του συστήματος χρησιμοποιήθηκαν δύο κινητές συσκευές διαφορετικού τύπου, ένα smartphone και ένα tablet, ενώ λήφθηκαν υπόψη: (α) η πτώση ακρίβειας στα μοντέλα λόγω βελτιστοποιήσεων, όπως είναι η κβαντοποίηση και (β) μετρικές σχετικές με την καθυστέρηση στην εκτέλεση, όπως είναι το 90th percentile.

1.2 Οργάνωση

Στο 2ο Κεφάλαιο θα παρουσιαστούν τα βασικά θεωρητικά στοιχεία που πρέπει να γνωρίζει κανείς για να μπορέσει να παρακολουθήσει με ευκολία τη ροή της εργασίας.

Στο 3ο Κεφάλαιο θα αναλυθούν οι βασικές διεργασίες (tasks) Βαθιάς Μάθησης που θα ενσωματωθούν στην εφαρμογή. Θα περιγραφούν τα πιο δημοφιλή σύνολα δεδομένων που χρησιμοποιούνται για κάθε διεργασία και οι σημαντικότερες μετρικές για την αξιολόγηση των μοντέλων.

Στο 4ο Κεφάλαιο θα περιγραφούν οι τεχνολογίες που αξιοποιήθηκαν για την εκπόνηση της εργασίας. Συνοπτικά, αναφέρεται η ιστορία της γλώσσας προγραμματισμού Java που χρησιμοποιήθηκε κατά τη δημιουργία της εφαρμογής στο προγραμματιστικό περιβάλλον Android Studio. Επίσης, θα παρουσιαστεί η γλώσσα προγραμματισμού Python με την οποία έγιναν οι μετατροπές των μοντέλων Βαθιάς Μάθησης που ενσωματώθηκαν στην εφαρμογή.

Στο 5ο Κεφάλαιο θα αναφερθούν τα δομικά στοιχεία και οι λειτουργίες της εφαρμογής. Θα δοθούν παραδείγματα και κάποια στιγμιότυπα από την εκτέλεση της εφαρμογής για να γίνει κατανοητή η λειτουργία της.

Στο 6ο Κεφάλαιο θα παρουσιαστούν οι κινητές συσκευές που χρησιμοποιήθηκαν και θα αναλυθούν τα μοντέλα και τα χαρακτηριστικά τους για κάθε διεργασία Βαθιάς Μάθησης.

Στο 7ο Κεφάλαιο θα παρουσιαστούν οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των μοντέλων, θα παρουσιαστούν συνοπτικές μετρήσεις και τα άμεσα συμπεράσματα που εξάγονται από αυτές.

Τέλος, στο 8ο Κεφάλαιο θα αναλυθούν τα γενικά συμπεράσματα που προκύπτουν και θα επισημανθούν βελτιώσεις και επεκτάσεις που θα μπορούσαν να αποτελέσουν μελλοντική εργασία.

Κεφάλαιο 2: Θεωρητικό Υπόβαθρο

Τις τελευταίες δεκαετίες η τεχνολογία έχει αναπτυχθεί σε τέτοιο βαθμό, ώστε να υπάρχει η δυνατότητα για ανάπτυξη αυτόνομων συστημάτων που σχετίζονται με ανθρώπινες λειτουργίες. Ουσιαστικά δημιουργούνται με κώδικα σε γλώσσα υπολογιστών μέθοδοι που ο ανθρώπινος εγκέφαλος πραγματοποιεί καθημερινά, όπως είναι η κατανόηση αντικειμένων. Το πεδίο που ασχολείται με αυτές τις λειτουργίες ονομάζεται Τεχνητή Νοημοσύνη (Artificial Intelligence - AI).

Η Τεχνητή Νοημοσύνη είναι ένας τομέας που σε συνδυασμό με την Επιστήμη των Υπολογιστών και των συνόλων δεδομένων επιλύει διάφορα προβλήματα. Περιλαμβάνει τα πεδία της Μηχανικής και της Βαθιάς Μάθησης και είναι στενά συνδεδεμένη με την Όραση Υπολογιστών, έννοιες που αναλύονται στο παρόν Κεφάλαιο.

2.1 Όραση Υπολογιστών

Η Όραση Υπολογιστών (Computer Vision - CV) είναι ένα πεδίο της Τεχνητής Νοημοσύνης που επιτρέπει στους υπολογιστές να αντλήσουν σημαντικές πληροφορίες από διάφορες ψηφιακές οπτικές εισόδους, όπως οι εικόνες και τα βίντεο. Με απλούς συσχετισμούς θα μπορούσε να ειπωθεί ότι η Τεχνητή Νοημοσύνη επιτρέπει στους υπολογιστές να σκέφτονται και η Όραση Υπολογιστών τους βοηθά να βλέπουν, να παρατηρούν και να κατανοούν.

Οι ηλεκτρονικοί υπολογιστές προσπαθούν να κατανοήσουν αντικείμενα σε ψηφιακές εικόνες και βίντεο με τον ίδιο τρόπο που ένας άνθρωπος βλέπει και διαισθητικά κατανοεί τις εικόνες που παρατηρεί κάθε στιγμή. Η ανθρώπινη όραση μελετά τη λειτουργία της αντίληψης οπτικών ερεθισμάτων κάτω από φυσιολογικές διαδικασίες ενώ η μηχανική όραση μελετά και περιγράφει τα τεχνητά συστήματα λογισμικού και υλικού υπολογιστών. Αυτές οι συσχετίσεις μεταξύ όρασης ανθρώπων ή ζώων και Όρασης Υπολογιστών βοηθούν στην κατανόηση και την εξέλιξη και των δύο τομέων.

Η Όραση Υπολογιστών υπάρχει σε πολλές δραστηριότητες που ο άνθρωπος έρχεται σε επαφή καθημερινά. Κάποια παραδείγματα είναι:

- Ψυχαγωγία, π.χ. οι κονσόλες παιχνιδιών.
- Ασφάλεια, π.χ. η αναγνώριση επικίνδυνων αντικειμένων ή συμπεριφορών, η πρόληψη ατυχημάτων στο οδικό δίκτυο, η ανάπτυξη των αυτόνομων οχημάτων.
- Ιατρική, με τις ακτινογραφίες και τις τομογραφίες όπου διακρίνονται λεπτομέρειες που δεν μπορεί να αντιληφθεί το ανθρώπινο μάτι.
- Βιομηχανία, π.χ. η βαθμονόμηση των βιομηχανικών ρομπότ, η ανίχνευση βεβλημένων αντικειμένων κατά τη διαδικασία παραγωγής τους, η απομάκρυνση ανεπιθύμητων ουσιών και υλικών από τα τρόφιμα της γεωργικής γραμμής παραγωγής.
- Στρατιωτικός τομέας, π.χ. τα συστήματα μη-επανδρωμένων αεροσκαφών, για την αναγνώριση αγνώστου εδάφους, εχθρικών οχημάτων και προσωπικού και άλλα στοιχεία εχθρικού ενδιαφέροντος.

Τα τελευταία χρόνια, η παραπάνω τεχνολογία συνεχώς εξελίσσεται αφού βασίζεται τόσο στο υλικό όσο και στο λογισμικό των συστημάτων που

χρησιμοποιούνται. Γίνεται συνεχώς έρευνα για να αναπτύσσονται κάμερες με μεγαλύτερη ανάλυση, για να χρησιμοποιούνται ισχυρότερα υλικά και για να υλοποιούνται γρηγορότερα τα λογισμικά. Επίσης, γίνεται προσπάθεια να βελτιώνονται οι ήδη υπάρχοντες αλγόριθμοι και οι μέθοδοι αναγνώρισης αντικειμένων ή να δημιουργούνται καινούργιες αποδοτικότερες μέθοδοι [2].

2.2 Μηχανική Μάθηση

Η Μηχανική Μάθηση (Machine Learning - ML) είναι ένα πεδίο της Τεχνητής Νοημοσύνης. Αναπτύχθηκε με σκοπό να βελτιώσει τις προβλέψεις των υπολογιστικών μηχανών χωρίς την επίβλεψη και την καθοδήγηση κάποιου ανθρώπου. Η βασική ιδέα της Μηχανικής Μάθησης είναι η ανάπτυξη αλγορίθμων που λαμβάνουν κάποια δεδομένα εισόδου και παράγουν μια πρόβλεψη σαν δεδομένο εξόδου. Αυτή η διαδικασία γίνεται με χρήση στατιστικών αναλύσεων.

Τα μοντέλα Μηχανικής Μάθησης δημιουργήθηκαν για να λύσουν το πρόβλημα που έχουν τα περισσότερα προγραμματιστικά εργαλεία. Συνήθως αυτά τα εργαλεία δεν εξελίσσονται ενώ τα δεδομένα που επεξεργάζονται μεταβάλλονται συνεχώς στο χρόνο. Έτσι η Μηχανική Μάθηση σχεδιάστηκε για να προσαρμόζεται συνεχώς στα δεδομένα εισόδου που επεξεργάζεται, ανεξάρτητα από το μέγεθος και την πολυπλοκότητα αυτών των δεδομένων. Ένα πρόγραμμα Μηχανικής Μάθησης προσαρμόζει συνεχώς τις ενέργειές του ανάλογα με τα μοτίβα που βρίσκει κατά τη σάρωση των δεδομένων.

Η Μηχανική Μάθηση έχει εισχωρήσει σε εφαρμογές που χρησιμοποιούν καθημερινά οι άνθρωποι. Για παράδειγμα, υπάρχουν στο διαδίκτυο μηχανισμοί συστάσεων (Recommender Systems - RE), με τους οποίους εμφανίζονται προσωποποιημένες διαφημίσεις ανάλογα με τις αναζητήσεις και τις αγορές του κάθε χρήστη. Άλλες εφαρμογές της Μηχανικής Μάθησης είναι οι μηχανές αναζήτησης στο διαδίκτυο, η ανίχνευση απάτης, η ανίχνευση απειλών για την ασφάλεια των δικτύων, η κατηγοριοποίηση των μηνυμάτων σε ανεπιθύμητων ή ασφαλών, η πρόβλεψη του καιρού, κ.α. [3].

Επειδή υπάρχουν πολλές διαφορετικές εφαρμογές της Μηχανικής Μάθησης, αναπτύχθηκαν πολλά διαφορετικά μοντέλα και αλγόριθμοι για κάθε σκοπό. Οι αλγόριθμοι της Μηχανικής Μάθησης μπορεί να είναι είτε περίπλοκοι στην ανάπτυξή τους και στη χρήση τους, είτε πολύ απλοί. Το ίδιο συμβαίνει και με τα μοντέλα Μηχανικής Μάθησης που χρησιμοποιούνται πιο συχνά. Τα πιο δημοφιλή μοντέλα είναι τα δένδρα αποφάσεων, ο αλγόριθμος k-means για ομαδοποίηση (clustering) και τα νευρωνικά δίκτυα (neural networks - NNs).

Οι αλγόριθμοι Μηχανικής Μάθησης κατηγοριοποιούνται σε τρεις κατηγορίες ανάλογα με τον τρόπο μάθησης:

1. **Επιβλεπόμενη Μάθηση (Supervised Learning)**. Είναι η διαδικασία όπου κατά την εκπαίδευση ο αλγόριθμος δέχεται δεδομένες εισόδους (σύνολο εκπαίδευσης) με γνωστές επιθυμητές εξόδους και προσπαθεί να «μάθει» τη συνάρτηση αντιστοίχισης. Στόχος είναι η γενίκευση της συνάρτησης για εισόδους που δεν έχουν γνωστές εξόδους. Δημοφιλή προβλήματα που χρησιμοποιούν επιβλεπόμενη μάθηση είναι:
    - Ταξινόμηση (Classification)
    - Διερμηνεία (Interpretation)
    - Πρόγνωση (Prediction)
2. **Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)**. Είναι η διαδικασία όπου κατά την εκπαίδευση ο αλγόριθμος δέχεται δεδομένες εισόδους και

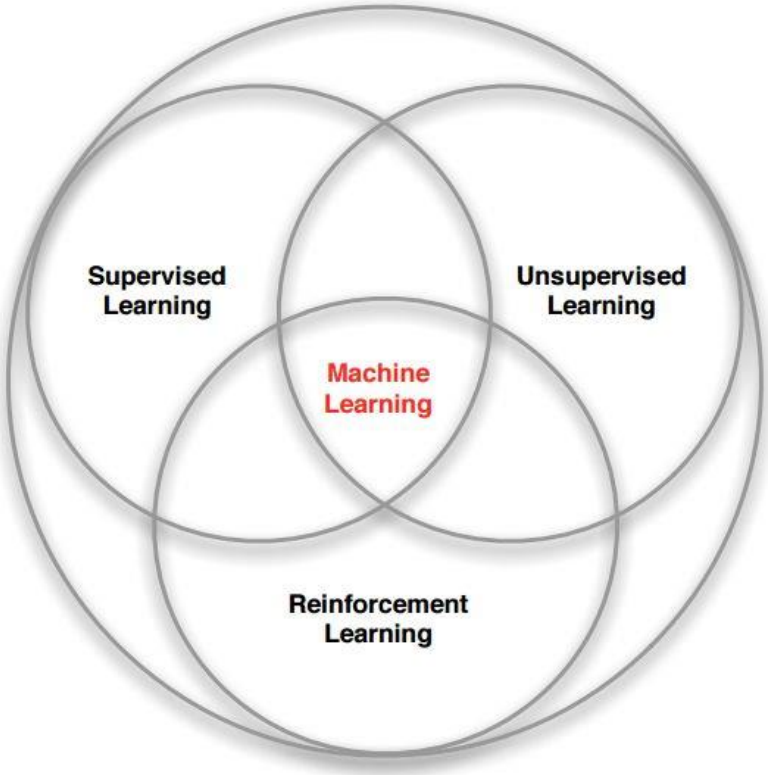
κατασκευάζει παρατηρήσεις χωρίς να γνωρίζει τις επιθυμητές εξόδους. Γνωστά προβλήματα που χρησιμοποιούν τη μη-επιβλεπόμενη μάθηση είναι:

- Ανάλυση Συσχετισμών (Association Analysis)
- Ομαδοποίηση (Clustering)

3. **Ενισχυτική Μάθηση (Reinforcement Learning)**. Η διαδικασία όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από την αλληλεπίδρασή του με το περιβάλλον. Συνήθως εφαρμόζεται σε προβλήματα Σχεδιασμού (Planning). Κάποια από αυτά είναι:

   - Βελτιστοποίηση εργασιών στη βιομηχανία
   - Έλεγχος κίνησης ρομπότ

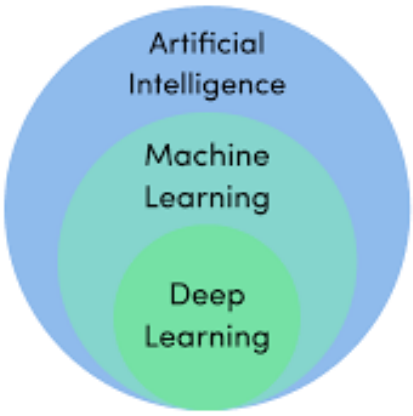
Στο Σχήμα 2.1 παρακάτω αναπαρίσταται σχηματικά ο διαχωρισμός της Μηχανικής Μάθησης στις τρεις κατηγορίες που παρουσιάστηκαν προηγουμένως [3].



Σχήμα 2.1: *Διάγραμμα Venn των διαφόρων κατηγοριών Μηχανικής Μάθησης*

2.3 Βαθιά Μάθηση

Η Βαθιά Μάθηση (Deep Learning - DL) αποτελεί υποσύνολο της Μηχανικής Μάθησης, όπως φαίνεται και στο Σχήμα 2.2 που παρατίθεται πιο κάτω. Ο όρος Βαθιά Μάθηση πρωτοεμφανίστηκε το 1986 από την Rina Dechter. Πολλές φορές, ο όρος Μηχανική Μάθηση χρησιμοποιείται για να αναφερθεί στη Βαθιά Μάθηση και αντίστροφα. Αυτό είναι λάθος αφού η δεύτερη είναι μια υποκατηγορία της πρώτης και έχουν αρκετές διαφοροποιήσεις μεταξύ τους ως προς την εκπαίδευση και τη συμμετοχή του ανθρώπου στην εξαγωγή αποτελεσμάτων.

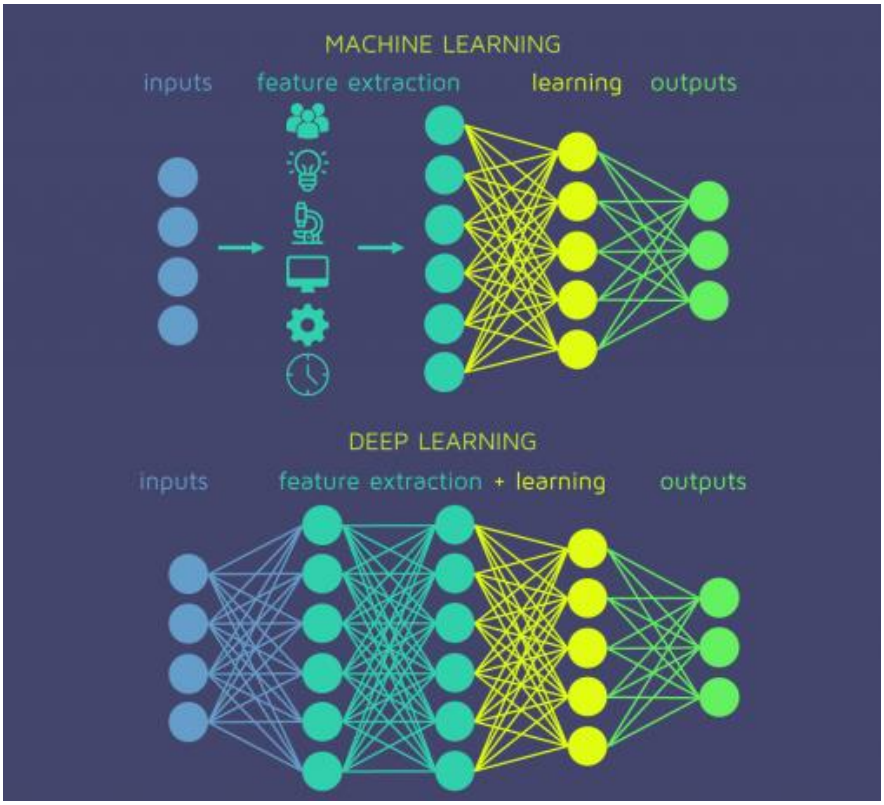


Σχήμα 2.2: *Πεδία Τεχνητής Νοημοσύνης*

Για να γίνει κατανοητή η Βαθιά Μάθηση θα αναφερθούν αναλυτικότερα οι διαφορές μεταξύ των δύο αυτών πεδίων:

1. Η πρώτη διαφορά είναι ο τρόπος που παρουσιάζονται τα δεδομένα. Στη Μηχανική Μάθηση απαιτούνται δομημένα δεδομένα που έχουν προηγουμένως επεξεργαστεί από τον άνθρωπο. Από την άλλη πλευρά, ένας αλγόριθμος Βαθιάς Μάθησης αποτελείται από την είσοδο, την έξοδο και κάποια κρυφά επίπεδα που δέχονται ακατέργαστα δεδομένα (raw data) και πραγματοποιούν επεξεργασία και ταξινόμηση αυτών των δεδομένων χωρίς την βοήθεια ανθρώπινου παράγοντα.
2. Μία άλλη σημαντική διαφορά είναι η εκπαίδευση. Η Βαθιά Μάθηση χρειάζεται μεγάλη υπολογιστική ισχύ και πολύ χρόνο εκπαίδευσης αφού εκτελεί απαιτητικές διαδικασίες. Από την άλλη πλευρά στη Μηχανική Μάθηση οι διεργασίες που απαιτούνται είναι πιο απλές αφού δέχονται δομημένα δεδομένα, μειώνοντας έτσι τόσο τον χρόνο εκπαίδευσης όσο και τις απαιτήσεις σε υπολογιστική ισχύ.
3. Τέλος, στη Βαθιά Μάθηση είναι απαραίτητη η χρήση μεγάλου όγκου δεδομένων για την σωστή λειτουργία των συστημάτων ενώ η Μηχανική Μάθηση μπορεί να πραγματοποιήσει αποδοτική εξαγωγή αποτελεσμάτων και με λιγότερα δεδομένα [3] [4].

Στο Σχήμα 2.3 παρακάτω αναπαρίστανται σχηματικά οι διαφορές μεταξύ Μηχανικής Μάθησης και Βαθιάς Μάθησης [5].



Σχήμα 2.3: *Διαφορές μεταξύ Μηχανικής Μάθησης και Βαθιάς Μάθησης*

Οι αρχιτεκτονικές που χρησιμοποιούνται στη Βαθιά Μάθηση έχουν πολλαπλά επίπεδα και το κάθε επίπεδο μετασχηματίζει τα δεδομένα εισόδου σε πιο αφηρημένη και σύνθετη αναπαράσταση [6]. Αυτό σημαίνει ότι οι μηχανές μαθαίνουν από εμπειρία και συσχετίζουν έννοιες με άλλες απλούστερες έννοιες για να κατανοήσουν τα δεδομένα που δέχονται [3]. Για παράδειγμα, σε ένα πρόβλημα επεξεργασίας εικόνων, τα χαμηλά επίπεδα μπορούν να κατανοήσουν ακμές ή γωνίες και τα υψηλότερα επίπεδα αριθμούς, αντικείμενα ή ψηφία [7].

Για την εκπαίδευση των νευρωνικών δικτύων απαιτείται μεγάλος αριθμός παράλληλων υπολογισμών λόγω του μεγάλου όγκου κρυφών επιπέδων και νευρώνων. Οι υπολογιστικές μονάδες που επιτυγχάνουν αυτούς τους υπολογισμούς ονομάζονται επιταχυντές (accelerators). Οι πιο γνωστοί επιταχυντές είναι:

- Μονάδες Επεξεργασίας Γραφικών (Graphics Processing Units - GPUs). Αρχικά σχεδιάστηκαν για τη απεικόνιση γραφικών αλλά γρήγορα αποδείχθηκαν καταλληλότερες από τις CPUs για την εκπαίδευση βαθιών νευρωνικών δικτύων.
- Μονάδες Επεξεργασίας Τανυστών (Tensor Processing Units - TPUs). Αναπτύχθηκαν το 2016 από την Google αποκλειστικά για τη συμπερασματολογία βαθιών νευρωνικών δικτύων και γρήγορα φάνηκε ότι μπορούν να επιταχύνουν και την εκπαίδευσή τους.

Η Βαθιά Μάθηση μπορεί να εφαρμοσθεί για αναγνώριση αντικειμένων, αναγνώριση φωνής, επεξεργασία φυσικής γλώσσας, ανάλυση κινήσεων, ανάλυση συναισθημάτων, επεξεργασία ιατρικών εικόνων, αυτόνομα αυτοκίνητα, κ.ά. Όλες οι παραπάνω εφαρμογές καθώς και πλήθος άλλων μπορούν να φανούν ιδιαίτερα επικερδείς, για αυτό παρατηρείται ότι οι μεγαλύτερες εταιρίες τεχνολογίας έχουν στραφεί στην αξιοποίηση της Βαθιάς Μάθησης. Μερικές από αυτές είναι η Google, η NVIDIA, η Amazon, η Facebook [8], η Amazon, η Netflix, η Apple, κ.ά. [3].

Με την ψηφιοποίηση όλο και περισσότερων δεδομένων στην σύγχρονη κοινωνία, τα σύνολα δεδομένων γίνονται ολοένα καλύτερα και μεγαλύτερα. Στη σημερινή εποχή βρίσκονται εύκολα πάνω από ένα δισεκατομμύριο δείγματα με ετικέτα τα οποία μπορούν να χρησιμοποιηθούν για την εκπαίδευση των μοντέλων Βαθιάς Μάθησης, βοηθώντας τους αλγορίθμους να γενικεύουν πιο εύκολα και πιο σωστά σε νέα άγνωστα δεδομένα. Για αυτό το λόγο η εποχή αυτή ονομάζεται και εποχή «Μεγάλων Δεδομένων» («Big Data») [3].

Οι πιο δημοφιλείς αρχιτεκτονικές Βαθιών Νευρωνικών Δικτύων είναι:

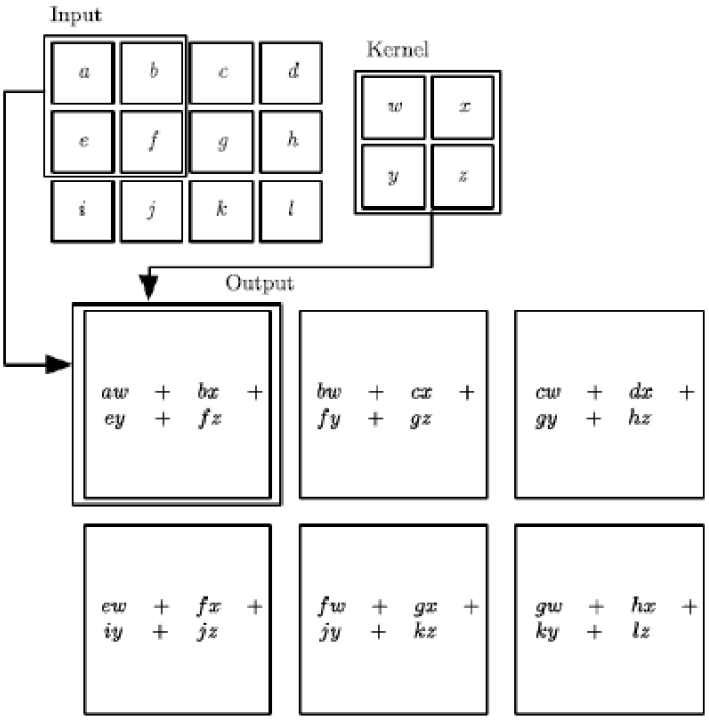
- Τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs), που μπορούν να επεξεργαστούν πληροφορία που αναπαρίσταται σε μορφή πλέγματος, όπως είναι το δισδιάστατο πλέγμα μιας εικόνας. Πιο αναλυτικά θα παρουσιαστούν στην Ενότητα 2.4.
- Τα Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNNs), που μπορούν να επεξεργαστούν ακολουθιακά δεδομένα, όπως είναι ο ήχος και το κείμενο.
- Τα Νευρωνικά Δίκτυα Εμπρόσθιας Τροφοδότησης (Feedforward Neural Networks - FNNs), που αποτελούν τα πιο απλά νευρωνικά δίκτυα. Η ονομασία τους προέρχεται από την ιδιότητά τους η πληροφορία να ρέει μόνο προς μια κατεύθυνση, έτσι δεν υπάρχουν βρόχοι.
- Τα Παραγωγικά Αντιπαραθετικά Δίκτυα (Generative Adversarial Networks - GANs), τα οποία μπορούν να διαχωριστούν σε δύο δίκτυα, τη Γεννήτρια (Generator) που παράγει λανθασμένα δεδομένα και τον Διαχωριστή (Discriminator) που αναγνωρίζει αν τα παραγόμενα δεδομένα είναι αληθή ή όχι.

