ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

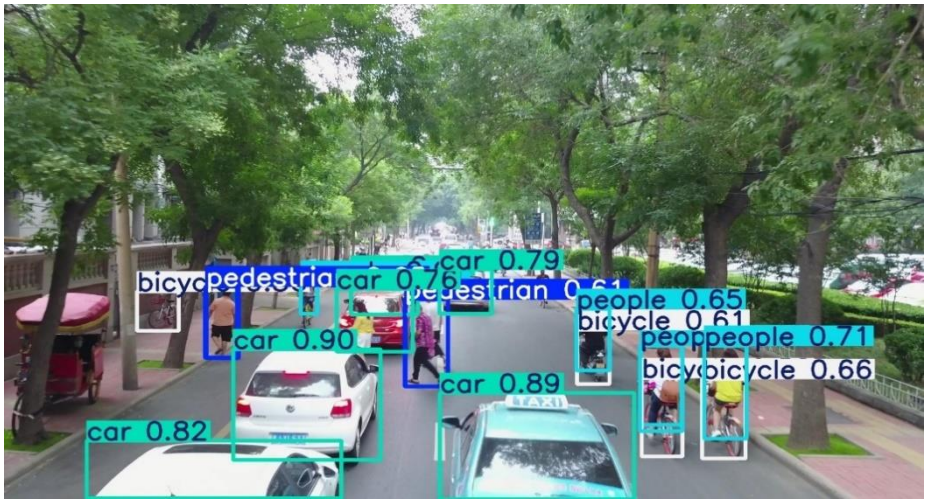
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ
(WEB INTELLIGENCE)

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανίχνευση μικρών αντικειμένων με μεθόδους μηχανικής μάθησης



Του φοιτητή
Κωνσταντίνου Τσομπανίδη
Αρ. Μητρώου: 26/2023

Επιβλέπων
Παναγιώτης Αδαμίδης
Καθηγητής

Θεσσαλονίκη, Σεπτέμβριος 2025

Τίτλος Μ.Δ.Ε. Ανίχνευση μικρών αντικειμένων με μεθόδους μηχανικής μάθησης
Κωδικός Δ.Ε. 24294
Ονοματεπώνυμο φοιτητή Κωνσταντίνος Τσομπανίδης
Ονοματεπώνυμο εισηγητή Αδαμίδης Παναγιώτης
Ημερομηνία ανάληψης Μ.Δ.Ε. 30-10-2024
Ημερομηνία περάτωσης Μ.Δ.Ε. 12-09-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως μεταπτυχιακή διπλωματική εργασία, στο ΠΜΣ «Ευφυείς Τεχνολογίες Διαδικτύου – Web Intelligence» του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Στην οικογένεια μου

Πρόλογος

Η παρούσα διπλωματική εργασία είναι το επιστέγασμα των σπουδών μου και επιλέχθηκε με στόχο την διερεύνηση ενός επιστημονικού πεδίου που εξελίσσεται ραγδαία και βρίσκει εφαρμογή σε πλήθος εφαρμογών. Δεν ήταν τυχαία η επιλογή του θέματος καθώς υπάρχει προσωπικό ενδιαφέρον για τον ανάπτυξη εφαρμογών που κάνουν χρήση της ανίχνευσης μικρών αντικειμένων σε συνδυασμό με δεδομένα που προέρχονται από UAVs ή εικόνων που συλλέγονται από το έδαφος. Η υπολογιστική όραση αποκτά όλο και μεγαλύτερο ενδιαφέρον τα τελευταία χρόνια, καθώς υπάρχει μια αυξανόμενη χρήση της από τα αυτόνομα οχήματα που σταδιακά μπαίνουν στην αγορά έως την βιομηχανία και τα UAVs που χρησιμοποιούνται σε πλήθος εφαρμογών τόσο της καθημερινότητας όσο και στην έρευνα.

Η ενασχόληση μου με τη παρούσα εργασία με βοήθησε να εμβαθύνω στο θεωρητικό υπόβαθρο της υπολογιστικής όρασης και των προηγμένων τεχνικών που εφαρμόζονται στην επεξεργασία και ενίσχυση των δεδομένων αλλά και στα κριτήρια με τα οποία επιλέγουμε τα μοντέλα που θα χρησιμοποιήσουμε καθώς και στην προσέγγιση μας στο σύνολο δεδομένων εκπαίδευσης. Επίσης, κατανόησα σε βάθος τις προκλήσεις που μπορεί να συναντήσει κάποιος υλοποιώντας ένα σύστημα ανίχνευσης μικρών αντικειμένων πάνω σε πραγματικές συνθήκες. Το σημαντικότερο όμως όφελος ήταν η ενίσχυση της ερευνητικής μου σκέψης, ο τρόπος που αντιμετωπίζω πολυδιάστατα προβλήματα και η ανάπτυξη δομημένης μεθοδολογικής προσέγγισης.

Η τριβή με μια τεχνολογία αιχμής που βρίσκεται ακόμα σε εξέλιξη προσφέρει την δυνατότητα επαγγελματικής εξέλιξης και προσφοράς στην επιστημονική κοινότητα και μπορεί να συνεισφέρει στην κοινωνία και στην οικονομία.

Περίληψη

Η υπολογιστική όραση εξελίσσεται σε μια από τις σημαντικότερες τεχνολογίες του Industry 4.0. με κρίσιμες εφαρμογές σε τομείς όπως η ανάλυση εναέριων εικόνων, η επιτήρηση δημόσιων χώρων, η οδική ασφάλεια και η αυτοκινητοβιομηχανία. Η ακρίβεια και η αποδοτικότητα των συστημάτων ανίχνευσης είναι καίριας σημασίας για την ασφαλή και ορθή λειτουργία των περιβαλλόντων όπου εφαρμόζεται.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανίχνευση μικρών αντικειμένων με μεθόδους μηχανικής μάθησης. Η ανίχνευση αντικειμένων μικρού μεγέθους διαφέρει από την γενική ανίχνευση αντικειμένων καθώς το μέγεθος και το είδος των εικόνων εμφανίζουν πολύ διαφορετικές ιδιότητες και προκλήσεις. Η παρούσα εργασία εστιάζει στην αποτελεσματική ανάπτυξη και αξιολόγηση μεθόδων ανίχνευσης μικρών αντικειμένων. Στόχοι της εργασίας είναι η διερεύνηση και ανάλυση σύγχρονων μεθόδων ανίχνευσης μικρών αντικειμένων, η κατανόηση των προκλήσεων στην ανάπτυξη των μεθόδων, η σύγκριση και αξιολόγηση καταλληλότερων τεχνικών εύρεσης, η παραμετροποίηση και βελτιστοποίηση υπαρχόντων μοντέλων, και τέλος η εξέταση της δυνατότητας υλοποίησης ενός μοντέλου ανίχνευσης μικρών αντικειμένων σε περιορισμένα υπολογιστικά περιβάλλοντα (π.χ. Raspberry Pi), αξιολογώντας την ταχύτητα και ακρίβεια αναγνώρισης.

Στην παρούσα εργασία χρησιμοποιούνται δυο καθιερωμένα και αξιόπιστα σύνολα δεδομένων, το VisDrone 2019, με εικόνες από UAVs σε διαφορετικές περιβαλλοντικές συνθήκες, και το DOTA το οποίο περιλαμβάνει δορυφορικές εικόνες σε ποικίλα περιβάλλοντα, κάνοντας χρήση Oriented Bounding Boxes. Η μεθοδολογία στηρίχθηκε στην επιλογή μοντέλων μέσω βιβλιογραφικής ανασκόπησης και δοκιμής τους με διαφορετικές παραμετροποιήσεις. Τα μοντέλα που εξετάστηκαν είναι τα YOLOv11n/s/m, RT-DETR-l, YOLOv11n/m-OBB. Όλα τα μοντέλα είναι προεκπαιδευμένα και στη συνέχεια γίνεται η εκπαίδευση τους με και χωρίς την χρήση υπερπαραμέτρων, με σκοπό την βελτίωση της ακρίβειας πάνω στα συγκεκριμένα σύνολα δεδομένων.

Από τα αποτελέσματα της ανάλυσης διαπιστώνουμε ότι τα μοντέλα, για το σύνολο δεδομένων VisDrone 2019, YOLOv11n και YOLOv11s είναι καταλληλότερα και ελαφρύτερα για συστήματα περιορισμένων πόρων, προσφέροντας υψηλή ταχύτητα παρά το χαμηλότερο mAP. Τα μοντέλα YOLOv11m είναι πιο αποδοτικά υστερώντας σε χρόνο. Αντίθετα το μοντέλο RT-DETR-l, αν και αποδοτικό, είναι ιδιαίτερα αργό και απαιτεί εξαιρετικά υψηλούς υπολογιστικούς πόρους. Για το σύνολο δεδομένων DOTA, τα μοντέλο YOLOv11n-OBB και YOLOv11m-OBB έδωσαν αξιόλογα αποτελέσματα με το δεύτερο να γίνεται μη διαχειρίσιμο υπό εκτεταμένη υπερπαραμετροποίηση.

Τέλος, οι αλγόριθμοι εφαρμόστηκαν δοκιμαστικά σε real time βίντεο με ικανοποιητικά αποτελέσματα, ακόμα και σε περιβάλλοντα χαμηλού φωτισμού και πυκνής διάταξης.

Small Object Detection with Machine Learning Methods

Konstantinos Tsompanidis

Abstract

Computer vision is evolving into one of the most significant technologies of Industry 4.0, with critical applications in fields such as aerial image analysis, public space surveillance, road safety, and the automotive industry. The accuracy and efficiency of detection systems are of vital importance for the safe and proper operation of the environments in which they are applied.

The subject of this thesis is the detection of small objects using machine learning methods. Small object detection differs from general object detection, as the size and type of images present very different properties and challenges. This work focuses on the effective development and evaluation of methods for detecting small objects. The objectives of the study are the investigation and analysis of state-of-the-art small object detection methods, the understanding of the challenges in their development, the comparison and evaluation of the most suitable detection techniques, the parameterization and optimization of existing models, and the deployment and testing of a functional version of the model on the Raspberry Pi platform, demonstrating the feasibility of real time inference on edge devices.

In this thesis, two established and reliable datasets are used: VisDrone 2019, which contains UAV images captured under diverse environmental conditions, and DOTA, which includes satellite images in various environments and makes use of Oriented Bounding Boxes (OBB). The methodology relied on the selection of models through a literature review and their testing under different parameterizations. The models examined are YOLOv11n/s/m, RT-DETR-l, and YOLOv11n/m-OBB. All models are pre-trained and subequenty trained further, both with and without the use of hyperparameters, in order to improve accuracy on the specific datasets.

The results of the analysis show that, for the VisDrone 2019 dataset, YOLOv11n and YOLOv11s are more suitable and lightweight for resource-constrained systems, offering high speed despite a lower mAP. The YOLOv11m models are more accurate but less efficient in terms of processing time. Conversely, the RT-DETR- model, although effective, is particularly slow and requires extremely high computational resources. For the DOTA dataset, the YOLOv11n-OBB and YOLOv11m-OBB models produced promising results, with the latter becoming unmanageable under extensive hyperparameter tuning.

Finally, the algorithms were tested on real-time video, yielding satisfactory results even in low-light and densely populated environments.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες σε όλους αυτούς που βοήθησαν να ολοκληρωθεί η παρούσα διπλωματική εργασία όπως και το μεταπτυχιακό πρόγραμμα σπουδών.

Αρχικά, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή μου, κύριο Παναγιώτη Αδαμίδη για τις πολύτιμες συμβουλές, την εμπιστοσύνη του, την διαρκή υποστήριξη και την καθοδήγηση που γενναιόδωρα μου πρόσφερε. Η επιστημονική του κατάρτιση και η στάση του αποτελούν κίνητρο και παράδειγμα, το ειλικρινές του ενδιαφέρον και η άμεση επικοινωνία ήταν καταλυτικές για την ολοκλήρωση της παρούσας εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τους διδάσκοντες και το επιστημονικό και διοικητικό προσωπικό του Τμήματος για τις γνώσεις που μου πρόσφεραν και το υψηλό επίπεδο σπουδών που παρείχαν σε όλη την διάρκεια των σπουδών.

Θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου για την συνεχή υποστήριξη και ενθάρρυνση στη διάρκεια όλης της μακράς περιόδου για την περάτωση των σπουδών μου και την ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Τέλος, ευχαριστώ τους φίλους και συμφοιτητές μου για την υποστήριξη, ενθάρρυνση, τα αναρίθμητα ξενύχτια και την ανταλλαγή απόψεων καθώς συνέβαλλαν στην ανάπτυξη ενός άριστου ακαδημαϊκού κλίματος.

Ένα μεγάλο ευχαριστώ σε όλους εσάς που συμβάλλατε στην ολοκλήρωση αυτής της εργασίας και που κάνατε αυτή την δύσκολη πορεία ευχάριστη, αυτή η εργασία είναι και δική σας.

Κωνσταντίνος Τσομπανίδης, 2025

Περιεχόμενα

Κατάλογος Εικόνων / Γραφημάτων

Κατάλογος Πινάκων

Συντομογραφίες

CNN	Convolutional Neural Network
YOLO	You Only Look Once
IoU	Intersection over Union
FPS	Frames Per Second
SAHI	Slicing Aided Hyper Inference
GAN	Generative Adversarial Network
Μ.Δ.Ε	Μεταπτυχιακή Διπλωματική Εργασία
ΔΙ.ΠΑ.Ε	Διεθνές Πανεπιστήμιο της Ελλάδος
Δ.Ε.	Διπλωματική Εργασία
Π.Μ.Σ.	Πρόγραμμα Μεταπτυχιακών Σπουδών

Κεφάλαιο 1ο: Εισαγωγή

Η ανίχνευση αντικειμένων σε εικόνες ή βίντεο είναι από τα σημαντικότερα και πιο απαιτητικά προβλήματα στον τομέα της Υπολογιστικής Όρασης. Τα τελευταία χρόνια η ανάγκη για ανάπτυξη αποδοτικών αλγορίθμων και αρχιτεκτονικών Υπολογιστικής Όρασης είναι μεγαλύτερη, εξαιτίας της ανάπτυξης των αυτόνομων συστημάτων που εκτείνονται από οχήματα και ρομπότ έως και βιομηχανικές εφαρμογές. Οι απαιτήσεις σε ταχύτητα και ακρίβεια είναι υψηλές, καθότι η μηχανική όραση εφαρμόζεται πλέον σε κρίσιμα συστήματα και επιστημονικούς τομείς που εστιάζουν στην ασφάλεια και τον άνθρωπο.

Έχουν γίνει σημαντικά βήματα στην ανίχνευση αντικειμένων μεσαίου και μεγάλου μεγέθους. Ωστόσο, η ανίχνευση μικρών αντικειμένων παραμένει μια πρόκληση και πεδίο έρευνας, καθώς τα αντικείμενα αυτά καταλαμβάνουν πολύ λίγα εικονοστοιχεία (pixels) σε μια εικόνα, καθιστώντας την εκπαίδευση απαιτητική καθώς παρατηρείται απώλεια πληροφορίας κατά την εκπαίδευση, ενώ ταυτόχρονα τα αντικείμενα συχνά παρουσιάζονται σε πυκνή διάταξη στην εικόνα.

Στην παρούσα εργασία επιδιώκεται η διερεύνηση των δυνατοτήτων που μας προσφέρει η μηχανική και βαθιά μάθηση στην υλοποίηση συστημάτων μηχανικής όρασης και πιο συγκεκριμένα σε συστήματα ανίχνευσης μικρών αντικειμένων.

1.1 Τεχνητή Νοημοσύνη και Μηχανική Μάθηση

Η μηχανική μάθηση είναι υποπεδίο της επιστήμης των υπολογιστών και αντικείμενο της είναι η έρευνα και υλοποίηση αλγορίθμων που μπορούν να «μάθουν» από δεδομένα και να παράγουν προβλέψεις χωρίς ρητές εντολές. Αυτό γίνεται μέσα από την διαδικασία της παρατήρησης, ή αλλιώς εκπαίδευση, πάρα πολλών παραδειγμάτων. Για να υλοποιηθεί ένα σύστημα μηχανικής μάθησης χρειάζεται: δεδομένα για την εκπαίδευση του αλγορίθμου, το μοντέλο που θα «δει» τα δεδομένα και θα μάθει τις συνδέσεις και τα μοτίβα μεταξύ τους και τέλος μια μέθοδο αξιολόγησης του μοντέλου (εικόνα 1).

Ο τρόπος με τον οποίο χωρίζονται οι αλγόριθμοι μηχανικής μάθησης χωρίζονται σε τρεις βασικές κατηγορίες:

- Αλγόριθμοι επιβλεπόμενης μάθησης (Supervised Learning)

Στην επιβλεπόμενη μάθηση απαιτείται ένας εξωτερικός παρατηρητής που γνωρίζει καλά το πρόβλημα και τα αποτελέσματα που περιμένει να έχει από τον αλγόριθμο. Πρώτον, ορίζει τα «ορθά» παραδείγματα και δεύτερον αξιολογεί αν τα αποτελέσματα που μας δίνει ο αλγόριθμος είναι σωστά και ικανοποιητικά και καθορίζει τον τερματισμό της εκπαίδευσης του αλγορίθμου [1].

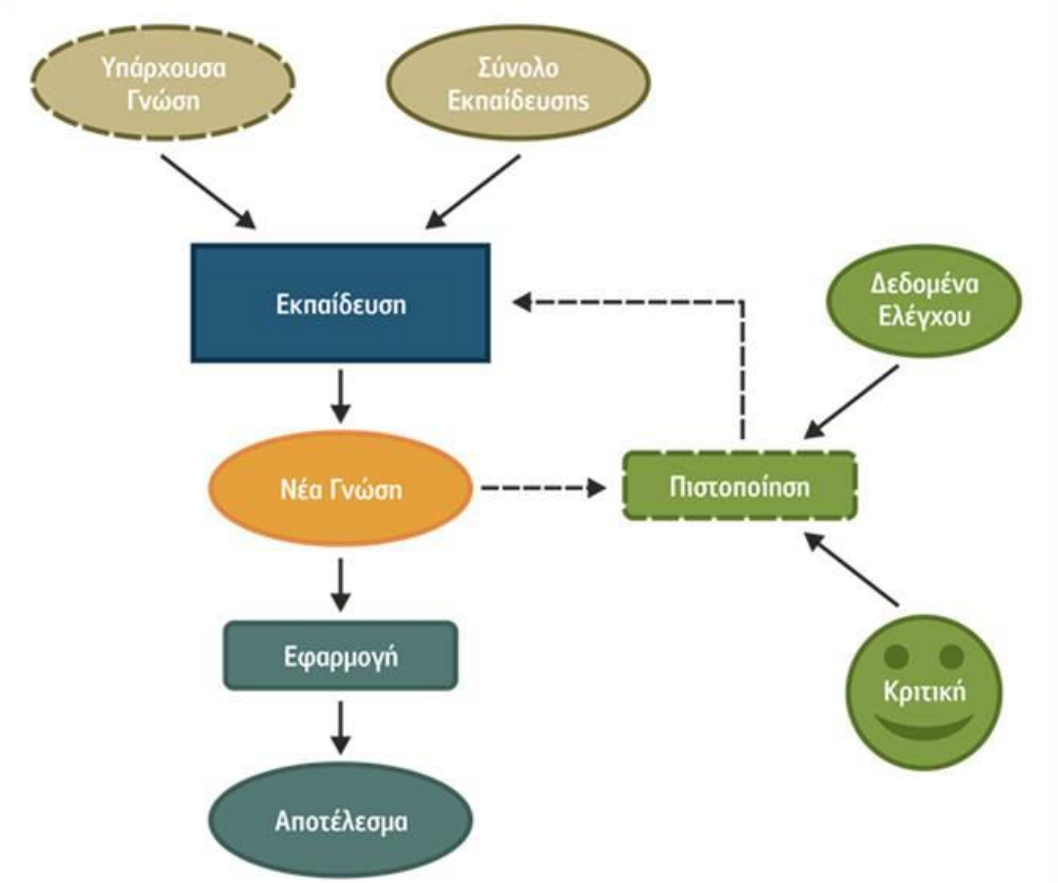
Ο αλγόριθμος κατασκευάζει μια συνάρτηση για συγκεκριμένες εισόδους με τις επιθυμητές εξόδους, σκοπεύοντας στη γενίκευση της σε εισόδους για τις οποίες οι έξοδοι δεν είναι εκ των προτέρων γνωστοί. Αλγόριθμοι επιβλεπόμενης εκπαίδευσης χρησιμοποιούνται σε προβλήματα ταξινόμησης (classification), πρόγνωσης (prediction) και διερμηνείας (Interpretation) [1].

- Αλγόριθμοι μη επιβλεπόμενης μάθησης (Unsupervised learning)

Οι αλγόριθμοι μη επιβλεπόμενης μάθησης δεν χρειάζονται κάποιον παρατηρητή, οργανώνονται και λειτουργούν μόνοι τους. Προσπαθούν να εντοπίσουν μοτίβα και ιδιότητες στα δεδομένα εξάγοντας συμπεράσματα [1]. Κατασκευάζουν ένα μοντέλο για ένα σύνολο δεδομένων ως απλές παρατηρήσεις χωρίς την γνώση γνωστών εξόδων, και η χρήση τους γίνεται κυρίως στην επίλυση προβλημάτων ανάλυσης συσχετισμών (Association Analysis) και ομαδοποίησης (Clustering) [1].

- Αλγόριθμοι ενισχυτικής μάθησης (Reinforcement learning)

Οι αλγόριθμοι αυτής της κατηγορίας λειτουργούν ως agents βασιζόμενοι στην εμπειρία τους. Πιο συγκεκριμένα οι αλγόριθμοι reinforcement learning έχουν ένα σύνολο κανόνων με τους οποίους αξιολογούν τις αποφάσεις τους. Για κάθε θετική τους απόφαση λαμβάνουν μια ανταμοιβή (reward) και για κάθε αρνητική λαμβάνουν κάποια ποινή (penalty). Είναι αλγόριθμοι που λειτουργούν χωρίς ανθρώπινη καθοδήγηση ή παρέμβαση και μόνοι τους αλληλοεπιδρούν με το περιβάλλον προσπαθώντας να επιτύχουν τους στόχους τους [1].