2.4 Συνελικτικά Νευρωνικά Δίκτυα

Τα Συνελικτικά Νευρωνικά Δίκτυα είναι ένα είδος νευρωνικών δικτύων που χρησιμοποιείται συχνά σε προβλήματα Όρασης Υπολογιστών. Είναι γενικά αρχιτεκτονικές εμπρόσθιας τροφοδότησης (feedforward) και το όνομα τους οφείλεται στο γεγονός ότι τουλάχιστον ένα από τα επίπεδα του δικτύου έχει σαν κύρια λειτουργία του την πράξη της συνέλιξης [9].

Τα Συνελικτικά Νευρωνικά Δίκτυα προτιμώνται σε χρήση από άλλα είδη νευρωνικών δικτύων γιατί κλιμακώνουν καλά ακόμα και σε μεγάλου μεγέθους εικόνες. Για παράδειγμα τα Πλήρως Συνδεδεμένα Νευρωνικά Δίκτυα (FCNs) για μεγάλες εικόνες απαιτούν τεράστιες ποσότητες νευρώνων και επομένως πολλές παραμέτρους για να ληφθούν υπόψιν κατά την εκπαίδευση. Από την άλλη πλευρά, τα CNNs κατανοούν τη δομή της εικόνας όταν επεξεργάζονται δεδομένα. Τα επίπεδά τους αποτελούνται από φίλτρα 3 διαστάσεων, που αντιπροσωπεύουν το ύψος, το πλάτος και το βάθος. Οι νευρώνες κάθε επιπέδου συνδέονται με μια μικρή περιοχή του προηγούμενου επιπέδου αντί με όλους τους νευρώνες όπως θα γινόταν στα FCNs.

Στα πρώτα επίπεδα εξάγονται αρκετά απλά χαρακτηριστικά των εικόνων, όπως είναι οι ακμές και οι γωνίες και όσο προχωράμε σε βαθύτερα επίπεδα εξάγονται χαρακτηριστικά υψηλότερου επιπέδου. Στη συνέλιξη επιλέγεται ένα φίλτρο ή πυρήνας συγκεκριμένων διαστάσεων, πολύ μικρότερων από την εικόνα εισόδου. Αυτός ο πυρήνας μετακινείται πάνω στην εικόνα με κάποιον σταθερό βηματισμό και παράγει την εικόνα για το επόμενο επίπεδο. Για να γίνει ποιο κατανοητή η πράξη της συνέλιξης, το Σχήμα 2.4 αναπαριστά σχηματικά ένα παράδειγμα υπολογισμού.



Σχήμα 2.4: *Ανάλυση υπολογισμού δισδιάστατης συνέλιξης*

Ένα Συνελικτικό Νευρωνικό Δίκτυο μπορεί να εκτελεί γενικά ένα μεγάλο πλήθος από διαφορετικές λειτουργίες, όμως 4 είναι οι πιο βασικές:

1. Συνέλιξη (Convolution)
2. Μη γραμμικότητα (Non-linearity)
3. Συγκέντρωση (Pooling)
4. Κανονικοποίηση (Normalization)

Συνελικτικό Επίπεδο

Το σημαντικότερο και μεγαλύτερο υπολογιστικό μέρος ενός συνελικτικού νευρωνικού δικτύου είναι το Συνελικτικό Επίπεδο (Convolutional Layer). Αυτό το επίπεδο αποτελείται από πολλά μικρά φίλτρα τα οποία διαμορφώνονται κατά τη διάρκεια της εκπαίδευσης του δικτύου. Αυτά τα φίλτρα ενώνονται με την πράξη της συνέλιξης με την είσοδο του συστήματος, η οποία είναι πολύ μεγαλύτερη σε σύγκριση με το μέγεθος των φίλτρων. Από αυτές τις συνελίξεις δημιουργούνται οι χάρτες ενεργοποίησης (activation maps) ή αλλιώς χάρτες χαρακτηριστικών (feature maps). Αυτοί οι χάρτες περιέχουν χαμηλού ή υψηλού επιπέδου χαρακτηριστικά [9].

Κάθε φίλτρο εξάγει κι ένα διαφορετικό χαρακτηριστικό, αφού κάθε νευρώνας εξόδου συνδέεται με ένα μικρό κομμάτι της εικόνας εισόδου γνωστό και ως τοπικό δεκτικό επίπεδο του νευρώνα (local receptive field) [10].

Το μέγεθος εξόδου ενός συνελικτικού επιπέδου εξαρτάται από ένα σύνολο παραμέτρων, οι οποίες ονομάζονται και υπερπαράμετροι. Ο σχεδιαστής του δικτύου με διάφορες τεχνικές βελτιστοποίησης επιλέγει τις καταλληλότερες για το εκάστοτε δίκτυο. Οι παράμετροι αυτές παρουσιάζονται παρακάτω:

- Το πλήθος των φίλτρων
- Το μέγεθος των φίλτρων
- Ο βηματισμός (stride)
- Το πλήθος των μηδενικών παραγεμίσματος (zero-padding)

Πιο αναλυτικά, ο αριθμός των φίλτρων καθορίζει το πόσα διαφορετικά χαρακτηριστικά θα εντοπισθούν αφού κάθε φίλτρο εξάγει ένα χαρακτηριστικό. Για το μέγεθος των φίλτρων συνηθέστερα επιλέγεται να έχει τετράγωνο σχήμα. Όσο για την ολίσθηση επιλέγεται πιο συχνά ο βηματισμός 1 ή 2 pixels. Τέλος, το πλήθος των μηδενικών είναι το γέμισμα του συνόρου της εικόνας με μηδενικά για να διατηρηθεί το μέγεθος της εικόνας αναλλοίωτο στην έξοδο αυτού του επιπέδου [9].

Για να μειωθούν οι απαιτήσεις μνήμης χρησιμοποιείται το σχήμα διαμοιρασμού παραμέτρων (parameter sharing). Σύμφωνα με αυτό το σχήμα αν εντοπισθεί κάποιο αντικείμενο σε μία θέση στην εικόνα, θα αναζητηθεί το συγκεκριμένο χαρακτηριστικό σε όλη την υπόλοιπη εικόνα. Με αυτόν τον τρόπο όλα τα φίλτρα χρησιμοποιούν τις ίδιες παραμέτρους για τον υπολογισμό κάθε νευρώνα εξόδου [9].

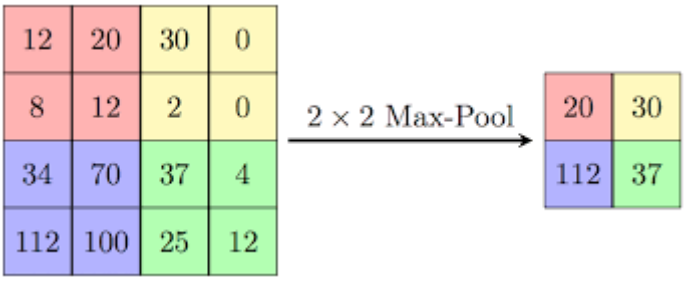
Μη γραμμικότητα

Σε ένα πρόβλημα Βαθιάς Μάθησης είναι αδύνατο όλες οι κλάσεις του προβλήματος να εξαρτώνται γραμμικά από τα δεδομένα. Για αυτόν τον λόγο κρίνεται απαραίτητη η χρήση μιας συνάρτησης ενεργοποίησης (activation fuction) ή μη-γραμμικότητας (non-linearity). Χρησιμοποιώντας τέτοιες συναρτήσεις προστίθεται μη-γραμμικότητα στο σύστημα ώστε να μπορεί να διαχειριστεί όλα τα δεδομένα [9].

Επίπεδο Συγκέντρωσης

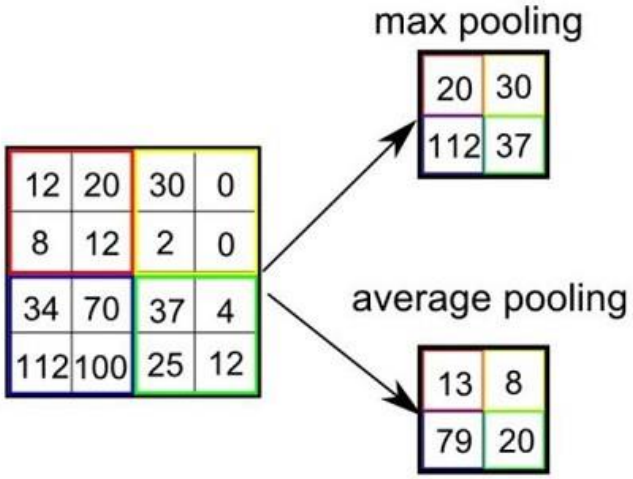
Το επίπεδο συγκέντρωσης (pooling layer) μειώνει το μέγεθος των αναπαραστάσεων ώστε να είναι πιο εύκολα διαχειρίσιμες. Οι διάφορες συναρτήσεις συγκέντρωσης υπολογίζουν συγκεκριμένες στατικές τιμές λαμβάνοντας υπόψιν τις τιμές των νευρώνων μια συγκεκριμένης περιοχής. Στη συνέχεια αυτή η τιμή αντικαθιστά όλη την περιοχή του νευρώνα αυτού. Οι πιο δημοφιλείς συναρτήσεις είναι:

- Μέγιστη συγκέντρωση (max pooling). Σε κάθε τετραγωνική γειτονιά βρίσκει και επιστρέφει τη μέγιστη τιμή της περιοχής. Στο Σχήμα 2.5 παρακάτω αναπαρίσταται σχηματικά ένα παράδειγμα μέγιστης συγκέντρωσης.



Σχήμα 2.5: *Παράδειγμα Μέγιστης Συγκέντρωσης*

- Μέση συγκέντρωση (average pooling). Σε κάθε τετραγωνική γειτονιά βρίσκει και επιστρέφει τη μέση τιμή της περιοχής. Στο Σχήμα 2.6 παρακάτω αναπαρίσταται σχηματικά η διαφορά της μέσης συγκέντρωσης και της μέγιστης συγκέντρωσης [11].



Σχήμα 2.6: *Διαφορά Μέγιστης και Μέσης Συγκέντρωσης*

- L2 συγκέντρωση (L2-norm pooling). Σε κάθε τετραγωνική γειτονιά υπολογίζει τα τετράγωνα των τιμών μιας περιοχής και επιστρέφει την τετραγωνική ρίζα του αθροίσματος αυτών των τιμών [3].

Επίπεδο Κανονικοποίησης

Η λειτουργία αυτού του επιπέδου είναι να μετασχηματίζει τις ενεργοποιήσεις ώστε ο μέσος όρος όλων των τιμών να είναι κοντά στο 0 και η τυπική απόκλιση να είναι κοντά στο 1. Το επίπεδο κανονικοποίησης (normalization layer) δεν επηρεάζει σε μεγάλο βαθμό την απόδοση του συστήματος. Οι ερευνητές έχουν προτείνει διάφορα επίπεδα κανονικοποίησης αλλά κανένα δεν συνεισφέρει σημαντικά στη βελτίωση του δικτύου [9].

Στο τέλος κάθε συνελικτικού νευρωνικού δικτύου χρησιμοποιείται ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα (fully connected layers) για την εξαγωγή της τελικής πρόβλεψης για την ετικέτα της εικόνας εισόδου. Το πρώτο πλήρως συνδεδεμένο επίπεδο είναι υπεύθυνο για τον μετασχηματισμό των χαρακτηριστικών που έχουν εντοπιστεί σε ένα ενιαίο διάνυσμα. Τα επόμενα πλήρως συνδεδεμένα επίπεδα πλην του τελευταίου συνήθως ακολουθούνται από κάποια συνάρτηση ενεργοποίησης. Το τελευταίο εξάγει τις πιθανότητες για κάθε κλάση και το μέγεθός του είναι ίδιο με τον αριθμό των κλάσεων του προβλήματος [9].

2.5 Βαθιά Μάθηση σε Κινητές Συσκευές

Τα τελευταία χρόνια οι κινητές συσκευές καταλαμβάνουν μεγάλο μέρος στην καθημερινότητα των ανθρώπων. Χρόνο με το χρόνο οι χρήστες κινητών συσκευών αυξάνονται και υπολογίζεται ότι το 2023 το 90% του ενήλικου πληθυσμού θα έχει στην κατοχή του τουλάχιστον μία κινητή συσκευή. Για αυτό το λόγο οι ερευνητές προσπαθούν να ξεπεράσουν τα προβλήματα που υπάρχουν στην ενσωμάτωση της Βαθιάς Μάθησης σε εφαρμογές που θα είναι χρήσιμες για τον χρήστη [12].

Με τη χρήση των κινητών συσκευών καθημερινά συλλέγονται δεδομένα που αφορούν τις προτιμήσεις, τη συμπεριφορά και τις συνήθειες του κάθε χρήστη. Παρατηρείται ότι οι ενήλικοι χρήστες σε αναπτυγμένες χώρες που χρησιμοποιούν καθημερινά εφαρμογές Βαθιάς Μάθησης ξεπερνούν το 70% [13]. Γίνεται αντιληπτό ότι ο τομέας αυτός είναι αναπτυσσόμενος και είναι επιτακτική η ανάγκη να ξεπερασθούν τα προβλήματα που δημιουργούνται από τις υψηλές απαιτήσεις των βαθιών νευρωνικών δικτύων ή από τα θέματα απορρήτου και ασφάλειας [12].

Οι εφαρμογές της Βαθιάς Μάθησης στις κινητές συσκευές μπορούν να χωριστούν σε δύο διεργασίες:

- Εκπαίδευση (training), χρησιμοποιώντας αλγορίθμους κατάβασης κλίσης (gradient descent) και ένα σύνολο πολλών δεδομένων το δίκτυο εκπαιδεύεται, δηλαδή τα βάρη του δικτύου παίρνουν κατάλληλες τιμές.
- Συμπερασματολογία (inference), δίνοντας άγνωστα δεδομένα σε ένα ήδη εκπαιδευμένο δίκτυο αναμένεται μια κατάλληλη πρόβλεψη ή η εξαγωγή μίας ετικέτας [12].

2.5.1 Εκπαίδευση

Μία κινητή συσκευή έχει περιορισμένη μνήμη και μικρή υπολογιστική ισχύ για να μπορέσει να εκπαιδεύσει ένα βαθύ νευρωνικό δίκτυο. Ένα νευρωνικό δίκτυο μπορεί να έχει εκατοντάδες εκατομμύρια παραμέτρους και για αυτόν το λόγο η εκπαίδευση τους γίνεται πριν την ενσωμάτωση σε μια εφαρμογή κινητής συσκευής. Την εκπαίδευση αναλαμβάνουν συνήθως κέντρα δεδομένων υπολογιστικού νέφους (cloud data centers) ή υπολογιστές υψηλών αποδόσεων (high-performance computers). Υπάρχουν περιπτώσεις όπου η εκπαίδευση πρέπει να γίνει με δεδομένα που παράγονται από την κινητή συσκευή. Σε αυτήν την περίπτωση τα δεδομένα στέλνονται σε κάποιον διακομιστή νέφους (cloud server) κι επιστρέφονται τα αποτελέσματα.

Για να λυθεί το πρόβλημα της έλλειψης πόρων προτείνονται διάφοροι τύποι εκπαίδευσης:

- Κατανεμημένη εκπαίδευση (distributed training)
- Ομοσπονδιακή εκπαίδευση (federated training)
- Μέθοδοι εκπαίδευσης με βασικό στόχο την προστασία της ιδιωτικότητας των δεδομένων (privacy-preserving accuracy) [12]

2.5.2 Συμπερασματολογία

Η συμπερασματολογία χρειάζεται ενέργεια, ισχύ και μνήμη άλλα μπορεί να εκτελεστεί σε μια κινητή συσκευή γιατί χρειάζεται λιγότερους πόρους σε σύγκριση με την εκπαίδευση ενός βαθιού συνελικτικού νευρωνικού δικτύου. Έτσι υπάρχει η δυνατότητα της συμπερασματολογίας να εκτελείτε τοπικά (on-device).

Άλλος ένας τρόπος να εκτελεστεί η συμπερασματολογία είναι απομακρυσμένα σε κάποιον διακομιστή νέφους (on-cloud). Για να επικοινωνήσει η συσκευή με το διακομιστή απαιτείται μία διεπαφή (interface) και η σύνδεση στο διαδίκτυο. Η εφαρμογή στέλνει δεδομένα στον διακομιστή ο οποίος με τη σειρά του απαντάει με το αποτέλεσμα [12]. Αυτός ο τρόπος έχει αρκετά πλεονεκτήματα αλλά και μειονεκτήματα. Τα θετικά της απομακρυσμένης συμπερασματολογίας είναι:

1. Η υλοποίηση της εφαρμογής είναι απλή.
2. Δεν απαιτείται πολύς χρόνος για την υλοποίηση της.
3. Το κόστος της υλοποίησης της εφαρμογής είναι χαμηλό.
4. Το νευρωνικό δίκτυο στο διακομιστή νέφους είναι ανεξάρτητο της εφαρμογής, οπότε μπορεί να τροποποιηθεί ανεξάρτητα από την εφαρμογή.

Τα αρνητικά της εκτέλεσης της συμπερασματολογίας σε έναν απομακρυσμένο διακομιστή νέφους είναι:

1. Απαιτείται η σύνδεση στο διαδίκτυο.

2. Υπάρχουν προβλήματα ασφάλειας απορρήτου κατά την επικοινωνία της κινητής συσκευής με τον διακομιστή, όπου είναι πιθανό να παραβιαστεί η ιδιωτικότητα του χρήστη.
3. Εισάγεται χρονική καθυστέρηση, η οποία μπορεί να οδηγεί σε προβλήματα σε εφαρμογές πραγματικού χρόνου (real-time applications) [12].

Αυτά τα μειονεκτήματα έχουν οδηγήσει τους ερευνητές να αναζητούν τρόπους να εισάγουν τη συμπερασματολογία τοπικά στην κινητή συσκευή. Έτσι τα εκπαιδευμένα βάρη είναι μέσα στην εφαρμογή και δεν είναι απαραίτητη η επικοινωνία με άλλες συσκευές. Για τη συμπερασματολογία στη συσκευή (on-device) οι μεγαλύτερες εταιρίες έχουν δείξει ιδιαίτερο ενδιαφέρον και έχουν αναπτύξει πλατφόρμες. Οι πιο γνωστές είναι από τη Google η Tensorflow Lite, από τη Facebook η Caffe2, η Apple έχει δημιουργήσει την Core ML, η Qualqomm έχει αναπτύξει το Snapdragon Neural Processing SDK και η Xiaomi δημιούργησε την Mobile AI Compute Engine (MACE).

Με την τοπική συμπερασματολογία δίνεται λύση στα παραπάνω προβλήματα αλλά υπάρχουν κάποια μειονεκτήματα που δεν μπορούν να παραλειφθούν:

- Κατά τη χρήση της εφαρμογής εξαντλείται μεγάλο ποσοστό ενέργειας, ισχύος και μνήμης της συσκευής.
- Τα μοντέλα είναι συνήθως βελτιστοποιημένα και άρα έχουν χαμηλότερη ακρίβεια.

Για να μην επιβαρύνεται η κεντρική μονάδα επεξεργασίας (CPU) της συσκευής, μπορεί αντί για αυτή να χρησιμοποιούνται οι λεγόμενοι επιταχυντές, GPUs, DSPs ή οι πιο νέες NPUs. Επίσης, διάφοροι μέθοδοι βελτιστοποίησης αναπτύσσονται ώστε να εξαλείψουν τα μειονεκτήματα της τοπικής εκτέλεσης της συμπερασματολογίας.

Επιταχυντές

Η υπολογιστική μονάδα που θα είναι υπεύθυνη για την εκτέλεση της συμπερασματολογίας σε τοπικό επίπεδο πρέπει να επιλεχθεί με βάση την απόδοση, το μέγεθος, το κόστος κατασκευής της, το βάρος της καθώς την ενέργεια που καταναλώνει. Τέτοιες μονάδες είναι οι CPUs, οι GPUs, τα FPGAs και τα ASICs.

Αρχικά, για να μην εισαχθεί νέο υλικό, χρησιμοποιούνταν οι CPUs και οι GPUs γιατί ήταν ευρέως διαδεδομένες και υπήρχαν σε όλες τις κινητές συσκευές. Οι CPUs θα μπορούσαν με χρήση πολλαπλών πυρήνων να υλοποιούν παράλληλους υπολογισμούς ώστε να θεωρούνται ικανές υπολογιστικές μονάδες για εφαρμογές Βαθιάς Μάθησης. Οι GPUs από την άλλη είναι κατασκευασμένες για να επεξεργάζονται παράλληλα δεδομένα άρα είναι πιο κατάλληλες για τα βαθιά νευρωνικά δίκτυα. Και οι δύο αυτές υπολογιστικές μονάδες, όπως προαναφέρθηκε, υπήρχαν ήδη στις κινητές συσκευές για να υλοποιούν άλλες διεργασίες. Οπότε η συμπερασματολογία θα είναι επιπλέον βάρος για αυτές.

Οι NPUs (Neural Processing Units) δημιουργήθηκαν για να μπαίνουν στις κινητές συσκευές και να βοηθούν στις διεργασίες της συμπερασματολογίας των βαθιών νευρωνικών εφαρμογών. Οι NPUs ανήκουν στην κατηγορία των ASICs (Application-Specific Integrated Circuits), είναι μικρές και ελαφριές μονάδες, δεν απαιτούν πολλή ενέργεια, αλλά χρειάζονται πολύ χρόνο για να κατασκευασθούν. Από το 2017 και μετά, οι μεγαλύτεροι κατασκευαστές ολοκληρωμένων κυκλωμάτων (chips) για κινητές συσκευές χρησιμοποιούν NPUs για την εκτέλεση των εφαρμογών Βαθιάς Μάθησης.

Μια άλλη κατηγορία υπολογιστικών μονάδων είναι τα FPGAs, τα οποία είναι μία συστοιχία προγραμματιζόμενων πυλών. Είναι ικανά να επαναπρογραμματιστούν άπειρες φορές από το υλικολογισμικό για να εκτελούν διάφορους υπολογισμούς αποδοτικότερα. Η κατανάλωση ενέργειας είναι μεγαλύτερη σε σύγκριση με τα ASICs.

Συμπερασματικά, οι CPUs και οι GPUs είναι πιο ευέλικτες και έτσι μπορούν εκτελούν υπολογισμούς αυξημένων απαιτήσεων αν και καταναλώνουν πολλούς πόρους. Τα ASICs έχουν χαμηλότερη κατανάλωση ενέργειας αλλά απαιτούν πολύ χρόνο στην κατασκευή τους. Τα FPGAs είναι κάπου ανάμεσα σε θέμα ευελιξίας και απόδοσης.