Εικόνα 1.1: Τα στάδια της μηχανικής μάθησης [1]

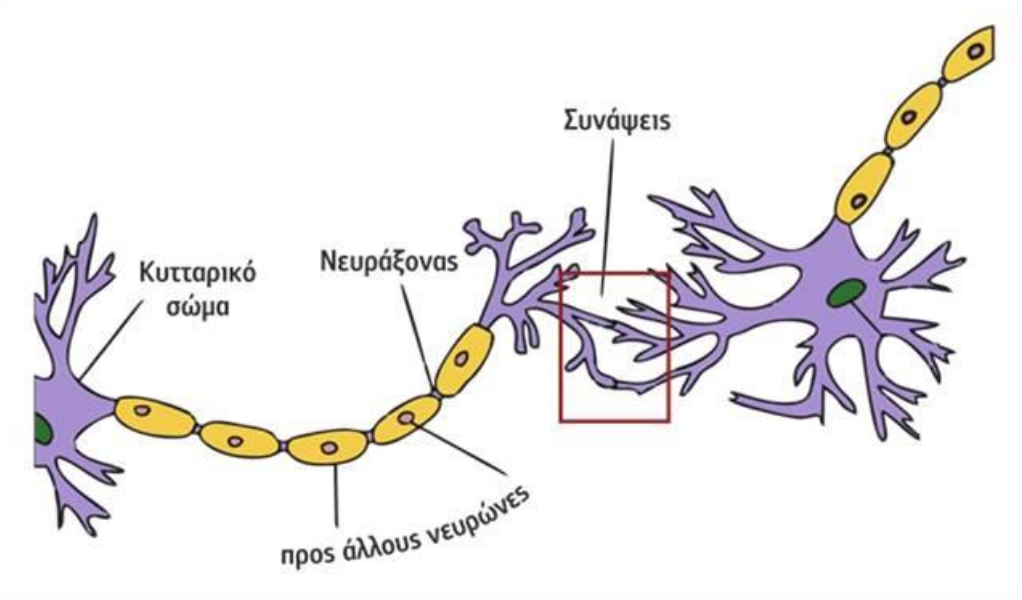
Στην Τεχνητή Νοημοσύνη ανήκουν και τα (τεχνητά) νευρωνικά δίκτυα τα οποία είναι εμπνευσμένα από τα ανθρώπινα κύτταρα (εικόνα 2) υλοποιώντας συχνά μαθηματικά μοντέλα που συναντώνται σε συμπεριφορές βιολογικών οργανισμών (εικόνα 3).

Τα νευρωνικά δίκτυα είναι διασυνδεδεμένα επίπεδα κόμβων. Κάθε νευρώνας ή κόμβος παίρνει πολλές εισόδους xi και δίνει μια μόνο έξοδο y. Οι είσοδοι του ζυγίζονται με βάρη wi και τα αποτελέσματα της συνάρτησης αθροίζονται μέσω μια συνάρτησης αθροίσματος F (συνάρτηση 1.1), όπου F είναι η συνάρτηση αθροίσματος, x_i είναι η είσοδος και w_i το βάρος.

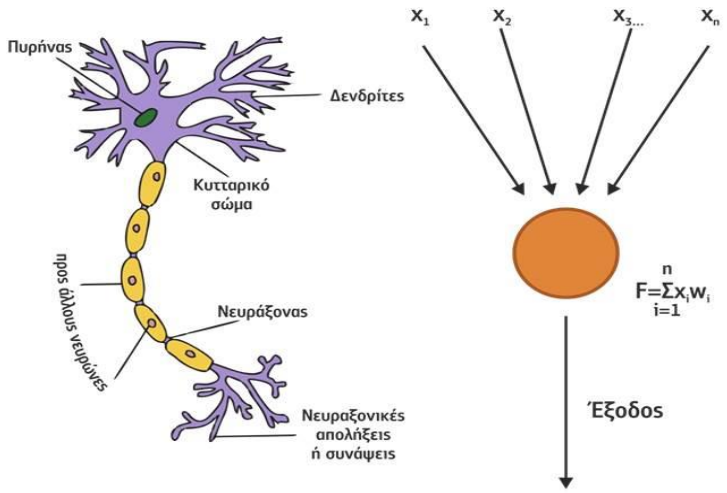
$$F = \sum_i^n x_i w_i \tag{1.1}$$

Αντίστοιχα, με τους βιολογικούς νευρώνες έτσι και ο τεχνητός δίνει κάποια έξοδο μόνο όταν το άθροισμα ικανοποιεί κάποια τιμή κατωφλιού όπου F είναι η συνάρτηση αθροίσματος, x_i είναι η είσοδος και w_i το βάρος και θ η τιμή του κατωφλιού.

$$F = \sum_i^n x_i w_i - \theta \tag{1.2}$$

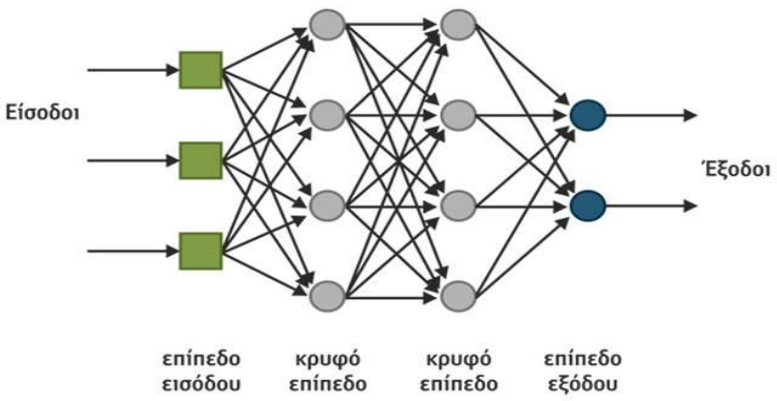


Εικόνα 1.2: Η δομή του ανθρώπινου νευρικού συστήματος [1]

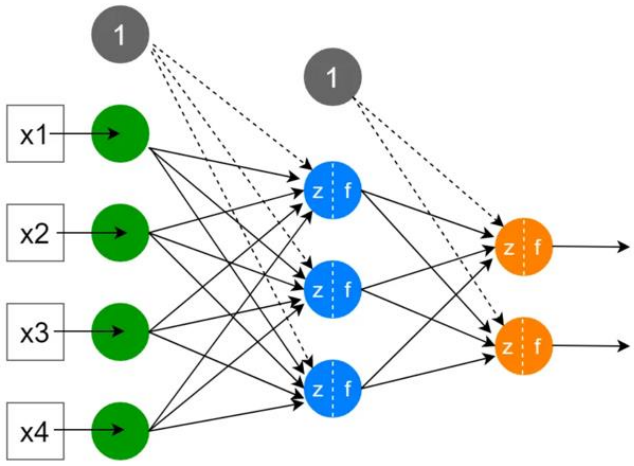


Εικόνα 1.3: Αριστερά η αναπαράσταση του βιολογικού νευρώνα και δεξιά η αναπαράσταση του τυπικού τεχνητού νευρώνα Perceptron [1].

Η είσοδος του κάθε κόμβου είναι η έξοδος του προηγούμενου ή από το περιβάλλον του, πχ μια είσοδος ενός αισθητήρα. Το πρώτο επίπεδο είναι αυτό της εισόδου (input layer) και είναι το σημείο από το οποίο η πληροφορία εισάγεται στο δίκτυο. Μετά από την επεξεργασία της πληροφορίας, στο πρώτο επίπεδο, περνάει στα επόμενα επίπεδα του δικτύου, τα οποία ονομάζονται κρυφά επίπεδα (hidden layers) και είναι τα κύρια και μεγαλύτερα μέρη του νευρωνικού δικτύου. Τέλος, το επίπεδο της εξόδου (output layer) παρέχει την τελική πληροφορία καθώς και την έξοδο του δικτύου (εικόνα 4 και εικόνα 5).



Εικόνα 1.4: Η δομή ενός νευρωνικού δικτύου με πολλά επίπεδα. Το συγκεκριμένο παράδειγμα είναι με δυο κρυμμένα επίπεδα και πλήρως συνδεδεμένους κόμβους [1].



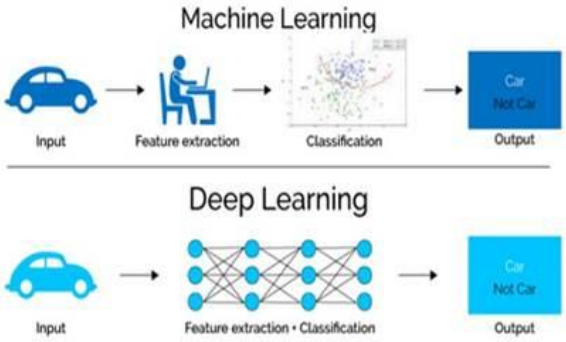
Εικόνα 1.5: Η δομή ενός νευρωνικού δικτύου με ένα κρυφό επίπεδο [2].

Η βαθιά μάθηση βασίζεται σε νευρωνικά δίκτυα και η βασική διαφορά της με τη μηχανική μάθηση είναι ότι έχει μεγαλύτερο μέγεθος δικτύου. Δίκτυα βαθιάς μάθησης κατηγοριοποιούνται σε επιβλεπόμενης και μη επιβλεπόμενης μάθησης. Χρήση της βαθιάς μάθησης γίνεται συνήθως όταν υπάρχει μεγάλος όγκος πληροφορίας. Εφαρμόζεται συνήθως σε τομείς της επεξεργασίας φυσικής γλώσσας (NLP), αναγνώριση φωνής (Speech Recognition) και η μηχανική όραση (Computer Vision).

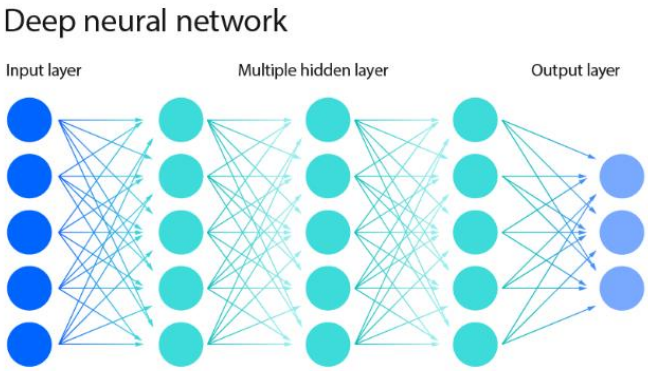
1.2 Βαθιά Μάθηση

Η βαθιά μάθηση είναι μια τεχνική μάθησης που δίνει την δυνατότητα στους υπολογιστές να μαθαίνουν μέσα από παραδείγματα, παρόμοια με τον τρόπο που μαθαίνουν οι άνθρωποι (εικόνα 6). Έχει την δυνατότητα να παίρνει αποφάσεις αυτόνομα, χωρίς την χρήση αλγορίθμων με τη μορφή βηματικής εκτέλεσης, παρέχοντας αυτόματα συστήματα που μπορούν να εξάγουν συμπεράσματα μέσα από την ανάλυση δεδομένων.

Η βαθιά μάθηση βασίζεται σε βαθιά νευρωνικά δίκτυα, δίκτυα δηλαδή με πολλαπλά κρυμμένα επίπεδα (εικόνα 7) και άρα με μεγαλύτερα μοντέλα, που δίνουν καλύτερη ακρίβεια στις προβλέψεις των μοντέλων. Βασικό της απαιτούμενο είναι η ύπαρξη μεγάλης και ικανής ποσότητας δεδομένων εκπαίδευσης. Παρατηρείται ότι όσο περισσότερα δεδομένα τροφοδοτήσουμε σε ένα μοντέλο βαθιάς μάθησης τόσο καλύτερα αυτό αποδίδει, αρκεί αυτά τα δεδομένα να είναι ποιοτικά κατάλληλα.



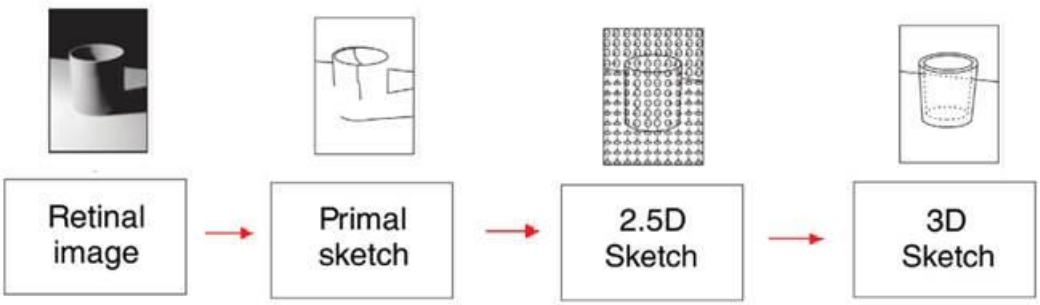
Εικόνα 1.6: Η διαφορά μηχανικής μάθησης και βαθιάς μάθησης [3].



Εικόνα 1.7: Ένα βαθύ νευρωνικό δίκτυο με πολλαπλά κρυμμένα επίπεδα [4].

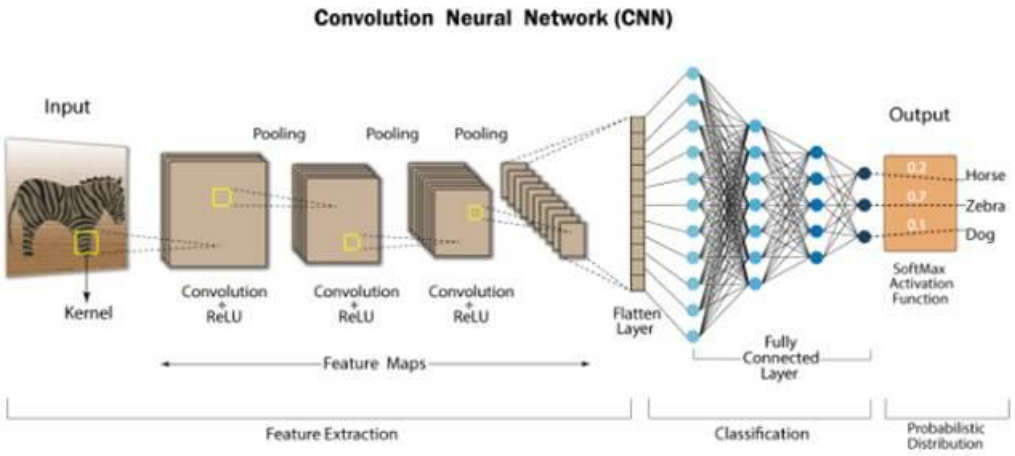
1.3 Ιστορικό Μηχανικής Όρασης

Η μηχανική όραση απασχολεί την επιστημονική κοινότητα από το 1959 όταν ερευνητές προσπάθησαν να κατανοήσουν πως μια γάτα αντιλαμβάνεται μια σειρά από εικόνες και ανακάλυψαν ότι ήταν ικανή να κατανοεί πρώτα τις πιο έντονες ακμές αντικειμένων. Στη συνέχεια, στη δεκαετία του 1960 με την εμφάνιση του πρώτου υπολογιστή και της Τεχνητής Νοημοσύνης ξεκίνησαν πειραματισμοί με δισδιάστατες εικόνες. Αυτό θεωρείται και το σημείο που ξεκίνησε η Μηχανική Όραση. Ο Larry Roberts εξήγαγε τρισδιάστατα χαρακτηριστικά από εικόνες δύο διαστάσεων. Ο David Marr το 1978 εισήγαγε την bottom-up προσέγγιση για την κατανόηση σκηνών με σκοπό την ανίχνευση ακμών και edge segmentation (εικόνα 8) [5]. Στη σύγχρονη εποχή η προσέγγιση του Marr θεωρείται υπολογιστικά κοστοβόρα και πλεονάζουσα καθώς στις περισσότερες περιπτώσεις δεν απαιτείται ολόκληρη η αναπαράσταση του αντικειμένου [6].

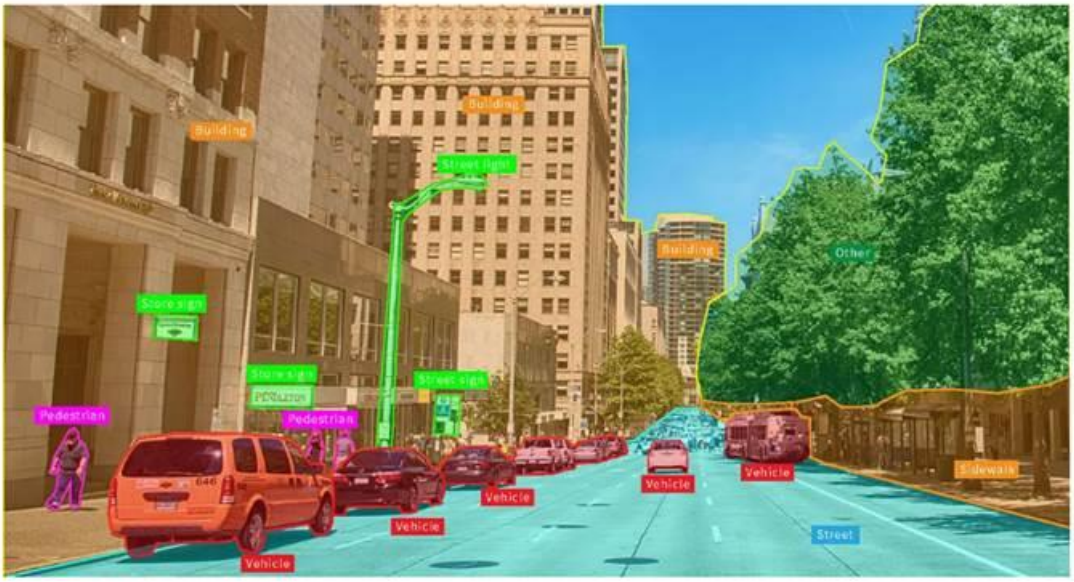


Εικόνα 1.8: Η ιδέα του David Marr [7]

Στη δεκαετία του 1970 η ανίχνευση μοτίβου είχε εδραιωθεί και μπορούσε να αναγνωρίσει σχήματα, αντικείμενα και υφές από εικόνες [8]. Πλέον στην σημερινή εποχή, που οι αλγόριθμοι βαθιάς μάθησης έχουν κυριαρχήσει στον χώρο, δίνουν αποτελέσματα με πολύ υψηλή ακρίβεια πάνω σε πολύ σύνθετα προβλήματα όπως η αναγνώριση αντικειμένων, η κατηγοριοποίηση τους, το image segmentation, κ.α. Στην επίλυση τους συνέβαλε η ανάπτυξη των συνελικτικών νευρωνικών δικτύων – CNN (εικόνα 9 και εικόνα 10).



Εικόνα 1.9: Η δομή ενός τυπικού συνελικτικού νευρωνικού δικτύου [9].



Εικόνα 1.10: Παράδειγμα εφαρμογής image segmentation [10].

1.4 Σκοπός – Στόχοι - Ερευνητικά ερωτήματα της Διπλωματικής Εργασίας

Ο σκοπός της παρούσας εργασίας είναι η ανάπτυξη και αξιολόγηση μεθόδων για την αποτελεσματική ανίχνευση μικρών αντικειμένων σε εικόνες, με εφαρμογή σε περιπτώσεις όπου η ακρίβεια είναι κρίσιμος παράγοντας, όπως για παράδειγμα στην αυτόνομη οδήγηση, στην επιτήρηση χώρων και σε ιατρικές διαγνώσεις. Η παρούσα εργασία εστιάζει μόνο σε εικόνες προερχόμενες από εναέριες λήψεις. Δεδομένου ότι τα μικρά αντικείμενα παρουσιάζουν προκλήσεις, όπως η δυσκολία στην αναγνώριση τους λόγω περιορισμένων χαρακτηριστικών και χαμηλής ανάλυσης, η εργασία επιδιώκει να βελτιώσει την απόδοση των υπαρχόντων αλγορίθμων ή να προτείνει νέες προσεγγίσεις που εστιάζουν ειδικά σε

αυτά τα ζητήματα. Επιπλέον, εξετάζει τεχνικές βελτιστοποίησης και παραμετροποίησης, και αξιολογεί προτεινόμενες λύσεις ως προς την ακρίβεια και αποδοτικότητα τους, με στόχο την διερεύνηση της πιθανότητας εφαρμογής τους σε πλατφόρμες χαμηλών πόρων, όπως το Raspberry Pi.

Οι κύριοι στόχοι της παρούσας διπλωματικής εργασίας είναι οι εξής:

- Η διερεύνηση και η ανάλυση σύγχρονων μεθόδων ανίχνευσης μικρών αντικειμένων και ιδιαίτερα η κατανόηση των τεχνικών και των προκλήσεων που αντιμετωπίζουν τα νευρωνικά δίκτυα σε περιπτώσεις εντοπισμού μικρής κλίμακας αντικείμενων σε εναέριες εικόνες.
- Η εύρεση και σύγκριση των καταλληλότερων αλγορίθμων για τον εντοπισμό μικρών αντικειμένων σε εικόνες, καθώς και η δυνατότητα εφαρμογής τους σε πραγματικό χρόνο.
- Η ανάπτυξη ενός αποτελεσματικού μοντέλου ή την παραμετροποίηση και βελτιστοποίηση υπάρχοντος που να βελτιώνει την ακρίβεια στην αναγνώριση μικρών αντικειμένων.
- Η διερεύνηση της ικανότητας του προτεινόμενου βελτιωμένου μοντέλου να εφαρμοστεί σε πραγματικό χρόνο, καθώς και η πιθανότητα να έχει χαρακτηριστικά που να το καθιστούν κατάλληλο να τρέξει σε πλατφόρμες περιορισμένων πόρων.
- Η εξέταση της δυνατότητας υλοποίησης ενός μοντέλου ανίχνευσης μικρών αντικειμένων σε υπολογιστικά περιορισμένα περιβάλλοντα (π.χ. Raspberry Pi), αξιολογώντας την ταχύτητα και ακρίβεια αναγνώρισης.
- Η αξιολόγηση και σύγκριση της προτεινόμενης λύσης σύμφωνα με κατάλληλους και κοινά αποδεκτούς δείκτες μέτρησης.

Τα ερευνητικά ερωτήματα που θέτει η παρούσα εργασία συνοψίζονται ως εξής:

- Ποιες είναι οι προκλήσεις στον εντοπισμό μικρών αντικειμένων και ποιες είναι οι στρατηγικές υπέρβασης τους;
- Ποιοι αλγόριθμοι μηχανικής μάθησης είναι πιο αποδοτικοί στον εντοπισμό μικρών αντικειμένων;
- Πως επηρεάζουν την απόδοση του συστήματος οι τύποι και τα χαρακτηριστικά των συνόλων δεδομένων;
- Ποιες είναι οι μελλοντικές κατευθύνσεις στον εντοπισμό μικρών αντικειμένων;
- Είναι δυνατή η αξιόπιστη λειτουργία σύγχρονων μοντέλων μικρού βάρους (lightweight) σε ενσωματωμένες πλατφόρμες περιορισμένων υπολογιστικών πόρων;

Δομή της παρούσας εργασίας:

Η παρούσα εργασία δομείται σε έξι κεφάλαια. Ονομαστικά τα κεφάλαια της εργασίας αποτελούνται από: Εισαγωγή, Βιβλιογραφική Ανασκόπηση, Δεδομένα και προεπεξεργασία, Μεθοδολογία & Αλγόριθμοι Ανίχνευσης, Πειραματική Αξιολόγηση & Αποτελέσματα, Συμπεράσματα & Μελλοντική Εργασία.

Στο πρώτο κεφάλαιο δίνεται η γενική περιγραφή του προβλήματος με ένα σύντομο ιστορικό της Τεχνητής Νοημοσύνης, της Μηχανικής και Βαθιάς Μάθησης. Τέλος, αναλύεται ο σκοπός, οι στόχοι και τα ερευνητικά ερωτήματα καθώς και βασική διάρθρωση που ακολουθεί η παρούσα Διπλωματική Εργασία.

Στο δεύτερο κεφάλαιο παρατίθεται μια βιβλιογραφική ανασκόπηση στις σύγχρονες τεχνικές ανίχνευσης μικρών αντικειμένων με τεχνολογίες όπως, το YOLO, Faster R-CNN, DETR, FPN και RPN. Αναφέρονται επίσης μερικές ακόμα State of the Art τεχνολογίες με μια επισκόπηση παρόμοιων ερευνών και μια συγκριτική αξιολόγηση υπαρχόντων μοντέλων.

Στο τρίτο κεφάλαιο αναλύονται ζητήματα που αφορούν τα δεδομένα και τις τεχνικές προεπεξεργασίας αυτών. Παρουσιάζονται διεξοδικά τα χαρακτηριστικά μερικών συχνά χρησιμοποιούμενων δημόσιων - ανοιχτών συνόλων δεδομένων, τα κριτήρια με τα οποία αυτά επιλέχθηκαν και πραγματοποιείται ανάλυση των αντικειμένων που απεικονίζονται και τα οποία αποτελούν δεδομένα εκπαίδευσης των αλγορίθμων.

Στο τέταρτο κεφάλαιο αναφέρεται η μεθοδολογία που ακολουθείται στην παρούσα εργασία. Αναφέρονται οι αλγόριθμοι που εκπαιδεύτηκαν και οι παραλλαγές τους, καθώς και ο λόγος που επιλέχθηκαν οι συγκεκριμένες τεχνολογίες. Στη συνέχεια ακολουθεί η υλοποίηση και βελτιστοποίηση των επιλεγμένων αλγορίθμων. Τέλος, δίνεται μια παρουσίαση των υπερπαραμέτρων που συμμετείχαν.

Στο πέμπτο κεφάλαιο γίνεται πειραματική αξιολόγηση και παρουσίαση των αποτελεσμάτων που προέκυψαν κατά την επικύρωση των αλγορίθμων. Συγκεκριμένα, αναλύονται μετρικές όπως η mAP και η IoU, η Precision-Recall, η F-Score και η μετρική του FPS για την αξιολόγηση της ταχύτητας. Εντοπίζονται περιπτώσεις False-Positives και False-Negatives. Στη συνέχεια πραγματοποιείται σύγκριση αυτών των μοντέλων, και επισημαίνονται τα πλεονεκτήματα και μειονεκτήματα τους. Προτείνονται, επίσης, πιθανές βελτιώσεις τους. Τέλος, αξιολογείται η εφαρμογή του τελικού μοντέλου σε embedded συστήματα με σκοπό τη διερεύνηση της απόδοσης και τους περιορισμούς της.

Στο έκτο κεφάλαιο παρουσιάζονται τα συμπεράσματα και οι κατευθύνσεις για μελλοντική εργασία. Συνοψίζονται τα κύρια ευρήματα της έρευνας και προτείνονται τρόποι βελτίωσης της εφαρμογής σε προβλήματα πραγματικού κόσμου. Ιδιαίτερη έμφαση δίνεται στην συνεισφορά της ενσωμάτωσης τεχνολογιών όπως το SAHI, οι Transformers και οι τεχνολογίες Hardware Acceleration.

Κεφάλαιο 2ο: Βιβλιογραφική Ανασκόπηση

Η ανίχνευση μικρών αντικειμένων είναι μια από τις δυσκολότερες προκλήσεις στον τομέα της υπολογιστικής όρασης ειδικά αν συνυπολογιστεί ότι πολλές φορές απαιτείται πολύ υψηλή ακρίβεια και ταχύτητα. Για να δοθούν τα καλύτερα δυνατά αποτελέσματα χρησιμοποιούνται οι πιο σύγχρονες τεχνικές ανίχνευσης αντικειμένων που υπάρχουν διαθέσιμες.

Στο παρόν κεφάλαιο, εξετάζονται οι πιο σύγχρονες μέθοδοι που εφαρμόζονται για την ανίχνευση μικρών αντικειμένων με μεθόδους μηχανικής και βαθιάς μάθησης. Αναλύονται θεμελιώδεις και καινοτόμες αρχιτεκτονικές μοντέλων νευρωνικών δικτύων καθώς και οι εξελίξεις τους στο πεδίο της ανίχνευσης μικρών αντικειμένων. Οι μέθοδοι αυτές αξιοποιούν νευρωνικά δίκτυα και τεχνικές επαύξησης δεδομένων με στόχο την βελτίωση της ακρίβειας και της γενίκευσης των μοντέλων. Παρουσιάζονται τόσο βασικές αρχιτεκτονικές όσο και προσεγγίσεις State of the Art που χρησιμοποιούνται από την βιομηχανία και την ερευνητική κοινότητα. Όλοι οι αλγόριθμοι που εξετάζονται ανήκουν στις κατηγορίες της μηχανικής και βαθιάς μάθησης και είναι εκπαιδευμένοι μέσω της επιβλεπόμενης μάθησης.

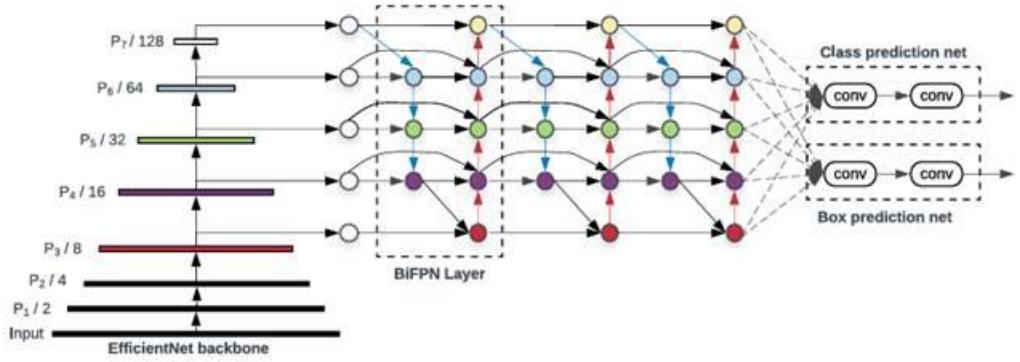
Παρακάτω γίνεται μια ανασκόπηση των βασικότερων μοντέλων και μεθόδων που χρησιμοποιούνται στην ανίχνευση μικρών αντικειμένων με σύγχρονες μεθόδους, αναφέρονται οι προκλήσεις που εντοπίζονται στην εφαρμογή τους και η αποτελεσματικότητα της κάθε μεθόδου.

2.1 Μοντέλα ανίχνευσης One-Stage

Τα μοντέλα one - stage (ενός βήματος) είναι μοντέλα ανίχνευσης αντικειμένων που αποτελούνται από ένα ενιαίο νευρωνικό δίκτυο. Το ενιαίο δίκτυο είναι υπεύθυνο τόσο για την εύρεση των περιοχών (bounding boxes) των αντικειμένων και της κατηγοριοποίησης τους σε ένα πέρασμα της εικόνας. Αυτό τα καθιστά γρήγορα και κατάλληλα για εφαρμογές πραγματικού χρόνου.

2.1.1 EfficientDet

Ο αλγόριθμος EfficientDet [11], που αναπτύχθηκε από τους Tan et al. [11], ερευνητές της Google, είναι μια οικογένεια μοντέλων για ανίχνευση αντικειμένων. Το μοντέλο χρησιμοποιεί ως backbone το EfficientNet και η καινοτομία που εφαρμόζει είναι το BiFPN (Bi-directional Feature Pyramid Network) το οποίο είναι ένα FPN που επιτρέπει την αποδοτικότερη συγχώνευση χαρακτηριστικών από διαφορετικές κλίμακες. Έτσι μπορεί να εντοπίζει καλύτερα αντικείμενα μικρών διαστάσεων με ακρίβεια χωρίς να χρειάζεται πολλούς πόρους.



Εικόνα 2.1: Η αρχιτεκτονική του EfficientDet [11]

Το EfficientNet (Εικόνα 2.1) είναι το backbone δίκτυο, το BiFPN το feature network με ένα κοινό δίκτυο για την πρόβλεψη των κλάσεων και των bounding boxes. Τα επίπεδα του δικτύου BiFPN με αυτά του δικτύου πρόβλεψης κλάσεων και των bounding boxes επαναλαμβάνονται ανάλογα με το μέγεθος της έκδοσης του μοντέλου. Στις δοκιμές που πραγματοποιήθηκαν με το σύνολο δεδομένων COCO παρατηρήθηκε ότι το μοντέλο EfficientDet-D0 είναι το μικρότερο της οικογένειας EfficientDet και έχει την ίδια απόδοση όσο το μοντέλο YOLOv3, ενώ απαιτεί αρκετές φορές μικρότερους υπολογιστικούς πόρους για να τρέξει.

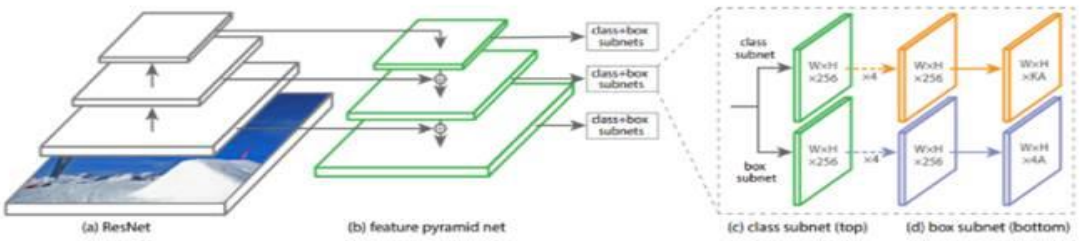
2.1.2 RetinaNet

Το μοντέλο RetinaNet [12] των ερευνητών Lin et al. [12] της Facebook AI Research (FAIR), αποτελεί μια καινοτομία, καθώς η αρχιτεκτονική του κάνει χρήση της τεχνικής Focal Loss.

Αρχικά, η τεχνική Focal Loss παρέχει βοήθεια στους αλγορίθμους single shot όταν υπάρχει μεγάλη ανισοκατανομή αντικειμένων ανάμεσα στο προσκήνιο (foreground) και στο φόντο (background). Όπως αναφέρουν οι συγγραφείς, η τεχνική Focal Loss μαθαίνει στον αλγόριθμο τα «δύσκολα παραδείγματα» (hard examples), τα αντικείμενα εκείνα δηλαδή που εμφανίζονται λίγες φορές. Αυτό το κάνει δίνοντας μεγαλύτερο βάρος σε αυτές τις εμφανίσεις μειώνοντας λίγο την επιρροή των εύκολα ανιχνεύσιμων αντικειμένων.

Το backbone δίκτυο του RetinaNet είναι συνήθως ένα Faster R-CNN οπότε και γίνεται χρήση δικτύου FPN για την διαχείριση αντικειμένων σε διαφορετικές κλίμακες. Ο συνδυασμός της Focal Loss με το FPN επιτρέπει την ανίχνευση αντικειμένων σε ποικίλα μεγέθη και την καλύτερη διαχείριση των μεταβολών της κλίμακας στις εικόνες.

Πιο αναλυτικά, το RetinaNet χωρίζεται σε τέσσερα τμήματα (Εικόνα 2.2). Πρώτο αποτελεί το τμήμα του ResNet που είναι μια bottom-up αρχιτεκτονική η οποία υπολογίζει τα feature maps σε διαφορετικές κλίμακες. Δεύτερο τμήμα του, είναι το δίκτυο FPN που είναι μια top-down αρχιτεκτονική και η οποία εντοπίζει αντικείμενα διαφορετικών μεγεθών, και ειδικά αυτά που εμφανίζονται σπάνια. Τρίτο τμήμα είναι το δίκτυο που κάνει την κατηγοριοποίηση και δίνει την πιθανότητα της ύπαρξης κάποιου αντικείμενου σε κάθε θέση της εικόνας για κάθε anchor box και για κάθε κλάση του ίδιου του αντικειμένου. Τέταρτο τμήμα είναι το Regression Network (υποδίκτυο παλινδρόμησης) που δίνει την ακριβή θέση των anchor boxes μέσω της μετατόπισης τους ώστε αυτά να προσαρμοστούν στα αντικείμενα της εικόνας με ακρίβεια. Για κάθε πραγματικό αντικείμενο υπολογίζονται οι συντεταγμένες που προσαρμόζουν τα anchor boxes ώστε αυτά να ταυτιστούν με τα πραγματικά αντικείμενα [12].



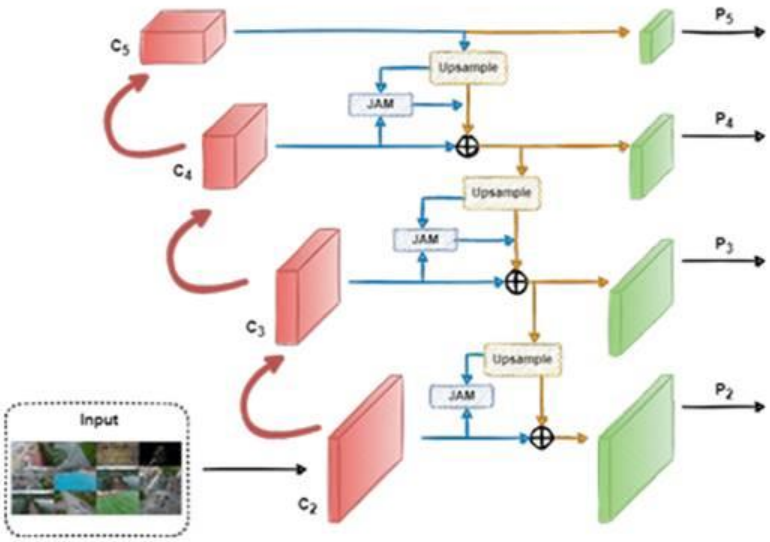
Εικόνα 2.2: Η αρχιτεκτονική του μοντέλου RetinaNet [12]

Το μοντέλο αξιολογήθηκε χρησιμοποιώντας το σύνολο δεδομένων COCO και αποδείχθηκε ότι η απόδοση του RetinaNet (η έκδοση RetinaNet-101-800) εμφανίζει καλύτερη απόδοση κατά 5.9 μονάδες στο AP όταν γίνεται σύγκριση με το μοντέλο single-stage (DSSD), και καλύτερο κατά 2.3 μονάδες AP όταν γίνεται σύγκριση με το two-stage μοντέλο (Faster R-CNN).

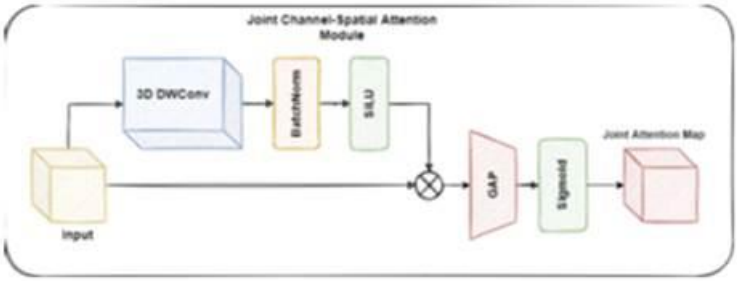
2.1.3 Το μοντέλο Drone - YOLO

Οι Faraji και Chen [13] προτείνουν μια καινούργια αρχιτεκτονική, την Drone - YOLO η οποία βασίζεται στο YOLO και στη χρήση επαύξησης των δεδομένων (data augmentation), προερχόμενων από εικόνες drone. Η μεθοδολογία τους αποτελείται από δύο τμήματα.

Η μεθοδολογία τους χρησιμοποιεί ένα Feature Proposal Network (FPN) με μηχανισμούς προσοχής (attention mechanism) (Εικόνα 2.3). Όπως αναφέρουν, οι περισσότεροι μηχανισμοί προσοχής διαχωρίζουν την χωρική πληροφορία και τα χαρακτηριστικά εξάγοντας διαφορετικά στοιχεία από το κάθε ένα και μετά συνδυάζονται σε ένα ενιαίο αποτέλεσμα. Προτείνουν μια ενιαία ή συνδυασμένη δομή που δεν διαχωρίζει και συνενώνει αλλά ψάχνει εξαρχής και εξάγει την χωρική πληροφορία και τα χαρακτηριστικά τους, διαδικασία την οποία ονομάζουν Joint Attention Module (JAM) [13] (Εικόνα 2.4). Το αποτέλεσμα (που είναι στην ουσία ένας πίνακας) περιέχει αρκετή πληροφορία τόσο στη χωρική πληροφορία τους όσο και στα χαρακτηριστικά τους, συμβάλλοντας στην αύξηση της απόδοσης για την ανίχνευση μικρών αντικειμένων.



Εικόνα 2.3: FPN οδηγούμενο από μηχανισμό προσοχής στη δομή JAM [13]



Εικόνα 2.4: Η δομή του Joint Attention Module (JAM) [13]

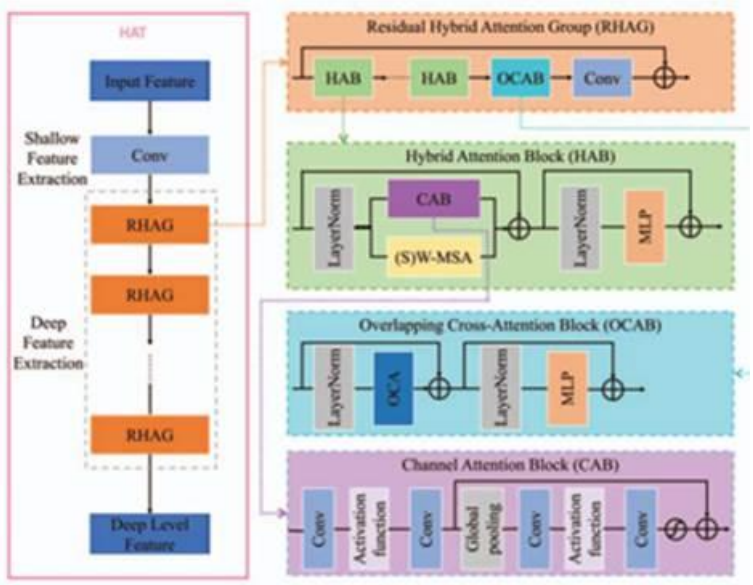
Το δεύτερο σκέλος της μεθοδολογίας τους αφορά στην ανισοκατανομή των δεδομένων ως προς την κλίμακα (scale imbalance). Το πρόβλημα στις εναέριες λήψεις είναι η ύπαρξη πολλών αντικειμένων διαφορετικού μεγέθους. Αυτή η ανομοιομορφία στην κλίμακα επιβαρύνει την απόδοση του μοντέλου στο να αντιληφθεί μικρότερα αντικείμενα. Η λύση που προτείνουν είναι η μέθοδος NN-Resample [13] που βασίζεται στον αλγόριθμο Nearest Neighbors. Η «κλασική» μέθοδος random copy paste ενώ παίρνει αντικείμενα και τα μεταφέρει σε άλλα σημεία με σκοπό την επαύξηση, δεν παίρνει υπόψη της το φόντο ή την κλίμακα αυτών οπότε το αποτέλεσμα μπορεί να είναι μια εικόνα ασύμβατη, από την άποψη ότι αντικείμενα διαφορετικού μεγέθους δεν βρίσκονται στη σωστή θέση ή περιοχή που προβλέπεται να είναι. Η προτεινόμενη μέθοδος παίρνει και επαυξάνει τα αντικείμενα που εμφανίζονται τις λιγότερες φορές και τα αντιγράφει σε θέσεις που ταιριάζουν σε σχέση με το μέγεθος τους αλλά και το μέγεθος των γειτόνων τους.