Βελτιστοποίηση

Οι ερευνητές έχουν δώσει προσοχή και στην πρόοδο του λογισμικού με την ανάπτυξη μεθόδων βελτιστοποίησης βαθιών νευρωνικών δικτύων. Από τις σημαντικότερες μεθόδους είναι η κβαντοποίηση (quantization), η οποία ασχολείται με τον αριθμό των ψηφίων που αναπαρίστανται τα βάρη ενός εκπαιδευμένου δικτύου. Η κβαντοποίηση είναι υλοποιήσιμη και αποδοτική γιατί τα βαθιά δίκτυα όταν εκπαιδευτούν έχουν προκαθορισμένες τιμές στα βάρη εξόδου. Οι τιμές των βαρών έχουν μικρές αυξομειώσεις οι οποίες είναι προβλέψιμες από το ίδιο το νευρωνικό δίκτυο. Αν γίνει κβαντοποίηση και είναι σωστή τότε θα υπάρχει μια μικρή μείωση της ακρίβειας αλλά θα είναι ελάχιστη σε σύγκριση με τα οφέλη που παρέχει. Πιο αναλυτικά θα παρουσιαστεί αυτή η μέθοδος στην Ενότητα 4.4.1 [14]

Κεφάλαιο 3: Διεργασίες

Τα τελευταία χρόνια γίνονται προσπάθειες να αναπτυχθούν και να εξελιχθούν εφαρμογές Βαθιάς Μάθησης που αφορούν την Όραση Υπολογιστών. Δίνεται μεγάλη έμφαση στην αναγνώριση αντικειμένων και στην επεξεργασία εικόνων για τη βελτίωση της ζωής των ανθρώπων αφού αυτές οι εφαρμογές χρησιμοποιούνται καθημερινά από την ιατρική μέχρι και τα αυτόνομα αυτοκίνητα. Η Κατηγοριοποίηση Εικόνας είναι η πιο απλή διεργασία Βαθιάς Μάθησης και τα τελευταία χρόνια έχει απασχολήσει πολύ τους ερευνητές. Για αυτό το λόγο, στην παρούσα διπλωματική εργασία, επίκεντρο είναι τρεις διεργασίες Βαθιάς Μάθησης που δεν έχουν αναλυθεί σε αντίστοιχο βαθμό:

- Ανίχνευση Αντικειμένων
- Κατάτμηση Εικόνας
- Εκτίμηση Ανθρώπινης Πόζας

3.1 Κατάτμηση Εικόνας

Στο Σχήμα 3.1 φαίνεται ένα παράδειγμα Κατάτμησης Εικόνας. Τα τελευταία χρόνια έχουν αναπτυχθεί διάφοροι αλγόριθμοι και τεχνικές για την αντιμετώπιση του προβλήματος της Κατάτμησης των Εικόνων (Image Segmentation). Οι ανάγκες και τα δεδομένα κάθε διαφορετικού τομέα επηρεάζουν σε μεγάλο βαθμό την αποδοτικότητα και την αποτελεσματικότητα της κάθε μεθόδου. Οι κύριες κατηγορίες τεχνικών Κατάτμησης Εικόνας είναι:

- Μέθοδος Κατωφλίου (Thresholding). Αποτελεί την πιο απλή τεχνική Κατάτμησης Εικόνας. Επιλέγεται μια τιμή κατωφλίου, ώστε να μετατραπεί μια ασπρόμαυρη εικόνα σε δυαδική. Πιο αναλυτικά, τα pixels της εικόνας που ξεπερνούν την τιμή του κατωφλίου περιλαμβάνονται σε ένα σύνολο με τιμή 1, ενώ τα υπόλοιπα σε ένα σύνολο με τιμή 0. Σημαντικό στοιχείο για αυτήν τη μέθοδο είναι η κατάλληλη επιλογή της τιμής του κατωφλίου. Διάσημες τέτοιες μέθοδοι είναι η μέθοδος μέγιστης εντροπίας και η μέθοδος μέγιστης διακύμανσης Otsu. Τα τελευταία χρόνια έχουν αναπτυχθεί και νέοι μέθοδοι με χρήση πολυδιάστατων, μη-γραμμικών, ασαφών κανόνων ως κατώφλια.
- Μέθοδοι Ομαδοποίησης (Clustering). Βασίζονται στην διαδικασία ομαδοποίησης των pixels σε συστοιχίες (clusters), έτσι ώστε τα pixels κάθε συστοιχίας να παρουσιάζουν μεγαλύτερες ομοιότητες μεταξύ τους από τα pixels στις υπόλοιπες συστοιχίες. Μία βασική μέθοδος ομαδοποίησης αποτελεί ο αλγόριθμος k-means, ο οποίος διαιρεί επαναληπτικά την εικόνα σε k συστοιχίες. Το πλεονέκτημα του αλγορίθμου είναι ότι προσφέρει σίγουρη σύγκλιση, όμως η ποιότητα του αποτελέσματος εξαρτάται σε μεγάλο βαθμό από τις αρχικές συστοιχίες και την παράμετρο k.
- Μέθοδοι βασισμένες στη Συμπίεση (Compression-based Methods). Θεωρείται ότι η βέλτιστη κατάτμηση μιας εικόνας είναι αυτή που ελαχιστοποιεί το μέγεθος της κωδικοποίησης των δεδομένων σε σχέση με άλλες κατατμήσεις. Η λογική βασίζεται στα μοτίβα στην εικόνα κι έτσι οποιαδήποτε κανονικότητα της εικόνας μπορεί να χρησιμοποιηθεί για τη συμπίεση. Μια συνάρτηση κατανομής διαμορφώνει την υφή και το σχήμα των ορίων κάθε τμήματος της εικόνας.
- Μέθοδοι βασισμένες σε Ιστογράμματα (Histogram-based Methods). Είναι πολύ πιο αποδοτικοί μέθοδοι Κατάτμησης Εικόνας σε σύγκριση με άλλες μεθόδους.

Κάθε pixel της εικόνας διαβάζεται μόνο μια φορά και παράγεται ένα ιστόγραμμα. Στη συνέχεια χρησιμοποιούνται οι κορυφές και οι κοιλάδες αυτού για να βρεθούν οι συστοιχίες (clusters). Υπάρχει η δυνατότητα να επαναλαμβάνεται η μέθοδος στις συστοιχίες για να διαχωρίζονται σε μικρότερες, ώσπου να μην είναι δυνατή η δημιουργία καινούργιων συστοιχιών.

- Ανίχνευση Ακμών (Edge Detection). Η χρήση αυτής της μεθόδου είναι συχνή γιατί συνήθως υπάρχει έντονη αλλαγή στην ένταση των pixels στα όρια των τμημάτων της εικόνας. Για να διαχωριστεί ένα αντικείμενο μέσα σε μία εικόνα πρέπει τα όρια του τμήματος να είναι κλειστά γιατί οι εντοπισμένες ακμές είναι συνήθως ασύνδετες μεταξύ τους.
- Μέθοδοι Ανάπτυξης Περιοχών (Region-growing Methods). Βασίζονται στην υπόθεση ότι τα γειτονικά pixels έχουν παρόμοιες τιμές. Η διαδικασία περιλαμβάνει τη σύγκριση ενός pixel με τα γειτονικά του. Αν ικανοποιείται ένα κριτήριο ομοιότητας τότε το pixel ανήκει στη συστοιχία, όπως και ένας ή περισσότεροι γείτονές του.
- Μετασχηματισμός Λεκάνης Απορροής (Watershed Transformation). Ο βαθμός κλίσης μιας εικόνας θεωρείται μια τοπογραφική επιφάνεια. Τα pixels που ανήκουν στα όρια των τμημάτων έχουν το μεγαλύτερο βαθμό κλίσης. Τα εσωτερικά pixels τείνουν σε ένα τοπικό ελάχιστο της έντασης κλίσης δημιουργώντας μια λεκάνη απορροής που αναπαριστά ένα τμήμα της εικόνας.
- Μέθοδοι Κατάτμησης Γράφων (Graph Partitioning Methods). Μοντελοποιούν την επίδραση των γειτονικών pixels σε δοσμένες συστοιχίες. Η εικόνα θεωρείται ένας μη κατευθυνόμενος σταθμισμένος γράφος. Κάθε pixel σχετίζεται με έναν κόμβο του γράφου και τα βάρη των ακμών ορίζουν την ομοιότητα μεταξύ γειτονικών pixels. Έπειτα, ο γράφος διαχωρίζεται και δημιουργούνται διάφορες συστοιχίες, κάθε μια από τις οποίες είναι ένα αντικείμενο της εικόνας [15] [16].

Σχήμα 3.1: *Κατάτμηση Εικόνας*

3.2 Ανίχνευση Αντικειμένων

Η Ανίχνευση Αντικειμένων (Object Detection) είναι μια δημοφιλής εφαρμογή της Βαθιάς Μάθησης κατά την οποία ένα υπολογιστικό σύστημα παίρνει ως είσοδο μία εικόνα και βγάζει ως έξοδο περιοχές της εικόνας που εμφανίζεται κάποιο συγκεκριμένο αντικείμενο. Το αποτέλεσμα της διαδικασίας αυτής αποτελείται από την κατηγορία κάθε αντικειμένου, τη θέση του και την πεποίθηση του δικτύου για την ύπαρξη αυτού.

Η Ανίχνευση Αντικειμένων συχνά μπερδεύεται με την Κατηγοριοποίηση Εικόνας αλλά είναι διαφορετική διαδικασία. Στην Κατηγοριοποίηση Εικόνας η έξοδος του υπολογιστικού συστήματος περιλαμβάνει μόνο την πληροφορία της ύπαρξης ενός συγκεκριμένου αντικειμένου και όχι τη θέση αυτού. Αντίθετα στην Ανίχνευση Αντικειμένων πρέπει να ελεγχθεί αν εμφανίζεται κάποιο αντικείμενο σε κάποια θέση της εικόνας και να προσδιορισθεί και το μέγεθός του. Στο Σχήμα 3.2 παρακάτω αναπαρίσταται ένα παράδειγμα μιας εικόνας στην οποία εφαρμόζεται η Ανίχνευση Αντικειμένων [17].

Σχήμα 3.2: *Ανίχνευση Αντικειμένων*

Κάθε εικόνα αποτελείται από ένα σύνολο pixels, που το καθένα παίρνει μία τιμή που αντιστοιχεί στη φωτεινότητα και στο χρώμα της συγκεκριμένης θέσης της εικόνας. Δηλαδή η εικόνα μπορεί να θεωρηθεί σαν μια αναπαράσταση μιας δισδιάστατης δομής από στοιχεία που παίρνουν διακριτές τιμές, άρα μπορεί να θεωρηθεί ως ένα ψηφιακό σήμα δύο διαστάσεων.

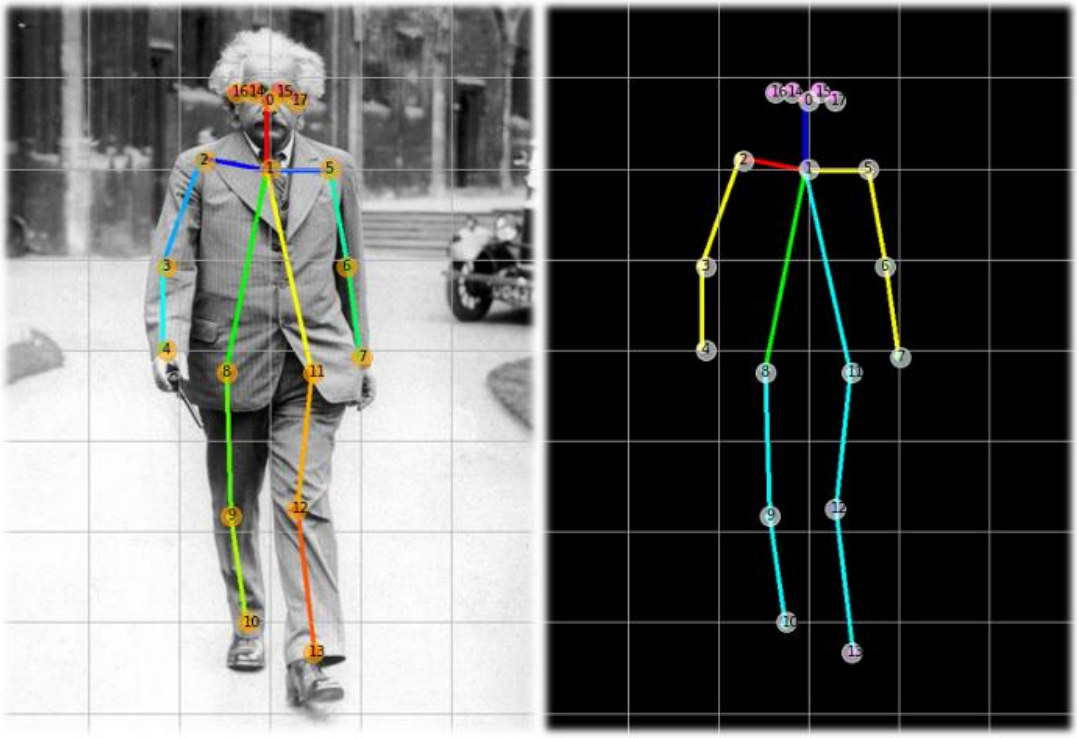
Παρατηρείται ότι η Ανίχνευση Αντικειμένων είναι μια περίπλοκη και σύνθετη διαδικασία που μπορεί να διαχωριστεί σε 3 στάδια:

- Επιλογή Περιοχών Πληροφορίας (Informative Region Selection). Ένα παράθυρο ολίσθησης διασχίζει την εικόνα για να ανιχνεύσει αντικείμενα σε διάφορες θέσεις και με διαφορετικές διαστάσεις. Αυτή η στρατηγική μπορεί να ανακαλύψει όλες τις θέσεις των αντικειμένων αλλά είναι υπολογιστικά ακριβή και παράγει πολλά περιττά παράθυρα. Αν από την άλλη χρησιμοποιηθεί μόνο συγκεκριμένος αριθμός παραθύρων, ενδέχεται τα αποτελέσματα να μην είναι ικανοποιητικά.
- Εξαγωγή Χαρακτηριστικών (Feature Extraction). Για την αναγνώριση διαφορετικών αντικειμένων πρέπει να εξαχθούν οπτικά χαρακτηριστικά που μπορούν να παρέχουν σημασιολογική αναπαράσταση. Κάποιες τεχνικές όπως οι SIFT, HOG και Haar-Lite παράγουν αναπαραστάσεις που σχετίζονται με σύνθετα κύτταρα του ανθρώπινου εγκεφάλου.
- Ταξινόμηση (Classification). Η ταξινόμηση μιας εικόνας γίνεται με την εφαρμογή της θεωρίας αναγνώρισης προτύπων. Μετά την περιγραφή της εικόνας ή ενός μέρους αυτής (πρότυπο), ελέγχεται αν η εικόνα ανήκει σε μία κατηγορία ενδιαφέροντος. Για τον έλεγχο αυτό μπορούν να χρησιμοποιηθούν διάφοροι αλγόριθμοι. Κάποιοι από αυτούς είναι τα νευρωνικά δίκτυα, τα SVMs (Support Vector Machines), ο αλγόριθμος Winnow και αλγόριθμοι ενδυνάμωσης (π.χ. AdaBoost) [18].

3.3 Εκτίμηση Ανθρώπινης Πόζας

Τα τελευταία χρόνια έχει δοθεί μεγάλη προσοχή σε μοντέλα Εκτίμησης της Ανθρώπινης Πόζας. Η ανίχνευση των κινήσεων του ανθρώπου σε ένα βίντεο ή μια εικόνα αποτελεί σημαντικό παράγοντα για την βελτίωση της Αλληλεπίδρασης Ανθρώπου-Υπολογιστή (Human-Computer Interaction - HCI). Μερικές δημοφιλείς εφαρμογές της Εκτίμησης Ανθρώπινης Πόζας είναι η αναγνώριση δραστηριοτήτων, η καταγραφή κίνησης, η επαυξημένη πραγματικότητα, η εκπαίδευση ρομποτικών συστημάτων, η καταγραφή κίνησης σε διαδραστικά βιντεοπαιχνίδια, κ.α.

Το πρόβλημα της Εκτίμησης της Ανθρώπινης Πόζας ορίζεται ως ένα πρόβλημα ανίχνευσης αρθρώσεων του ανθρώπινου σώματος που αναπαρίστανται ως σημεία (keypoints) με τις συντεταγμένες τους. Κάθε σύνδεση δύο τέτοιων σημείων ονομάζεται ζεύγος σημείων (pair). Το σύνολο όλων των σημείων που συνδέονται μεταξύ τους σχηματίζουν έναν σκελετό που αποτελεί μια γραφική αναπαράσταση της πόζας του ανθρώπου. Σημαντικό είναι να επισημανθεί ότι δεν δίνουν όλοι οι συνδυασμοί των σημείων έγκυρα ζευγάρια. Στο Σχήμα 3.3 παρακάτω αναπαρίσταται σχηματικά η κίνηση ενός ανθρώπου σαν σκελετός σημείων [19].



Σχήμα 3.3: *Γράφημα ανθρώπινου σώματος με σημεία και ακμές*

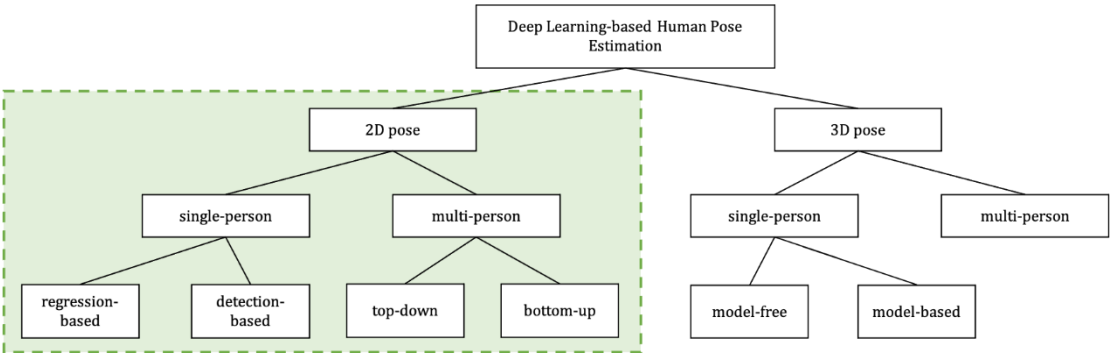
Το πρόβλημα της Εκτίμησης της Ανθρώπινης Πόζας μπορεί να χωριστεί σε 2 κατηγορίες:

- Εκτίμηση δισδιάστατης (2D) πόζας: Αφορά την εκτίμηση συντεταγμένων σημείων (x, y) από μία RGB εικόνα.
- Εκτίμηση τρισδιάστατης (3D) πόζας: Πρόκειται για πιο περίπλοκα μοντέλα, τα οποία μπορεί να βασίζονται είτε σε μία RGB εικόνα, είτε σε ένα σύστημα πολλών προβολών της ίδιας τρισδιάστατης εικόνας, είτε ακόμα και σε RGB-D κάμερες που δίνουν μαζί με την εικόνα και έναν χάρτη βάθους (depth map) με πληροφορίες για τη θέση του αντικειμένου στο χώρο.

Σε αυτό το σημείο πρέπει να σημειωθεί ότι η 3D ανάλυση είναι πιο δύσκολα ανιχνεύσιμη από την 2D για αυτό και επιλέγεται συνήθως να εφαρμοσθεί σε εργαστηριακές εικόνες και όχι σε εξωτερικούς χώρους. Η Εκτίμηση Ανθρώπινης Πόζας μπορεί να χωρισθεί σε 2 κατηγορίες με βάση τον αριθμό των ανθρώπων που υπάρχουν σε μία εικόνα.

- Αν υπάρχει μόνο ένας άνθρωπος τότε υπάρχουν δύο κατηγορίες μεθόδων:
  - Μέθοδοι Οπισθοδρόμησης. Δημιουργείται άμεσα μια χαρτογράφηση εικόνων εισόδου σε συντεταγμένες σημείων - αρθρώσεων του σώματος.
  - Μέθοδοι ανίχνευσης μελών του σώματος. Αρχικά δημιουργείται ένας χάρτης θερμότητας (heatmap) βασικών σημείων του σώματος και στη συνέχεια ενώνονται τα σημεία σε ολόκληρο το σώμα.
- Αν υπάρχουν περισσότεροι άνθρωποι τότε καταλήγουμε σε δύο είδη μεθόδων:
  - Top-Down μέθοδος. Πρώτα ανιχνεύεται ένας άνθρωπος και στη συνέχεια προβλέπονται τα keypoints για τον κάθε άνθρωπο χωριστά.
  - Bottom-Up μέθοδος. Πρώτα ανιχνεύονται σημεία αρθρώσεων χωρίς να υπάρχει γνώση για τον αριθμό των ατόμων που υπάρχουν στην εικόνα και στη συνέχεια κατηγοριοποιούνται σε ανεξάρτητες στάσεις [20].

Στο Σχήμα 3.4 παρακάτω αναπαρίσταται σχηματικά ο διαχωρισμός κατηγοριών του προβλήματος της Εκτίμησης της Ανθρώπινης Πόζας.

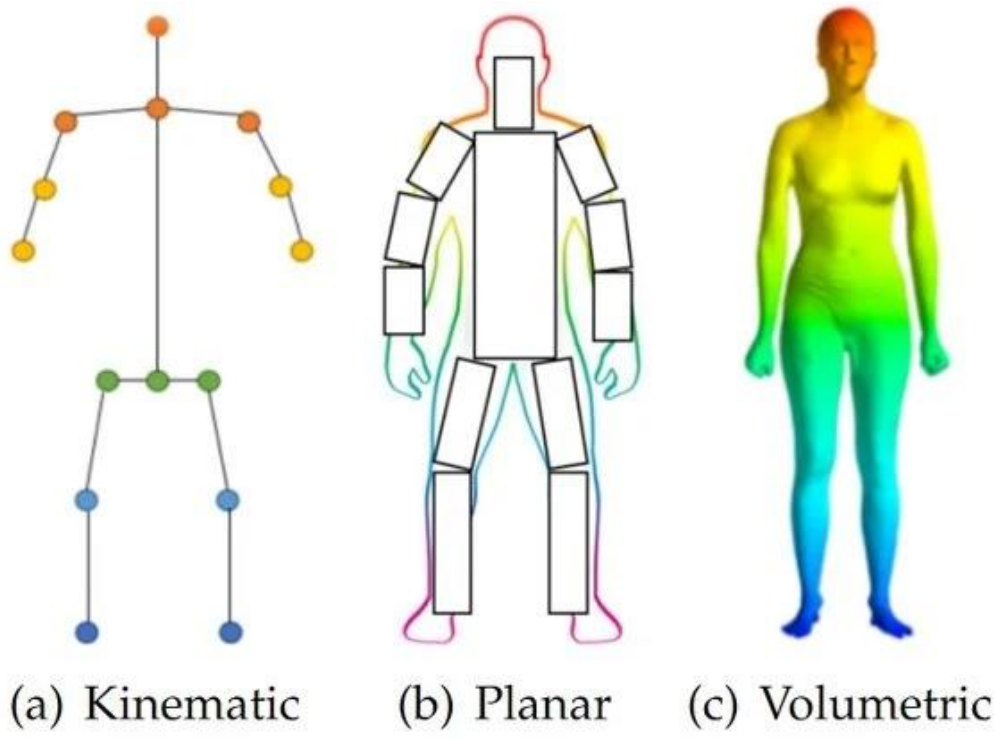


Σχήμα 3.4: *Κατηγοριοποίηση της Εκτίμησης Ανθρώπινης Πόζας*

Στις περισσότερες μεθόδους Εκτίμησης Ανθρώπινης Πόζας το άτομο αναπαρίσταται ως αρθρώσεις και άκρα. Παρόλα αυτά υπάρχουν και άλλοι τρόποι αναπαράστασης της πόζας του ανθρώπινου σώματος. Αυτές είναι:

- Κινηματικό ή Σκελετικό μοντέλο (Kinematic model). Αποτελείται από σημεία και ζεύγη σημείων. Μπορεί να χρησιμοποιηθεί τόσο σε 2D όσο και σε 3D εκτιμήσεις. Είναι αρκετά δημοφιλές και εύχρηστο μοντέλο αλλά υστερεί στην απεικόνιση υφών και σχημάτων.
- Επίπεδο μοντέλο (Planar model). Αναπαριστά το μέρη του σώματος σαν ορθογώνια κουτιά. Μπορεί να χρησιμοποιηθεί μόνο σε 2D εκτιμήσεις.
- Ογκομετρικό μοντέλο (Volumetric model). Δεν χρησιμοποιείται συχνά αλλά είναι πολύ αναλυτικό μοντέλο. Μπορεί να χρησιμοποιηθεί σε 3D αναλύσεις πόζας ανθρώπου [20].

Στο Σχήμα 3.5 παρακάτω παρουσιάζονται σχηματικά τα τρία παραπάνω μοντέλα και οι αναπαραστάσεις τους.



Σχήμα 3.5: *Είδη μοντέλων Εκτίμησης Ανθρώπινης Πόζας*

3.4 Σύνολα Δεδομένων

Ένα καλό σύνολο δεδομένων (dataset) είναι το πρώτο βήμα για να καταλήξουμε σε ένα ισχυρό και ακριβές μοντέλο Βαθιάς Μάθησης. Σε αυτήν την εργασία χρησιμοποιήθηκαν κάποια από τα πιο δημοφιλή σύνολα δεδομένων για την εκπαίδευση και την αξιολόγηση των μοντέλων.

Το **Pascal Visual Object Classes (Pascal VOC)** είναι ένα από τα δημοφιλέστερα σύνολα δεδομένων στην Όραση Υπολογιστών, παρέχοντας εικόνες με σχολιασμό για πέντε εργασίες: Ταξινόμηση, Κατάτμηση Εικόνας, Ανίχνευση Αντικειμένων, Αναγνώριση Δράσεων και Περιγράμματος Ανθρώπων. Σχεδόν όλοι οι γνωστοί αλγόριθμοι κατάτμησης έχουν αξιολογηθεί σε αυτό το σύνολο δεδομένων. Περιέχει 21 διαφορετικές κλάσεις αντικειμένων (μαζί με το φόντο της εικόνας):

[αεροπλάνο, ποδήλατο, πουλί, πλοίο, μπουκάλι, λεωφορείο, αυτοκίνητο, γάτα, καρέκλα, αγελάδα, τραπέζι, σκύλος, άλογο, μοτοσικλέτα, άνθρωπος, γλάστρα με φυτό, πρόβατο, καναπές, τραίνο, τηλεόραση]

Αν κάποιο pixel της εικόνας δεν ανήκει σε κάποια από τις παραπάνω κλάσεις, τότε θεωρείται φόντο (background). Το σύνολο δεδομένων χωρίζεται στο σύνολο εκπαίδευσης με 1464 εικόνες και το σύνολο αξιολόγησης με 1449 εικόνες [15].

Το **Microsoft Common Objects in Context (MS COCO)** περιλαμβάνει εικόνες πολύπλοκων καθημερινών σκηνών που περιέχουν αντικείμενα στο φυσικό τους πλαίσιο. Είναι από τα πιο συχνά χρησιμοποιούμενα σύνολα δεδομένων αφού χρησιμοποιείται για εκπαίδευση, αξιολόγηση και έλεγχο σε εργασίες Όρασης Υπολογιστών όπως η Κατάτμηση Εικόνας, η Ανίχνευση Αντικειμένων και η Εκτίμηση Ανθρώπινης Πόζας. Αυτό το σύνολο δεδομένων περιλαμβάνει εικόνες από 91

διαφορετικά είδη αντικειμένων από τα οποία τα 82 περιέχουν πάνω από 5000 περιγραφές το καθένα. Το σύνολο δεδομένων χωρίζεται στο σύνολο εκπαίδευσης με 82000 εικόνες, στο σύνολο επικύρωσης με 40500 εικόνες και στο σύνολο αξιολόγησης του μοντέλου με περισσότερες από 80000 εικόνες [15].

Το **Cityscapes** είναι ένα μεγάλης κλίμακας σύνολο δεδομένων που εστιάζει στη σημασιολογική κατανόηση σκηνικών από δρόμους πόλεων. Περιέχει πολλά διαφορετικά βίντεο που καταγράφηκαν σε 50 διαφορετικές πόλεις και ένα σετ από 20000 πλαίσια (frames) με σχολιασμό. Περιέχει 19 διαφορετικές κλάσεις:

[δρόμος, πεζοδρόμιο, κτίριο, τοίχος, φράχτης, στύλος, φανάρι, πινακίδες τροχαίας, βλάστηση, έδαφος, ουρανός, άνθρωπος, αναβάτης μοτοσικλέτας/ποδηλάτου, αυτοκίνητο, φορτηγό, λεωφορείο, τραίνο, μοτοσικλέτα, ποδήλατο]

οι οποίες κατηγοριοποιούνται σε 8 ομάδες: έδαφος, φύση, ουρανός, κατασκευές, άνθρωπος, οχήματα, αντικείμενα και κενό [15].