Δοκιμές που έγιναν στα dataset των MS COCO και VisDrone έδειξαν βελτιωμένα αποτελέσματα. Πιο συγκεκριμένα στο MS COCO η αύξηση στο Average Precision ήταν 1% – 3 % σε σχέση με το YOLOv5. Μεγαλύτερο κέρδος όμως είναι ότι η πρόταση τους δίνει καλύτερο mAP στα μικρά αντικείμενα αποδεικνύοντας έτσι ότι η αρχιτεκτονική τους λειτουργεί και είναι αποτελεσματική. Όσον αφορά τα αποτελέσματα στο VisDrone ήταν καλύτερα κατά 10.01% σε σχέση με το YOLOv5. Σε μια επιπλέον σύγκριση που κάνανε εφαρμόζοντας το JAM στο MobileNet, έγινε κατανοητό ότι ακόμα και σε ένα ελαφρύ μοντέλο μπορούν να πάρουν καλύτερα αποτελέσματα αν ενσωματώσουν τη τεχνική JAM. Τέλος, δοκιμάσανε την μέθοδο επαύξησης NN-Resample. Τα αποτελέσματα έδειξαν ότι υπήρξε βελτίωση του Average Precision σε όλες τις κατηγορίες (mAP, APsmall, APmedium, APlarge). Παράλληλα μετρήθηκε και ο χρόνος εκτέλεσης (runtime) και η αύξηση του ήταν πολύ μικρή σε σχέση με το αποτέλεσμα που δίνει η μέθοδος στην συνολική απόδοση του αλγορίθμου.

2.1.4 Το μοντέλο HATSC - YOLOv10

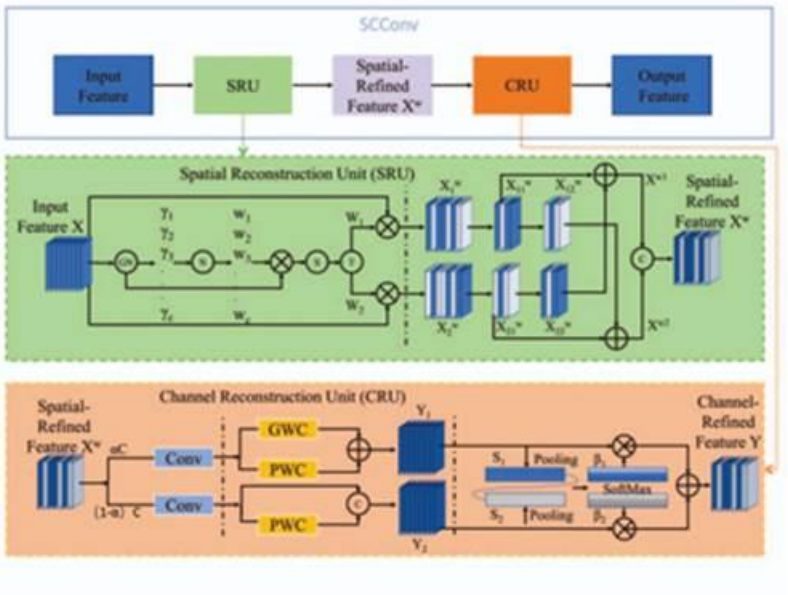
Στη μελέτη των Xie et al. [14] προτείνεται ένα μοντέλο βασισμένο στο YOLOv10 το HATSC – YOLOv10, για τον καλύτερο εντοπισμό μικρών αντικειμένων σε δορυφορικές εικόνες. Η βελτιωμένη αυτή αρχιτεκτονική συνδυάζει έναν Hybrid Attention Transformer (HAT) [14] (Εικόνα 2.5) με την αρχιτεκτονική Spatial and Channel reconstruction Convolution (SCconv) [14] (Εικόνα 2.6). Το στοιχείο HATSC (Εικόνα 2.7) χρησιμοποιεί καλύτερα, κατόπιν επανασχεδίασης του, τις δυνατότητες του HAT [14] και του SCconv για τον καλύτερο και αποδοτικότερο εντοπισμό της χωρικής πληροφορίας και των high level στοιχείων. Επίσης, χρησιμοποιήθηκε η μετρική Normalized Wasserstein Distance (NWD) [14] αντί της Complete Intersection over Union (CIoU) για τον υπολογισμό της ομοιότητας των πλαισίων οριοθέτησης, με σκοπό η αρχιτεκτονική να είναι λιγότερο ευαίσθητη στην χωρική απόκλιση των πλαισίων των μικρών αντικειμένων.

Το HATSC χρησιμοποιείται ως ένα δίκτυο εξαγωγής χαρακτηριστικών, εστιάζοντας τόσο στα ρηχά χαρακτηριστικά όσο και στα βαθιά, με σκοπό την καλύτερη εξαγωγή των λεπτομερών χαρακτηριστικών της εικόνας. Πολλαπλά Hybrid Attention Blocks (HAB) συνθέτουν κάθε ένα από τα Residual Hybrid Attention Group (RHAG) μαζί με ένα Overlapping Cross – Attention Block (OCAB) [14]. Το HAB έχει δομή όμοια με το τυπικό μπλοκ ενός Swin Transformer και ενσωματώνει Channel Attention Block (CAB). Το τελευταίο (CAB) λειτουργεί παράλληλα με το Window – based Multihead Self – Attention block (W-MSA). Το OCAB επαυξάνει τις δυνατότητες του μηχανισμού Window Self – Attention.

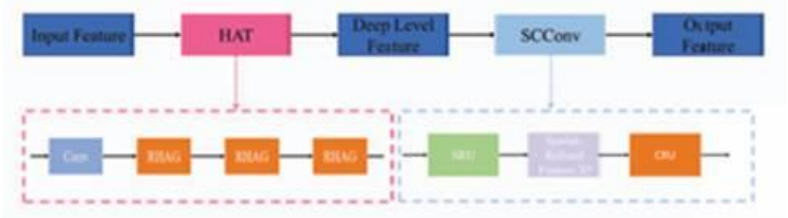


Εικόνα 2.5: Η δομή του δικτύου HAT για την εξαγωγή χαρακτηριστικών [14]

Το δίκτυο SCconv βοηθάει στη μείωση της πλεονάζουσας πληροφορίας από τα εξαγόμενα χαρακτηριστικά και στη βελτιώση της απόδοσης του μοντέλου μέσα από την συμπίεση του. Το δίκτυο λειτουργεί με τον παρακάτω τρόπο: Τα χαρακτηριστικά εισόδου περνάνε από ένα Spatial Reconstruction Unit (SRU) για να βελτιωθούν χωρικά, και έπειτα περνάνε από ένα Channel Reconstruction Unit (CRU) με σκοπό την βελτιστοποίηση των χαρακτηριστικών τους.



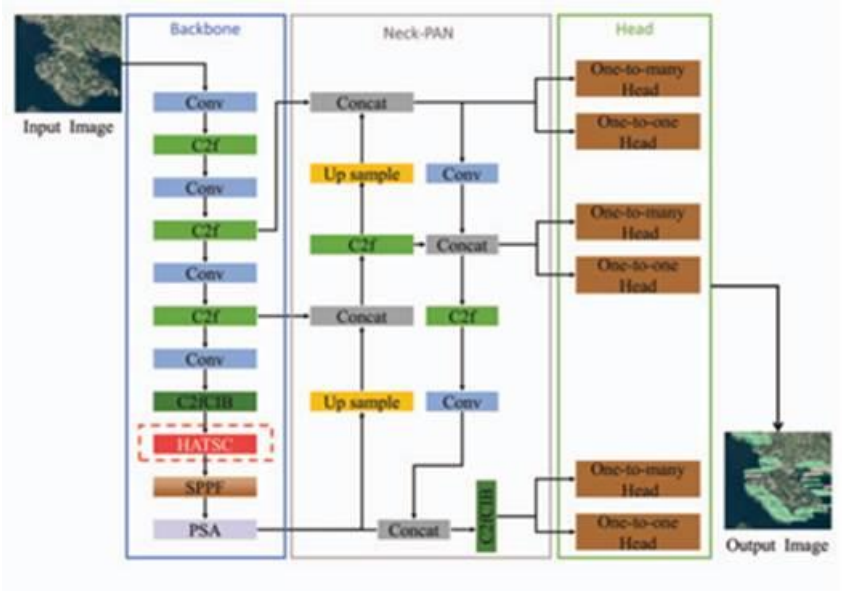
Εικόνα 2.6: Η δομή του δικτύου SCconv για την βελτιστοποίηση των χαρακτηριστικών [14]



Εικόνα 2.7: Η δομή του δικτύου HATSC όπως περιγράφεται στην πρόταση των συγγραφέων [14]

Σε ότι αφορά την συνάρτηση απώλειας (Loss Function) χρησιμοποιείται η μετρική Normalized Wasserstein Distance (NWD) καθώς, όπως αναφέρουν οι συγγραφείς, είναι καλύτερη για εφαρμογές που απαιτούν ανίχνευση μικρών αντικειμένων από δορυφορικές εικόνες. Το αρχικό μοντέλο YOLOv10 χρησιμοποιεί ως μετρική την Complete Intersection over Union (CIoU) για την αξιολόγηση της ομοιότητας των πλαισίων οριοθέτησης και αποτελεί μια βελτίωση της Intersection over Union (IoU). Παρ' όλα αυτά μπορεί να οδηγήσει σε μια μεροληψία ως προς την θέση (positional bias) και να μειώσει την αποτελεσματικότητα και την απόδοση του μοντέλου.

Οι ερευνητές στην πρόταση τους ενσωματώνουν το HATSC στο backbone του YOLOv10 (Εικόνα 2.8) για να βελτιώσουν την εξαγωγή χαρακτηριστικών στην ανίχνευση μικρών αντικειμένων από δορυφορικές εικόνες. Το μοντέλο YOLOv10 έχει ένα βελτιστοποιημένο Cross Stage Partial Network (CSP-Net) [14], για την καλύτερη βελτίωση του gradient flow και του υπολογιστικού κόστους. Στο τέλος της δομής του backbone χρησιμοποιείται η αρχιτεκτονική Spatial Pyramid Pooling Fast module (SPPF) [14] για τον συνδυασμό χαρακτηριστικών διαφόρων κλιμάκων. Προτείνουν, λοιπόν, την ενσωμάτωση του HATSC πριν από την δομή του SPPF module [14] για την ενίσχυση του μοντέλου και την ικανότητα του να εστιάζει σε σημαντικά χαρακτηριστικά και χωρικές περιοχές πριν αυτά συγχωνευτούν.



Εικόνα 2.8: Η δομή του δικτύου HATSC – YOLOv10 [14]

Η αξιολόγηση της πρότασης τους έγινε στο DOTA dataset. Χρησιμοποιήθηκαν οι μετρικές Precision, Recall και τα mean Average Precision (mAP50 και mAP50-95). Οι δοκιμές έδειξαν ότι το Precision βελτιώθηκε κατά 2.91%, το Recall βελτιώθηκε κατά 2.09%, το mAP50 βελτιώθηκε κατά 2.3% και το mAP50-95 βελτιώθηκε κατά 1.68% σε σχέση με το αρχικό.

2.2 Μοντέλα Two-Stage

Τα μοντέλα two - stage διαχωρίζουν τη διαδικασία πρόβλεψης αντικειμένων σε δύο διακριτά στάδια. Στο πρώτο στάδιο βρίσκουν τις προτάσεις περιοχών, δηλαδή περιοχές που είναι πιθανό να περιέχουν αντικείμενα (region proposals), και στο δεύτερο στάδιο γίνεται αξιολόγηση και κατηγοριοποίηση της κάθε πρότασης. Ο συγκεκριμένος τρόπος εργασίας επιτρέπει την αναζήτηση μεγαλύτερης ακρίβειας στις προβλέψεις, υπό τον συμβιβασμό μεγαλύτερου υπολογιστικού κόστους στην εκτέλεση του

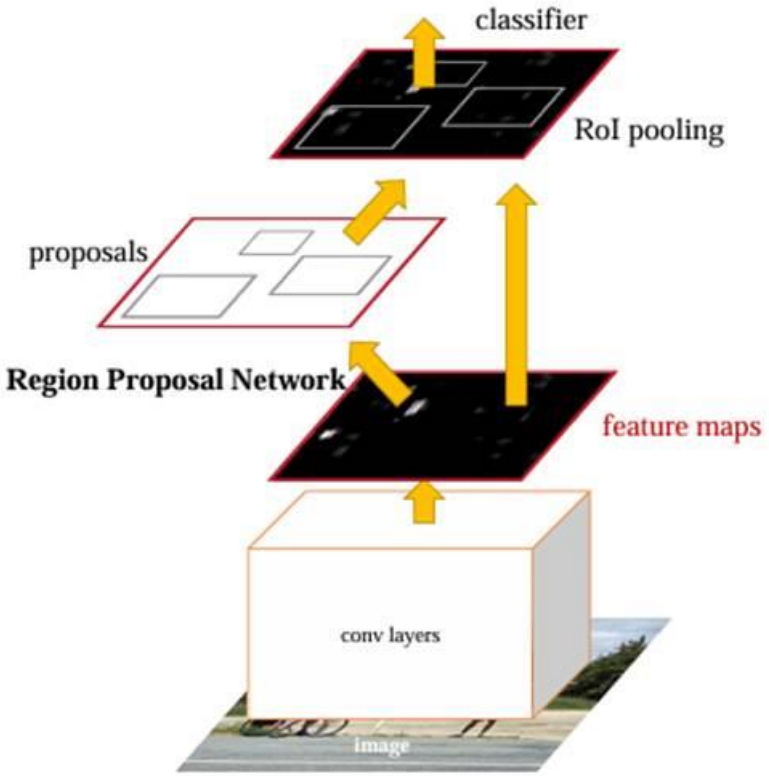
μοντέλου και μικρότερης ταχύτητας εκτέλεσης στην διαδικασία της ανίχνευσης, περιορίζοντας τη δυνατότητα εφαρμογής τους σε προβλήματα πραγματικού χρόνου.

2.2.1 Faster R-CNN

Το μοντέλο Faster R-CNN προτάθηκε από τους Ren et al. [15], είναι ένα μοντέλο ανίχνευσης αντικειμένων και αποτελεί την τρίτη εξέλιξη του αρχικού αλγορίθμου R-CNN (με δεύτερο να αποτελεί τον Fast R-CNN). Ο σκοπός του Faster R-CNN είναι να αυξηθεί η ταχύτητα και η ακρίβεια ανίχνευσης αντικειμένων.

Στον R-CNN ο όγκος εργασιών εκτελείται από τον αλγόριθμο Selective Search για τον εντοπισμό τον προτεινόμενων περιοχών ενδιαφέροντος (region proposals) καθώς και θέσεων των αντικειμένων. Το κυρίως δίκτυο, ένα μεγάλο CNN υπολογίζει και επαναϋπολογίζει τα χαρακτηριστικά (features) των εικόνων στο αρχικό επίπεδο (layer). Ο αλγόριθμος Fast R-CNN βελτίωσε αυτή την διαδικασία εκτελώντας την εξαγωγή των χαρακτηριστικών σε ολόκληρη την εικόνα πριν κάνει την επιλογή των προτεινόμενων περιοχών ελαττώνοντας το υπολογιστικό κόστος.

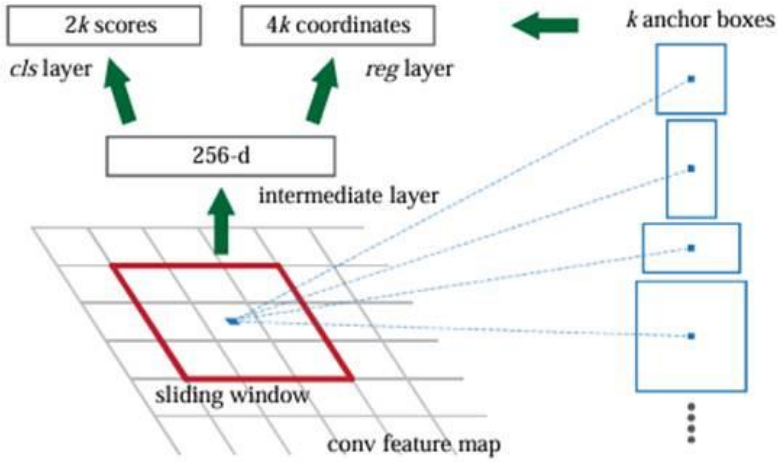
Η αλλαγή που εισάγει ο αλγόριθμος Faster R-CNN είναι η χρήση ενός νευρωνικού δικτύου, του Region Proposal Network (RPN) που αναλαμβάνει τις εργασίες του αλγορίθμου Selective Search. Το RPN είναι ένα αυτόνομο συνελικτικό δίκτυο που προτείνει περιοχές κάνοντας χρήση και αξιοποιώντας τα ίδια feature maps που χρησιμοποιούνται τόσο για την πρόταση των περιοχών όσο και για την τελική ανίχνευση των αντικειμένων κάνοντας το γρηγορότερο και αποδοτικότερο.



Εικόνα 2.9: Το Faster R-CNN ως ένα ενιαίο δίκτυο ανίχνευσης αντικειμένων. Το τμήμα του RPN λειτουργεί ως μηχανισμός προσοχής [15]

Το μοντέλο αποτελείται από δύο τμήματα (Εικόνα 2.9), (α) το πρώτο τμήμα είναι ένα βαθύ πλήρως συνελικτικό δίκτυο (RPN) που εντοπίζει πιθανές περιοχές που περιέχουν αντικείμενα και (β) το δεύτερο τμήμα είναι ο ανιχνευτής του Fast R-CNN που ταξινομεί τις προτεινόμενες περιοχές και ορίζει τα όρια των πλαισίων τους (bounding boxes) [15]. Τα δύο αυτά τμήματα λειτουργούν ως ένα ενιαίο δίκτυο για τον εντοπισμό αντικειμένων. Το νευρωνικό δίκτυο RPN εκτελεί αυτόματα και αποδοτικά την πρόταση περιοχών κάνοντας χρήση του μηχανισμού προσοχής (attention mechanism)˙ όχι όμως με την κλασική πλέον έννοια που χρησιμοποιείτε από τους Transformers αλλά ως ο τρόπος που το δίκτυο εστιάζει σε περιοχές ενδιαφέροντος μέσα στην εικόνα. Το βασικό δίκτυο CNN που κάνει την ανίχνευση (συνήθως το VGG16 ή το ResNet) χρησιμοποιεί αυτή την πληροφορία για να εντοπίσει τις πιθανές περιοχές που περιέχουν αντικείμενα, βελτιώνοντας την απόδοση του σε σχέση με τις προηγούμενες εκδόσεις.

Ένα RPN δίκτυο (Εικόνα 2.10) παίρνει ως είσοδο του μια εικόνα και βγάζει ως έξοδο προτάσεις αντικειμένων με τη μορφή ορθογωνίων πλαισίων όπου το κάθε ορθογώνιο πλαίσιο, έχοντας αντιστοιχισμένο ένα σκορ, συμβολίζει τις κλάσεις «στόχους» που υπάρχουν έναντι του φόντου της εικόνας. Το RPN υλοποιείται με ένα πλήρως συνελικτικό δίκτυο. Για την παραγωγή αυτών των περιοχών ένα μικρό δίκτυο μετακινείται ή «σκανάρει» το τελευταίο εξαγώμενο feature map του τελευταίου επιπέδου (από το κύριο CNN). Το δίκτυο παίρνει ως είσοδο ένα παράθυρο n x n από την είσοδο του feature map. Στη συνέχεια κάθε παράθυρο μετασχηματίζεται σε μικρότερη διάσταση και περνιέται σε δυο δίδυμα πλήρως συνελικτικά επίπεδα, ένα επίπεδο box-regression (reg) και ένα box-classification (cls). Επειδή το μικρό αυτό δίκτυο δουλεύει με τη φιλοσοφία του κυλιόμενου παραθύρου (sliding window) τα πλήρως συνελικτικά επίπεδα είναι κοινά σε όλες τις περιοχές (spatial locations). Η αρχιτεκτονική υποστηρίζεται με ένα nxn συνελικτικό επίπεδο ακολουθούμενο από δύο δίδυμα 1x1 συνελικτικά επίπεδα (ένα για το reg και ένα για το cls). Επιπλέον, η εφαρμογή του Non-Maximum Suppression (NMS) βοηθάει το μοντέλο να μη προτείνει επικαλυπτόμενες περιοχές και περιττές προτάσεις.



Εικόνα 2.10: Το δίκτυο RPN [15]

Άγκυρες

Σε κάθε σημείο του κυλιόμενου παραθύρου ο αλγόριθμος προβλέπει προτάσεις περιοχών, με τον μέγιστο αριθμό των δυνατών προτάσεων για κάθε σημείο να συμβολίζεται με k. Οπότε το επίπεδο (layer) reg έχει 4 x k εξόδους που κωδικοποιούν τις συντεταγμένες των k κουτιών, και το επίπεδο (layer) cls παράγει 2 x k σκορ με την εκτίμηση της πιθανότητας ένα κουτί να είναι αντικείμενο ή όχι. Έτσι, οι k προτάσεις παραμετροποιούνται ανάλογα με τα k κουτιά αναφοράς, τα οποία ονομάζονται

άγκυρες. Η άγκυρα είναι κεντραρισμένη στο κυλιόμενο παράθυρο που εξετάζουμε και συσχετίζεται με την κλίμακα και την αναλογία διαστάσεων. Αυτό επιτρέπει το RPN να ανιχνεύει αντικείμενα διαφόρων μεγεθών και σχημάτων. Οι προβλέψεις που γίνονται από το RPN περιλαμβάνουν offsets που προσαρμόζουν τα πλαίσια ώστε να ταιριάζουν καλύτερα στα πραγματικά αντικείμενα της εικόνας.

Translation-Invariant Anchors

Μια ακόμα σημαντική ιδιότητα της λύσης (σε σχέση με μια «κλασική λύση multibox») που προτείνουν είναι ότι οι άγκυρες και οι συναρτήσεις, που υπολογίζουν τις προτάσεις σε σχέση με τις άγκυρες, είναι αναλλοίωτες από μετάφραση [15]. Εάν μεταφραστεί ένα αντικείμενο σε μια εικόνα, απαιτείται να μεταφραστεί και η πρόταση, και η συνάρτηση να προβλέπει την πρόταση σε κάθε θέση. Η ιδιότητα αυτή επιτρέπει στο σύστημα να βρει ένα αντικείμενο σε οποιαδήποτε θέση. Για παράδειγμα, αν ένα αυτοκίνητο εμφανίζεται στο αριστερό μέρος μιας εικόνας και μετά μετακινηθεί στο δεξί μέρος της ίδιας εικόνας, το σύστημα είναι ικανό να το εντοπίσει. Ένα multibox σύστημα δεν παρουσιάζει την ίδια ικανότητα.

Multi-Scale Anchors as Regression References

Η μέθοδος που προτείνεται ταξινομεί και παλινδρομεί τα πλαίσια οριοθέτησης χρησιμοποιώντας κουτιά άγκυρας πολλαπλών κλιμάκων και αναλογιών διαστάσεων. Η μέθοδος στηρίζεται σε feature maps μιας κλίμακας και χρησιμοποιεί φίλτρα σταθερού μεγέθους, όπως των κυλιόμενων παραθύρων (στο feature map). Εξαιτίας του σχεδιασμού της μεθόδου με άγκυρες πολλαπλών κλιμάκων, μπορεί να γίνει χρήση συνελικτικών χαρακτηριστικών που υπολογίζονται σε εικόνα μιας κλίμακας, εξαλείφοντας την ανάγκη για επεξεργασία σε πολλαπλές κλίμακες.

Οι άγκυρες πολλαπλής κλίμακας επιτρέπουν, επίσης, τον διαμοιρασμό χαρακτηριστικών χωρίς επιπλέον κόστος για την αντιμετώπιση διαφορετικών κλιμάκων (μεγεθών των αντικειμένων). Σαν μέθοδος είναι πιο αποδοτική για την εύρεση αντικειμένων διαφορετικών διαστάσεων. Αντί να μεταβάλλεται το μέγεθος της εικόνας αρκετές φορές ή να χρησιμοποιούνται κυλιόμενα παράθυρα πολλαπλών διαστάσεων, αξιοποιούνται «κουτιά» άγκυρες προκαθορισμένων μεγεθών, ποικίλων όμως διαστάσεων και αναλογιών, επιτρέποντας την δυναμική προσαρμογή στην εικόνα και, κατά συνεπεία, υψηλότερη ταχύτητα.

Οι προτεινόμενες περιοχές από το RPN μετατρέπονται σε σταθερού μεγέθους μέσω ROI Pooling / ROI Align με το detection head να εκτελεί μια ταξινόμηση (classification) με softmax και bounding box regression για να κατηγοριοποιήσει και να οριοθετήσει τα αντικείμενα στην εικόνα (προσδιορίζοντας και τα πλαίσια του αντικειμένου). Το μοντέλο Faster R-CNN εκπαιδεύεται σε δυο στάδια. Αρχικά εκπαιδεύεται το RPN στη παραγωγή προτάσεων περιοχών και μετά εκπαιδεύεται το Fast R-CNN που χρησιμοποιεί τις περιοχές του RPN για να ανιχνεύει τα αντικείμενα και κάνει την ταξινόμηση. Οι δυο εκπαιδεύσεις τρέχουν ως ένα ενιαίο μοντέλο.

Δίκτυο RPN

Κατά τη μελέτη των Nguyen et al. [16] τονίζεται η αξία της βελτίωσης που έφερε η μέθοδος με τον αλγόριθμο Region Proposal Network (RPN) και ο οποίος είναι ικανός να παράγει πολύ καλύτερες ποιοτικά προτάσεις περιοχών. Όπως αναλύουν οι Nhat-Duy Nguyen et. al. [16] από τα πρώτα συνελικτικά επίπεδα εξάγονται βαθιά χαρακτηριστικά της εικόνας και πάνω σε αυτά ο RPN μετακινεί ένα παράθυρο για να εξάγει χαρακτηριστικά για κάθε περιοχή. Ο αλγόριθμος RPN δίνει ταυτόχρονα

και τις προτεινόμενες περιοχές και την βαθμολογία (objectness score) της κάθε προτεινόμενης περιοχής.

Το δίκτυο RPN παίρνει ως είσοδο το feature map από το πέμπτο συνελικτικό επίπεδο και εφαρμόζει ένα κυλιόμενο παράθυρο μεγέθους 3 x 3. Μετέπειτα, στο ενδιάμεσο επίπεδο τα δεδομένα οδηγούνται σε δυο παράλληλα κλαδιά, στο ένα τα δεδομένα δίνονται ως είσοδος για να υπολογιστεί η βαθμολογία του κάθε αντικειμένου και στο άλλο η παλινδρόμηση του που καθορίζει το πως θα πρέπει να προσαρμοστεί το σχήμα του πλαισίου οριοθέτησης.

Ο αλγόριθμος RPN αυξάνει την ακρίβεια και μειώνει τον χρόνο εκτέλεσης ενώ ταυτόχρονα αποφεύγει να κάνει υπερβολικό αριθμό προτάσεων αξιοποιώντας την κοινή χρήση των συνελικτικών χαρακτηριστικών [17]. Το βασικό μειονέκτημα του αλγόριθμου Faster R-CNN είναι ότι αν και είναι βελτιωμένος σε σχέση με τις προηγούμενες εκδόσεις παραμένει αργός για εφαρμογές πραγματικού χρόνου, κυρίως επειδή είναι ένας αλγόριθμος δυο σταδίων (two stage detector) [17].

Η λύση της εύρεσης αντικειμένων και ειδικότερα μικρών βασιζόμενη σε CNN έχει μειονεκτήματα και περιορισμούς. Τα περισσότερα μοντέλα CNN σχεδιάζονται ιεραρχικά από διαφορετικά επίπεδα, όπως για παράδειγμα συνελικτικά (convolutional), υποδειγματοληψίας (pooling layers) και πλήρως συνδεδεμένα (fully connected), που τοποθετούνται διαδοχικά όπου ορισμένα από αυτά έχουν συγκεκριμένη σειρά. Όλα αυτά τα επίπεδα συνθέτουν τον ανιχνευτή (detector) ο οποίος εξάγει τα σημαντικότερα χαρακτηριστικά από τις εικόνες προς ταξινόμηση.

Υπάρχει παραπάνω δυσκολία εντοπισμού μικρών αντικειμένων στα CNN. Η διαδικασία και η δομή των μοντέλων CNN όπου η πληροφορία περνάει από πολλά διαδοχικά επίπεδα κάνει τον εντοπισμό δυσκολότερο, καθώς τα μικρά αντικείμενα έχουν πολύ μικρή επιφάνεια αναπαράστασης και τα χαρακτηριστικά τους είναι λιγότερο εμφανή. Η διαδικασία που προαναφέρθηκε για την δομή του δικτύου και την πληροφορία να περνάει από πολλά επάλληλα επίπεδα έχει επίπτωση, γιατί η εικόνα μετασχηματίζεται και αλλάζει μέγεθος πολλές φορές. Αν ο εντοπισμός αφορά αντικείμενα μεσαίου ή μεγάλου μεγέθους το πρόβλημα του μετασχηματισμού των εικόνων δεν είναι τόσο έντονο στα αντικείμενα αυτά, καθώς αντικείμενα μεγαλύτερου μεγέθους είναι πιο ανεκτικά λόγω του ότι διαθέτουν περισσότερη πληροφορία και τα χαρακτηριστικά τους είναι πιο έντονα.

Αυτό το πρόβλημα λύνεται εν μέρη από τα δίκτυα GAN που αν και χρησιμοποιούν τα ίδια επίπεδα, έχουν λίγο καλύτερη προσέγγιση, επιτρέπουν την παραγωγή παραπάνω δεδομένων και διευκολύνουν στην επίλυση του προβλήματος της περιορισμένης πληροφορίας και της ανισορροπίας δεδομένων κατά την εκπαίδευση [16]. Τα δίκτυα GAN μπορούν να χρησιμοποιηθούν για την υποστήριξη και ενίσχυση των CNN δικτύων ειδικά σε προβλήματα εντοπισμού μικρών αντικειμένων. Στο πλαίσιο της παρούσας διπλωματικής εργασίας, τα δίκτυα GAN δεν εξετάζονται περαιτέρω.

Απόδοση

Αρχικά, υλοποιήθηκαν αρκετά ablation studies για να αξιολογηθεί η απόδοση του μοντέλου με διαφορετικές παραμέτρους. Με την αντικατάσταση του Selective Search με 300 RPN προτάσεις το ποσοστό του mAP έφτασε στο 56,8%. Επιπλέον, με την χρήση των 100 καλύτερων προτάσεων το mAP βρέθηκε στο 55,1% και στον αντίποδα χρησιμοποιώντας τις καλύτερες 6000 προτάσεις, αλλά χωρίς την χρήση Non-Maximum Suppression (NMS), το mAP ανέβηκε στο 55,2%. Η ανάλυση IoU έδωσε των βέλτιστο αριθμό προτάσεων (300), στο dataset PASCAL VOC 2012.

2.3 Μοντέλα βασισμένα σε Transformers

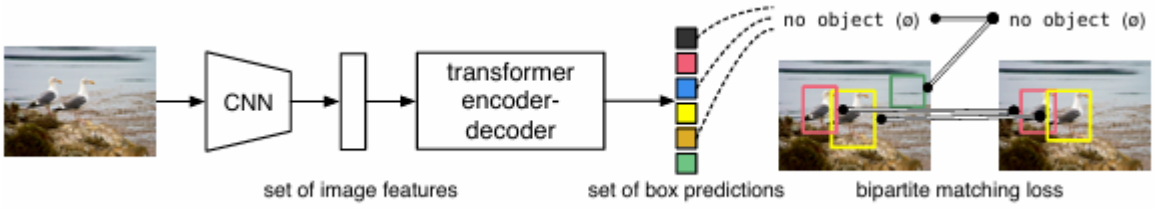
Τα μοντέλα ανίχνευσης αντικειμένων που βασίζονται σε transformers φέρνουν μια νέα προσέγγιση στην μηχανική όραση εισάγοντας τον μηχανισμό του self - attention. Η αρχιτεκτονική παρουσιάστηκε από τους Vaswani et al. [18] στην εργασία τους με τίτλο “Attention is All You Need” και περιγράφει μια μέθοδο, που έκτοτε καθιερώθηκε, για εργασίες ακολουθιών (sequence to sequence) για την επεξεργασία φυσικής γλώσσας. Η αρχιτεκτονική των transformers βασισμένη στην προσοχή (attention) τους επιτρέπει να επεξεργάζονται μεγάλο πλήθος δεδομένων (ακολουθίες) ταυτόχρονα και με αυτό τον τρόπο να αντιλαμβάνονται αποτελεσματικά συσχετίσεις μεγάλης εμβέλειας.

2.3.1 Αλγόριθμος DETR και Drone - DETR

Γενικά στοιχεία για τον αλγόριθμο DETR

Ο αλγόριθμος DETR είναι ένας αλγόριθμος ανίχνευσης αντικειμένων πάνω σε αρχιτεκτονική transformer. Δεν χρειάζεται να χρησιμοποιήσει Non-Maximum Suppression (NMS) ή άγκυρες (anchors). Δεν απαιτεί τη χρήση ειδικών βιβλιοθηκών όπως άλλα μοντέλα και παρουσιάζει καλή απόδοση και ταχύτητα. Η αρχιτεκτονική του είναι αυτή ενός encoder-decoder βασισμένη σε transformer.

Η δομή του αποτελείται από τρία βασικά μέρη (Εικόνα 2.11): το CNN ως το δίκτυο κορμού (CNN backbone), το οποίο πολλές φορές είναι ένα ResNet, που εξάγει και δίνει το feature representation, τον encoder-decoder transformer όπου ο encoder αναλαμβάνει να μετατρέψει τον χάρτη χαρακτηριστικών (feature map) που προκύπτει από το CNN backbone, τον decoder που αναλαμβάνει να παράξει ένα σύνολο προβλέψεων των αντικειμένων με βάση τις αναπαραστάσεις που έδωσε ο encoder και τέλος, το τρίτο τμήμα, το feed forward network (FFN) που παίρνει την έξοδο του decoder και δίνει την πρόβλεψη μαζί με τα πλαίσια οριοθέτησης και τις ετικέτες της κλάσης του κάθε αντικειμένου που έχει ανιχνευτεί. Η υλοποίηση του μπορεί να γίνει σε οποιοδήποτε deep learning framework που λειτουργεί με CNN και transformers, όπως για παράδειγμα το PyTorch ή το TensorFlow.



Εικόνα 2.11: Ο αλγόριθμος DETR κάνει απευθείας προβλέψεις με την χρήση ενός CNN δικτύου και μια αρχιτεκτονική Transformer. Κατά την διάρκεια της εκπαίδευσης αντιστοιχίζονται μοναδικά οι προβλέψεις, τα αντικείμενα, με τα πλαίσια. Όσα αντικείμενα δεν έχουν αντιστοίχιση παίρνουν την τιμή της κλάσης “no object” [19].

Ο αλγόριθμος DETR (DEtection TRansformer) χρησιμοποιεί τον συνδυασμό των τεχνικών bipartite matching loss και βασίζεται σε έναν αλγόριθμο αντιστοίχισης ανάμεσα στα προβλεπόμενα και πραγματικά αντικείμενα και στους transformers με μη αυτοπαλλιδρομική παράλληλη αποκωδικοποίηση. Η τεχνική bipartite matching αναλαμβάνει να βρει τη καλύτερη αντιστοίχιση ανάμεσα στα αντικείμενα που προβλέφθηκαν και στα ground truth αντικείμενα πάνω σε βαθμολογίες ομοιότητας (similarity score). Οι βαθμολογίες ομοιότητας βγαίνουν από το Intersection over Union

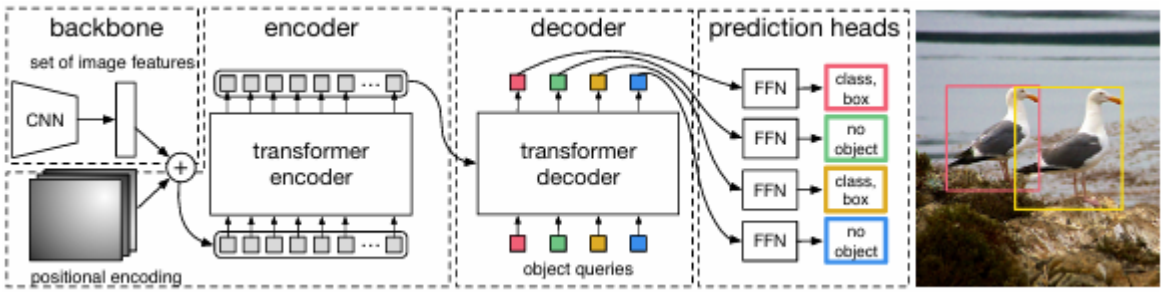
(IoU) των κουτιών (bounty boxes) που έχουν προβλεφθεί και, των κουτιών (bounty boxes) που είναι αληθή (ground truth). Με τη χρήση λοιπόν bipartite matching, κάθε ένα αντικείμενο που προκύπτει από πρόβλεψη, διαθέτει τουλάχιστον ένα μοναδικό αληθές (ground truth) αντικείμενο και το αντίστροφο. Εδώ να τονιστεί ότι επειδή η αρχιτεκτονική transformer επιτρέπει τη παράλληλη αποκωδικοποίηση, το μοντέλο παράγει ταυτόχρονα όλες τις προβλέψεις και η μη αυτοπαλινδρομική λειτουργία του transformer επιτρέπει την επεξεργασία όλων των ανιχνεύσεων παράλληλα και όχι σειριακά.

Το πρόβλημα με αυτή την τεχνική είναι η βελτιστοποίηση της συνάρτησης που χρησιμοποιεί για να βρει τη βέλτιστη αντιστοίχιση των προβλεπόμενων αντικειμένων και η οποία μετά χρησιμοποιείται για την έξοδο του τελικού συνόλου προβλέψεων του μοντέλου. Η διαδικασία αυτή είναι υπολογιστικά ακριβή και η πολυπλοκότητα αυξάνεται εκθετικά όσο αυξάνονται και τα αντικείμενα στην εικόνα.

Το μοντέλο για να μπορέσει να πραγματοποιήσει προβλέψεις χρειάζεται δύο πολύ βασικά στοιχεία. Πρώτον, χρειάζεται μια συνάρτηση απώλειας που εξαναγκάζει μια μοναδική αντιστοίχιση μεταξύ των προβλεπόμενων και των πραγματικών πλαισίων. Δεύτερον, χρειάζεται μια αρχιτεκτονική που προβλέπει σε ένα πέρασμα ένα σύνολο αντικειμένων και, μοντελοποιεί τις σχέσεις τους. Αυτό προκύπτει και από την αρχιτεκτονική των transformers που με τον μηχανισμό της προσοχής (attention mechanism) κατανοεί και βελτιώνει των χωρικό συσχετισμό ανάμεσα στα αντικείμενα.

Ο αλγόριθμος αρχικά αποφασίζει ένα σταθερό σύνολο προβλέψεων (N) με ένα μόνο πέρασμα μέσα από τον αποκωδικοποιητή όπου το N έχει οριστεί να είναι αρκετά μεγαλύτερο από τον μέσο αριθμό των αντικειμένων που βρίσκονται μέσα σε μια εικόνα. Για κάθε μια πρόβλεψη ο αλγόριθμος δίνει την κλάση του αντικειμένου και τη θέση του με τη μορφή πλαισίου. Μια πρόκληση στην εκπαίδευση του μοντέλου αποτελεί η σωστή εύρεση των αντικειμένων τόσο ως προς την κλάση, το είδος δηλαδή του αντικειμένου αλλά και ως προς τη θέση και το μέγεθος. Η διαδικασία της εύρεσης αντιστοίχισης έχει τον ίδιο ρόλο με τους ευρετικούς κανόνες που χρησιμοποιούνται για την εύρεση προτάσεων ή αγκυρών σε αληθή αντικείμενα (ground truth) στους ανιχνευτές. Η κύρια διαφορά έγκειται στο ότι πρέπει να βρεθεί αντιστοίχιση για την άμεση πρόβλεψη ένα προς ένα χωρίς διπλότυπα. Στη συνέχεια υπολογίζει τη loss function χρησιμοποιώντας την συνάρτηση Hungarian loss [20], [21]. Η απώλεια (loss) ορίζεται όπως και οι απώλειες άλλων κοινών ανιχνευτών. Αυτό γίνεται με ένα γραμμικό συνδυασμό μιας πιθανοφάνειας κλάσης με τη χρήση ενός αρνητικού λογαρίθμου και μιας απώλειας κουτιού ή πλαισίου [20], [21]. Ο τρόπος εκπαίδευσης στον DETR εξισορροπεί τις προτάσεις με υποδειγματοληψία.

Ο DETR χρησιμοποιεί αρχικά ένα δίκτυο CNN στο backbone και στέλνει την έξοδο του σε έναν encoder. Ακολούθως, ο decoder παίρνει έναν μικρό αριθμό από γνωστά embeddings, τα οποία στην ουσία είναι τα object queries, και τα εφαρμόζει στην έξοδο του encoder (Εικόνα 2.12). Τέλος, περνάει τις εξόδους από τα embeddings του decoder σε ένα feed forward network που κάνει τις τελικές προβλέψεις οι οποίες είναι κλάση και πλαίσιο οριοθέτησης για κάθε αντικείμενο.



Εικόνα 2.12: Η δομή του αλγορίθμου [19]

Η απώλεια των οριοθετημένων πλαισίων είναι το δεύτερο μέρος του κόστους αντιστοίχισης. Αντίθετα

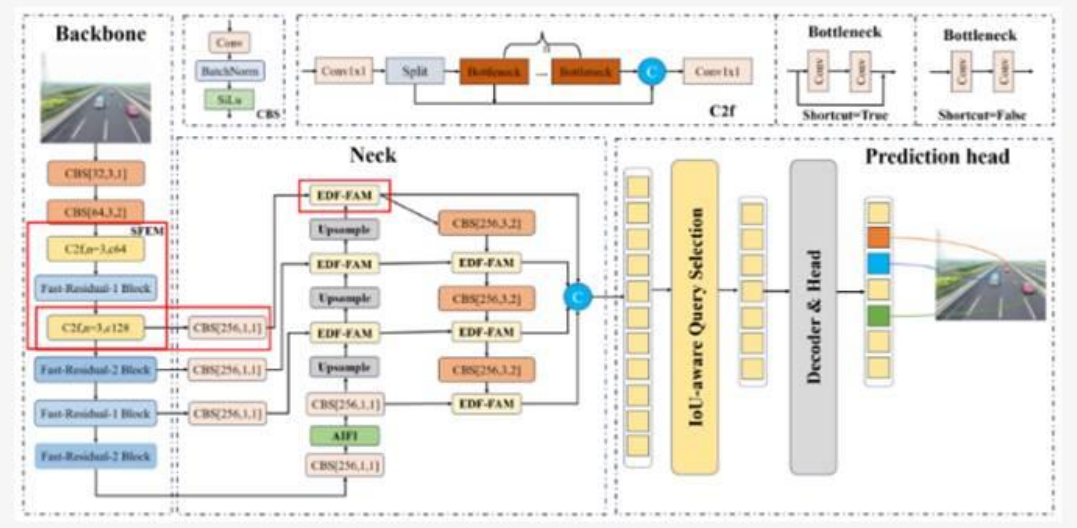
με άλλους ανιχνευτές που προβλέπουν τα πλαίσια αρχικά με κάποιες υποθέσεις (π.χ. anchor boxes), εδώ οι προβλέψεις πραγματοποιούνται απευθείας (υπολογίζονται δηλαδή απευθείας οι συντεταγμένες του πλαισίου). Αν και αυτό (η άμεση πρόβλεψη) απλοποιεί την υλοποίηση ενέχει τον κίνδυνο που έχει σχέση με την κλιμάκωση της απώλειας [20], [21]. Η συχνότερη χρησιμοποιούμενη απώλεια l1(mean absolute error) έχει διαφορετικές κλίμακες για μικρά και μεγάλα πλαίσια ακόμα και αν τα σφάλματα του είναι όμοια. Η τιμή της απώλειας l1 επηρεάζεται από το μέγεθος των πλαισίων [20], [21]. Ένα, δηλαδή, σφάλμα σε μεγάλο πλαίσιο μπορεί να έχει παρόμοια απώλεια με ένα μεγαλύτερο σχετικό σφάλμα σε ένα μικρό πλαίσιο.