Επιπλέον, χρησιμοποιήθηκαν κάποια σύνολα δεδομένων για δυαδική κατάτμηση πορτρέτων ανθρώπων και περιβάλλοντος χώρου. Το **AISegment** είναι το μεγαλύτερο σύνολο δεδομένων αυτού του είδους με 34427 εικόνες. Έχει ξεχωρίσει για την υψηλή ποιότητα αποτελεσμάτων από το Beijing Play Star Convergence Technology Co., Ltd. Περιέχει δύο κλάσεις: άνθρωπος και φόντο. Μπορεί να συνδυαστεί και με άλλα σύνολα δεδομένων αυτού του είδους για καλύτερη απόδοση και μεγαλύτερη ακρίβεια, όπως είναι τα δύο σύνολα δεδομένων που παρουσιάζονται στη συνέχεια.

Γνωστό σύνολο δεδομένων προσώπων είναι το **PFCN** που περιέχει 1800 εικόνες. Οι εικόνες κατηγοριοποιούνται σε δυο ομάδες, στο σύνολο εκπαίδευσης με 1500 εικόνες και στο σύνολο αξιολόγησης με 300 εικόνες. Λόγω του μικρού μεγέθους αυτού του συνόλου δεδομένων δεν χρησιμοποιείται μόνο του, αλλά μπορεί να χρησιμοποιηθεί σε συνδυασμό με άλλα δυαδικά σύνολα δεδομένων προσώπων [21].

Το **Supervisely** είναι ένα σύνολο δεδομένων κατάτμησης προσώπων. Συλλέγει εικόνες από δημόσιες βάσεις δεδομένων που περιέχουν κυρίως φωτογραφίες προσώπων που τραβήχτηκαν με την μπροστινή κάμερα ενός κινητού τηλεφώνου (self-portrait). Περιέχει υψηλής ποιότητας εμφανίσεις σχολιασμένων προσώπων. Έχει 2258 εικόνες πορτρέτων διαφορετικών μεγεθών εκ των οποίων οι 1858 εικόνες επιλέγονται τυχαία για εκπαίδευση και οι υπόλοιπες 400 χρησιμοποιούνται για την αξιολόγηση του μοντέλου [22].

Το **DAGM2007** είναι ένα συνθετικό σύνολο δεδομένων για τον εντοπισμό ελαττωμάτων σε επιφάνειες με υφή. Το όνομά του υποδηλώνει τη δημιουργία του για έναν διαγωνισμό του 2007 από το Deutsche Arbeitsgemeinschaft für Mustererkennung, το γερμανικό κεφάλαιο της Διεθνούς Ένωσης για την Αναγνώριση Προτύπων. Τα δεδομένα παράγονται τεχνητά, αλλά είναι παρόμοια με προβλήματα του πραγματικού κόσμου. Τα πρώτα έξι από τα δέκα σύνολα δεδομένων, που δηλώνονται ως σύνολα δεδομένων ανάπτυξης χρησιμοποιούνται για την ανάπτυξη των αλγορίθμων, ενώ τα υπόλοιπα τέσσερα, τα οποία αναφέρονται ως σύνολα δεδομένων ανταγωνισμού, μπορούν να χρησιμοποιηθούν για την αξιολόγηση της απόδοσης [23].

Το **AI Challenger - Human Keypoint Detection (AIC-HKD)** είναι η μεγαλύτερη βάση δεδομένων εκπαίδευσης για δισδιάστατη Εκτίμηση Ανθρώπινης Πόζας. Έχει 300000 εικόνες με ετικέτα για ανίχνευση σημείων στο σώμα που συλλέχθηκαν από μηχανές αναζήτησης στο διαδίκτυο και εστιάζουν σε καθημερινές ανθρώπινες δραστηριότητες. Περιλαμβάνει 210000 εικόνες για εκπαίδευση, 30000 για επικύρωση και 60000 για αξιολόγηση [20].

Το **Max Planck Institute for Informatics (MPII) Human Pose Dataset** είναι μία δημοφιλής βάση δεδομένων για την αξιολόγηση της Εκτίμησης Ανθρώπινης Πόζας. Η βάση δεδομένων περιλαμβάνει περίπου 25000 εικόνες που περιλαμβάνουν πάνω από 40000 αρθρώσεις σώματος. Οι εικόνες συλλέχθηκαν από μια ιεραρχική μέθοδο δύο επιπέδων για την αποτύπωση των καθημερινών ανθρώπινων δραστηριοτήτων από βίντεο στο Youtube. Επιπλέον, στη βάση δεδομένων περιέχονται σχολιασμοί για 14 αρθρώσεις σώματος και τρισδιάστατοι προσανατολισμοί κορμού και κεφαλιού που δημιουργήθηκαν από εργαζόμενους της Amazon Mechanical Turk. Οι εικόνες της βάσης αυτής είναι κατάλληλες για δισδιάστατη Εκτίμηση Ανθρώπινης Πόζας ενός ή πολλών ατόμων [20].

Τέλος, το **Active Dataset** είναι ένα σύνολο δεδομένων με εικόνες που συλλέχθηκαν από βίντεο στο Youtube και αφορούν γυμναστική, γιόγκα και χορό. Περιλαμβάνει εικόνες ανθρώπων σε διάφορες στάσεις και κινήσεις. Περιλαμβάνει 23500 εικόνες για εκπαίδευση και 1161 εικόνες για αξιολόγηση μοντέλων που ανιχνεύουν μέχρι ένα άτομο και 120000 εικόνες για εκπαίδευση και 1900 εικόνες για αξιολόγηση μοντέλων που ανιχνεύουν πολλούς ανθρώπους.

3.5 Μετρικές

Στην ιδανική περίπτωση ένα μοντέλο πρέπει να αξιολογείται από πολλές απόψεις, όπως για παράδειγμα η ποσοτική ακρίβεια, η ταχύτητα και οι απαιτήσεις αποθήκευσης. Ωστόσο, οι περισσότερες ερευνητικές εργασίες επικεντρώνονται στις μετρήσεις για την αξιολόγηση της ακρίβειας του μοντέλου. Αν και οι ποσοτικές μετρήσεις χρησιμοποιούνται για τη σύγκριση διαφορετικών μοντέλων σε σημεία αναφοράς, η οπτική ποιότητα των εξόδων του μοντέλου είναι επίσης ένας σημαντικός παράγοντας για να αποφασιστεί ποιο είναι το καλύτερο μοντέλο για προβλήματα Όρασης Υπολογιστών.

Η πρώτη μετρική που αξίζει να αναφερθεί είναι η **Pixel Accuracy (PA)**, κατά την οποία υπολογίζεται το κλάσμα των σωστών ταξινομημένων pixels ως προς το συνολικό αριθμό των pixels μιας εικόνας. Η εκτεταμένη εκδοχή αυτής της μετρικής είναι η **mean Pixel Accuracy (mPA)** που είναι η μέση τιμή όλων των PA κάθε κλάσης:

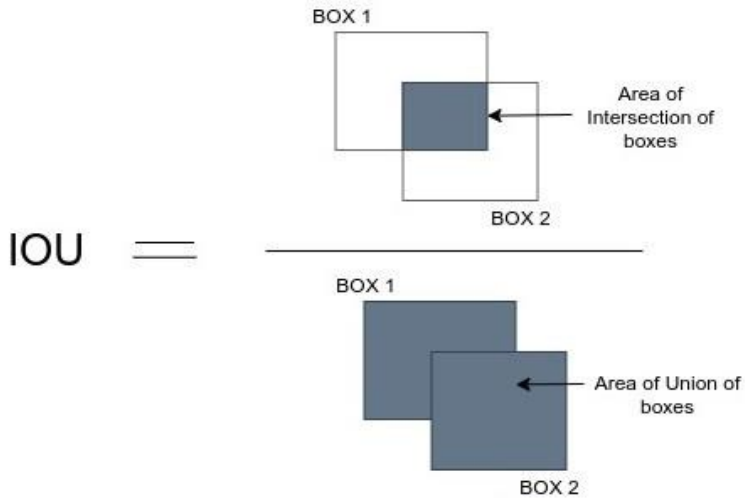
$$PA = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij}}, \qquad MPA = \frac{1}{K+1} \sum_{i=0}^{K} \frac{p_{ii}}{\sum_{j=0}^{K} p_{ij}}$$

όπου K είναι ο αριθμός των κλάσεων (χωρίς το φόντο) και p_{ij} το πλήθος των pixels που ανήκουν στην κλάση i αλλά το μοντέλο προέβλεψε ότι ανήκουν στην κλάση j.

Η **Intersection over Union (IoU)** ή **Jaccard Index** είναι η πιο συχνά χρησιμοποιούμενη μετρική στην Κατάτμηση Εικόνας. Ορίζεται ως η τομή μεταξύ προβλεπόμενης και πραγματικής κατάτμησης, διαιρούμενη με την περιοχή της ένωσης τους:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

Στο Σχήμα 3.6 παρακάτω αναπαρίσταται σχηματικά αυτός ο δείκτης αξιολόγησης.



Σχήμα 3.6: *Διαγραμματική απεικόνιση του δείκτη αξιολόγησης IoU*

Η **mean IoU** είναι ο μέσος όρος της IoU όλων των κλάσεων και είναι εξίσου δημοφιλής μετρική που χρησιμοποιείται ευρέως στην αξιολόγηση σύγχρονων αλγορίθμων Κατάτμησης Εικόνων.

Σε προβλήματα Ανίχνευσης Αντικειμένων, ένα ορισμένο κατώφλι για τη μετρική IoU -συνήθως μεγαλύτερο ή ίσο από την τιμή 0.5- μπορεί να καθορίσει αν η πρόβλεψη για ένα αντικείμενο είναι τελικά σωστή ή όχι. Έστω ότι στο δείγμα εισόδου υπάρχει το αντικείμενο x. Αν το μοντέλο βρήκε αυτό το αντικείμενο και η μετρική IoU είναι μεγαλύτερη από το κατώφλι, τότε η τελική πρόβλεψη θεωρείται Αληθώς Θετική (True Positive - TP), ενώ αν η μετρική IoU είναι μικρότερη από το κατώφλι, τότε η τελική πρόβλεψη θεωρείται Ψευδώς Θετική (False Positive - FP). Αν το μοντέλο εντόπισε το αντικείμενο x στην σωστή θέση, αλλά η κατηγοριοποίηση είναι λάθος, τότε η τελική πρόβλεψη θεωρείται Ψευδώς Αρνητική (False Negative - FN). Η τελευταία περίπτωση δεν είναι χρήσιμη καθώς αφορά όλα τα μέρη της εικόνας εισόδου όπου το αντικείμενο x είναι απόν και το μοντέλο όντως δεν προβλέπει ότι υπάρχει. Με βάση τα παραπάνω, υπολογίζονται οι δείκτες **Ακρίβειας (Precision)** και **Ανάκλησης (Recall)**:

$$Presicion = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Ο δείκτης της **Μέσης Ακρίβειας (Average Precision - AP)** ορίζεται ως η περιοχή κάτω από την καμπύλη Ακρίβειας-Ανάκλησης μετά από παρεμβολή (interpolation). Συνήθως η τιμή της Ανάκλησης διαιρείται σε έναν αριθμό από σημεία και στη συνέχεια υπολογίζεται ο μέσος όρος των Ακριβειών σε αυτές τις τιμές. Αν υπολογιστεί στη συνέχεια ο μέσος όρος των AP σε όλες τις κατηγορίες του προβλήματος, προκύπτει η μετρική **mean Average Precision (mAP)**.

Μιας και τα μοντέλα Ανίχνευσης Αντικειμένων που χρησιμοποιούνται στην παρούσα διπλωματική είναι εκπαιδευμένα στο σύνολο δεδομένων COCO 2017, η μετρική αξιολόγησης mAP σε αυτό υπολογίζεται χρησιμοποιώντας 101 σημεία για τον υπολογισμό των AP που αντιστοιχούν σε διαφορετικά κατώφλια IoU. Το προκαθορισμένο εύρος κατωφλιών είναι από 0.5 μέχρι 0.95 με βήμα 0.05, οπότε προκύπτει η τελική μετρική AP@[.50:.05:.95] [24].

Για προβλήματα Εκτίμησης 2D Ανθρώπινης Πόζας έχουν αναπτυχθεί οι ακόλουθες βασικές μετρικές που καταφέρνουν και αποδίδουν αρκετά καλά κατά την εκπαίδευση.

Το **Percentage of Correct Parts (PCP)** είναι μια μετρική που χρησιμοποιούνταν συχνά παλαιότερα. Ο εντοπισμός ενός άκρου θεωρείται σωστός όταν η απόσταση μεταξύ πραγματικής και προβλεπόμενης θέσης είναι μικρότερη από ένα ορισμένο κλάσμα του μήκους του άκρου. Το κλάσμα κυμαίνεται μεταξύ των τιμών 0.1 και 0.5 και όσο μεγαλύτερο PCP έχει το μοντέλο τόσο πιο ακριβές θεωρείται. Ωστόσο, η συγκεκριμένη μετρική «τιμωρεί» τα άκρα με μικρό μήκος που είναι δύσκολο να εντοπιστούν. Για αυτό τον λόγο εισάχθηκε η μετρική **Percentage of Detected Joints (PDJ)**, με βάση την οποία μια προβλεπόμενη άρθρωση θεωρείται ότι σωστή αν η διαφορά μεταξύ πραγματικής και προβλεπόμενης θέσης βρίσκεται εντός ενός ορισμένου κλάσματος της διαμέτρου του κορμού.

Μια άλλη μετρική που χρησιμοποιεί κατώφλια είναι η **Percentage of Correct Keypoints (PCK)**. Όταν το κατώφλι είναι 0.5, τότε προκύπτει η μετρική PCKh@0.5 με βάση την οποία ο εντοπισμός ενός άκρου είναι σωστός αν η διαφορά πραγματικής και προβλεπόμενης θέσης είναι μικρότερη από το μισό του μήκους του κεφαλιού. Αν το κατώφλι καθοριστεί 0.2, τότε προκύπτει η μετρική PCK@0.2 που αντιστοιχεί στην μετρική PDJ με βάση την οποία η διαφορά θα πρέπει να είναι μικρότερη από το 20% της διαμέτρου του κορμού. Όσο υψηλότερη είναι η τιμή του PCK, τόσο πιο ακριβές είναι και το μοντέλο [20].

Τέλος, εκτός από τα παραπάνω, για μοντέλα Εκτίμησης Πόζας μπορούν να χρησιμοποιηθεί και η μετρική AP. Για παράδειγμα, για τα μοντέλα MoveNet και BlazePose της Google, η αξιολόγηση στο σύνολο δεδομένων COCO keypoints γίνεται χρησιμοποιώντας τη μετρική **Keypoint mean Average Precision** με **Object Keypoint Similarity (OKS)** [25].

Κεφάλαιο 4: Τεχνολογίες

Για την ανάπτυξη της παρούσας εργασίας χρησιμοποιήθηκαν τεχνολογίες που είναι πολύ δημοφιλείς κατά την περίοδο της συγγραφής της. Οι γλώσσες προγραμματισμού, το προγραμματιστικό περιβάλλον και οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι σημαντικά εργαλεία για την ανάπτυξη εφαρμογών σε λειτουργικό σύστημα Android.

4.1 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που δημιουργήθηκε από την εταιρεία Sun Microsystems. Η ιστορία της ξεκινά στις αρχές του 1991, όταν η Sun προσπαθούσε να βρει μια γλώσσα ανεξάρτητης πλατφόρμας για υλοποίηση λογισμικού μικρών συσκευών. Εκείνη την εποχή ήταν δημοφιλής η γλώσσα C++, η οποία ήταν σχεδιασμένη να μεταγλωττίζεται σε έναν μοναδικό τύπο CPU, πράγμα που οδηγούσε στην ανάγκη κάθε συσκευή να έχει διαφορετικό μεταγλωττιστή. Ο James Gossling θεωρείται ο πατέρας της Java, αφού παίρνοντας σαν βάση την γλώσσα C++ δημιούργησε κάποιες πειραματικές γλώσσες που οδήγησαν στην γλώσσα Oak. Η γλώσσα αυτή ήταν σχεδιασμένη να δημιουργεί κώδικα που ήταν ανεξάρτητος της CPU.

Το 1995 η γλώσσα μετονομάστηκε σε Java και παρουσιάστηκε επίσημα για πρώτη φορά στο συνέδριο Sun World 1995. Την ίδια περίοδο αναπτυσσόταν ο Παγκόσμιος Ιστός (World Wide Web), ο οποίος ζητούσε την ύπαρξη φορητών προγραμμάτων. Έτσι η γλώσσα άλλαξε μορφή για να καλύψει τις ανάγκες του Διαδικτύου. Η μορφή που έχει σήμερα το διαδίκτυο και από πλευράς πελάτη (client) και από πλευράς εξυπηρετητή (server) οφείλεται στην γλώσσα Java.

Ο τρόπος λειτουργίας της γλώσσας ήταν πολύ επαναστατικός για την εποχή κι έτσι το Δεκέμβριο του 1995 ξεκίνησαν μεγάλες εταιρείες κατασκευής λογισμικού να χρησιμοποιούν τη Java, όπως είναι η IBM, η Mitsubishi Electronics, κ.α. Από τότε η γλώσσα γινόταν όλο και πιο δημοφιλής [26].

Στις 13 Νοεμβρίου του 2006 η Java έγινε γλώσσα ανοιχτού κώδικα (GPL) όσον αφορά τον μεταγλωττιστή javac και το πακέτο ανάπτυξης JDK. Μέχρι το Μάιο του 2007 όλος ο πυρήνας της γλώσσας είχε μετατραπεί σε ανοιχτού κώδικα [26]. Ο μεταγλωττιστής της Java μετατρέπει τον κώδικα σε bytecode, που διερμηνεύεται από το Java Runtime Environment (JRE) ή τη Java Virtual Machine (JVM). Αυτά μεταφράζουν το bytecode σε κάθε υπολογιστή. Με αυτόν τον τρόπο ο κώδικας δεν χρειάζεται να αλλάζει μορφή για διαφορετικές πλατφόρμες [27].

Το 2009 η Oracle αγόρασε την εταιρία Sun Microsystems και είχε συνεχώς ανοδική πορεία στη δημοτικότητά της. Το 2015 έκανε την εμφάνισή της η γλώσσα Python που άρχισε να γίνεται πολύ δημοφιλής. Σήμερα βλέπουμε ότι η Python ξεπερνά σε δημοτικότητα την Java αλλά η Java παραμένει δεύτερη [28]. Συγκεκριμένα συνεχίζει να χρησιμοποιείται ευρέως σε εφαρμογές κινητών συσκευών, στο Διαδίκτυο, σε υπολογιστές, σε εφαρμογές νέφους, κ.α. Για αυτό η Oracle κάθε έξι μήνες κυκλοφορεί μία ανανεωμένη έκδοση, με την πιο πρόσφατη να είναι η Java SE 17, που κυκλοφόρησε το Σεπτέμβριο του 2021 [29].

4.2 Android

Το Android είναι ένα λειτουργικό σύστημα για κινητές συσκευές που βασίζεται στον πυρήνα Linux. Αναπτύχθηκε από τη Google και στη συνέχεια εξελίχθηκε από την Open Handset Alliance. Είναι ένα λογισμικό ανοιχτού κώδικα (open source), επιτρέποντας στους κατασκευαστές λογισμικού να γράφουν κώδικα σε γλώσσα Java.

Η ιστορία του Android ξεκινάει το 2003, πριν την εμφάνιση των έξυπνων κινητών συσκευών. Εκείνη τη χρονιά ιδρύθηκε στην Καλιφόρνια της Αμερικής η Android Inc. με σκοπό να αναπτύξει έξυπνες κινητές συσκευές που γνωρίζουν περισσότερο την τοποθεσία και τις προτιμήσεις του κατόχου της. Το 2005 γίνεται η αγορά της εταιρείας Android Inc. από την Google. Η Google είχε σκοπό να χρησιμοποιήσει το πυρήνα Linux ως βάση. Μετά από δύο χρόνια αποφασίστηκε η ίδρυση της Open Handset Alliance, μία συνύπαρξη πολλών εταιριών τηλεπικοινωνιών, λογισμικού και κατασκευής υλικού, με σκοπό την ανάπτυξη του λογισμικού αυτού.

Τον Σεπτέμβριο του 2008 ανακοινώθηκε η πρώτη κινητή συσκευή με λειτουργικό σύστημα Android 1.0. Από τότε η Google κάθε χρόνο κάνει αναβαθμίσεις ώστε να βελτιώνεται και να παρέχει συνεχώς καινούργιες δυνατότητες στον χρήστη. Το 2009 εκδίδεται το Android 1.5 Cupcake, η πρώτη έκδοση που πήρε δημόσιο κωδικό όνομα. Στη συνέχεια όλες οι εκδόσεις έπαιρναν ονομασίες από γλυκά μέχρι και την έκδοση 9.0 Pie. Η τελευταία έκδοση είναι η Android 12 που κυκλοφόρησε τον Οκτώβριο του 2021. Σήμερα το Android κατέχει το μεγαλύτερο ποσοστό χρηστών παγκοσμίως με ποσοστό 72% και ακολουθεί το σύστημα iOS της Apple [30].

4.2.1 Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (Integrated Development Environment - IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Το 2013 η Google έκανε διαθέσιμη τη δοκιμαστική έκδοση 0.1 για προεπισκόπηση και το Δεκέμβριο του 2014 λάνσαρε την πρώτη σταθερή έκδοση 1.0 [31]. Το Android Studio βασίζεται στο IntelliJ IDEA και είναι ελεύθερο για τον χρήστη με την άδεια Apache License 2.0 [32].

Ο χρήστης μπορεί εύκολα να δημιουργήσει εφαρμογές για κάθε συσκευή Android. Παρέχει στο χρήστη πολλά εργαλεία όπως έναν εξομοιωτή κινητών συσκευών, εργαλεία για να δοκιμάζεται και να υπολογίζεται η απόδοση, ακόμη και διασύνδεση με το Github για εύκολη χρήση κώδικα από αυτό. Η Google έχει προσπαθήσει να κάνει το Android Studio όσο πιο εύκολο στη χρήση. Δίνει συμβουλές σε πραγματικό χρόνο κατά την συγγραφή του κώδικα, δείχνοντας προτάσεις για διόρθωση λαθών ή για αυτόματη συμπλήρωση εντολών [31]. Παραδοσιακά, η πιο δημοφιλής γλώσσα προγραμματισμού για το Android Studio είναι η Java, όμως σήμερα όλο και περισσότεροι προγραμματιστές χρησιμοποιούν την Kotlin.

4.3 Python

Η Python είναι μια γλώσσα προγραμματισμού γενικού σκοπού και υψηλού επιπέδου. Η εκμάθηση της γλώσσας είναι απλή και εύκολη για αυτό επιλέγεται η χρήση της τόσο από έμπειρους προγραμματιστές όσο και από αρχάριους. Η σύνταξη της γλώσσας είναι κομψή και οι τύποι που χρησιμοποιούνται σε αυτήν είναι δυναμικοί. Ο κώδικας αυτής της γλώσσας ομαδοποιείται σε ενότητες (modules) και πακέτα (packages) και επεξεργάζεται από έναν διερμηνέα (interpreter). Με τη χρήση της μπορούν να αναπτυχθούν απλές εφαρμογές εκπαιδευτικού σκοπού μέχρι και ολοκληρωμένες σύνθετες εφαρμογές.

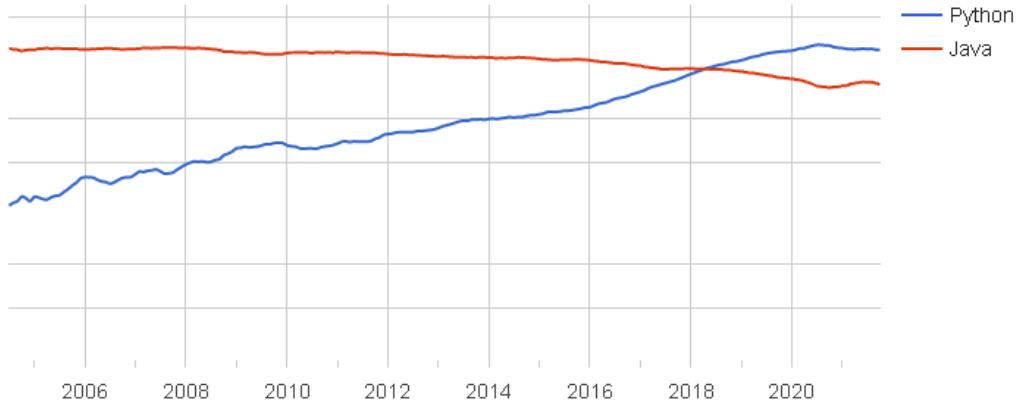
Η Python υποστηρίζει πολλές γνωστές αποτελεσματικές προγραμματιστικές προσεγγίσεις όπως είναι ο αντικειμενοστραφής, ο διαδικαστικός και ο συναρτησιακός προγραμματισμός, διαθέτοντας δομές δεδομένων υψηλού επιπέδου. Διαθέτει πολλές έτοιμες βιβλιοθήκες που είναι εύκολα προσβάσιμες. Επιπλέον, δίνεται η δυνατότητα επέκτασης νέων βιβλιοθηκών σε γλώσσα προγραμματισμού C και C++. Σε σύγκριση με γλώσσες όπως οι C, C++ και Java, τα προγράμματα που γράφονται σε Python είναι συμπαγή, ευανάγνωστα και γράφονται και συντηρούνται γρηγορότερα.

Εκτός από πολλά πλεονεκτήματα που διαθέτει η Python έχει και κάποια μειονεκτήματα που θα αναφερθούν παρακάτω. Εφόσον πρόκειται για μια διερμηνευόμενη γλώσσα προγραμματισμού, ο χρόνος εκτέλεσης των προγραμμάτων είναι πιο αργός από άλλες γλώσσες προγραμματισμού που χρησιμοποιούν μεταγλωττιστές (compilers) αντί για διερμηνείς. Για αυτόν τον λόγο δεν είναι αποδοτική γλώσσα προγραμματισμού για τη δημιουργία λειτουργικών συστημάτων. Αυτό το μειονέκτημα αντισταθμίζεται με το γεγονός ότι ο χρόνος ανάπτυξης μιας εφαρμογής σε γλώσσα Python είναι πολύ μικρότερος σε σύγκριση με άλλων κωδίκων σε γλώσσα C, C++ ή Java.

Η ιστορία της Python ξεκινάει το 1989 στην Ολλανδία και συγκεκριμένα στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) από τον Guido van Rossum. Η βασική πηγή έμπνευσης της Python ήρθε από την γλώσσα προγραμματισμού ABC. Αρχικά δημιουργήθηκε σαν γλώσσα σεναρίων για το κατανεμημένο λειτουργικό σύστημα Amoeba. Σχεδιάστηκε με τέτοιο τρόπο ώστε να επεκτείνεται εύκολα, να παρέχει ενσωματωμένα στοιχεία όπως εντολές και τύπους αντικειμένων και να δίνει την δυνατότητα στους προγραμματιστές να προσθέτουν επιπλέον στοιχεία για την κάλυψη αναγκών των συστημάτων που χρησιμοποιούν.

Η πρώτη έκδοση κυκλοφόρησε το 1991 και εξελίσσεται γρήγορα με νέες εκδόσεις. Σε όλες τις εκδόσεις διατηρείται ίδια γλώσσα με διαφορές σε λεπτομέρειες όπως στη λειτουργία των ενσωματωμένων αντικειμένων (λεξικά, συμβολοσειρές, κτλ.). Στις 16 Οκτωβρίου του 2000 κυκλοφορεί η έκδοση 2.0 και στις 3 Δεκεμβρίου του 2008 κυκλοφορεί η έκδοση 3.0. Επειδή η γλώσσα είναι συμβατή και προς τα πίσω, πολλά χαρακτηριστικά της Python 3.0 υιοθετήθηκαν από τις εκδόσεις 2.6 και 2.7 για να διορθωθούν κάποια λάθη και να γίνει σαφής ο απλός τρόπος που εκτελούνται κάποια πράγματα. Τη στιγμή της συγγραφής αυτής της εργασίας η πιο πρόσφατη έκδοση είναι η 3.9.2 που δημοσιεύτηκε στις 26 Φεβρουαρίου του 2021.

Τα τελευταία πέντε χρόνια η Python γίνεται η δημοφιλέστερη γλώσσα προγραμματισμού και ακολουθείται από τη Java που μέχρι το 2018 κρατούσε τα ηνία στον τομέα του προγραμματισμού γενικής χρήσης, όπως φαίνεται και στο Σχήμα 4.4 παρακάτω [33].



Σχήμα 4.1: *Δημοτικότητα γλωσσών προγραμματισμού Java και Python*

4.4 TensorFlow

Το TensorFlow είναι μια μαθηματική βιβλιοθήκη που σχεδιάστηκε για εφαρμογές Μηχανικής Μάθησης πάνω σε δύσκολα προβλήματα σε ετερογενή περιβάλλοντα. Σχεδιάστηκε από την Google για να αντιμετωπίσει τα προβλήματα που παρουσιάζονταν κατά την εκπαίδευση των νευρωνικών δικτύων με το παλαιότερο σύστημα DistBelief. Η δημόσια διανομή του TensorFlow έγινε επίσημα στις 9 Νοεμβρίου 2015 και από τότε γίνεται εκτεταμένη χρήση του σε διάφορες εφαρμογές όπως αναγνώριση φωνής, ανακάλυψη νέων φαρμάκων, κ.α. [34].

Για να εκτελούνται γρηγορότερα οι αριθμητικοί υπολογισμοί η βιβλιοθήκη TensorFlow παρέχει στον προγραμματιστή τη δυνατότητα αξιοποίησης γράφων ροής δεδομένων. Ένας γράφος είναι μία μαθηματική κατασκευή που αναπαριστά τις μαθηματικές πράξεις σαν κόμβους και τις σχέσεις των δεδομένων σαν ακμές. Οι ακμές απεικονίζουν πολυδιάστατους πίνακες δεδομένων που ονομάζονται τανυστές (tensors). Επίσης, προσφέρεται η δυνατότητα παράλληλης επεξεργασίας δεδομένων από πολλές διαφορετικές CPU και GPU.

Η βιβλιοθήκη TensorFlow έχει γίνει πολύ δημοφιλής από την δημόσια διανομή της και έπειτα καθώς προσφέρει πολλά πλεονεκτήματα. Το TensorFlow μπορεί να χαρακτηριστεί ως ένα ισχυρό πακέτο (framework) Μηχανικής Μάθησης για ευρεία γκάμα πεδίων, όπως η ιατρική και η δασολογία, που εξειδικεύεται σε μεγάλα σύνολα δεδομένων Μηχανικής Μάθησης και σε εφαρμογές Βαθιάς Μάθησης. Επιπλέον είναι πολύ εύκολο στην ανάπτυξη κώδικα και στην μεταποίηση της δομής του δικτύου το οποίο χρησιμοποιεί ο κάθε προγραμματιστής.

Με το TensorFlow και το -πλέον ενσωματωμένο σε αυτό- Keras μπορούν εύκολα να δημιουργηθούν περίπλοκα και εκτεταμένα νευρωνικά δίκτυα με χρήση λίγων γραμμών κώδικα. Ο κώδικας αυτός μπορεί να γραφτεί σε διάφορες γλώσσες προγραμματισμού όπως είναι Python, Java, C, C++ κ.α.

Επιπλέον, στη βιβλιοθήκη TensorFlow υπάρχει γρήγορη, εύκολη και αποδοτική εισαγωγή και επεξεργασία των δεδομένων χαρακτηριστικό που την καθιστά σημαντικό εργαλείο για μοντέλα Μηχανικής Μάθησης. Τέλος, ένα άλλο χαρακτηριστικό είναι ότι έχει μεγάλη ευελιξία αφού με μικρές αλλαγές στον ίδιο κώδικα μπορεί να μετατραπεί αξιοποιήσιμος από πολλές αρχιτεκτονικές, διαφορετικές φορητές συσκευές μέχρι και διανεμημένα συμπλέγματα υπολογιστών. Στο TensorFlow Hub υπάρχουν πολλά μοντέλα Μηχανικής Μάθησης που είναι διαθέσιμα σε όλους [35].

4.4.1 TensorFlow Lite

Το TensorFlow Lite είναι ένα σύνολο εργαλείων που βοηθάει τον προγραμματιστή να τρέξει μοντέλα Βαθιάς Μάθησης σε κινητές συσκευές ή συσκευές IoT. Δίνει τη δυνατότητα σε εκπαιδευμένα μοντέλα TensorFlow να ενσωματωθούν σε συσκευές αιχμής, όπως είναι οι κινητές συσκευές και να εκτελούν συμπερασματολογία τόσο τοπικά (on device) όσο και στην αιχμή (at the edge) [36]. Το TensorFlow Lite έχει δύο βασικά συστατικά:

1. TensorFlow Lite Converter. Πρώτη και κύρια λειτουργία του είναι η μετατροπή των TensorFlow μοντέλων σε FlatBuffers. Με αυτόν τον τρόπο παίρνει μεγάλα σε μέγεθος μοντέλα και μπορεί να μειώσει το μέγεθος χωρίς να μειώνει την ορθότητά τους. Επιπλέον με την μετατροπή εισάγονται βελτιστοποιήσεις, όπως η κβαντοποίηση ή η μείωση των παραμέτρων, για να μειωθεί ο χρόνος εκτέλεσης και το μέγεθος.

2. TensorFlow Lite Interpreter. Τα ελαφριά μοντέλα που παράγει ο μετατροπέας παίρνει ο διερμηνέας και τα εκτελεί σε πολλούς διαφορετικούς τύπους υλικού. Επίσης χρησιμοποιούνται «εκπρόσωποι» (delegates) για να στείλουν μέρη από το δίκτυο σε κάποιον επιταχυντή όπως μια GPU ή NPU. Έτσι μειώνεται ο χρόνος εκτέλεσης και επιτυγχάνονται υψηλότερες επιδόσεις [37].

Κβαντοποίηση

Για την διαδικασία της κβαντοποίησης που πραγματοποιεί ο TensorFlow Lite Converter υπάρχουν πολλά διαφορετικά είδη. Στον Πίνακα 4.1 παρακάτω παρατίθενται τα 4 πιο βασικά είδη κβαντοποίησης, οι τύποι μεταβλητών στην είσοδο, στην έξοδο και στα βάρη του μοντέλου, καθώς και η μείωση του μεγέθους κάθε μοντέλου.

Τεχνική	Είσοδος	Βάρη	Έξοδος	Μέγεθος
Μη κβαντισμένα	fp32 / int64	fp32	fp32 / int64	-
Dynamic Range (DR)	fp32 / int64	int8*	fp32 / int64	4x μικρότερο
Integer (INT)	fp32 / int64	int8	fp32 / int64	
Full Integer (FULL)	int8	int8	int8	
Float 16 (FP16)	fp32 / int64	fp16	fp32 / int64	2x μικρότερο

Πίνακας 4.1: *Είδη κβαντοποίησης και τύποι μεταβλητών*

Η πιο απλή μέθοδος κβαντοποίησης είναι η μετατροπή των βαρών του δικτύου από κινητής υποδιαστολής των 32 bits σε ακεραίους των 8 bits, χωρίς αλλαγές στην είσοδο ή την έξοδο του δικτύου. Όπως φαίνεται και από τον παραπάνω πίνακα, αυτό συμβαίνει κατά την dynamic range και την integer κβαντοποίηση. Η διαφορά μεταξύ των δύο τεχνικών έρχεται από το γεγονός ότι για πλήρη κβαντοποίηση ακεραίων πρέπει το εύρος όλων των τανυστών κινητής υποδιαστολής στο μοντέλο να βαθμονομηθεί. Ωστόσο, οι μη στατικοί τανυστές, όπως είναι οι ενδιάμεσες ενεργοποιήσεις, δεν μπορούν να βαθμονομηθούν αν δεν εκτελεστούν πρώτα κάποιοι κύκλοι συμπερασματολογίας. Επομένως, για την integer κβαντοποίηση απαιτείται ένα μικρό αντιπροσωπευτικό σύνολο δεδομένων για την βαθμονόμηση, ενώ για την dynamic range κβαντοποίηση δεν απαιτείται κάτι τέτοιο, καθώς πριν τη συμπερασματολογία, ο διερμηνέας μετατρέπει τα κβαντισμένα βάρη σε αριθμούς κινητής υποδιαστολής, ώστε η εκτέλεση να γίνει και από πυρήνες κινητής υποδιαστολής (floating-point kernels). Αυτή η διαδικασία γίνεται μια φορά και αποθηκεύεται στην προσωρινή μνήμη, ώστε να μην προστίθεται σε κάθε εκτέλεση ο επιπρόσθετος χρόνος της μετατροπής.

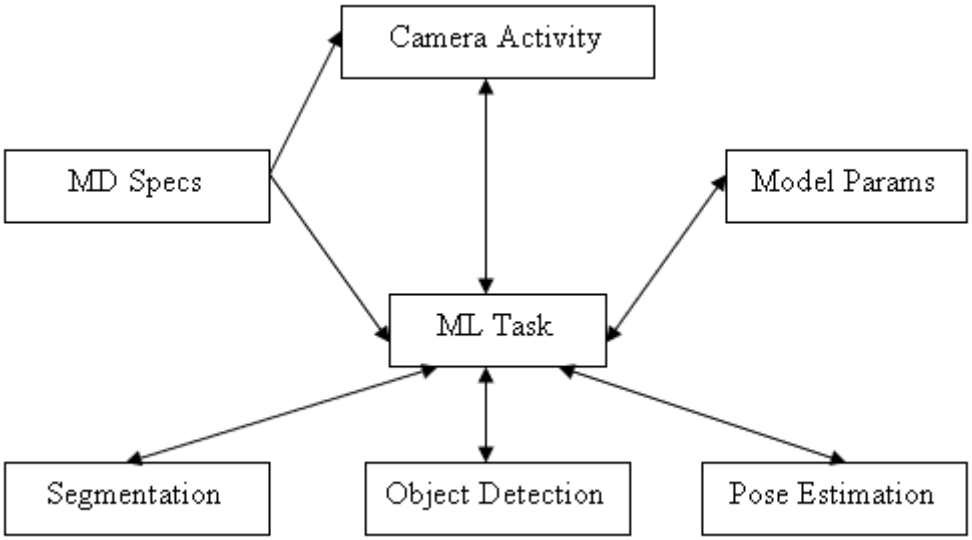
Στην περίπτωση που το μοντέλο προορίζεται να ενσωματωθεί σε κάποιον μικροεπεξεργαστή (microcontroller), χαρακτηριστικό των οποίων είναι η έλλειψη πυρήνων κινητής υποδιαστολής, εκτός από τα βάρη, είναι απαραίτητο και οι είσοδοι και οι έξοδοι να αναπαρίστανται από ακεραίους αριθμούς. Έτσι, προκύπτει η full integer κβαντοποίηση. Και με τις τρεις παραπάνω τεχνικές επιτυγχάνεται 4 φορές μικρότερο μέγεθος μοντέλου, ενώ αναμένεται επιτάχυνση 2 με 3 φορές σε σχέση με το αρχικό μοντέλο.

Τέλος, η κβαντοποίηση των βαρών του μοντέλου σε αριθμούς κινητής υποδιαστολής των 16 bits προτιμάται για την εκτέλεση της συμπερασματολογίας σε GPU, οι οποίες μπορούν να διαχειριστούν απευθείας αριθμούς των 16 bits. Το μέγεθος του μοντέλου μειώνεται στο μισό και η ακρίβεια επηρεάζεται ελάχιστα σε σύγκριση με τα τρία προηγούμενα είδη.

Κεφάλαιο 5: Περιγραφή Εφαρμογής

Η εφαρμογή αναπτύχθηκε για να λειτουργεί σε κινητές συσκευές με λειτουργικό σύστημα Android. Συγκεκριμένα, πρόκειται για μια «έξυπνη» κάμερα, η οποία επεξεργάζεται σε πραγματικό χρόνο οτιδήποτε βρίσκεται μπροστά από τον αισθητήρα της κινητής συσκευής. Με την υποστήριξη διαφορετικών διεργασιών Βαθιάς Μάθησης, η εφαρμογή μπορεί να εκτελεί κατάτμηση του τωρινού frame, ανίχνευση των αντικειμένων που βρίσκονται σε αυτό ή εκτίμηση της πόζας ενός ανθρώπου.

Η εφαρμογή απαρτίζεται συνολικά από 7 Java κλάσεις, ενώ η δομή της επιτρέπει την εύκολη επέκτασή της και για την υποστήριξη επιπρόσθετων διεργασιών. Στο Σχήμα 5.1 παρακάτω αναπαρίσταται ένα διάγραμμα των κλάσεων της εφαρμογής και των αλληλεπιδράσεων μεταξύ τους, ενώ στις επόμενες Ενότητες περιγράφονται τα βασικά χαρακτηριστικά τους.



Σχήμα 5.1: *Διάγραμμα κλάσεων της εφαρμογής*

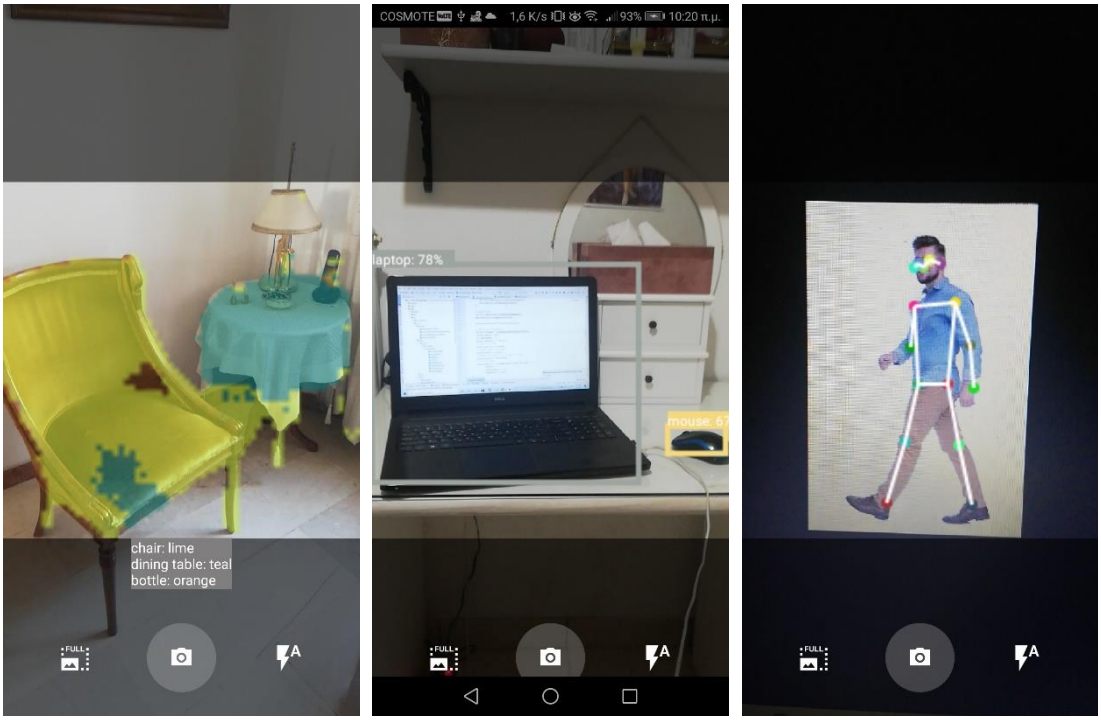
5.1 CameraActivity

Η CameraActivity είναι η κύρια (main) δραστηριότητα της εφαρμογής και σε αυτή καθορίζεται η οθόνη που εμφανίζεται όταν ο χρήστης εκκινεί την εφαρμογή. Εφόσον η οθόνη της εφαρμογής είναι μια κάμερα, ένα μεγάλο μέρος αυτής της κλάσης ασχολείται με την ανάπτυξη των λειτουργιών της κάμερας.

Οι δραστηριότητες (activities) είναι ένα από τα βασικά δομικά στοιχεία μιας Android εφαρμογής και χαρακτηρίζονται από έναν ξεχωριστό κύκλο ζωής που καθορίζει τον τρόπο δημιουργίας και καταστροφής τους. Για τη μετάβαση μεταξύ των σταδίων του κύκλου ζωής μιας δραστηριότητας υπάρχει ένα σύνολο από έξι βασικές κλήσεις (callbacks): onCreate(), onStart(), onResume(), onPause(), onStop() και onDestroy(). Το σύστημα καλεί κάθε μια από αυτές τις μεθόδους καθώς η δραστηριότητα εισέρχεται σε μια νέα κατάσταση.

Στιγμιότυπα

Ανάλογα με τη διεργασία που εκτελεί η εφαρμογή, παρουσιάζει και το αντίστοιχο αποτέλεσμα στον χρήστη. Στο Σχήμα 5.2 φαίνονται παραδείγματα του αποτελέσματος που θα λάβει ο χρήστης ανάλογα με τη διεργασία Βαθιάς Μάθησης.



(α) *(β)* *(γ)*

Σχήμα 5.2: *Στιγμιότυπα οθόνης για τις τρεις διεργασίες*

Όπως φαίνεται στο Σχήμα 5.2 (α), στην Κατάτμηση Εικόνας, τα αντικείμενα που αναγνωρίζονται χρωματίζονται με μια συγκεκριμένη απόχρωση και κάτω από την εικόνα αναγράφεται σε γκρι πλαίσιο το αντικείμενο που αναγνωρίσθηκε και το χρώμα που του αντιστοιχεί. Αντίστοιχα, στο Σχήμα 5.2 (β) για την Ανίχνευση Αντικειμένων, τα αντικείμενα που εντοπίζονται οριοθετούνται από χρωματισμένα πλαίσια, στο πάνω μέρος των οποίων αναγράφεται η κλάση του αντικειμένου και η πεποίθηση του μοντέλου. Τέλος, στο Σχήμα 5.2 (γ) φαίνεται το αποτέλεσμα της Εκτίμησης Ανθρώπινης Πόζας. Τοποθετούνται σημεία στις συντεταγμένες των αρθρώσεων που έχουν εντοπιστεί και ενώνοντάς τα, σχηματίζεται τελικά ο σκελετός του ανθρώπου.

Στις παραπάνω εικόνες, υπάρχουν τρία εικονίδια που αφορούν -από τα αριστερά προς τα δεξιά- την αναλογία προεπισκόπησης κάμερας, την λήψη εικόνας και την λειτουργία του φλας. Η εφαρμογή υποστηρίζει μέχρι τέσσερις διαφορετικές αναλογίες που εμφανίζονται σε όσες συσκευές τις υποστηρίζουν. Οι αναλογίες αυτές είναι 1:1, 4:3, 16:9 και FULL, η οποία είναι η αναλογία της οθόνης εφόσον δεν συγκαταλέγεται σε μια από τις προηγούμενες αναλογίες. Αντίστοιχα για το φλας, η εφαρμογή υποστηρίζει μέχρι πέντε διαφορετικές λειτουργίες (modes):

- SINGLE: Το flash ενεργοποιείται στιγμιαία κατά τη λήψη της φωτογραφίας.
- OFF: Το flash είναι απενεργοποιημένο.
- TORCH: Το flash είναι ενεργοποιημένο συνέχεια σαν φακός.
- AUTO: Το flash ενεργοποιείται κατά τη λήψη της φωτογραφίας μόνο αν ο φωτισμός είναι χαμηλός. Για να ενεργοποιηθεί αυτή η λειτουργία γίνεται

κατάλληλος έλεγχος της συσκευής αν υποστηρίζει τη λειτουργία αυτόματης έκθεσης (auto exposure).

- AUTO_REDEYE: Είναι η λειτουργία AUTO με ένα επιπλέον χαρακτηριστικό την αυτόματη μείωση των κόκκινων ματιών.

Τέλος, μια επιπρόσθετη λειτουργία που έχει ορισθεί είναι η λειτουργία μεγέθυνσης - σμίκρυνσης της προεπισκόπησης της κάμερας (zoom) που πραγματοποιείται με τη μέθοδο pinch-to-zoom. Για τη μεγέθυνση τα δύο δάχτυλα πρέπει να απομακρύνονται το ένα από το άλλο, ενώ για τη σμίκρυνση τα δάχτυλα τείνουν να ενώνονται.

5.2 MDSpecs

Η συγκεκριμένη κλάση παρέχει πληροφορίες σχετικά με τη συσκευή, όπως είναι ο επεξεργαστής, η ανάλυση της κάμερας, οι διαθέσιμες αναλογίες προεπισκόπησης, κ.α. Η λειτουργία αυτής της κλάσης είναι να αποθηκεύει και να διαχειρίζεται δεδομένα που σχετίζονται με την αλληλεπίδραση του χρήστη με συνειδητό τρόπο ως προς τον κύκλο ζωής κάθε δραστηριότητας. Με άλλα λόγια, είναι υπεύθυνη για την διατήρηση των δεδομένων σε περίπτωση αλλαγών της διαμόρφωση του UI, όπως είναι η αλλαγή της αναλογίας της προεπισκόπησης της κάμερας κατά την οποία διαγράφεται η κύρια δραστηριότητα και επανεκκινείται.

Όπως φαίνεται και στο Σχήμα 5.1, η MDSpecs επικοινωνεί με τις κλάσεις CameraActivity και MLTask -που περιγράφεται παρακάτω- παρέχοντάς τους τις απαραίτητες πληροφορίες που σχετίζονται με την εκάστοτε κινητή συσκευή.

5.3 MLTask

Αυτή η κλάση είναι υπεύθυνη για το κομμάτι της Μηχανικής Μάθησης. Κατά την αρχικοποίηση της κλάσης MLTask αρχικοποιείται ο Interpreter που θα εκτελεί τη συμπερασματολογία και διαβάζονται τα metadata από το μοντέλο Βαθιάς Μάθησης που έχει επιλεγεί. Καθώς διαβάζονται τα metadata, αυτά εισάγονται σε ένα αντικείμενο της κλάσης ModelParams, η οποία έχει τρεις εσωτερικές κλάσεις ανάλογα με τη διεργασία Βαθιάς Μάθησης.

Η MLTask έχει άμεση επικοινωνία με τις κλάσεις που είναι υπεύθυνες για κάθε διεργασία. Για την Ανίχνευση Αντικειμένων υπεύθυνη είναι η κλάση ObjectDetection, για την Κατάτμηση Εικόνας υπεύθυνη είναι η κλάση Segmentation και για την Εκτίμηση Ανθρώπινης Πόζας υπεύθυνη είναι η κλάση PoseEstimation. Σε κάθε μια από αυτές τις κλάσεις εκτελείται η συμπερασματολογία και επεξεργάζεται το αποτέλεσμα. Για παράδειγμα, στην κλάση PoseEstimation υπάρχει η μέθοδος drawPose(), η οποία παίρνει την έξοδο του μοντέλου και ζωγραφίζει τα σημεία και τους συνδέσμους, ώστε να δημιουργηθεί η εικόνα εξόδου που θα δει ο χρήστης.

Κεφάλαιο 6: Ανάπτυξη

Η εφαρμογή που περιεγράφηκε πιο πάνω ενσωματώθηκε σε μία κινητή συσκευή και ένα tablet ώστε να μετρηθεί ο χρόνος εκτέλεσης κάθε μοντέλου Βαθιάς Μάθησης. Τα τεχνικά χαρακτηριστικά των συσκευών και τα βασικά στοιχεία των μοντέλων είναι σημαντικές πληροφορίες για τα συμπεράσματα που αναμένονται από αυτήν την εργασία.

6.1 Συσκευές

Στην συγκεκριμένη εργασία χρησιμοποιήθηκαν δύο διαφορετικές κινητές συσκευές για την αξιολόγηση της εφαρμογής. Στον Πίνακα 6.1 έχουν συγκεντρωθεί τα χαρακτηριστικά κάθε συσκευής.