Για τον έλεγχο του προβλήματος το DETR συνδυάζει την απώλεια l1 με την απώλεια Intersection over Union (IoU). Συγκεκριμένα χρησιμοποιεί τη Generalized IoU (GIoU) που δεν επηρεάζεται από την κλίμακα (scale invariant). Η απώλεια IoU (IoU loss) είναι αμετάβλητη στην επικάλυψη μεταξύ των πλαισίων των αντικειμένων οπότε σαν μετρική δεν επηρεάζεται από το μέγεθος των πλαισίων. Επομένως, η συγκεκριμένη συνάρτηση απώλειας δεν επηρεάζεται σε μεγάλο βαθμό όπως άλλες συναρτήσεις από το μέγεθος των αντικειμένων που ανιχνεύονται. Συνδυάζοντας τις δυο απώλειες υπάρχει πιο σταθερή εκπαίδευση και καλύτερη ακρίβεια ανίχνευσης.

2.3.2 Drone - DETR

Στο άρθρο τους οι Yaning et al. παρουσιάζουν το μοντέλο Drone – DETR [22] (Εικόνα 2.13) ένα μοντέλο πραγματικού χρόνου transformer (RT – DETR). Ο αλγόριθμος DETR χρησιμοποιεί backbone δίκτυα όπως το ResNet50/101 για την εξαγωγή των χαρακτηριστικών και αυτά τα χαρακτηριστικά περνάνε μέσα από ένα encoder που παρέχει τη θέση του αντικειμένου, ενώ μέσα από τον decoder παρέχει την ταξινόμηση, τη κλάση δηλαδή που ανήκει το κάθε αντικείμενο.

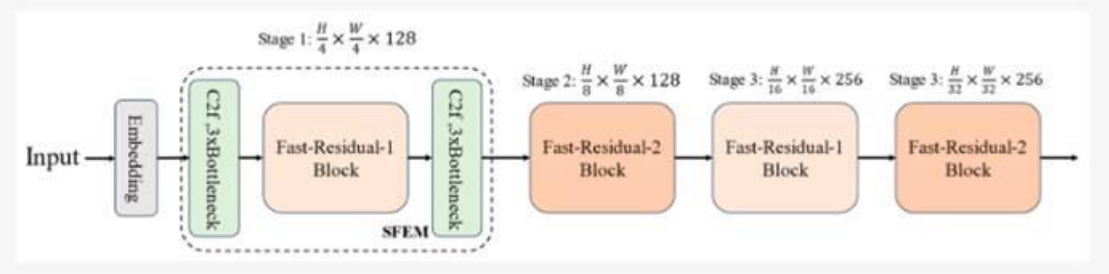
Στην ίδια μελέτη αναφέρουν επίσης ότι το DETR είναι ένα μοντέλο end – to – end training και μπορεί να δώσει πολύ καλά αποτελέσματα χωρίς να χρειάζεται post – processing όπως για παράδειγμα Non – Maximum Suppression (NMS) και αυτό το πετυχαίνει μέσα από τον μηχανισμό προσοχής (self – attention mechanism). Η αρχιτεκτονική όμως έχει δύο μεγάλα μειονεκτήματα. Αρχικά, απαιτεί πολύ μεγάλο χρόνο για την εκπαίδευση του (long training cycles). Επιπλέον, παρατηρείται το φαινόμενο της αργής σύγκλισης, η διαδικασία δηλαδή που ένα μοντέλο δε βελτιώνεται άλλο ή το σφάλμα δε φτάνει σε ένα σταθερό νούμερο. Μια ακόμα παρατήρηση που δίνεται είναι ότι δεν είναι αποδοτικό για την ανίχνευση μικρών αντικειμένων.



Εικόνα 2.13: Η δομή του αλγορίθμου Drone – DETR [22]

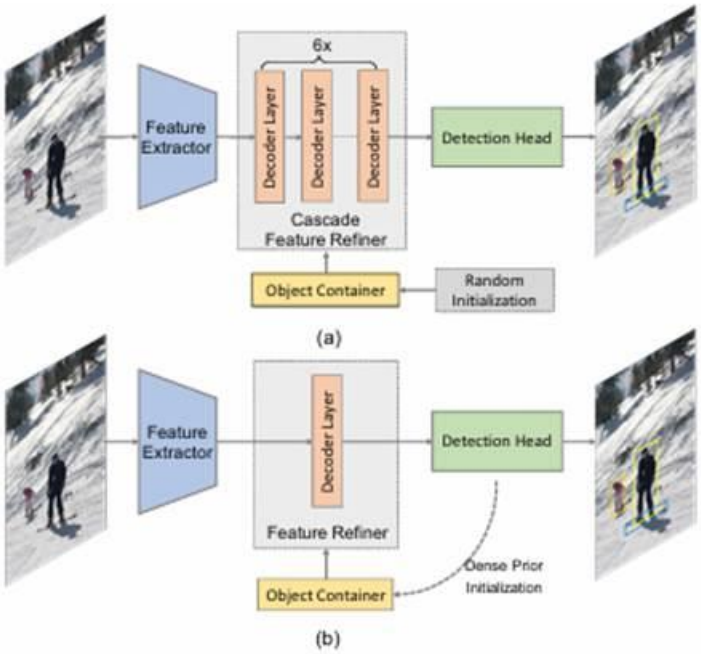
Για την επίλυση αυτών των προβλημάτων αναπτύχθηκε μια αρχιτεκτονική, η Drone – DETR, που στόχος της είναι να βοηθήσει στον εντοπισμό μικρών αντικειμένων. Το νέο μοντέλο δοκιμάστηκε στο σύνολο δεδομένων VisDrone2019, το οποίο αναλύεται στο 3ο κεφάλαιο. Συνοπτικά, το VisDrone2019 αποτελεί ένα σύνολο εικόνων που είναι αρκετά απαιτητικό καθώς έχει αρκετές κατηγορίες όπου τα αντικείμενα έχουν πολύ μικρές διαστάσεις στην εικόνα. Τα αποτελέσματα έδειξαν ότι το μοντέλο (Drone – DETR) συγκλίνει γρηγορότερα σε σχέση με το αρχικό μοντέλο κάνοντας χρήση του VisDrone2019. Σημειώνουν, επίσης, ότι το μοντέλο πέτυχε σύγκλιση μέσα σε 25 εποχές, ενώ στις 100 εποχές η μετρική mAP@50 συνέχισε να αυξάνεται με ρυθμό καλύτερο σε σχέση με το αρχικό μοντέλο.

Δοκιμές έγιναν και με το σύνολο δεδομένων DOTA (το οποίο επίσης αναλύεται στο 3ο κεφάλαιο) και έδειξαν ότι το Drone – DETR παρουσιάζει καλύτερα αποτελέσματα σε σχέση με τον αρχικό αλγόριθμο. Η βελτίωση οφείλεται εν μέρη στο δίκτυο ESDNet [22] (Εικόνα 2.14) το οποίο ενισχύει την ικανότητα ανίχνευσης και διατηρεί τα χαρακτηριστικά που το κάνουν κατάλληλο για εφαρμογές πραγματικού χρόνου.



Εικόνα 2.14: Η δομή του ESDNet [22]

2.3.3 Ο αλγόριθμος Efficient DETR



Εικόνα 2.15: Επάνω (α) μια τυπική δομή και κάτω (β) η πρόταση του νέου μοντέλου [23]

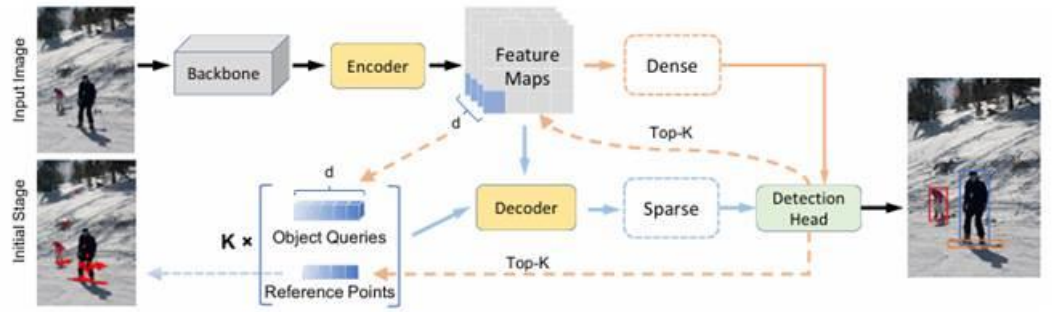
Η μελέτη των Yao et al. [23] προτείνει μια παραλλαγή του DETR, το Efficient DETR. Το μοντέλο αυτό παίρνει χαρακτηριστικά των dense detection και sparse detection. Πρωτίστως, αναφέρεται ότι ο encoder

έχει τη μορφή multi head self-attention με feed forward network και ο decoder έχει παρόμοια αρχιτεκτονική με τον encoder (αλλά είναι πιο ελαφριά καθώς έχει μόνο ένα επίπεδο σε σχέση με τα έξι επίπεδα του αρχικού αλγορίθμου DETR)˙ και τα δύο μέρη έχουν αρχιτεκτονική cascade (Εικόνα 2.15). Μια διαφορά του encoder από τον decoder είναι ότι ο δεύτερος δεν χρησιμοποιεί Multi – head Self – attention. Όπως αναφέρουν, από δοκιμές που διεξήχθησαν, φαίνεται ότι ο DETR είναι πιο ευαίσθητος στον αριθμό των επιπέδων του decoder και έτσι θεωρούμε ότι ο decoder είναι πιο σημαντικός.

Ο Efficient DETR [23] έχει μια διαφοροποιημένη δομή με τρία επίπεδα στον encoder και ένα επίπεδο στον decoder, σε αντίθεση με τα τρία επίπεδα στον decoder στον αρχικό DETR. Η νέα πρόταση έχει δύο μέρη, το πυκνό (dense) και το αραιό (sparse).

Το πυκνό τμήμα, και το οποίο κάνει τις προβλέψεις, περιλαμβάνει το backbone (ένα ResNet), τον encoder και το detection head (Εικόνα 2.16). Στο αραιό τμήμα εισάγονται τα object containers (που είναι τα object queries και τα reference points) και αρχικοποιούνται τα object queries. Στο πυκνό τμήμα δημιουργούνται οι άγκυρες για κάθε μια θέση στα feature maps, τα οποία είναι multi – scale. Εδώ βρίσκεται και το detection head που προμηθεύει την κατηγοριοποίηση και την βαθμολογία της.

Το αραιό τμήμα δέχεται ως εισαγωγή την έξοδο από το πυκνό τμήμα και επιτρέπει τη επιλογή ενός τμήματος της με βάση τη βαθμολογία (objectness score) που αντιπροσωπεύει την πιθανότητα μια περιοχή να έχει αντικείμενο και όχι φόντο. Επίσης, εισάγονται τα object queries που υπάρχουν στα feature maps από τον encoder. Το αραιό τμήμα επιτελεί επί της ουσίας ότι και το πυκνό, με την διαφορά στο μέγεθος των εξόδων τους (output size). Σε ότι αφορά την απώλεια και τα δυο τμήματα χρησιμοποιούν την ίδια συνάρτηση απώλειας και πιο συγκεκριμένα την Lcls η οποία χρησιμοποιείται για το focal loss της κατηγοριοποίησης και τις L1 και Lgiou για το localization.



Εικόνα 2.16: Αναπαράσταση της αρχιτεκτονικής. Οι άγκυρες με το καλύτερο σκορ από το πυκνό δίκτυο χρησιμοποιούνται για την επιλογή των object queries από τον encoder και επιλέγονται τα σημεία αναφοράς από τις προτάσεις περιοχών. [23]

Δοκιμές του αλγορίθμου έγιναν στο σύνολο δεδομένων MS COCO με τη μετρική αξιολόγησης mAP (mean average precision). Χρησιμοποιήθηκε (για το backbone δίκτυο) ένα προ εκπαιδευμένο μοντέλο ResNet-50 με τα βάρη του ImageNet και η εκπαίδευση ήταν για 36 εποχές. Τα αποτελέσματα έδειξαν ότι το μοντέλο αποδίδει καλύτερα από άλλα μοντέλα ανίχνευσης αντικειμένων με mAP 44,2%. Η σύγκλιση επίσης είναι πολύ καλύτερη, φτάνοντας το μοντέλο σε σταθερή κατάσταση, δέκα φορές γρηγορότερα σε σχέση με τον αρχικό DETR. Για να πετύχει αυτές τις μετρικές ο Efficient DETR χρησιμοποιεί μόνο εκατό προτάσεις (object queries). Το μοντέλο έχει καλύτερη ακρίβεια και απόδοση από άποψη υπολογιστικού κόστους καθώς έχουν αφαιρεθεί επίπεδα από τον decoder.

2.4 Αρχιτεκτονικές MobileNet, Squeezenet και DarkNet

Οι αρχιτεκτονικές MobileNet και SqueezeNet ανήκουν σε μια κατηγορία ελαφρών αρχιτεκτονικών αλλά αποδοτικών συνελικτικών δικτύων με χαμηλό υπολογιστικό κόστος και ικανοποιητική ακρίβεια. Σκοπός τους είναι η ικανότητα να εκτελέσουν εργασίες σε περιβάλλοντα με περιορισμένους υπολογιστικούς πόρους όπως φορητές συσκευές και ενσωματωμένα συστήματα. Η αρχιτεκτονική DarkNet χρησιμοποιείται από την οικογένεια των μοντέλων YOLO και σχεδιάστηκε με σκοπό την εφαρμογή σε εργασίες πραγματικού χρόνου με υψηλή ακρίβεια και ταχύτητα .

2.4.1 MobileNet

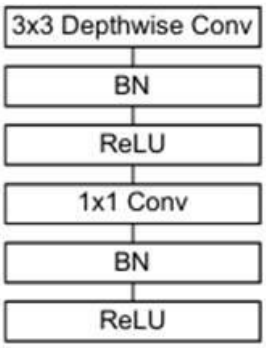
Το μοντέλο MobileNet, που προτάθηκε από τους Howard et al. [24], είναι μια αρχιτεκτονική που σχεδιάστηκε για κινητές συσκευές και συσκευές με περιορισμένους πόρους. Βασικός στόχος της αρχιτεκτονικής είναι η αναγνώριση εικόνας μέσα από το χαμηλό υπολογιστικό κόστος και χαμηλή κατανάλωση ενέργειας.

Η καινοτομία του είναι η τεχνική Depthwise Separable Convolution [24], η οποία στην μειώνει τον αριθμό των παραμέτρων και άρα το υπολογιστικό κόστος. Η τεχνική Depthwise Separable Convolution χωρίζεται σε δύο τμήματα, όπου το πρώτο της τμήμα είναι το Depthwise Convolution [24] ενώ το δεύτερο είναι το Pointwise Convolution [24]. Στο Depthwise Convolution κάθε φίλτρο εφαρμόζεται μόνο σε ένα κανάλι – είσοδο. Στο Pointwise Convolution έχουμε μια σύγκλιση (convolution) 1x1 για την καλύτερη σύνδεση των αποτελεσμάτων των επιπέδων Depthwise Convolution. Η γενική αρχιτεκτονική του μοντέλου (Πίνακας 2.1 και Πίνακας 2.2) είναι μια ακολουθία από Depthwise separable convolutions, ακολουθούμενα από πλήρως συνδεδεμένα επίπεδα (fully connected layer) και ολοκληρώνεται από μια softmax για την ταξινόμηση.

Πίνακας 2.1: Η δομή του μοντέλου MobileNet [24]

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$



Εικόνα 2.17: Συνοπτικά η δομή του μοντέλου MobileNet [24]

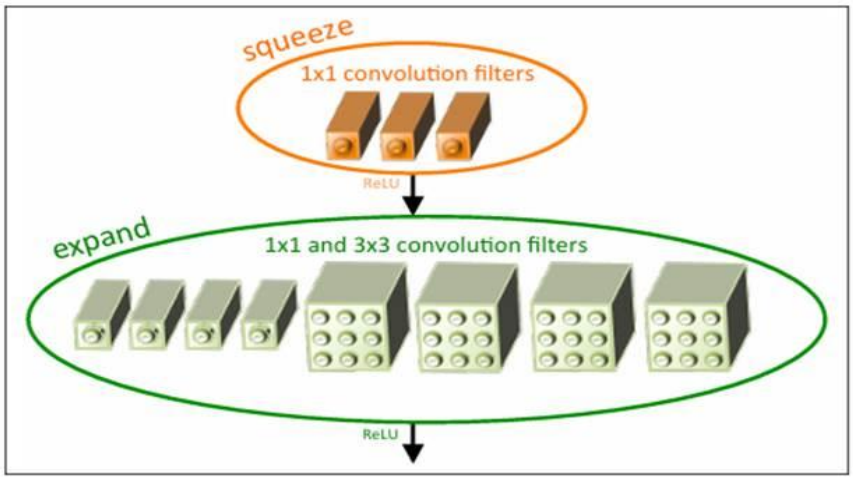
Στις δοκιμές που έγιναν με το σύνολο δεδομένων ImageNet έδειξαν ότι αποδίδει καλύτερα από το VGG16 και το GoogleNet έχοντας σημαντικά μικρότερο αριθμό παραμέτρων [24].

2.4.2 SqueezeNet

Το SqueezeNet είναι ένα μοντέλο που προτάθηκε από τους Iandola et al. [25]. Αποτελεί ένα πολύ ελαφρύ αλλά πολύ αποδοτικό μοντέλο νευρωνικών δικτύων το οποίο στοχεύει τη λειτουργία του σε συστήματα με περιορισμένους υπολογιστικούς πόρους. Αναπτύχθηκε αρχικά με σκοπό τη χρήση του στην αυτοκινητοβιομηχανία και την εφαρμογή του σε μικροελεγκτές.

Η καινοτομία που εφαρμόζει το SqueezeNet είναι το Fire Module [25], το οποίο είναι και ο πυρήνας του. Αποτελείται από δύο μέρη (Εικόνα 2.19), το πρώτο είναι το Squeeze Layer [25] και το δεύτερο είναι το Expand Layer [25]. Το πρώτο είναι ένα σύνολο από convolution layers 1x1 με στόχο τη μείωση των features, γεγονός που οδηγεί στη μείωση των παραμέτρων. Το δεύτερο μέρος αποτελείται από φίλτρα μεγαλύτερου μεγέθους, 1x1 και 3x3, με σκοπό να επεκτείνει και να επαναφέρει τη διάσταση των καναλιών (features) δίνοντας περισσότερη λεπτομέρεια. Τα επίπεδα Squeeze τροφοδοτούνε τα επίπεδα Expand. Η αναλογία των επιπέδων Squeeze και Expand είναι 1/16. Το SqueezeNet δεν έχει πλήρως συνδεδεμένα (fully connected) επίπεδα στην αρχιτεκτονική του (Πίνακας 2.3) [25].

Η αξιολόγηση έγινε συγκριτικά με το μοντέλο AlexNet και το σύνολο δεδομένων ImageNet [25]. Τα αποτελέσματα ήταν εφάμιλλα ή καλύτερα ως προς την απόδοση. Παρ' όλα αυτά, το SqueezeNet έχει πολύ λιγότερες παραμέτρους (50 φορές λιγότερες) και αντιστοιχεί σε πολύ μικρότερο μέγεθος στη μνήμη του συστήματος [25].



Εικόνα 2.18: Η δομή των επιπέδων Squeeze και Expand [25]

Πίνακας 2.2: Η αρχιτεκτονική του SqueezeNet και τα μεγέθη των παραμέτρων ανά επίπεδο [25]

layer name/type	output size	filter size / stride (if not a fire layer)	depth	s_{1x1} (#1x1 squeeze)	e_{1x1} (#1x1 expand)	e_{3x3} (#3x3 expand)	s_{1x1} sparsity	e_{1x1} sparsity	e_{3x3} sparsity	# bits	#parameter before pruning	#parameter after pruning
input image	224x224x3										-	-
conv1	111x111x96	7x7/2 (x96)	1				100% (7x7)			6bit	14,208	14,208
maxpool1	55x55x96	3x3/2	0									
fire2	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	11,920	5,746
fire3	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	12,432	6,258
fire4	55x55x256		2	32	128	128	100%	100%	**33%**	6bit	45,344	20,646
maxpool4	27x27x256	3x3/2	0									
fire5	27x27x256		2	32	128	128	100%	100%	**33%**	6bit	49,440	24,742
fire6	27x27x384		2	48	192	192	100%	**50%**	**33%**	6bit	104,880	44,700
fire7	27x27x384		2	48	192	192	**50%**	100%	**33%**	6bit	111,024	46,236
fire8	27x27x512		2	64	256	256	100%	**50%**	**33%**	6bit	188,992	77,581
maxpool8	13x12x512	3x3/2	0									
fire9	13x13x512		2	64	256	256	**50%**	100%	**30%**	6bit	197,184	77,581
conv10	13x13x1000	1x1/1 (x1000)	1				**20%** (3x3)			6bit	513,000	103,400
avgpool10	1x1x1000	13x13/1	0									
	activations		parameters				compression info				1,248,424 (total)	**421,098** (total)

2.4.3 DarkNet

Το δίκτυο DarkNet-53 προτάθηκε από τους Redmon & Farhadi [26] ως το backbone δίκτυο του YOLOv3. Το DarkNet-53 (Πίνακας 2.3) είναι ένα δίκτυο που αποτελείται από 53 συνελικτικά επίπεδα. Η αρχιτεκτονική του έχει ομοιότητες με αυτή του ResNet και δίνει βάση σε υπολειπόμενα επίπεδα (residual block) [26]. Η τεχνική των residual block συμβάλλει στη βελτίωση της εκπαίδευσης και στο περιορισμό φαινομένων όπως το vanishing και exploding gradient, με το πρώτο να αποτελεί το κυρίως πρόβλημα που πρέπει να περιοριστεί [26].

Τα residual connections επιτρέπουν τα δεδομένα να παρακάμπτουν ορισμένα από τα επίπεδα και να προχωράνε προς τα εμπρός, δηλαδή προς τα επόμενα επίπεδα του δικτύου [26]. Στα συνελικτικά επίπεδα έχουμε δύο τύπους επιπέδων, τα 1x1 convolutions και τα 3x3 convolutions. Τα μεν πρώτα χρησιμοποιούνται για την εξαγωγή των χαρακτηριστικών, τα μεν δεύτερα για να μειώσουν τον όγκο των παραμέτρων και να βελτιώσουν την ροή της πληροφορίας.