Κινητή Συσκευή	**Samsung Galaxy A20e (κινητό)**	**Samsung Galaxy Tab S7 (tablet)**
Κυκλοφορία	Μάιος 2019	Αύγουστος 2020
System on Chip (Soc)	Samsung Exynos 7 Octa 7884	Qualcomm Snapdragon 865+
CPU	2x 1.60 GHz ARM Cortex-A73 6x 1.35 GHz ARM Cortex-A53	1x 3.09 GHz Kryo 585 Prime 3x 2.42 GHz Kryo 585 Gold 4x 1.80 GHz Kryo 585 Silver
GPU	ARM Mali-G71	Qualcomm Adreno 650
RAM	3 GB	6 GB
Έκδοση Android	11 (API Level 30)	10 (API Level 29)
Back Facing Camera Hardware Level	LIMITED	LEVEL_3

Πίνακας 6.1: *Χαρακτηριστικά συσκευών*

Για λόγους απλότητας στη συνέχεια της εργασίας η συσκευή Samsung Galaxy A20e θα αναφέρεται ως κινητό και η συσκευή Samsung Galaxy Tab S7 ως tablet. Από τον παραπάνω Πίνακα αναμένεται ότι το tablet θα έχει υψηλότερες αποδόσεις, κυρίως λόγω της μεγαλύτερης μνήμης και του πιο σύγχρονου SoC που περιέχει. Επίσης, βασική διαφορά των δύο συσκευών είναι ότι το κινητό δεν περιέχει κάποια NPU, ενώ ο Snapdragon 865+ που βρίσκεται στο tablet έρχεται εξοπλισμένος με ισχυρό AI Engine που χρησιμοποιεί συνδυαστικά τη CPU, τη GPU και τον Hexagon 698 DSP (Digital Signal Processor). Οπότε περιμένουμε άρτια απόδοση όταν χρησιμοποιείται το NNAPI (βλ. Ενότητα 7.2) για την συμπερασματολογία στο tablet σε αντίθεση με το κινητό, όπου το NNAPI δεν θα δίνει ικανοποιητικά αποτελέσματα.

6.2 Μοντέλα

Για την αξιολόγηση του συστήματος επιλέχθηκαν διάφορες αρχιτεκτονικές βαθιών νευρωνικών δικτύων με διαφορετική πολυπλοκότητα και διαφορετικά σχήματα κβαντοποίησης. Στους παρακάτω Πίνακες:

- Η παύλα (-) στη στήλη Κβαντοποίηση υποδηλώνει ότι το μοντέλο της συγκεκριμένης γραμμής δεν είναι κβαντισμένο.
- Όλες οι τιμές πλήθους παραμέτρων προσεγγίστηκαν από τα tflite αρχεία αναλύοντας τους ενταμιευτές (buffers) και τις λειτουργίες (operations) που

εκτελούνται μεταξύ αυτών. Το μεγαλύτερο ποσοστό των παραμέτρων προέρχεται φυσικά από τα επίπεδα που εκτελούν πράξεις συνέλιξης.

- Όλα τα μεγέθη των μοντέλων αναφέρονται στα μεγέθη των tflite αρχείων μετά την προσθήκη των μεταδεδομένων (βλέπε Ενότητα 6.2.1).
- Όλες οι τιμές ακρίβειας που θα αναφερθούν παρακάτω υπολογίστηκαν χειροκίνητα στα αντίστοιχα σύνολα επικύρωσης (validation sets).

Κατάτμηση Εικόνας

Για την Κατάτμηση Εικόνας επιλέχθηκαν τα 14 μοντέλα που φαίνονται στον Πίνακα 6.2 ταξινομημένα κατά φθίνουσα σειρά ακρίβειας με βάση τη μετρική mIoU. Για κάθε μοντέλο φαίνεται η μέθοδος κβαντοποίησης, το μέγεθος εισόδου, το σύνολο ή τα σύνολα δεδομένων στα οποία έχει εκπαιδευτεί και σε παρένθεση ο αριθμός των τελικών κλάσεων, ο δείκτης ακρίβειας mIoU, το πλήθος των παραμέτρων και το μέγεθος του tflite αρχείου στον δίσκο σε MB.

Μοντέλο	Κβαντοποίηση	Μέγεθος Εισόδου	Σύνολα Δεδομένων	mIoU (%)	Αριθμός Παραμέτρων	Μέγεθος (MB)
Bilinear MUNet	-	128x128	PFCN / Supervisely / Web (2)	96.47	3.62 M	14.5
	DR			96.37		3.7
PrismaNet	-	256x256		95.61	0.92 M	3.7
	DR			95.61		1.7
	FULL			67.73		1.2
DeepLab v3 Xception 65	-	513x513	Pascal VOC (21)	87.12		164.0
	INT			N/A		43.8
DeepLab v3 MobileNet v2	-	513x513		80.70	2.09 M	8.4
	INT			N/A		2.7
UNet Industrial	-	512x512	DAGM2007 (2)	77.38	1.85 M	7.4
	DR			76.64		1.9
BiSeNet v2	-	256x256	Cityscapes (19)	62.36	2.45 M	9.9
ERFNet	-	256x512		72.10	2.05 M	8.3
	FULL			N/A		2.3

Πίνακας 6.2: *Μοντέλα Κατάτμησης Εικόνας*

Τα πέντε πρώτα μοντέλα βασίζονται στη δομή κωδικοποιητή - αποκωδικοποιητή (encoder-decoder) της αρχιτεκτονικής U-Net και έχουν εκπαιδευτεί για το ειδικό πρόβλημα της Κατάτμησης Πορτρέτων (Portrait Segmentation). Το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση είναι μια μίξη του PFCN μαζί με επίλεκτες εικόνες πορτρέτων από το Supervisely και τυχαίες εικόνες selfie από το διαδίκτυο.

Για το **Bilinear MUnet** χρησιμοποιείται ως κωδικοποιητής - εξαγωγέας χαρακτηριστικών ένα προεκπαιδευμένο μοντέλο MobileNet v2 με πολλαπλασιαστή βάθους (depth multiplier) 0.5, ενώ στον αποκωδικοποιητή έχουμε μπλοκ διεύρυνσης (upsampling blocks) που χρησιμοποιούν την μέθοδο της διγραμμικής παρεμβολής (bilinear interpolation) ακολουθούμενα από συνελικτικά επίπεδα (Upsample2D + Conv2D).

Οι αλλαγές που έχουν γίνει στην αρχιτεκτονική U-Net για τη δημιουργία του **PrismaNet** περιλαμβάνουν (α) την αντικατάσταση της συνένωσης χαρακτηριστικών (feature concatenation) με την πρόσθεση ανά στοιχείο (element-wise addition), (β) αντί

για το συνηθισμένο μπλοκ συνέλιξης ακολουθούμενο από τη συνάρτηση ενεργοποίησης ReLU (Conv2D + ReLU), χρησιμοποιείται ένα υπολειπόμενο μπλοκ (residual block) με διαχωριζόμενες κατά βάθος συνελίξεις (depth-wise separable convolutions) και (γ), για να βελτιωθεί η ακρίβεια, το τμήμα του αποκωδικοποιητή περιέχει περισσότερα μπλοκ από ότι ο κωδικοποιητής [38].

Στη συνέχεια, έχουμε δύο **DeepLab v3** μοντέλα με διαφορετικούς εξαγωγείς χαρακτηριστικών (backbones): το Xception 65 και το MobileNet v2. Χαρακτηριστικό της DeepLab αρχιτεκτονικής είναι οι διεσταλμένες συνελίξεις (dilated ή atrous convolutions) που χρησιμοποιούνται για τη διατήρηση της χωρικής ανάλυσης. Η τρίτη έκδοση του μοντέλου χρησιμοποιεί ένα δίκτυο Atrous Spatial Pyramid Pooling (ASPP) ως αποκωδικοποιητή για την ανίχνευση αντικειμένων σε διαφορετικές κλίμακες (scales), καθώς και χαρακτηριστικά σε επίπεδο εικόνας (image-level features) που κωδικοποιούν την καθολική εικόνα και αυξάνουν την ακρίβεια [39].

Στα μοντέλα συμπεριλαμβάνεται και μια τροποποιημένη U-Net αρχιτεκτονική, η οποία ονομάστηκε **UNet Industrial** και σχεδιάστηκε ώστε να αποδίδει πολύ καλά στο σύνολο δεδομένων DAGM2007, ένα μικρό σύνολο δεδομένων που δημιουργήθηκε για τον εντοπισμό ελαττωμάτων σε επιφάνειες με υφή, όπως περιεγράφηκε και στην Ενότητα 3.4. Το μοντέλο αποτελείται από έναν κωδικοποιητή με 3 μπλοκ περιορισμού (downsampling) αποτελούμενα από 2D συνελίξεις ακολουθούμενες από 2D μέγιστη συγκέντρωση και έναν αντίστοιχο αποκωδικοποιητή με 3 μπλοκ διεύρυνσης (upsampling) [40].

Οι τελευταίες δύο αρχιτεκτονικές έχουν εκπαιδευτεί στο σύνολο δεδομένων Cityscapes, ώστε να εκτελούν κατάτμηση εικόνας σε σκηνές δρόμων πόλεων. Το **BiSeNet v2** (Bilateral Segmentation Network) περιλαμβάνει 2 κλάδους: ο πρώτος ονομάζεται Κλάδος Λεπτομερειών (Detail Branch) και έχει ένα ευρύ κανάλι και ρηχά στρώματα με στόχο να απεικονισθούν οι χαμηλού επιπέδου λεπτομέρειες και να παραχθεί μία υψηλής ανάλυσης παρουσίαση χαρακτηριστικών και ο δεύτερος κλάδος ονομάζεται Σημασιολογικός Κλάδος (Semantic Branch), έχει ρηχά κανάλια και βαθιά στρώματα και είναι υπεύθυνος να διατηρήσει το υψηλό επίπεδο σημασιολογικού πλαισίου [41].

Το μοντέλο **ERFNet** χρησιμοποιεί 3 διαφορετικά πεδία (modules): (1) ένα παραγοντοποιημένο υπολειπόμενο δίκτυο (factorized residual network) με διεσταλμένες συνελίξεις, (2) μια μονάδα περιορισμού (downsampling) εμπνευσμένη από μια μονάδα Inception, και (3) μια μονάδα διεύρυνσης (upsampling) [42].

Ανίχνευση Αντικειμένων

Για τη διεργασία της Ανίχνευσης Αντικειμένων χρησιμοποιήθηκαν 3 από τις πιο δημοφιλείς και ευρέως χρησιμοποιούμενες αρχιτεκτονικές αυτή τη στιγμή. Έγινε προσπάθεια να συγκεντρωθούν οι πιο ελαφριές και οι αντίστοιχες πιο βαριές εκδοχές τους. Όλα τα μοντέλα έχουν εκπαιδευτεί στο σύνολο δεδομένων COCO2017. Τα μοντέλα παρουσιάζονται στον Πίνακα 6.3 ταξινομημένα με φθίνουσα σειρά ακρίβειας ως προς το δείκτη mAP (mean Average Precision). Το mAP που παρουσιάζεται είναι ο μέσος όρος όλων των μέσων όρων 10 IoU επιπέδων σε 80 κατηγορίες του περιεχόμενου ετικετών της COCO.

Για κάθε μοντέλο φαίνεται η μέθοδος κβαντοποίησης, το μέγεθος εισόδου, ο μέγιστος αριθμός αντικειμένων που μπορεί να εντοπίσει το μοντέλο σε μια εικόνα, ο δείκτης ακρίβειας mAP, το πλήθος των παραμέτρων και το μέγεθος του tflite αρχείου στον δίσκο σε MB.

Μοντέλο	Κβαντοποίηση	Μέγεθος Εισόδου	Μέγιστος Αριθμός Αντικειμένων	mAP (%)	Αριθμός Παραμέτρων	Μέγεθος (MB)
EfficientDet Lite4	INT	640x640	25	41.96	17.81 M	20.8
EfficientDet Lite1	INT	384x384		30.55	4.84 M	6.1
YOLO v5s	-	320x320	85	36.10	7.26 M	29.1
	DR			N/A		7.6
SSD MobileDet	-	320x320	100	28.90	7.08 M	28.4
	INT			28.80		7.3
SpaghettiNet Large	INT	320x320	10	28	5.72 M	6.0
SpaghettiNet Small	INT			26.30	3.45 M	3.7
SSD MobileNet v3 Large	-	320x320	100	22.60	3.24 M	13.0
SSD MobileNet v3 Small	-			15.40	1.78 M	7.2

Πίνακας 6.3: *Μοντέλα Ανίχνευσης Αντικειμένων*

Η **EfficientDet Lite** είναι η mobile-friendly εκδοχή των γνωστών EfficientDets και σχεδιάστηκε ώστε να επιτυγχάνει καλές αποδόσεις με περιορισμένους πόρους. Έρχεται σε 6 διαφορετικές εκδόσεις με την Lite0 να είναι η πιο ελαφριά και την Lite4 η πιο βαριά. Η κλασική αρχιτεκτονική ενός EfficientDet είναι ένας συνδυασμός της αρχιτεκτονικής EfficientNet με ένα αμφίδρομο δίκτυο χαρακτηριστικών τύπου πυραμίδας (bidirectional feature pyramid network - BiFPN). Το BiFPN επιτρέπει τη γρήγορη και εύκολη σύντηξη χαρακτηριστικών πολλαπλής κλίμακας. Επίσης, χρησιμοποιείται μια σύνθετη μέθοδος κλιμάκωσης που ελέγχει την ανάλυση, το βάθος και το πλάτος σε όλα τα υποδίκτυα (backbone, εξαγωγέας χαρακτηριστικών και πρόβλεψης) [43].

Τα μοντέλα **YOLO** (You Only Look Once) έχουν το πλεονέκτημα ότι απαιτούν μόνο μια διάδοση προς τα εμπρός (forward propagation) για να προβλέψουν τις πιθανότητες των κλάσεων αλλά και για να οριοθετήσουν τα πλαίσια των αντικειμένων. Αυτό τα κάνει πολύ γρήγορα και ακριβή μοντέλα. Η λειτουργία ενός τέτοιου μοντέλου βασίζεται στην ιδέα του χωρισμού της εικόνας εισόδου σε ένα πλέγμα (grid). Κάθε κελί (cell) αυτού του πλέγματος είναι υπεύθυνο να εντοπίσει το αντικείμενο του οποίο το κέντρο βρίσκεται μέσα σε αυτό και πιο συγκεκριμένα το περίγραμμα του αντικειμένου (bounding box), αλλά και την πιθανότητα το αντικείμενο να βρίσκεται μέσα σε αυτό το περίγραμμα [18].

Τα υπόλοιπα μοντέλα του πίνακα βασίζονται στην αρχιτεκτονική **SSD** (Single Shot Detector). Τα SSD μοντέλα ακολουθούν επίσης τη λογική του ενός περάσματος (single shot), όπως και τα YOLO μοντέλα. Μπορούν να διαχωριστούν σε δύο μέρη: (1) το backbone δίκτυο, που λειτουργεί για την εξαγωγή των χαρακτηριστικών και (2) την κεφαλή (SSD head), που περιλαμβάνει έναν αριθμό από συνελικτικά επίπεδα

διαφορετικής κλίμακας για τον καθορισμό των περιγραμμάτων των αντικειμένων και των κατηγοριών τους [44].

Εκτίμηση Ανθρώπινης Πόζας

Για τη διεργασία της Εκτίμησης Ανθρώπινης Πόζας ενσωματώθηκαν στην εφαρμογή τα μοντέλα που παρουσιάζονται στον Πίνακα 6.4. Για κάθε μοντέλο παρουσιάζεται η μέθοδος κβαντοποίησης, το μέγεθος εισόδου, το σύνολο ή τα σύνολα δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση μαζί με το πλήθος των σημείων - αρθρώσεων (keypoints) που μπορούν να ανιχνευτούν σε παρένθεση, ο δείκτης ακρίβειας, το πλήθος των παραμέτρων και το μέγεθος του tflite αρχείου στον δίσκο σε MB. Για την αξιολόγηση της ακρίβειας της αρχιτεκτονικής MoveNet χρησιμοποιείται ο δείκτης Keypoint mAP, για την BlazePose ο δείκτης PCK@0.2 σε τρία διαφορετικά σύνολα επικύρωσης (Yoga, Dance και HIIT), ενώ για τις υπόλοιπες ο δείκτης PCKh@0.5. Επίσης, πρέπει να αναφερθεί ότι όλα τα μοντέλα μπορούν να ανιχνεύουν μέχρι μια ανθρώπινη πόζα στην εικόνα εισόδου, εκτός από το μοντέλο Movenet Multipose Lightning, το οποίο μπορεί να εντοπίσει μέχρι 6.

Μοντέλο	**Κβαντοποίηση**	**Μέγεθος Εισόδου**	**Σύνολα Δεδομένων**	**Ακρίβεια (%)**	**Αριθμός Παραμέτρων**	**Μέγεθος (MB)**
MoveNet Singlepose Thunder	-	256x256	COCO 2017 Keypoint / Active Dataset (17)	78.7	6.23 M	25.0
MoveNet Singlepose Lightning	-	192x192		67.4	2.32 M	9.4
MoveNet Multipose Lightning	-	192x192		60.2	4.72 M	19.1
BlazePose Lite	-	256x256	*Private* (33)	90.2/92.5 /93.5	1.44 M	5.8
	DR			N/A		1.8
	FULL			N/A		1.9
EfficientPoseII Lite	-	368x368	MPII (16)	87.1	1.41 M	5.7
	DR			N/A		1.9
	INT			N/A		2.1
	FP16			N/A		2.9
EfficientPoseRT Lite	-	224x224		80.6	0.39 M	1.5
	DR			N/A		0.56
	INT			N/A		0.62
	FP16			N/A		0.79
CPM	-	192x192	AI Challenger (14)	93.78	0.56 M	2.4
	DR			N/A		0.96
Hourglass	-	192x192		91.81	0.4 M	1.7
	DR			N/A		0.87

Πίνακας 6.4: *Μοντέλα Εκτίμησης Ανθρώπινης Πόζας*

Η πρώτη αρχιτεκτονική που χρησιμοποιήθηκε είναι η **MoveNet**, η οποία προβλέπει απευθείας τις θέσεις των αρθρώσεων ενός (Singlepose) η περισσότερων

(Multipose) ανθρώπων σε μια εικόνα. Για τη Singlepose εκδοχή έχουμε δύο υπο-εκδοχές: την ελαφριά Lightning με πολλαπλασιαστή βάθους 1.0 και την πιο βαριά Thunder με πολλαπλασιαστή βάθους 1.75. Η Multipose εκδοχή αυτή τη στιγμή κυκλοφορεί μόνο στην Lightning εκδοχή με πολλαπλασιαστή βάθους 1.5. Τα μοντέλα MoveNet χρησιμοποιούν ένα δίκτυο MobileNet v2 για την εξαγωγή των χαρακτηριστικών και ένα Δίκτυο Χαρακτηριστικών τύπου Πυραμίδας (Feature Pyramid Network) για την αποκωδικοποίηση. Στο τέλος, υπάρχει ένα δίκτυο CenterNet για την ταξινόμηση [45], [46].

Επόμενη αρχιτεκτονική είναι το **BlazePose** της MediaPipe στην Lite εκδοχή του. Αποτελείται από δύο στάδια, έναν ανιχνευτή κι έναν ιχνηλάτη. Ο ανιχνευτής εντοπίζει αρχικά την περιοχή ενδιαφέροντος όπου βρίσκεται ο άνθρωπος. Στη συνέχεια, ο ιχνηλάτης προβλέπει τα σημεία της πόζας χρησιμοποιώντας σαν είσοδο μόνο την περιοχή ενδιαφέροντος που καθόρισε ο ανιχνευτής [47].

Τα **EfficientPose** είναι μια οικογένεια μοντέλων των οποίος στόχος είναι οι εφαρμογή σε προβλήματα πραγματικού χρόνου. Χρησιμοποιούν την αρχιτεκτονική των γνωστών EfficientNets για εξαγωγή χαμηλού και υψηλού επιπέδου χαρακτηριστικών, τα οποία στη συνέχεια συνενώνονται. Ο εντοπισμός των σημείων των αρθρώσεων γίνεται μέσω μιας επαναληπτικής διαδικασίας ανίχνευσης που περιλαμβάνει δίκτυα Mobile DenseNets, υπολειπόμενες συνδέσεις (residual connections) και χωρίζεται σε δύο φάσεις: στην πρώτη αναμένεται η συνολική πόζα του ατόμου (εκτίμηση σκελετού), ενώ στη δεύτερη δημιουργούνται οι χάρτες θερμότητας (heatmaps) για τα σημεία ενδιαφέροντος [48].

Με βάση μια εντελώς διαφορετική προσέγγιση λειτουργούν τα μοντέλα **CPM (Convolutional Pose Machines)**. Λειτουργούν ακολουθιακά και παράγουν όλο και πιο ακριβείς προβλέψεις στο τέλος κάθε σταδίου. Κάθε στάδιο παράγει $P + 1$ χάρτες θερμότητας, όπου P ο αριθμός των αρθρώσεων, οι οποίοι δίνονται στο επόμενο στάδιο μαζί με νέα χαρακτηριστικά της αρχικής εικόνας. Με αυτό τον τρόπο, οι μηχανές καταφέρνουν και βελτιώνουν τις προβλέψεις σε κάθε στάδιο. Οι μηχανές πόζας υπήρχαν πριν ακόμα η Βαθιά Μάθηση εκτοξευθεί σε δημοτικότητα και η απόδοσή τους αυξήθηκε όταν οι ταξινομητές που χρησιμοποιούσαν αντικαταστάθηκαν από συνελικτικά επίπεδα.

Η ίδια περίπου τακτική ακολουθείται και στην αρχιτεκτονική των μοντέλων **Hourglass**. Πρόκειται για μοντέλα κωδικοποιητή - αποκωδικοποιητή, όπου ο κωδικοποιητής παράγει έναν πίνακα χαρακτηριστικών και ο αποκωδικοποιητής τον συνδυάζει με προηγούμενα επίπεδα -του κωδικοποιητή- που έχουν καλύτερη χωρική κατανόηση. Τα blocks που εκτελούν αυτές τις συνδέσεις είναι ένας τύπος υπολειπόμενων μπλοκ που ονομάζονται μπλοκ συμφόρησης (bottleneck blocks) [20].

6.2.1 Metadata

Πριν την ενσωμάτωση του κάθε μοντέλου στην εφαρμογή πραγματοποιείται μια προ επεξεργασία και εισάγονται μεταδεδομένα (metadata) που αφορούν χρήσιμες πληροφορίες σχετικά με το εκάστοτε μοντέλο και τη διεργασία για την οποία εχει εκπαιδευτεί. Τα metadata για κάθε μοντέλο βρίσκονται σε ένα text αρχείο το οποίο ενσωματώνεται στο tflite μοντέλο ως σχετιζόμενο αρχείο (associated files) με την python βιβλιοθήκη tflite-support. Σε κάθε metadata αρχείο βρίσκονται ανά γραμμή ζευγάρια παραμέτρων σε μορφή κλειδιού:τιμής (key:value). Οι παράμετροι χωρίζονται σε 3 κατηγορίες:

1. **Γενικές (General).** Αφορούν πιο γενικές πληροφορίες για κάθε μοντέλο. Σε αυτήν την κατηγορία ανήκουν το όνομα (name) του μοντέλου και το πρόβλημα (task) Βαθιάς Μάθησης που αντιμετωπίζει.
2. **Εισόδου (Input).** Αφορούν πληροφορίες σχετικές με την είσοδο του μοντέλου:
   - Διαστάσεις: inputHeight, inputWidth και inputNumChannels.
   - Κβαντοποίηση: inputQuantization, με δυνατές τιμές FP32, INT64, INT8 και UINT8.
   - Αν η είσοδος δεν είναι κβαντισμένη, τότε απαιτούνται επιπρόσθετες παράμετροι σχετικά με την κλιμάκωση και την κανονικοποίηση των εισόδων. Για την κλιμάκωση μας ενδιαφέρει το εύρος τιμών (inputMin και inputMax), ενώ για την κανονικοποίηση ο μέσος όρος (inputNormMean) και η τυπική απόκλιση (inputNormStd). Ο τύπος που εφαρμόζεται είναι ο ακόλουθος:

$$normalized\_input = (input - mean)/std$$

3. **Εξόδου (Output)**. Αφορούν κυρίως την κβαντοποίηση στην έξοδο και τη διεργασία.
   - Κβαντοποίηση: outputQuantization, με δυνατές τιμές FP32, INT64, INT8 και UINT8.
   - Αν η έξοδος είναι κβαντισμένη, απαιτούνται επιπρόσθετες παράμετροι για την μετατροπή των κβαντισμένων ακέραιων αριθμών σε κινητής υποδιαστολής. Αυτές οι παράμετροι είναι η κλίμακα (outputQuantScale) και το σημείο μηδέν (outputQuantZeroPoint). Η μετατροπή γίνεται με βάση τον τύπο:

$$f = (q - zeroPoint) * scale$$

   - Ανάλογα με τη διεργασία ορίζονται στην έξοδο του μοντέλου κάποιες επιπρόσθετες παράμετροι. Αν η διεργασία είναι η Κατάτμηση Εικόνας, τότε ορίζονται ο αριθμός των κλάσεων (numClasses) και οι διαστάσεις των χαρτών θερμότητας (outputHeight και outputWidth). Αν η διεργασία είναι η Εκτίμηση Ανθρώπινης Πόζας, τότε αρχικά ορίζονται ο μέγιστος αριθμός των ποζών που μπορούν να εντοπιστούν σε μία εικόνα (maxNumDetections), το πλήθος των σημείων - αρθρώσεων σε κάθε πόζα (numBodyParts) και τα ζευγάρια σημείων που ενώνονται μεταξύ τους για να σχηματιστεί ο σκελετός (pairs). Επιπλέον, ορίζεται η μέθοδος που χρησιμοποιεί το μοντέλο, method, με δυνατές τιμές points και heatmaps. Στην περίπτωση που το μοντέλο εξάγει heatmaps, τότε απαιτούνται και οι διαστάσεις εξόδου (outputHeight και outputWidth).

Εκτός από τα text αρχεία που περιέχουν τα μεταδεδομένα για κάθε μοντέλο, υπάρχουν και επιπρόσθετα text αρχεία που περιέχουν ετικέτες και χρώματα ανάλογα με το σύνολο εκπαίδευσης. Για παράδειγμα, στα μοντέλα DeepLab που έχουν εκπαιδευτεί στο σύνολο δεδομένων Pascal VOC2017 προστίθενται εκτός από το metadata file και τα δύο αρχεία voc17_labels.txt και voc17_colors.txt που περιέχουν τις 21 κλάσεις εξόδου και 21 διαφορετικά χρώματα RGB για κάθε κλάση αντίστοιχα.

Κεφάλαιο 7: Αξιολόγηση

Οι μετρικές χρόνου που παρουσιάζονται σε αυτό το Κεφάλαιο σε συνδυασμό με τις πληροφορίες του προηγούμενου Κεφαλαίου με τα τεχνικά χαρακτηριστικά των συσκευών και των μοντέλων βοηθούν στην εξαγωγή σωστών συμπερασμάτων.

7.1 Μετρικές

Οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των μοντέλων είναι οι ακόλουθες:

1. Minimum, ο ελάχιστος χρόνος που καταγράφηκε στο σύνολο των μετρήσεων.
2. Maximum, ο μέγιστος χρόνος που καταγράφηκε στο σύνολο των μετρήσεων.
3. Mean, ο μέσος όρος όλων των μετρήσεων.
4. Median, η τιμή που βρίσκεται ακριβώς στη μέση των ταξινομημένων μετρήσεων.
5. 90^{th} percentile, η τιμή κάτω από την οποία βρίσκεται το 90% των μετρήσεων. Με άλλα λόγια το 90% των μετρήσεων παρουσίασαν χρόνο μικρότερο ή ίσο από αυτή την τιμή.

Από τις παραπάνω, η πιο κατάλληλη μετρική για την μέτρηση καθυστέρησης είναι η 90^{th} percentile. Αυτό ισχύει διότι μετρικές όπως η mean και η median μπορεί να είναι παραπλανητικές αφού δεν λαμβάνουν και τόσο υπόψη τη χειρότερη περίπτωση.

7.2 Μετρήσεις

Περιβάλλον

Για τις μετρήσεις του χρόνου εκτέλεσης χρησιμοποιήθηκε η μέθοδος currentTimeMillis() της κλάσης System της Java, η οποία επιστρέφει την τρέχουσα ώρα σε milliseconds (ms). Συγκεκριμένα, απομονώνοντας τα επιθυμητά μπλοκ κώδικα, μπορούμε να υπολογίσουμε τον χρόνο που έχει παρέλθει για την εκτέλεσή τους αφαιρώντας την ώρα που καταγράφεται πριν εισέλθουμε στο μπλοκ από την ώρα που καταγράφεται όταν εξέλθουμε από το μπλοκ.

Επίσης, πριν ξεκινήσουν οι μετρήσεις έπρεπε να καθοριστούν ο αριθμός των warm-up runs, δηλαδή οι επαναλήψεις εκτέλεσης μέχρι η συσκευή να προσαρμοστεί στις λειτουργίες της εφαρμογής. Για αυτήν την εργασία ο αριθμός των warm-up runs καθορίστηκε στις 16 επαναλήψεις. Τέλος, επιλέχθηκαν 200 επαναλήψεις εκτέλεσης μετά το πέρας των 16 warm-up runs ώστε να προκύψουν οι τελικές μετρήσεις.

Στην παρούσα εργασία μετρήθηκε για κάθε μοντέλο και διαμόρφωση ο χρόνος εκτέλεσης της συμπερασματολογίας (inference latency), αλλά και ο συνολικός χρόνος εκτέλεσης ενός κύκλου (total latency), δηλαδή από τη στιγμή που γίνεται διαθέσιμη η εικόνα από την κάμερα μέχρι να εμφανιστεί στην οθόνη του χρήστη το αποτέλεσμα. Ο συνολικός χρόνος μπορεί να χωριστεί σε τρία στάδια: (1) την προετοιμασία της εικόνας εισόδου (preprocessing), (2) τη συμπερασματολογία και (3) την επεξεργασία της εξόδου του μοντέλου για την παρουσίαση του αποτελέσματος στον χρήστη (postprocessing).

Για κάθε μοντέλο πραγματοποιήθηκαν μετρήσεις για όλες τις διαφορετικές τιμές δύο βασικών παραμέτρων:

1. Ο επεξεργαστής εκτέλεσης του μοντέλου, με πιθανές τιμές CPU, GPU και NNAPI. Το NNAPI (Neural Networks API) είναι στην πραγματικότητα ένας «εκπρόσωπος» που είναι διαθέσιμος στις εκδόσεις του Android από την 8.1 και πάνω και προσφέρει επιτάχυνση στα μοντέλα χρησιμοποιώντας τη GPU, τον DSP ή την NPU, ανάλογα με το διαθέσιμο υλικό στην εκάστοτε συσκευή.
2. Το πλήθος των παράλληλων νημάτων στη CPU, με τιμές 1, 2, 4 ή 8.

Στο Παράρτημα Α παρουσιάζονται αναλυτικοί πίνακες μετρήσεων με τις διαμορφώσεις (configurations) στις οποίες σημειώθηκαν οι καλύτεροι και οι αμέσως επόμενοι καλύτεροι χρόνοι για κάθε μοντέλο που βοηθούν στην περαιτέρω κατανόηση της συμπεριφοράς των μοντέλων και σχολιάζονται στη συνέχεια για κάθε διεργασία.

Στους παρακάτω Πίνακες 7.1, 7.2 και 7.3 έχουν συγκεντρωθεί οι καλύτεροι χρόνοι για κάθε διεργασία με βάση τη μετρική 90th percentile. Οι μετρικές mean και median εξάγουν παρόμοια συμπεράσματα. Τις δύο μετρήσεις συνοδεύουν η διαμόρφωση -επεξεργαστής και αριθμός νημάτων σε παρένθεση- στην οποία παρατηρήθηκαν οι καλύτεροι χρόνοι για κάθε μοντέλο, καθώς και η διαφορά του χρόνου συμπερασματολογίας από τον συνολικό, που αντιστοιχεί στον χρόνο προεπεξεργασίας - προετοιμασίας της εικόνας εισόδου συν τον χρόνο επεξεργασίας του αποτελέσματος.

Κατάτμηση Εικόνας

Μοντέλο	mIoU	Κινητό				Tablet			
		Επεξεργαστής	Inf	Total	Diff	Επεξεργαστής	Inf	Total	Diff
Bilinear MUNet	96	GPU	177	291	114	GPU / NNAPI	17	62	45
Bilinear MUNet (DR)	96	CPU (4)	235	306	71	GPU / NNAPI	18	63	45
PrismaNet	96	GPU	127	391	264	GPU	15	144	129
PrismaNet (DR)	96	GPU	352	583	231	GPU	15	145	130
PrismaNet (FULL)	96	CPU (2)	355	564	209	GPU / NNAPI	15	141	126
DeepLab v3 Xception 65	87.8	-	-	-	-	-	-	-	-
DeepLab v3 Xception 65 (INT)		-	-	-	-	-	-	-	-
DeepLab v3 MobileNet v2	80.25	CPU (8)	820	946	126	GPU	56	124	68
DeepLab v3 MobileNet v2 (INT)		CPU (8)	774	901	127	CPU (2)	178	224	46
UNet Insudtrial	75.5	-	-	-	-	GPU	97	418	321
UNet Insudtrial (DR)		GPU	654	1462	808	GPU	95	418	323
BiSeNet v2	72.6	CPU (4)	234	486	252	NNAPI	53	138	85
ERFNet	72.1	CPU (8)	626	1222	596	GPU	41	280	239
ERFNet (FULL)		CPU (8)	662	1173	511	GPU	44	272	228

Πίνακας 7.1: *Inference και Total Latency για την Κατάτμηση Εικόνας*

Αρχικά, πρέπει να αναφερθεί ότι οι μετρήσεις που λείπουν για κάποια μοντέλα είναι μετρήσεις που ξεπερνούσαν τα 2 δευτερόλεπτα στον χρόνο συμπερασματολογίας. Επίσης, το κβαντισμένο μοντέλο UNet Insudtrial στο κινητό κατάφερε να τρέξει μόνο

στη GPU. Έτσι, τα συγκεκριμένα μοντέλα παρουσιάζονται σε αυτόν τον πίνακα για να δηλωθούν ακατάλληλα για real-time εφαρμογές.

Όπως παρατηρείται, στο κινητό οι καλύτερες μετρήσεις εντοπίστηκαν κυρίως κατά την χρήση της CPU με πολλαπλά παράλληλα νήματα και σε ορισμένες περιπτώσεις με χρήση της GPU. Η απουσία του NNAPI προκύπτει από την έλλειψη NPU. Σε αυτό οφείλεται και το γεγονός ότι σε πολλές περιπτώσεις η χρήση κβαντισμένων εκδοχών -στο Bilinear MUNet, στο PrismaNet και στο ERFNet- οδηγεί σε μεγαλύτερους χρόνους.

Στο tablet παρατηρείται ότι η GPU κυριαρχεί και με βάση τους Πίνακες του Παραρτήματος Α.1, οι διαφορές από τους αντίστοιχους χρόνους με CPU είναι σημαντικές, ενώ όταν η δεύτερη καλύτερη επιλογή είναι το NNAPI, αυτές οι διαφορές είναι μικρότερες. Με την πρόσφατη δυνατότητα των GPUs να εκτελούν αποδοτικά και κβαντισμένα μοντέλα, κάτι που παλαιότερα δεν ήταν εφικτό, φαίνεται ότι για μοντέλα Κατάτμησης Εικόνας η χρήση νέων GPUs πρέπει να προτιμάται.

Από τον ίδιο πίνακα παρατηρείται και η διαφορά στις CPUs των δύο συσκευών, αφού ο χρόνος που απαιτείται για τα υπόλοιπα στάδια (preprocessing και postprocessing) στο tablet είναι ο κατά μέσο όρο ο μισός από ότι στο κινητό.

Ανίχνευση Αντικειμένων

Μοντέλο	mAP	Κινητό				Tablet			
		Επεξεργαστής	Inf	Total	Diff	Επεξεργαστής	Inf	Total	Diff
EfficientDet Lite4 (INT)	41.96	GPU	1429	1555	126	NNAPI	177	204	27
EfficientDet Lite1 (INT)	30.55	GPU	290	366	76	NNAPI	49	62	13
YOLOv5s	36.1	GPU	261	337	76	GPU / NNAPI	66	89	23
YOLOv5s (DR)		GPU	261	336	75	GPU	65	92	27
SSD MobileDet	28.9	GPU	197	276	79	GPU / NNAPI	30	56	26
SSD MobileDet (INT)	28.8	GPU	211	278	67	NNAPI	16	26	10
SpaghettiNet Large (INT)	28	GPU	144	210	66	NNAPI	17	28	11
SpaghettiNet Small (INT)	26.3	GPU	108	172	64	NNAPI	14	25	11
SSD MobileNetv3 Large	22.6	GPU	92	154	62	GPU	19	42	23
SSD MobileNetv3 Small	15.4	CPU (4)	51	115	64	GPU	16	40	24

Πίνακας 7.2: *Inference και Total Latency για την Ανίχνευση Αντικειμένων*

Από τον παραπάνω πίνακα βλέπουμε ότι στο κινητό η GPU φέρνει τα καλύτερα αποτελέσματα. Ακόμη και για το SSD MobileNetv3 Small, οι διαφορές στο χρόνο μεταξύ χρήσης GPU και CPU είναι μικρότερες από 3 ms.

Αυτό που αλλάζει στο tablet είναι ότι για τα μοντέλα με integer κβαντοποίηση το NNAPI δίνει τις καλύτερες μετρικές με την GPU σε αυτές τις περιπτώσεις να έρχεται δεύτερη με μικρές διαφορές. Στην πραγματικότητα, για τα μοντέλα SSD MobileNetv3 Large και SSD MobileNetv3 Small η χρήση του NNAPI δεν ήταν εφικτή λόγω ενδεχομένως λειτουργιών (operations) που δεν υποστηρίζονται.

Αν δεν είχαμε στη διάθεσή μας τις μετρικές από τις μετρήσεις στο κινητό, το μοντέλο EfficientDet Lite4 (INT) στο tablet φαίνεται ότι έχει σχετικά καλή απόδοση και η καθυστέρησή του ίσως είναι ανεκτή για κάποιες εφαρμογές. Ωστόσο, η

καθυστέρηση του ίδιου μοντέλου σε παλιότερες συσκευές φτάνει τα 1.5 δευτερόλεπτα συνολικά, και αυτό είναι απαγορευτικό για εφαρμογές πραγματικού χρόνου.

Εκτίμηση Ανθρώπινης Πόζας

Μοντέλο	Ακρίβεια	Κινητό				Tablet			
		Επεξεργαστής	Inf	Total	Diff	Επεξεργαστής	Inf	Total	Diff
Movenet Singlepose Thunder	78.7	GPU	121	195	74	GPU	17	45	18
Movenet Singlepose Lightning	67.4	GPU	54	115	61	GPU / NNAPI	10	29	19
Movenet Multipose Lightning	60.2	CPU (4)	112	160	48	GPU	30	50	20
Blazepose Lite	45.0	CPU (4)	100	171	71	GPU	45	70	25
Blazepose Lite (DR)		GPU	102	177	75	GPU / NNAPI	46	61	15
Blazepose Lite (FULL)		CPU (4)	117	170	53	NNAPI	25	37	12
EfficientposeII Lite	87.1	GPU	735	870	135	GPU / NNAPI	196	229	33
EfficientposeII Lite (DR)		GPU	770	920	150	GPU	194	245	51
EfficientposeII Lite (INT)		GPU	789	925	136	GPU	196	251	55
EfficientposeII Lite (FP16)		GPU	735	868	133	GPU	195	246	51
EfficientposeRT Lite	80.6	GPU	264	334	70	NNAPI	70	87	17
EfficientposeRT Lite (DR)		GPU	264	337	73	GPU / NNAPI	93	111	18
EfficientposeRT Lite (INT)		GPU	264	337	73	GPU	92	118	26
EfficientposeRT Lite (FP16)		CPU (4)	249	325	76	GPU	93	120	27
CPM	93.78	GPU	120	170	50	NNAPI	25	35	10
CPM (DR)		GPU	120	170	50	GPU	36	54	18
Hourglass	91.81	CPU (4)	59	108	49	GPU / NNAPI	16	28	12
Hourglass (DR)		GPU	73	133	60	GPU	16	37	21

Πίνακας 7.3: *Inference και Total Latency για την Εκτίμηση Ανθρώπινης Πόζας*

Από τον παραπάνω Πίνακα για τα μοντέλα Εκτίμησης Πόζας προκύπτουν παρόμοια συμπεράσματα. Στο κινητό η GPU και η CPU με 4 νήματα έχουν παρόμοια αποτελέσματα, ενώ στο tablet η GPU με το NNAPI.

Μια ενδιαφέρουσα παρατήρηση που αφορά και τις τρεις διεργασίες είναι ότι σε όλα τα μοντέλα με dynamic range κβαντοποίηση η GPU φέρνει τα καλύτερα αποτελέσματα.

Κεφάλαιο 8: Επίλογος

Στον επίλογο της εργασίας παρουσιάζονται τα τελικά συμπεράσματα και οι μελλοντικές επεκτάσεις της.

8.1 Συμπεράσματα

Σε αυτή την Ενότητα γίνεται αρχικά μια προσπάθεια να εξαχθεί ένα γενικότερο συμπέρασμα με βάση τα αποτελέσματα του προηγούμενου Κεφαλαίου. Ωστόσο, από τα αποτελέσματα για κάθε διεργασία είναι προφανές ότι δεν υπάρχει ένα μοντέλο που να ξεχωρίζει, δηλαδή ένα μοντέλο που να υπερτερεί από όλες τις απόψεις ταυτόχρονα.

Για παράδειγμα, στον Πίνακα 8.1 παρακάτω συνοψίζονται τρία μοντέλα Ανίχνευσης Αντικειμένων -εκπαιδευμένα στο ίδιο σύνολο δεδομένων- που όμως υπερτερούν σε ξεχωριστούς τομείς.

Μοντέλο	**mAP (%)**	**Μέγεθος (MB)**	**90th percentile Total latency**	
			Κινητό	**Tablet**
EfficientDet Lite4	41.96	20.359	1555	204
SSD MobileNetv3 Small	15.4	7.123	118	40
SpaghettiNet Small	26.3	3.585	172	25

Πίνακας 8.1: *Βέλτιστα μοντέλα για Ανίχνευση Αντικειμένων*

Το πρώτο μοντέλο, αν και θεωρείται το state-of-the-art στο συγκεκριμένο task από άποψη ακρίβειας, εμφανίζει καθυστέρηση που είναι απαγορευτική για εφαρμογές πραγματικού χρόνου, όπως αναφέρθηκε και νωρίτερα. Από την άλλη, τα δύο επόμενα μοντέλα μπορεί να ικανοποιούν τους περιορισμούς σχετικά με την καθυστέρηση και έχουν επίσης μικρότερο μέγεθος, όμως υστερούν στην ακρίβεια των προβλέψεών τους.

Ακόμη, από τα αποτελέσματα προκύπτει ότι η χρήση του NNAPI δεν έχει πάντα τα καλύτερα αποτελέσματα στα κβαντισμένα μοντέλα για τα οποία υπόσχεται υψηλή επιτάχυνση. Αυτό συμβαίνει γιατί σε κάθε κινητή συσκευή το NNAPI έρχεται αντιμέτωπο με το διαφορετικό υλικό (hardware), αλλά και τα διαφορετικά σχήματα συνεργασίας μεταξύ των διαφόρων υπολογιστικών μονάδες στο κεντρικό τσιπ. Σε αυτό οφείλεται και το γεγονός ότι οι λειτουργίες που μπορεί να υποστηρίξει το NNAPI είναι συγκεκριμένες και υποσύνολο των λειτουργιών που μπορεί να εκτελέσει μια CPU ή μια GPU.

Κάτι άλλο που πρέπει να ληφθεί υπόψη είναι η επίδραση της ίδιας της εφαρμογής στην απόδοση των επεξεργαστών. Για παράδειγμα, στη συγκεκριμένη εργασία, η οποία είχε να κάνει με μια εφαρμογή κάμερας, εκτός από την εκτέλεση της συμπερασματολογίας, η GPU της συσκευής είχε τον επιπρόσθετο φόρτο να διατηρεί την προεπισκόπηση της κάμερας στο επιθυμητό frame rate. Η ταυτόχρονη χρήση της υπολογιστικής μονάδας από πολλαπλές διεργασίες μειώνει την απόδοσή της, οπότε σε κάποια διαφορετική εφαρμογή, τα αποτελέσματα με χρήση GPU ίσως ήταν ακόμη πιο ικανοποιητικά.

Με βάση όλα τα παραπάνω, είναι εμφανές ότι η στατική επιλογή του μοντέλου, της υπολογιστικής μονάδας και των γενικότερων παραμέτρων της εφαρμογής είναι μη αποδεκτή. Αυτό ισχύει επειδή το περιβάλλον της εφαρμογής είναι δυναμικό και άρα

θα πρέπει η εν λόγω απόφαση να παίρνεται επίσης δυναμικά ανάλογα με τους διαθέσιμους πόρους και τους επιθυμητούς στόχους κάθε στιγμή.

8.2 Μελλοντικές Επεκτάσεις

Όσα αναπτύχθηκαν στα πλαίσια της διπλωματικής εργασίας επιδέχονται βελτιώσεις και επεκτάσεις με προσθήκες ή παραλλαγές στα βασικά δομικά στοιχεία τους.

Άλλες Εφαρμογές

Όπως αναφέρθηκε και στην προηγούμενη Ενότητα, τα αποτελέσματα επηρεάζονται από την εκάστοτε εφαρμογή, όπως για παράδειγμα στην παρούσα εφαρμογή, όπου η ταυτόχρονη λειτουργία της κάμερας και του διερμηνέα για την εκτέλεση της συμπερασματολογίας φορτώνουν παραπάνω τον επιλεγμένο επεξεργαστή. Εκτός από αυτό, διαφορετικές εφαρμογές έχουν και διαφορετικούς στόχους επίδοσης. Για παράδειγμα, σε εφαρμογές μη πραγματικού χρόνου, όπως ένα gallery app, η είσοδος στο μοντέλο μπορεί να περιλαμβάνει περισσότερες από μια εικόνες και ίσως αυτό είναι και πιο αποδοτικό από άποψη καθυστέρησης. Οπότε, είναι αναγκαίο να μελετηθούν και άλλα σενάρια εφαρμογών.

Μηχανισμός Επιλογής Μοντέλου και Προσαρμογής Παραμέτρων

Από το γενικό συμπέρασμα της παρούσας εργασίας, η στατική επιλογή μοντέλου και παραμέτρων κρίνεται μη αποδοτική. Αυτό ισχύει επειδή είναι αδύνατο ένα μόνο μοντέλο να μπορεί να εκτελεστεί σε κάθε κινητή συσκευή και να μπορεί να καλύπτει τις απαιτήσεις απόδοσης κάθε εφαρμογής. Μια λύση σε αυτό είναι η ενσωμάτωση ενός πλήθους από μοντέλα με διαφορετικά χαρακτηριστικά απόδοσης και η ύπαρξη μιας μονάδας (module), η οποία θα προσαρμόζει δυναμικά τις παραμέτρους του συστήματος με βάση αλλαγές στην κατάσταση του περιβάλλοντος εκτέλεσης, όπως για παράδειγμα μια αύξηση στον φόρτο ή μείωση στη συχνότητα του επεξεργαστή.

Υβριδικά Συστήματα Κατανεμημένης Συμπερασματολογίας

Ο μηχανισμός αυτόματης προσαρμογής που αναφέρθηκε παραπάνω μπορεί εύκολα να επεκταθεί με την προσθήκη της δυνατότητας αποστολής δειγμάτων εισόδου προς συμπερασματολογία σε έναν απομακρυσμένο εξυπηρετητή (server). Έτσι το συνολικό σύστημα γίνεται κατανεμημένο. Έχοντας περισσοτέρους πόρους, ο εξυπηρετητής μπορεί να χρησιμοποιεί πιο απαιτητικά μοντέλα που έχουν και υψηλότερη ακρίβεια. Ένα απλό αλλά ταυτόχρονα χαρακτηριστικό παράδειγμα τέτοιου συστήματος είναι τα Ζεύγη Νευρωνικών Δικτύων, στα οποία χρησιμοποιούνται ένα «ελαφρύ» μοντέλο στην κινητή συσκευή και ένα πιο «βαρύ» στον εξυπηρετητή.

Παράρτημα Α: Αναλυτικοί Πίνακες Μετρήσεων

Στους παρακάτω πίνακες παρουσιάζονται αναλυτικές μετρήσεις για κάθε διεργασία και κάθε συσκευή με τους καλύτερους και τους αμέσως επόμενους καλύτερους χρόνους για κάθε μοντέλο. Με πράσινο χρώμα έχουν σημειωθεί οι καλύτεροι χρόνοι για πιο εύκολη ανάγνωση κι επίσης κάτω από τους δύο καλύτερους χρόνους έχει υπολογιστεί η διαφορά τους. Υπενθυμίζεται ότι το κινητό είναι η συσκευή Samsung Galaxy A20e, ενώ το tablet η συσκευή Samsung Galaxy Tab S7.