Η δομή του δικτύου αναλύεται σε τέσσερεις βασικές ομάδες ή αλλιώς μπλοκ επιπέδων [26]. Η πρώτη ομάδα (μπλοκ) αποτελείται από την πρώτη σύγκλιση χαρακτηριστικών και την δημιουργία αναπαραστάσεων. Η δεύτερη ομάδα αποτελείται από την εκτέλεση μεγαλύτερων συνελικτικών επιπέδων και την εξαγωγή πιο σύνθετων χαρακτηριστικών. Η τρίτη ομάδα παρέχει πιο σύνθετες αναπαραστάσεις καθώς έχει τα 3x3 συνελικτικά επίπεδα και τις υπολειπόμενες συνδέσεις. Στην τέταρτη ομάδα εξάγει τα χαρακτηριστικά που θα χρησιμοποιηθούν από τον ανιχνευτή. Συνάρτηση ενεργοποίησης είναι η Leaky RELU, η οποία επιτρέπει μια μικρή αρνητική κλίση (αρνητικές τιμές) χωρίς να μηδενίζει την έξοδο [27],[28].

Στην αξιολόγηση του δικτύου οι συγγραφείς αναφέρουν ότι το DarkNet-53 δεν είναι τόσο αποδοτικό όσο το RetinaNet. Ωστόσο, παραμένει πιο γρήγορο από το SSD και βελτιώνει την απόδοση του μοντέλου YOLO στα μικρά αντικείμενα, αν και αυτό γίνεται με κόστος της απώλειας ενός μέρους της απόδοσης του στα μεσαίου και μεγάλου μεγέθους αντικείμενα.

Πίνακας 2.3: Η δομή του δικτύου DarkNet-53 [26]

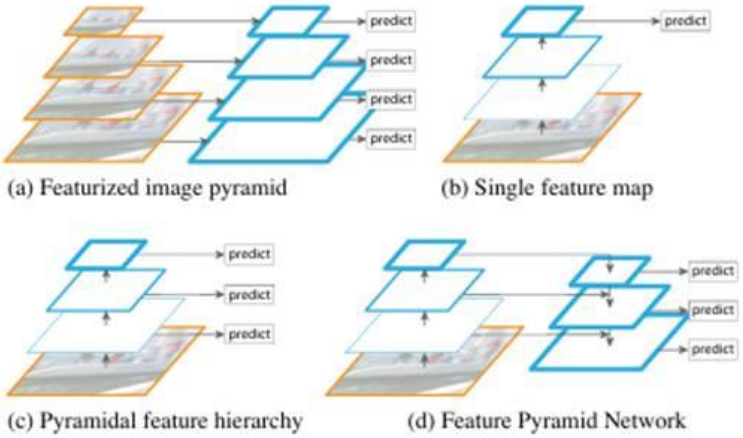
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

2.5 Αρχιτεκτονικές ενίσχυσης μοντέλων - Δίκτυο FPN

Το δίκτυο FPN ανήκει στις αρχιτεκτονικές βελτίωσης των δυνατοτήτων συνελικτικών νευρωνικών δικτύων (CNN) για να ανιχνεύουν αντικείμενα σε πολλαπλές κλίμακες. Αξιοποιεί την αρχή ότι, τα CNN κατασκευάζουν ιεραρχικές δομές αναπαραστάσεων στις οποίες στα χαμηλά επίπεδα αποδίδεται η λεπτομερής χωρική πληροφορία και στα ανώτερα η χαμηλότερη ανάλυση με υψηλότερης αξίας σημασιολογική πληροφορία. Χρησιμοποιείται σε προβλήματα ανίχνευσης αντικειμένων και βελτιώνει την απόδοση των μοντέλων με σχετικά χαμηλό υπολογιστικό κόστος.

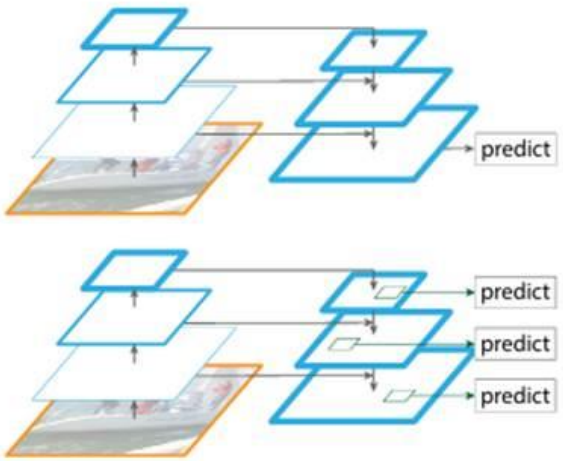
Κατά τη μελέτη των Lin et al. [29] παρουσιάζεται η τεχνική Feature Pyramid Networks (FPN) και αναλυτικότερα τη τεχνική inherent multi-scale pyramidal hierarchy. Η τεχνική των feature pyramids δεν είναι καινούργια, όμως με την διάδοση και την διαρκή ανάπτυξη της έρευνας σε τεχνικές γύρω από την βαθιά μάθηση, προτείνεται μια καινούργια μέθοδος η οποία με μικρό υπολογιστικό κόστος βελτιώνει υπάρχουσες αρχιτεκτονικές.

Η αρχιτεκτονική FPN (Εικόνα 2.22) λειτουργεί με εφαρμογές ανίχνευσής αντικειμένων σε όλες τις κλίμακες. Το πλεονέκτημα της είναι ότι εξάγει χαρακτηριστικά (features) από κάθε επίπεδο της εικόνας, δημιουργώντας μια πολύ-επίπεδη αναπαράσταση που περιλαμβάνει επίπεδα υψηλής ανάλυσης. Με αυτό το τρόπο διατηρείται σε όλα τα επίπεδα το ίδιο σημασιολογικό βάρος (semantically strong). Ωστόσο, ο χρόνος που χρειάζεται για την παραγωγή αποτελέσματος ή πρόβλεψης (inference time) είναι αρκετά μεγαλύτερος, περίπου κατά τέσσερις φορές σε σχέση με το ίδιο μοντέλο που δε χρησιμοποιεί FPN, με αποτέλεσμα να κάνει τη χρήση του δύσκολη τουλάχιστον σε εφαρμογές πραγματικού χρόνου. Άλλο ένα πρόβλημα αποτελεί η επιβάρυνση της μνήμης ειδικά όταν η τεχνική χρησιμοποιείται σε συστήματα περιορισμένων πόρων.



Εικόνα 2.19: (a) Image pyramid to feature pyramid, (b) Single scale features όπως χρησιμοποιούνται από τους σύγχρονους αλγόριθμους για γρηγορότερο εντοπισμό, (c) Μια εναλλακτική δομή της πυραμιδικής δομής με τη χρήση δικτύου ConvNet, (d) Η πρόταση με τη χρήση FPN που είναι το ίδιο γρήγορη με τα (b) και (c) αλλά με καλύτερη ακρίβεια. Με μπλε απεικονίζονται τα feature maps και με πιο παχιά γραμμή τα πιο semantically stronger features [29]

Στην μελέτη [29] τους το μοντέλο που χρησιμοποιούν είναι το Fast R-CNN το οποίο είναι ένα region based μοντέλο και χρησιμοποιεί το Region Proposal Network (RPN). Σε ότι αφορά την μεθοδολογία, έχουν μια εικόνα με τυχαίο μέγεθος ως είσοδο και δίνει ως αποτέλεσμα feature maps ανάλογα με το μέγεθος (proportionally sized feature maps) σε πολλαπλά επίπεδα με τη μορφή ενός συνελικτικού δικτύου. Η μεθοδολογία τους (αρχιτεκτονική FPN) δεν εξαρτάται από την backbone αρχιτεκτονική και στη μελέτη τους χρησιμοποιήθηκαν μοντέλα της οικογένειας ResNets. Οι πυραμίδες εφαρμόστηκαν με αρχιτεκτονικές bottom-up, top-down και με πλευρικές συνδέσεις (lateral connections) (Εικόνα 2.23).



Εικόνα 2.20: Στο πάνω μέρος top – down αρχιτεκτονική με skip connections και στο κάτω μέρος η πρόταση τους με τη χρήση FPN [29]

Από τα πειράματα αποδείχθηκε ότι η χρήση του Feature Pyramid Network (FPN) βελτιώνει την απόδοση του μοντέλου τόσο στη μετρική του average precision (AP) όσο και στη μετρική AP@50. Η μεθοδολογία τους επίσης αν και προσθέτει ένα πολύ μικρό υπολογιστικό κόστος, έχει ένα πιο ελαφρύ detection head που αποτέλεσμα έχει ο χρόνος εκτέλεσης να είναι μικρότερος. Η υλοποίηση δεν είναι υπερβολικά πολύπλοκη (not heavily engineered). Με top down pathway δίνει «ψευδαισθήσεις»

(hallucinations) των υψηλότερων ανάλυσης χαρακτηριστικών ενώ αντίθετα με τη bottom up pathway είναι πιο αδύναμη σημασιολογικά, αλλά πολύ πιο «εύστοχη» και με καλύτερο χωρικό προσδιορισμό των αντικειμένων. Στον Πίνακα 2.4 παρουσιάζονται τα χαρακτηριστικά των τεχνολογιών που αναφέρθηκαν.

Πίνακας 2.4: Συγκριτικός πίνακας των αλγορίθμων (* Αναφέρεται στο βασικό μοντέλο YOLOv10)

Αρχιτεκτονική/Κριτήρια	Faster R-CNN	Drone DETR	Efficient DETR	YOLOv10
Μετρική mAP	0.568	53,90%	42,00%	45,28%
Μετρική FPS	17	76	28	280
Αποδοτικό στην ανίχνευση μικρών αντικειμένων	Ναι	Ναι	Ναι	Ναι
Ταχύτητα εκπαίδευσης	Ναι	Οχι	Οχι	Ναι
Anchor based - Anchor free	Με anchors	Χωρίς anchors	Χωρίς anchors	Χωρίς anchors
Data augmentation	Τυπική	Τυπική	Τυπική	Καλή
Αντοχή στο occlusion	Μεσσαία	Καλή	Καλή	Καλή
Υπολογιστική πολυπλοκότητα	Υψηλή	Μεσσαία	Χαμηλή	Μεσσαία
Real-Time	Οχι	Ναι	Ναι	Ναι

2.6 Συγκριτική αξιολόγηση βιβλιογραφικής ανασκόπησης

Η συγκριτική αξιολόγηση αρχιτεκτονικών και μοντέλων ανίχνευσης μικρών αντικειμένων είναι εξαιρετικά δύσκολη διαδικασία. Για να θεωρηθεί η αξιολόγηση τυπικά ορθή και αντικειμενική πρέπει όλοι οι αλγόριθμοι να φιλτραριστούν μέσα από την ίδια διαδικασία, πράγμα που δεν μπορεί να ισχύσει σε αυτή τη περίπτωση. Η κάθε ερευνητική ομάδα έτρεξε τους αλγορίθμους της σε διαφορετικό υλικό υπολογιστή και σε διαφορετικές χρονικές περιόδους που σημαίνει ότι κάθε μια ομάδα είχε υλικό διαφορετικής δυναμικής και επιδόσεων, γεγονός που παίζει σημαντικό ρόλο σε πλήθος μετρικών και επηρεάζει αρκετά κριτήρια. Ένα άλλο ζήτημα είναι ότι οι ομάδες εκπαιδεύσανε τους αλγορίθμους τους σε διαφορετικά δεδομένα και με διαφορετικό τρόπο. Αυτό δεν είναι αρχικά αρνητικό αλλά είναι ένας παράγοντας που διαφοροποιεί το τελικό αποτέλεσμα καθώς μπορεί να δώσει διαφορετικές τεχνικές λύσεις σε επίπεδο αρχιτεκτονικής και κώδικα. Στην ίδια λογική ακόμα σημαντικότερο και πιο δύσκολο στην διαδικασία της συγκριτικής αξιολόγησης είναι ότι οι ερευνητές αξιολογήσανε τους αλγορίθμους τους σε διαφορετικά σύνολα δεδομένων και σε πολλές περιπτώσεις συμπεριέλαβαν διαφορετικές μετρικές ο ένας από τον άλλον. Όλοι οι αλγόριθμοι που περιλήφθησαν θεωρούνται αξιόπιστοι και θεωρούνται State of the Art από την ερευνητική κοινότητα.

Ο αλγόριθμος Faster R-CNN βασίζεται σε ένα δίκτυο ResNet και με την ενσωμάτωση του RPN είναι ένα ισχυρό μοντέλο με πολύ καλή ακρίβεια και δυνατότητες ανίχνευσης σε εικόνες με αντικείμενα διαφόρων κατηγοριών. Όμως, χρειάζεται υπολογιστικούς πόρους και δεν έχει την δυνατότητα να λειτουργήσει σε real time εφαρμογές όπως και να χειριστεί πολύ καλά εικόνες με πυκνή στοίχιση αντικειμένων.

Ο αλγόριθμος Drone – DETR βασίζεται σε αρχιτεκτονική Transformer και είναι ειδικά σχεδιασμένος για εφαρμογές με εναέριες εικόνες. Έχει πολύ καλή απόδοση στην ανίχνευση μικρών αντικειμένων ακόμα και σε σύνθετα περιβάλλοντα με πολλά και πυκνά αντικείμενα.

Ο αλγόριθμος Efficient – DETR αν και έχει λίγο μικρότερη απόδοση σε σχέση με τον Drone – DETR εκπαιδεύεται ευκολότερα και χειρίζεται καλά σκηνές με πυκνά αντικείμενα. Είναι πιο απλός στη δομή του σε σχέση με τους αλγορίθμους της οικογένειας DETR και πετυχαίνει γρηγορότερα σύγκληση. Επίσης, είναι πιο εύκολο να εφαρμοστεί σε συστήματα με περιορισμένους πόρους και εφαρμογές πραγματικού χρόνου. Ο βασικός αλγόριθμος RT – DETR εμφανίζει πολύ καλά χαρακτηριστικά τόσο ως προς την απόδοση του όσο και ως προς την ικανότητα του να παρέχει υψηλά FPS ακόμα και για μικρά αντικείμενα. Αν και απαιτεί περισσότερο χρόνο για την εκπαίδευση, είναι κατάλληλος για εφαρμογές πραγματικού χρόνου και περιβάλλοντα με πολλά πυκνά αντικείμενα. Ένα ακόμη χαρακτηριστικό της οικογένειας αλγορίθμων DETR είναι ότι παρουσιάζουν καλύτερες τεχνικές επαύξησης δεδομένων σε σχέση με τους κλασικούς αλγορίθμους.

Οι αρχιτεκτονικές FPN και RPN βασίζονται σε άγκυρες και έχουν δοκιμαστεί σε εφαρμογές ανίχνευσης αντικειμένων. Έχει αποδειχθεί ότι αποδίδουν καλύτερα όταν ενσωματώνονται σε πιο ισχυρές δομές και καταφέρνουν να βελτιώσουν κατά πολύ την απόδοση των μοντέλων.

Το μοντέλο HATSC-YOLOv10 (βασίζεται στο μοντέλο YOLOv10) είναι ένα μοντέλο που ισορροπεί πολύ καλά ανάμεσα στην ακρίβεια και στην ταχύτητα. Είναι ιδανικό για εφαρμογές πραγματικού χρόνου και για εργασίες που αφορούν τόσο αντικείμενα μεγάλου μεγέθους όσο και αντικείμενα μικρού μεγέθους. Έχει σχεδιαστεί ώστε να δίνει προβλέψεις πολύ γρήγορα για να μπορεί να εφαρμοστεί σε συστήματα πραγματικού χρόνου. Το backbone δίκτυο του μοντέλου είναι το CSP-Net και οι συγγραφείς προσθέσανε το δίκτυο HATSC για να βελτιώσουν ακόμα περισσότερο την ακρίβεια του στον εντοπισμό μικρών αντικειμένων. Το μοντέλο μπορεί να εκπαιδεύεται αρκετά γρήγορα και να διαχειρίζεται πολύ καλά καταστάσεις σκηνών με πολλά πυκνά αντικείμενα παρέχοντας πολλές και εξελιγμένες τεχνικές επαύξησης δεδομένων. Θεωρείται αυτή τη στιγμή από τα δυνατότερα και πιο ισορροπημένα μοντέλα ανίχνευσης μικρών αντικειμένων (State of the Art) με εφαρμογές όπου η ταχύτητα είναι βασικό κριτήριο, όπως, σε αυτόνομα οχήματα, την βιομηχανία και την επιτήρηση χώρων. Είναι αρκετά διαχειρίσιμο για να εφαρμοστεί σε συστήματα που έχουν περιορισμένους πόρους παραχωρώντας όμως ένα μέρος από την ακρίβεια και την ταχύτητα του.

Για εφαρμογές πραγματικού χρόνου απαιτείται ένα μοντέλο που να δίνει υψηλό αριθμό FPS καθώς έχουν την ανάγκη να επεξεργάζονται όσες περισσότερες εικόνες μπορούν ανά μονάδα χρόνου. Σε περιπτώσεις που κρίνεται αναγκαία η εφαρμογή πραγματικού χρόνου, μια αρχιτεκτονική Faster R-CNN είναι κατάλληλη καθώς έχει υψηλή και σταθερή απόδοση. Η τελική επιλογή πρέπει να είναι ολιστική και να λαμβάνει υπόψη της τα ισχυρά σημεία και τις αδυναμίες της κάθε επιλογής. Κάθε περίπτωση εφαρμογής είναι διαφορετική και πρέπει να μελετάται ξεχωριστά, γνωρίζοντας ότι κάθε επιλογή έχει ένα ποσοστό συμβιβασμού.

2.6.1 Κριτική σύγκριση μοντέλων και αρχιτεκτονικών

Παρακάτω παραδίδεται μια κριτική σύγκριση των μοντέλων και τον αρχιτεκτονικών (πίνακας 2.5) που παρουσιάστηκαν παραπάνω στο κεφάλαιο. Για την σύγκριση χρησιμοποιούνται κάποια κριτήρια ως μέτρο και αυτά ορίζονται στα παρακάτω πέντε:

- Αρχιτεκτονική (CNN, Transformer, Hybrid)
- Απόδοση σε μικρά αντικείμενα (precision, recall mAP)
- Υπολογιστικό κόστος (training time, inference speed, απαίτηση σε υλικό GPU/CPU/RAM)
- Γενίκευση (robustness σε διαφορετικά σύνολα δεδομένων και συνθήκες αναπαράστασης)
- Καταλληλότητα για συσκευές edge

Σύντομη σύνοψη

Μοντέλα Two - Stage: Ο Faster R-CNN παρέχει υψηλή ακρίβεια αλλά αδυνατεί να εφαρμοστεί σε εφαρμογές πραγματικού χρόνου ή σε συσκευές edge.

Μοντέλα One - Stage: Τα μοντέλα YOLO, EfficientDet, RetinaNet έχουν καλή απόδοση και είναι κατάλληλα για εφαρμογές πραγματικού χρόνου. Οι τροποποιημένες εκδόσεις του YOLO, τα Drone - YOLO και HATSC - YOLOv10 είναι καλύτερα σε εφαρμογές με εικόνες που έχουν ληφθεί από UAV με δυνατότητες εφαρμογής πραγματικού χρόνου.

Ελαφριές αρχιτεκτονικές και δομές: Τα δίκτυα MobileNet, SqueezeNet και DarkNet είναι καίρια για την ανάπτυξη εφαρμογών μηχανικής όρασης σε υλικό με περιορισμένους πόρους με δυνατότητες εφαρμογής πραγματικού χρόνου. Η δομή FPN βελτιώνει τα μοντέλα ως προς την ανίχνευση μικρών αντικειμένων καθώς τα βοηθά με την διαχείριση του προβλήματος του multi - scale.