Α.1 Κατάτμηση Εικόνας

Bilinear MUNet

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	157	264	185	299	169.2	282.4	171	283	177	291
	CPU (4)	201	267	251	316	216.1	286.2	215	285	226	298
	Διαφορά					46.9	3.8	44	2	49	7
Tablet	GPU	15	51	24	84	16.3	67	16	67	17	72
	NNAPI	30	54	41	67	34.3	59.3	34	59	37	62
	Διαφορά					18	7.7	18	8	20	10

Πίνακας Α.1: *Bilinear MUNet*

Bilinear MUNet (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	210	270	246	318	226.2	292.6	227	295	235	306
	CPU (2)	244	302	279	354	255.9	319	253	315	272	341
	Διαφορά					29.7	26.4	26	20	37	35
Tablet	GPU	15	53	21	82	16.5	66.8	16	66	18	73
	NNAPI	31	53	40	68	34.4	59.6	34	59	37	63
	Διαφορά					17.9	7.2	18	7	19	10

Πίνακας Α.2: *Bilinear MUNet (DR)*

PrismaNet

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	96	329	135	403	113.4	375.1	110	374	127	391
	CPU (4)	180	393	235	446	199.6	414.4	199	414	211	426
	Διαφορά					86.2	39.3	89	40	84	35
Tablet	GPU	13	97	18	153	14.3	135.7	14	137	15	144
	NNAPI	74	143	88	163	77.2	150.2	77	150	80	154
	Διαφορά					62.9	14.5	63	13	65	10

Πίνακας Α.3: *PrismaNet*

PrismaNet (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	314	543	364	607	334	563.8	331	561	352	583
	CPU (2)	433	655	474	724	451.4	698.7	451	698	460	708
	Διαφορά					117.4	134.9	120	137	108	125
Tablet	GPU	13	118	20	153	14.5	136.2	14	137	15	145
	CPU2	76	151	109	190	90.9	171.6	92	172	100	180
	Διαφορά					76.4	35.4	78	35	85	35

Πίνακας Α.4: *PrismaNet (DR)*

PrismaNet (FULL)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (2)	344	543	370	582	351.7	557.3	351	557	355	564
	CPU (4)	343	676	466	801	371.1	719.2	369	719	384	736
	Διαφορά					19.4	161.9	18	162	29	172
Tablet	GPU	14	117	22	152	14.8	137.2	15	138	15	144
	NNAPI	54	132	60	147	56.4	138.2	56	138	58	141
	Διαφορά					41.6	10	41	0	43	3

Πίνακας Α.5: *PrismaNet (FULL)*

DeepLabv3 MobileNetv2

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (8)	682	802	890	1014	776.4	903.2	775	903	820	946
	CPU (4)	855	982	1012	1139	887.1	1014.7	884	1012	906	1033
	Διαφορά					110.7	111.5	109	109	86	87
Tablet	GPU	52	109	57	132	54.4	119.5	54	120	56	124
	CPU (4)	172	213	209	262	185.5	231.6	185	231	195	242
	Διαφορά					131.1	112.1	131	111	139	118

Πίνακας Α.6: *DeepLabv3 MobileNetv2*

DeepLabv3 MobileNetv2 (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (8)	710	838	829	960	754.5	879.9	755	880	774	901
	CPU (4)	857	982	950	1067	882.6	1009.2	880	1008	899	1026
	Διαφορά					128.1	129.3	125	128	125	125
Tablet	CPU (2)	170	208	187	229	175.5	219.6	175	220	178	224
	CPU (4)	146	195	219	270	172.2	221.1	171	220	183	233
	Διαφορά					3.3	1.5	4	0	5	9

Πίνακας Α.7: *DeepLabv3 MobileNetv2 (DR)*

UNet Industrial

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	-	-	-	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-
	Διαφορά					-	-	-	-	-	-
Tablet	GPU	94	394	99	431	96	411.6	96	412	97	418
	NNAPI	519	786	536	819	528.8	799.3	529	799	531	802
	Διαφορά					432.8	387.7	433	387	434	384

Πίνακας Α.8: *UNet Industrial*

UNet Industrial (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	610	1356	660	1480	633.5	1436	626	1437	654	1462
	-	-	-	-	-	-	-	-	-	-	-
	Διαφορά					-	-	-	-	-	-
Tablet	GPU	92	380	98	427	94.3	410.9	94	411	95	418
	CPU (4)	344	615	1494	2795	379.3	673.6	371	663	386	675
	Διαφορά					285	262.7	277	252	291	257

Πίνακας Α.9: *UNet Industrial (DR)*

BiseNetv2

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	210	421	249	494	227.8	474.7	228	477	234	486
	CPU (8)	192	437	375	629	237.8	487.3	233	483	273	522
	Διαφορά					10	12.6	5	6	39	36
Tablet	NNAPI	45	126	60	145	50	133.6	50	133	53	138
	CPU (4)	48	127	87	177	61.8	145.2	61	144	71	156
	Διαφορά					11.8	11.6	11	11	18	18

Πίνακας Α.10: *BiseNet v2*

ERFNet

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	550	1052	862	1457	590.9	1175.1	592	1190	623	1225
	CPU (8)	494	1098	676	1288	572.6	1173.4	568	1171	626	1222
	Διαφορά					18.3	1.7	24	19	3	3
Tablet	GPU	35	247	44	291	38.9	270.9	39	271	41	280
	CPU (4)	118	294	152	344	131.3	319.9	130	319	140	330
	Διαφορά					92.4	49	91	48	99	50

Πίνακας Α.11: *ERFNet*

ERFNet (FULL)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (8)	570	1071	742	1246	634.7	1144.4	634	1143	662	1173
	CPU (4)	574	1084	830	1391	631.7	1180	639	1207	666	1238
	Διαφορά					3	35.6	5	64	4	65
Tablet	GPU	38	245	46	282	42.2	262.7	42	262	44	272
	CPU (4)	88	267	121	305	102.3	282.9	102	282	111	293
	Διαφορά					60.1	20.2	60	20	67	21

Πίνακας Α.12: *ERFNet (FULL)*

Α.2 Ανίχνευση Αντικειμένων

EfficientDet Lite4

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	1372	1494	1461	1597	1404	1527	1398	1521	1429	1555
	-	-	-	-	-	-	-	-	-	-	-
	Διαφορά					-	-	-	-	-	-
Tablet	NNAPI	169	195	183	210	173.3	200.3	173	200	177	204
	GPU	198	235	220	283	209	259.3	210	259	214	267
	Διαφορά					35.7	59	37	59	37	63

Πίνακας Α.13: *EfficientDet Lite4*

EfficientDet Lite1

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	218	288	301	377	247.8	323.3	246	323	290	366
	CPU (2)	306	376	328	408	313	388.4	313	388	317	395
	Διαφορά					65.2	65.1	67	65	27	29
Tablet	NNAPI	42	53	50	64	46.6	59.2	47	60	49	62
	GPU	41	58	69	98	56.9	86	57	86	63	93
	Διαφορά					10.3	26.8	10	26	14	31

Πίνακας Α.14: *EfficientDet Lite1*

YOLO v5s

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	223	293	303	373	246.2	319	248	318	261	337
	CPU4	265	334	316	388	279.7	352	278	350	291	364
	Διαφορά					33.5	33	30	32	30	27
Tablet	GPU	39	57	77	109	61.2	87.1	62	87	66	91
	NNAPI	69	78	87	97	73.5	85.3	73	85	76	89
	Διαφορά					12.3	1.8	11	2	10	2

Πίνακας Α.15: *YOLO v5s*

YOLO v5s (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	225	293	304	385	246.6	320	248	320	261	336
	CPU (2)	321	390	379	458	341.4	413.8	341	413	355	427
	Διαφορά					94.8	93.8	93	93	94	91
Tablet	GPU	39	52	72	102	60.8	86.6	61	86	65	92
	CPU (2)	71	95	104	131	84.2	110.3	84	110	95	121
	Διαφορά					23.4	23.7	23	24	30	29

Πίνακας Α.16: *YOLO v5s (DR)*

SSD MobileDet

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	153	213	222	300	178.9	253.1	172	246	197	276
	CPU (8)	197	269	427	497	242	314.1	240	311	271	340
	Διαφορά					63.1	61	68	64	74	64
Tablet	GPU	24	49	33	71	28	55.8	28	56	30	59
	NNAPI	38	48	46	61	40.4	52.6	40	52	42	56
	Διαφορά					12.4	3.2	12	4	12	3

Πίνακας Α.17: *SSD MobileDet*

SSD MobileDet (FULL)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	161	221	226	299	187.5	251.7	184	249	211	278
	CPU (4)	197	254	317	379	228.2	290	225	287	251	315
	Διαφορά					40.7	38.3	41	38	40	37
Tablet	NNAPI	12	21	17	28	14.6	24.6	14	25	16	26
	GPU	23	30	32	75	28.4	51.3	29	51	30	55
	Διαφορά					13.8	26.7	15	26	14	29

Πίνακας Α.18: *SSD MobileDet (FULL)*

SpaghettiNet Large

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	111	174	158	232	132.3	196.4	128	196	144	210
	CPU (2)	171	230	193	259	180.5	242.2	180	241	183	248
	Διαφορά					48.2	45.8	52	45	39	38
Tablet	NNAPI	12	21	18	33	14.9	25.4	15	25	17	28
	GPU	20	29	26	60	22.5	45.1	22	45	24	48
	Διαφορά					7.6	19.7	7	20	7	20

Πίνακας Α.19: *SpaghettiNet Large*

SpaghettiNet Small

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	79	134	124	199	94.6	158.5	93	157	108	172
	CPU (2)	126	185	145	209	135.5	197.9	136	198	138	203
	Διαφορά					40.9	39.4	43	41	30	31
Tablet	NNAPI	10	19	16	30	12.6	22.8	12	23	14	25
	GPU	14	33	21	52	17	39.3	17	39	18	42
	Διαφορά					4.4	16.5	5	16	4	17

Πίνακας Α.20: *SpaghettiNet Small*

SSD MobileNetv3 Large

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	57	115	113	177	76.4	138.6	73	136	92	154
	CPU (4)	77	134	143	192	90.1	151.3	91	150	96	159
	Διαφορά					13.7	12.7	18	14	4	5
Tablet	GPU	13	29	21	53	16.6	39.1	17	39	19	42
	CPU (2)	27	47	52	78	37.9	59.5	38	59	43	65
	Διαφορά					21.3	20.4	21	20	24	23

Πίνακας Α.21: *SSD MobileNetv3 Large*

SSD MobileNetv3 Small

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	41	102	59	123	47.3	109.2	47	109	51	115
	GPU	33	92	68	130	45.6	109	47	110	52	118
	Διαφορά					1.7	0.2	0	1	1	3
Tablet	GPU	9	29	20	48	14	36.5	14	36	16	40
	CPU (2)	13	25	29	55	20.4	42.2	20	42	25	47
	Διαφορά					6.4	5.7	6	6	9	7

Πίνακας Α.22: *SSD MobileNetv3 Small*

Α.3 Εκτίμηση Ανθρώπινης Πόζας

MoveNet Singlepose Thunder

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	84	151	134	207	103.7	176.4	100	175	121	195
	CPU (4)	129	191	170	232	140.2	204.9	139	204	147	212
	Διαφορά					36.5	28.5	39	29	26	17
Tablet	GPU	14	36	20	53	16.2	41.6	16	41	17	45
	NNAPI	29	39	40	59	32.9	44.6	33	45	35	47
	Διαφορά					16.7	3	17	4	18	2

Πίνακας Α.23: *MoveNet Singlepose Thunder*

MoveNet Singlepose Lightning

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	31	80	59	122	44.8	102.8	45	103	54	115
	CPU (4)	41	80	77	124	47.4	92.4	46	91	55	100
	Διαφορά					2.6	10.4	1	12	1	15
Tablet	GPU	8	16	12	45	9	28.9	9	28	10	32
	NNAPI	13	21	23	33	16.6	25.6	16	25	20	29
	Διαφορά					7.6	3.3	7	3	10	3

Πίνακας Α.24: *MoveNet Singlepose Lightning*

MoveNet Multipose Lightning

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	91	134	128	184	104.2	149.4	104	148	112	160
	CPU (2)	115	150	131	178	117	162.1	117	162	118	168
	Διαφορά					12.8	12.7	13	14	6	8
Tablet	GPU	17	26	33	68	27.5	45.7	29	46	30	50
	CPU (2)	30	44	58	72	44.8	60.5	45	60	50	66
	Διαφορά					17.3	14.8	16	14	20	16

Πίνακας Α.25: *MoveNet Multipose Lightning*

BlazePose Lite

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	82	145	107	183	93.6	162.6	93	162	100	171
	GPU	82	125	130	193	98.1	168.5	99	169	104	179
	Διαφορά					4.5	5.9	6	7	4	8
Tablet	NNAPI	34	44	45	56	38	49.9	38	50	41	53
	GPU	31	52	52	76	37.4	60.3	36	59	45	70
	Διαφορά					0.6	10.4	2	9	4	17

Πίνακας Α.26: *BlazePose Lite*

BlazePose Lite (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	82	151	125	192	96.7	167.6	98	168	102	177
	CPU (2)	113	178	137	208	125.7	194.8	126	195	131	201
	Διαφορά					29	27.2	28	27	29	24
Tablet	GPU	30	52	52	80	37.6	61.2	36	59	46	71
	NNAPI	42	51	52	65	45.8	57.4	46	57	49	61
	Διαφορά					8.2	3.8	10	2	3	10

Πίνακας Α.27: *BlazePose Lite (DR)*

BlazePose Lite (FULL)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	97	145	127	180	111	162.8	110	162	117	170
	CPU (2)	113	155	126	186	118.9	169.6	119	169	123	176
	Διαφορά					7.9	6.8	9	7	6	6
Tablet	NNAPI	20	31	26	39	23.1	34.5	23	34	25	37
	GPU	29	44	53	73	37.5	55.6	38	55	44	63
	Διαφορά					14.4	21.1	15	21	19	26

Πίνακας Α.28: *BlazePose Lite (FULL)*

EfficientPoseII Lite

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	684	820	759	893	712.9	848.3	719	853	735	870
	CPU (8)	856	989	1123	1248	912.8	1047	913	1045	943	1079
	Διαφορά					199.9	198.7	194	192	208	209
Tablet	GPU	157	199	209	256	187.2	236.4	187	237	196	247
	NNAPI	185	214	209	240	192.2	223.7	192	224	197	229
	Διαφορά					5	12.7	5	13	1	18

Πίνακας Α.29: *EfficientPoseII Lite*

EfficientPoseII Lite (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	687	832	780	932	728.9	876.5	729	876	770	920
	CPU (2)	1102	1234	1150	1282	1111.7	1246.4	1110	1245	1117	1254
	Διαφορά					382.8	369.9	381	369	347	334
Tablet	GPU	169	219	220	266	187.2	236.8	187	237	194	245
	CPU (2)	260	304	282	332	268.7	317.6	268	318	273	324
	Διαφορά					81.5	80.8	81	81	79	79

Πίνακας Α.30: *EfficientPoseII Lite (DR)*

EfficientPoseII Lite (INT)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	707	844	804	935	759.5	895.1	780	914	789	925
	CPU (2)	957	1085	986	1126	967.3	1102	967	1101	973	1109
	Διαφορά					207.8	206.9	187	187	184	184
Tablet	GPU	164	216	221	279	187.1	242.1	186	242	196	251
	CPU (2)	208	250	224	281	214.2	261.6	214	261	219	267
	Διαφορά					27.1	19.5	28	19	23	16

Πίνακας Α.31: *EfficientPoseII Lite (INT)*

EfficientPoseII Lite (FP16)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	685	819	757	888	714.7	848.9	721	854	735	868
	CPU (8)	865	1002	1061	1193	921.1	1056.2	920	1054	957	1091
	Διαφορά					206.3	207.3	199	200	222	223
Tablet	GPU	169	215	204	256	186.7	237	186	237	195	246
	CPU (4)	206	248	255	303	223	270.7	221	270	234	282
	Διαφορά					36.3	33.7	35	33	39	36

Πίνακας Α.32: *EfficientPoseII Lite (FP16)*

EfficientPoseRT Lite

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	235	309	258	349	243.7	327	243	326	250	336
	GPU	217	295	273	343	248	319.5	247	317	264	334
	Διαφορά					4.3	7.5	4	9	14	2
Tablet	NNAPI	62	75	74	94	67.6	82.6	67	82	70	87
	GPU	63	76	103	129	85.4	110.6	86	111	94	121
	Διαφορά					17.8	28	19	29	24	34

Πίνακας Α.33: *EfficientPoseRT Lite*

EfficientPoseRT Lite (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	209	289	278	357	249	321.5	247	320	264	337
	CPU (2)	292	369	315	398	299	378.1	299	378	302	383
	Διαφορά					50	56.6	52	58	38	46
Tablet	GPU	66	79	101	129	84.1	109.4	85	110	93	119
	NNAPI	85	98	103	122	91.8	105.7	92	105	96	111
	Διαφορά					7.7	3.7	7	5	3	8

Πίνακας Α.34: *EfficientPoseRT Lite (DR)*

EfficientPoseRT Lite (INT)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	209	289	278	357	249	321.5	247	320	264	337
	CPU (2)	267	327	286	358	273.3	344.8	273	345	277	351
	Διαφορά					24.3	23.3	26	25	13	14
Tablet	GPU	68	93	105	130	84.8	110.1	86	111	92	118
	CPU (2)	71	93	104	130	86.5	112.2	87	112	95	121
	Διαφορά					1.7	2.1	1	1	3	3

Πίνακας Α.35: *EfficientPoseRT Lite (INT)*

EfficientPoseRT Lite (FP16)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	234	297	268	339	243.8	316.1	244	316	249	325
	GPU	205	289	281	370	250.2	327.6	248	324	266	343
	Διαφορά					6.4	11.5	4	8	17	18
Tablet	GPU	60	83	103	128	85.5	110.8	86	112	93	120
	CPU (2)	73	85	113	137	89.7	115	89	115	98	125
	Διαφορά					4.2	4.2	3	3	5	5

Πίνακας Α.36: *EfficientPoseRT Lite (FP16)*

CPM

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	87	137	130	185	105.8	156.2	105	156	120	170
	CPU (4)	151	196	191	240	164.8	212.9	165	213	175	224
	Διαφορά					59	56.7	60	57	55	54
Tablet	NNAPI	20	28	28	46	22.5	32.3	22	32	25	35
	GPU	21	37	41	66	30.2	48.4	30	48	36	55
	Διαφορά					7.7	16.1	8	16	11	20

Πίνακας Α.37: *CPM*

CPM (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	82	130	126	178	104.8	154.7	103	154	120	170
	CPU (2)	135	179	157	212	142.3	190.6	142	189	147	198
	Διαφορά					37.5	35.9	39	35	27	28
Tablet	GPU	21	29	64	81	29.8	47.7	29	48	36	54
	NNAPI	40	49	50	65	43.6	53.2	43	53	46	56
	Διαφορά					13.8	5.5	14	5	10	2

Πίνακας Α.38: *CPM (DR)*

Hourglass

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	CPU (4)	47	86	87	128	55.6	100.9	55	100	59	108
	GPU	46	96	88	141	62	118.1	60	117	75	134
	Διαφορά					6.4	17.2	15	17	16	26
Tablet	GPU	9	17	19	41	14.6	33.3	15	34	16	37
	NNAPI	14	21	21	34	16.4	25.6	16	25	18	28
	Διαφορά					1.8	7.7	1	9	2	9

Πίνακας Α.39: *Hourglass*

Hourglass (DR)

Συσκευή	Επεξεργαστής	Min		Max		Mean		Median		90th percentile	
		Inf	Total	Inf	Total	Inf	Total	Inf	Total	Inf	Total
Κινητό	GPU	45	95	86	140	61.4	117.8	60	116	73	133
	CPU (2)	126	169	262	325	139.3	184.5	136	181	143	193
	Διαφορά					77.9	66.7	76	65	70	60
Tablet	GPU	10	24	19	43	14.6	33.5	15	34	16	37
	NNAPI	33	40	43	53	38.4	47.5	38	48	40	50
	Διαφορά					23.8	14	23	14	24	13

Πίνακας Α.40: *Hourglass (DR)*

Βιβλιογραφία

[1] «Mobile Operating System Market Share Worldwide» [Ηλεκτρονικό]. Διαθέσιμο: https://gs.statcounter.com/os-market-share/mobile/worldwide. [Πρόσβαση: 13 Σεπτεμβρίου 2021].

[2] Jean Ponce, David A. Forsyth, «Computer Vision: A Modern Approach (2nd Edition)». Pearson. 2011.

[3] Ian J. Goodfellow, Yoshua Bengio και Aaron C. Courville. «Deep Learning. Adaptive computation and Machine Learning. MIT Press». 2016.

[4] «A.I. Technical - Machine vs Deep Learning». 2019. [Ηλεκτρονικό]. Διαθέσιμο: https://lawtomated.com/a-i-technical-machine-vs-deep-learning/. [Πρόσβαση: Σεπτέμβριος 2021].

[5] «Deep Learning». QuantDare, 13 June 2019. [Ηλεκτρονικό]. Διαθέσιμο: https://quantdare.com/what-is-the-difference-between-deep-learning-and-machine-learning/deep_learning. [Πρόσβαση: 21 Οκτωβρίου 2021].

[6] Li Deng και Dong Yu. «Deep Learning: Methods and Applications,» Found. Trends Signal Process. 2014.

[7] «What is Deep Learning?» 2021. [Ηλεκτρονικό]. Διαθέσιμο: https://www.mathworks.com/discovery/deep-learning.html. [Πρόσβαση: 3 Σεπτεμβρίου 2021].

[8] Carole-Jean Wu, et al. «Machine Learning at Facebook: Understanding Inference at the Edge». 25th IEEE International Symposium on High Performance Computer Architecture, HPCA 2019. Washington, DC, USA.

[9] «Notes for the Stanford CS Class "CS231n Convolutional Neural Networks for Visual Recognition"». [Ηλεκτρονικό]. Διαθέσιμο: https://cs231n.github.io/. [Πρόσβαση: 6 Σεπτεμβρίου 2021].

[10] Michael Nielsen. «Neural Networks and Deep Learning». Determination Press. 2015.

[11] Vishal Passricha, et al. «Average Polling». Science Direct. [Ηλεκτρονικό]. Διαθέσιμο: https://www.sciencedirect.com/topics/computer-science/average-pooling. [Πρόσβαση: 8 Σεπτεμβρίου 2021].

[12] Ji Wang, et al. «Deep Learning Towards Mobile Applications». 2018. 38th IEEE International Conference on Distributed Computing Systems, ICDCS 2018, Vienna, Austria.

[13] Paul Sallomi, et al. «Technology, Media and Telecommunications predictions 2020». Deloitte. 2020.

[14] M. Sahni. «8-Bit Quantization and TensorFlow Lite: Speeding up mobile inference with low precision». [Ηλεκτρονικό]. Διαθέσιμο: https://heartbeat.fritz.ai/8-bit-quantization-and-tensorflow-lite-speeding-up-mobile-inference-with-low-precision-a882dfcafbbd. [Πρόσβαση: 13 Σεπτεμβρίου 2021].

[15] Shervin Minaee, et al. «Image Segmentation Using Deep Learning: A Survey». CoRR abs/2001.05566. 2020.

[16] Jianbo Shi και Jitendra Malik. «Normalized Cuts and Image Segmentation». IEEE Trans. Pattern Anal. Mach. Intell. 2000.

[17] Tsung-Yi Lin, et al. «Microsoft COCO: Common Objects in Context». Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland.

[18] Zhong-Qiu Zhao, et al. «Object Detection with Deep Learning: A Review». IEEE Transactions on Neural Networks and Learning Systems. 2019.

[19] «An Overview of Human Pose Estimation with Deep Learning». BeyondMinds. [Ηλεκτρονικό]. Διαθέσιμο: https://beyondminds.ai/blog/an-overview-of-human-pose-estimation-with-deep-learning. [Πρόσβαση: 22 Σεπτεμβρίου 2021].

[20] Ce Zheng, et al. «Deep Learning-Based Human Pose Estimation: A Survey». CoRR abs/2012.13392. 2021.

[21] Yuezun Li, Ao Luo και Siwei Lyu. «Fast Portrait Segmentation With Highly Light-Weight Network». IEEE International Conference on Image Processing, ICIP 2020. Abu Dhabi, United Arab Emirates.

[22] Song-Hai Zhanga, et al. «PortraitNet: Real-time Portrait Segmentation Network for Mobile Device» Computers & Graphics, τόμ. 80, pp. 104-113, 2019.

[23] «DAGM 2007» [Ηλεκτρονικό]. Διαθέσιμο: https://conferences.mpi-inf.mpg.de/dagm/2007. [Πρόσβαση: 30 Οκτωβρίου 2021].

[24] J. Hui. «mAP (mean Average Precision) for Object Detection». 2018. [Ηλεκτρονικό]. Διαθέσιμο: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173. [Πρόσβαση: 1 Νοεμβρίου 2021].

[25] «Keypoint Evaluation». COCO Common Object in Context. [Ηλεκτρονικό]. Διαθέσιμο: https://cocodataset.org/#keypoints-eval. [Πρόσβαση 3 Νοεμβρίου 2021].

[26] H. Schildt. «Java: The complete Reference», 9th edition, McGraw-Hill Education.

[27] «Java: Computer Programming Language». [Ηλεκτρονικό]. Διαθέσιμο: https://www.britannica.com/technology/Java-computer-programming-language. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[28] «PYPL PopularitY of Programming Language». [Ηλεκτρονικό]. Διαθέσιμο: https://pypl.github.io/PYPL.html. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[29] A. Zeichick. «Java 17 is here: 14 JEPs with exciting new language and JVM features». [Ηλεκτρονικό]. Διαθέσιμο: https://blogs.oracle.com/javamagazine/java-jdk-17-generally-available. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[30] J. Callaham. «The history of Android: The evolution of the biggest mobile OS in the world». [Ηλεκτρονικό]. Διαθέσιμο: https://www.androidauthority.com/history-android-os-name-789433/. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[31] «Meet Android Studio». [Ηλεκτρονικό]. Διαθέσιμο: https://developer.android.com/studio/intro. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[32] «Android Studio». [Ηλεκτρονικό]. Διαθέσιμο: https://searchmobilecomputing.techtarget.com/definition/Android-Studio. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[33] V. Sheromova. «A brief history of Python». 2020. [Ηλεκτρονικό]. Διαθέσιμο: https://exyte.com/blog/a-brief-history-of-python. [Πρόσβαση: 6 Οκτωβρίου 2021].

[34] Martin Abadi, et al. «TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems». CoRR abs/1603.04467. 2016.

[35] C. Kozyrkov. «9 Things You Should Know About TensorFlow». 2018. [Ηλεκτρονικό]. Διαθέσιμο: https://hackernoon.com/9-things-you-should-know-about-tensorflow-9cf0a05e4995. [Πρόσβαση: 6 Οκτωβρίου 2021].

[36] R. Khandelwal. «A Basic Introduction to TensorFlow Lite». [Ηλεκτρονικό]. Διαθέσιμο: https://towardsdatascience.com/a-basic-introduction-to-tensorflow-lite-59e480c57292. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[37] «TensorFlow Lite Guide». [Ηλεκτρονικό]. Διαθέσιμο: https://www.tensorflow.org/lite/guide. [Πρόσβαση: 14 Σεπτεμβρίου 2021].

[38] «Github-Portrait Segmentation». [Ηλεκτρονικό]. Διαθέσιμο: https://github.com/anilsathyan7/Portrait-Segmentation. [Πρόσβαση: 27 Οκτωβρίου 2021].

[39] Liang-Chieh Chen, et al. «DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs». IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018.

[40] «Github - UNet Industrial». [Ηλεκτρονικό]. Διαθέσιμο: https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/Segmentation/UNet_Industrial. [Πρόσβαση: 1 Νοεμβρίου 2021].

[41] Changqian Yu, et al. «BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation». International Journal of Computer Vision. 2021.

[42] Eduardo Romera, et al. «ERFNet: Efficient Residual Factorized ConvNet for Real-time Semantic Segmentation». IEEE Transactions on Intelligent Transportation Systems. 2018.

[43] Mingxing Tan, Ruoming Pang και Quoc V. Le. «EfficientDet: Scalable and Efficient Object Detection». 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA.

[44] Joseph Redmon, et al. «You Only Look Once: Unified, Real-Time Object Detection». 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA.

[45] «MoveNet.Singlepose model card». [Ηλεκτρονικό]. Διαθέσιμο: https://storage.googleapis.com/movenet/MoveNet.SinglePose%20Model%20Card.pdf. [Πρόσβαση: 26 Οκτωβρίου 2021].

[46] «MoveNet.MultiPose model card». [Ηλεκτρονικό]. Διαθέσιμο: https://storage.googleapis.com/movenet/MoveNet.MultiPose%20Model%20Card.pdf. [Πρόσβαση: 26 Οκτωβρίου 2021].

[47] «MediaPipe Pose». [Ηλεκτρονικό]. Available: https://google.github.io/mediapipe/solutions/pose.html. [Πρόσβαση: 26 Οκτωβρίου 2021].

[48] Daniel Groos, Heri Ramampiaro και Espen A. F. Ihlen. «EfficientPose: Scalable single-person pose estimation». Applied Intelligence, vol. 51. 2021.