Πίνακας 2.5: Συγκριτικός πίνακας χαρακτηριστικών των μοντέλων και περιγραφών

Μοντέλο / Δομή	Αρχιτεκτονική	Απόδοση σε μικρά αντικείμενα	Υπολογιστικό κόστος / Ταχύτητα	Καταλληλότητα για συσκευές edge
Faster R-CNN	Two-stage	Καλός εντοπισμός αλλά χρειάζεται καλά anchors και scales	Μέτριο προς υψηλό, δεν εφαρμόζεται σε πραγματικό χρόνο	Πολύ δύσκολη η εφαρμογή του, απαιτεί ισχυρούς υπολογιστικούς πόρους
RT-DETR	Transformer	Σχεδιασμένο για γενική ανίχνευση, με βελτιώσεις σε multi-scale δείχνει καλύτερη ισορροπία μεταξύ ταχύτητα και ακρίβειας	Βελτιστοποιημένο για real-time (έκδοση RT επιτυγχάνει υψηλά FPS με τη χρήση GPU)	Πολύ δύσκολη η εφαρμογή του ακόμα και στις μικρές εκδόσεις
Drone-DETR	Transformer	Βελτιωμένο για μικρά αντικείμενα σε εικόνες από UAV	Ελαφρύτερο από το DETR σχεδιασμένο με λιγότερες παραμέτρους	Πολύ δύσκολη η εφαρμογή του ακόμη, χρειάζεται βελτιστοποίηση για edge συσκευές
Efficient DETR	Transformer	Πολύ καλύτερος χρόνος εκπαίδευσης, συνολικά βελτιωμένο σε σχέση με το αρχικό μοντέλο DETR	Καλύτερη απόδοση σε σχέση με τον κανονικό DETR αλλά παραμένει με μεγάλο υπολογιστικό κόστος	Δεν είναι φιλικό για εφαρμογή σε συσκευές edge
Efficient DET	One-stage	Πολύ καλό στη διασχείριση multi-scale, καλή απόδοση σε μικρά αντικείμενα	Σχεδιασμένο για την βελτίωση της αποδοτικότητας	Καλή επιλογή για edge συσκευές αν επιλεγεί κάποια από τις μικρές εκδόσεις
RetinaNet	One-stage	Καλή απόδοση για μικρά αντικείμενα και αντικείμενα που εμφανίζονται σπάνια με την χρήση του focal loss	Μέτριο υπολογιστικό κόστος	Μπορεί να εφαρμοστεί σε συσκευές edge αλλά όχι σε πραγματικό χρόνο
FPN (Feature Pyramid Network)	Δομή που μας βοηθάει με προβλήματα multi-scale	Πολύ σημαντικό για μικρά αντικείμενα	Μικρό επιπρόσθετο κόστος σε σχέση με τα οφέλη	Δεν επηρεάζει αρνητικά την δυνατότητα εφαρμογής του μοντέλου σε συσκευές edge
MobileNet (backbone)	Πολύ ελαφρύ δίκτυο CNN	Απαιτεί προσεκτικό fine tuning για μικρά αντικείμενα	Πολύ χαμηλό υπολογιστικό κόστος	Σημαντική επιλογή για συσκευές edge
SqueezeNet	Πάρα πολύ μικρό δίκτυο CNN	Πολύ μικρό μέγεθος μοντέλου αντίστοιχα όμως μειωμένη και η δυνατότητα ανίχνευσης μικρών αντικειμένων σε πολύ δύσκολους στόχους	Εξαιρετικά μικρό και γρήγορο σε συστήματα περιορισμένων πόρων	Καλή βάση για edge συσκευές αλλά πιθανόν να χρειαστεί βελτιστοποίηση
DarkNet	Πλήρες CNN που χρησιμοποιείτε στις αρχικές εκδόσεις του YOLO	Καλή απόδοση σε μικρά αντικείμενα, θέλει ενίσχυση σε πολύ μικρά αντικείμενα (π.χ. με την χρήση SAHI)	Πολύ αποδοτικό real-time, πολύ καλή ισορροπία ανάμεσα σε ταχύτητα και απόδοση με την χρήση GPU	Πολύ καλή επιλογή για edge συσκευές όταν χρησιμοποιούνται ελαφριές εκδόσεις
Drone-YOLO	One-stage	Εστιασμένο για εντοπισμό αντικειμένων από UAV, καλή απόδοση σε μικρά αντικείμενα	Το υπολογιστικό κόστος είναι μέτριο προς ελαφρύ, κατάλληλο για εφαρμογές πραγματικού χρόνου	Μπορεί να εφαρμοστεί σε συσκευές edge
HATSC-YOLOv10	One-stage	Πολύ καλή απόδοση στον εντοπισμό μικρών αντικειμένων σε σχέση με το αρχικό μοντέλο	Μέτριο υπολογιστικό κόστος, βελτιωμένη ταχύτητα απαιτεί όμως με τη χρήση GPU	Υπάρχει δυσκολία να εφαρμοστεί σε συσκευές edge

Μοντέλα Two - Stage

Το μοντέλο Faster R-CNN είναι ένα κλασικό παράδειγμα μοντέλου δυο σταδίων και από τα πιο ισχυρά αυτού του τύπου. Η βελτίωση του και η καλή του απόδοση έγκειται στη χρήση του δικτύου RPN. Υστερεί σε ταχύτητα εκπαίδευσης και απαιτεί προσεκτική ρύθμιση των αγκυρών για μικρά αντικείμενα. Με την ενσωμάτωση του δικτύου FPN έχουμε σημαντική βελτίωση της αντίληψης σε πολλαπλές κλίμακες. Η χρήση του σε συσκευές περιορισμένων πόρων είναι περιορισμένη καθώς έχει

υψηλό υπολογιστικό κόστος. Επίσης, είναι περιορισμένη η δυνατότητα χρήσης του Faster R-CNN σε προβλήματα πραγματικού χρόνου.

Μοντέλα One - Stage

RetinaNet: Το μοντέλο RetinaNet δείχνει καλή απόδοση σε σκηνές με αντικείμενα μικρού μεγέθους και με πυκνή διάταξη. Χρησιμοποιεί την τεχνική Focal Loss με την οποία αντιμετωπίζει την ανισορροπία ανάμεσα στο φόντο (background) και τα μικρά αντικείμενα (στόχοι). Δεν είναι πλήρως συμβατό με εφαρμογές πραγματικού χρόνου και για χρήση του με edge συσκευές.

EfficientDet: Το μοντέλο EfficientDet εμφανίζει ικανοποιητική απόδοση στην ανίχνευση χαρακτηριστικών από αντικείμενα μικρού μεγέθους. Χρησιμοποιεί ως backbone δίκτυο το EfficientNet και BiFPN για να βελτιώσει την απόδοση του σε εικόνες διαφορετικών κλιμάκων. Το μοντέλο EfficientDet έχει εκδόσεις (D0 - D7) και επιτρέπει τη χρήση της έκδοσης (μέγεθος μοντέλου) που είναι η πιο κατάλληλη για την εφαρμογή που θέλει να αναπτύξει αλλά και το υλικό που διαθέτει κάποιος. Μπορεί να εφαρμοστεί σε edge συσκευές με τις εκδόσεις D0 ή D1.

Drone-YOLO: Είναι μια παραλλαγή του YOLO προσαρμοσμένη για εικόνες από UAV. Πετυχαίνει καλύτερη απόδοση με την εφαρμογή βελτιστοποιημένων modules για μικρά αντικείμενα. Είναι βελτιωμένο σε σχέση με το αρχικό μοντέλο και διατηρεί την ικανότητα υλοποίησης του σε συστήματα περιορισμένων πόρων και σε εφαρμογές πραγματικού χρόνου.

HATSC-YOLOv10: Το μοντέλο HATSC-YOLOv10 είναι και αυτό τροποποίηση του αρχικού μοντέλου. Οι αλλαγές που εισάγει είναι τα συστήματα Hybrid Attention και Spatio - Channel που βελτιώνουν την απόδοση του. Σε σχέση με το αρχικό μοντέλο συνεχίζει να είναι ένα «ελαφρύ» μοντέλο με καλή απόδοση και διατηρεί την ικανότητα του σε εφαρμογές πραγματικού χρόνου.

Μοντέλα Transformers

DETR & Efficient DETR: Τα μοντέλα DETR και Efficient DETR χρησιμοποιούν αρχιτεκτονική transformers για την ανίχνευση μικρών αντικειμένων. Δεν έχουν την ανάγκη να χρησιμοποιήσουν άγκυρες και NMS στη δομή τους. Έχουν αρκετά καλή απόδοση, ωστόσο το αρχικό μοντέλο DETR έχει πολύ μεγάλο υπολογιστικό κόστος και απαιτεί πολύ μεγάλη χρονικά εκπαίδευση. Το μοντέλο Efficient DETR αντιμετωπίζει ως ένα βαθμό αυτά τα προβλήματα.

RT-DETR: Το μοντέλο RT-DETR είναι μια παραλλαγή του αρχικού μοντέλου DETR με στόχο την βελτίωση του χρόνου και την αύξηση της ταχύτητας. Είναι μια προσπάθεια για την δημιουργία ενός μοντέλου DETR που να ισορροπεί ανάμεσα σε ακρίβεια και απόδοση. Παραμένει ένα μοντέλο με πολύ μεγάλο υπολογιστικό κόστος (ειδικά σε σύγκριση με τα μοντέλα της οικογένειας YOLO) και μια επιδίωξη ενσωμάτωσης μοντέλων transformer σε εφαρμογές πραγματικού χρόνου.

Drone-DETR: Το μοντέλο Drone-DETR είναι μια παραλλαγή του μοντέλου DETR προσανατολισμένο για δεδομένα εικόνας προερχόμενα από UAV και στοχεύει στην αναγνώριση μικρών αντικειμένων. Η βελτιστοποίηση έγκειται στον μηχανισμό προσοχής (attention mechanism) και στην σχεδίαση ελαφρύτερης δομής που το καθιστούν καλύτερη επιλογή σε εφαρμογές με εικόνες από UAV. Η εφαρμογή του σε πραγματικό χρόνο είναι ακόμα δύσκολη υπόθεση καθώς έχει μεγάλες υπολογιστικές απαιτήσεις.

Ελαφριές αρχιτεκτονικές και δομές (Lightweight Backbones και Modules)

MobileNet: Το MobileNet είναι ένα αποδοτικό backbone δίκτυο που είναι βασισμένο σε separable convolutions, δηλαδή διαχωρίζει το spatial filtering από το feature combing. Έχει πολύ μικρό μέγεθος μοντέλου, με καλή απόδοση σε συστήματα edge. Αποτελεί το πιο ισορροπημένο σε σχέση με τα ελαφριά μοντέλα, αλλά υστερεί στο να μάθει πολύπλοκες σκηνές.

SqueezeNet: Το SqueezeNet είναι ένα δίκτυο με σχετικά λίγες παραμέτρους ώστε να μπορεί να τρέξει σε συστήματα με αρκετά περιορισμένους πόρους. Αντιμετωπίζει δυσκολίες σε μεγάλα ή σύνθετα σύνολα δεδομένων με μεγάλες διαφορές στις κλίμακες των εικόνων.

DarkNet: Το DarkNet είναι το backbone δίκτυο του αρχικού μοντέλου YOLO. Είναι αρκετά ισχυρό και ισορροπεί πολύ καλά ανάμεσα σε ταχύτητα και απόδοση. Δεν είναι τόσο ελαφρύ όσο το MobileNet ή το SqueezeNet. Με την χρήση τεχνικών SAHI ή με tile based inference ενισχύει την ακρίβεια του στην ανίχνευση μικρών αντικειμένων.

FPN: Το FPN (Feature Pyramid Network) βελτιώνει την απόδοση των backbone δικτύων ως προς την αντίληψη αντικειμένων σε multi scale προβλήματα. Συνδυάζει high level semantic features (από deep layers) με fine grained spatial features (από shallow layers). Η ενσωμάτωση βελτιώνει την απόδοση σε εργασίες ανίχνευσης μικρών αντικειμένων, προσθέτοντας μικρή πολυπλοκότητα. Είναι απλό στην υλοποίηση.

Επομένως μπορούν να οριστούν τα μοντέλα που αποτελούν τις καταλληλότερες λύσεις ανά είδος εργασίας και υλικού:

- Της οικογένειας YOLO είναι αποτελεσματικά για την ανίχνευση μικρών αντικειμένων και μπορούν να ενισχυθούν περαιτέρω με την χρήση τεχνικών tile inference, όπως το SAHI.
- Τα μοντέλα Drone-YOLO και HATSC-YOLOv10 έχουν καλή απόδοση λόγω της εξειδίκευσης τους σε δεδομένα προερχόμενα από UAV.
- RT-DETR και Drone-DETR είναι υλοποιήσεις σε transformers και πρέπει να διερευνηθούν περαιτέρω καθώς προς το παρόν είναι πολύ απαιτητικά για χρήση ώστε να μπορούν να λειτουργήσουν σε συστήματα με μικρότερους πόρους.

2.7 Συμπεράσματα

Η ανίχνευση αντικειμένων γενικότερα όπως και η ανίχνευση μικρών αντικειμένων παρουσιάζει κάποιες ιδιαίτερες προκλήσεις. Αρχικά, τα αντικείμενα αντιπροσωπεύονται από πολύ λίγα εικονοστοιχεία (pixel) στην εικόνα και αυτό κάνει εξ' ορισμού δύσκολη την εύρεση τους από τα μοντέλα βαθιάς μάθησης. Η υψηλή ακρίβεια που απαιτείται συνήθως σε εφαρμογές ανίχνευσης αντικειμένων εκφράζεται από την θέση και το μέγεθος των πλαισίων αναφοράς (bounding boxes) και η τυχόν παραμικρή απόκλιση από την πραγματική θέση και μέγεθος έχουν σημαντική επίδραση στην ακρίβεια [19].

Η πληροφορία που είναι διαθέσιμη σε εφαρμογές ανίχνευσης μικρών αντικειμένων για το μοντέλο είναι πολύ περιορισμένη γιατί όπως αναφέρθηκε το αντικείμενο περιγράφεται από πολύ λίγα pixel, προκαλώντας το πρόβλημα της «τοπικής» χαμηλής ανάλυσης. Επιπροσθέτως, το πλαίσιο οριοθέτησης μπορεί να συμπεριλάβει γειτονικά pixel ή ακμές μαζί με το αντικείμενο ενδιαφέροντος, οπότε και μεγαλώνει το μέγεθος του πλαισίου. Ένα άλλο συνηθισμένο πρόβλημα είναι το spatial context (η χωρική συσχέτιση μεταξύ των αντικειμένων), που είναι η πιθανότητα να βρεθεί ένα αντικείμενο σε θέση κατάλληλη σε σχέση με τη θέση ενός άλλου, όπως για παράδειγμα οι ώμοι και ο λαιμός ενός ανθρώπου να βρίσκονται σε κοντινή απόσταση από κεφάλι του.

Η ανισοκατανομή (imbalance) είναι ένα ακόμη στοιχείο που επιδρά αρνητικά στην ακρίβεια των μοντέλων σε εργασίες ανίχνευσης μικρών αντικειμένων, και χωρίζεται σε δύο είδη. Το πρώτο είναι ανάμεσα στα αντικείμενα που βρίσκονται μπροστά σε σχέση με αυτά που βρίσκονται στο φόντο. Αυτό δημιουργεί ιδιαίτερο πρόβλημα στα μοντέλα που βασίζουν τη λειτουργία τους στο Region Proposal Network (RPN), καθώς αν οι άγκυρες δεν ταυτίζονται ικανοποιητικά με το ground truth χαρακτηρίζονται αρνητικά. Μόνο οι άγκυρες με υψηλό Intersection over Union (IoU) σε σχέση με το ground truth χαρακτηρίζονται αντικείμενο. Το δεύτερο είδος ανισοκατανομής αφορά στον αριθμό των αντικειμένων στις κλάσεις ενός συνόλου δεδομένων (dataset) που παρέχεται ως εκπαίδευση στο μοντέλο. Η συχνότητα εμφάνισης των αντικειμένων ανά κατηγορία δεν είναι όμοια. Αυτό μπορεί να οδηγήσει ένα μοντέλο, ακόμα και μετά από μια καλή εκπαίδευση από άποψη παραμέτρων, να αναγνωρίζει πολύ καλά αντικείμενα που βρίσκονται σε μεγάλο πλήθος (καλή εκπροσώπηση) στο σετ εκπαίδευσης και να μην αναγνωρίζει εύκολα αντικείμενα που στο σετ εκπαίδευσης είναι μικρότερα σε αριθμό αλλά μεγάλα σε μέγεθος (υπο-εκπροσώπηση). Το μοντέλο τείνει να είναι πολύ καλό στις κλάσεις που έχουν καλή εμφάνιση ανεξάρτητα από το μέγεθος ή τη σημασία τους [30], [31].

Πιο συγκεκριμένα σε ότι αφορά τις εικόνες που λαμβάνονται από εναέριες λήψεις (π.χ. UAV) προσθέτουν επιπλέον δυσκολίες στην ανίχνευση μικρών αντικειμένων λόγω διαφόρων παραγόντων. Αυτές προέρχονται από το γεγονός ότι οι εικόνες αυτές μεταξύ τους έχουν αντικείμενα πολλών ετερογενών κατηγοριών με επικάλυψη (occlusion), τυχαίες ποικίλες κλίσεις και γωνίες ως προς το αντικείμενο σε αντίθεση με εικόνες που λαμβάνονται από σταθερές κάμερες στο έδαφος όπου οι λήψεις μεταξύ τους έχουν συνέπεια. Τα αντικείμενα που αποτυπώνονται από εναέριες λήψεις πολλές φορές έχουν πιο πυκνή και ανομοιογενή κατανομή στην εικόνα, όπως για παράδειγμα ο κόσμος σε μια διασταύρωση ή οχήματα σταθμευμένα σε ένα πάρκινγκ, σε σχέση με την υπόλοιπη εικόνα. Σε αντίθεση, μικρά αντικείμενα σε λήψεις από το έδαφος μπορούν να θεωρηθούν μικροσκοπικά (tiny objects) οπότε και έχουμε ακόμα μικρότερη διαθέσιμη πληροφορία για την εκπαίδευση και εξαγωγή χαρακτηριστικών τους [31].

Οι εικόνες αυτές έχουν το χαρακτηριστικό ότι έχουν πολύ μεγάλη γωνία θέασης (wide angle) οπότε εκτός από τη παραμόρφωση του φακού εμφανίζεται και πολύ μεγαλύτερο φόντο για γίνει διάκριση των αντικειμένων ενδιαφέροντος καθιστώντας δυσκολότερο τον εντοπισμό ειδικά όταν είναι πυκνά και μικρά σε μέγεθος.

Τα ποιοτικά χαρακτηριστικά των εικόνων επηρεάζονται και από τις φυσικές συνθήκες, όπως για παράδειγμα ο φωτισμός και η συννεφοκάλυψη, που μπορεί να επικρατούν στην περιοχή την στιγμή της λήψης. Τέλος, η έλλειψη αρκετών ικανών χαρακτηριστικών οδηγεί στο συμπέρασμα ότι ένα επίπεδο νευρωνικού δικτύου δεν μπορεί να περιέχει αρκετή πληροφορία [31]. Αυτό σημαίνει ότι υπάρχει ανάγκη για αρχιτεκτονικές που εξάγουν χαρακτηριστικά σε πολλαπλές κλίμακες ώστε να έχουν τα μοντέλα ενισχυμένη ικανότητα ανίχνευσης μικρών αντικειμένων.

Κεφάλαιο 3ο: Δεδομένα και Προεπεξεργασία

Η ανίχνευση μικρών αντικειμένων αποτελεί από τις δυσκολότερες προκλήσεις στον τομέα της υπολογιστικής όρασης καθώς τα αντικείμενα προς ανίχνευση έχουν μικρό μέγεθος και οι εικόνες αρκετές φορές δεν έχουν υψηλή ανάλυση. Παράλληλα ο εντοπισμός αυτών των αντικειμένων απαιτεί ακρίβεια και σαφή προσδιορισμό της κλάσης που ανήκουν.

Για την καλύτερη εκπαίδευση των αλγορίθμων στην ανίχνευση μικρών αντικειμένων η ποιότητα και η προετοιμασία των εικόνων παίζουν πολύ μεγάλο ρόλο. Τις περισσότερες φορές τα σύνολα δεδομένων πρέπει να περαστούν από προεπεξεργασία για να ελεγχθεί ο θόρυβος που πιθανόν να υπάρχει, να γίνει κανονικοποίηση των δεδομένων και να διερευνηθούν οι στρατηγικές επαύξησης για την ενίσχυση του συνόλου δεδομένων. Αυτά τα βήματα βοηθούν στην βελτίωση της αποτελεσματικότητας και της αύξησης ακρίβειας ενός μοντέλου.

Στο παρόν κεφάλαιο αναλύονται οι πηγές των δεδομένων της παρούσας εργασίας, τα κριτήρια με τα οποία πρέπει να επιλέγονται, ποια σύνολα δεδομένων επιλέχθηκαν καθώς τα είδη και η διάρθρωση των αντικειμένων που συμπεριλαμβάνονται. Στη συνέχεια, αναφέρονται τεχνικές προεπεξεργασίας και κανονικοποίησης και, τέλος, τεχνικές επαύξησης δεδομένων όπως και προβλημάτων ανισοκατανομής αντικειμένων ανάμεσα σε κλάσεις και τεχνικές που εφαρμόζονται για την εξισορρόπηση τους.

Οι προκλήσεις που υπάρχουν στη σωστή δημιουργία και επιλογή συνόλου δεδομένων δεν είναι λίγες ούτε και εύκολες στην επίλυση τους. Πρέπει ο μελετητής που αναλαμβάνει να υλοποιήσει μια εφαρμογή ανίχνευσης αντικειμένων να λάβει υπόψιν του θεωρητικές και πρακτικές παραμέτρους αλλά και περιορισμούς που ορίζονται από τη φύση του προβλήματος και το είδος της εφαρμογής. Η κατανόηση του προβλήματος είναι το θεμελιώδες πρόβλημα και μέσα από αυτή πρέπει να γίνει η σωστή κατασκευή ή επιλογή συνόλου δεδομένων για την εκπαίδευση και επικύρωση των αποτελεσμάτων του μοντέλου.

3.1 Πηγές δεδομένων

Οι κυριότερες πηγές δεδομένων για την ανίχνευση μικρών αντικειμένων είναι συνήθως ιστοσελίδες πανεπιστημίων και ερευνητικών ιδρυμάτων, ιστοσελίδες ερευνητικών ομάδων και ιστοσελίδες – αποθετήρια δεδομένων. Για την παρούσα εργασία έγινε αναζήτηση κυρίως για ελεύθερα δεδομένα καθώς υπάρχουν και κάποια κλειστά που χρησιμοποιούνται από οργανισμούς και ερευνητές για ανάπτυξη προϊόντων. Μια αρχική αναζήτηση έγινε στο Kaggle το οποίο προσφέρει σύνολα δεδομένων για όλες τις περιοχές της επιστήμης της μηχανικής μάθησης. Στην συνέχεια, αναζητήθηκαν δεδομένα στο Paperswithcode το οποίο προσφέρει δεδομένα μαζί με ερευνητικές δημοσιεύσεις και τον κώδικα που χρησιμοποιήθηκε για την εκάστοτε προτεινόμενη λύση από ερευνητές. Το Paperswithcode είναι η βασική πηγή της εργασίας καθώς από εκεί αντλήθηκαν τα περισσότερα σύνολα δεδομένων με υπερσυνδέσμους προς τα αποθετήρια τους (σε κάποιες περιπτώσεις είναι είτε το GitHub είτε το Google Drive). Αναζητήσεις επίσης έγιναν στις παρακάτω ιστοσελίδες:

- Google Dataset Search, εξειδικευμένη μηχανή αναζήτησης για σύνολα δεδομένων
- IEEE Data Port, το αποθετήριο δεδομένων του IEEE
- UCI Machine Learning Repository, αποθετήριο συνόλων δεδομένων που συντηρείται από το πανεπιστήμιο Irvine στην Καλιφόρνια.
- Huggingface, ιστοσελίδα που φιλοξενεί τόσο έτοιμα μοντέλα όσο και σύνολα δεδομένων
- Harvard Dataverse, ένα αρκετά εκτενές αποθετήριο από το πανεπιστήμιο του Harvard

3.1.1 Ανοιχτά ή δημόσια δεδομένα

Η εργασία ασχολείται με δεδομένα που είναι είτε ελεύθερα προσβάσιμα είτε απαιτούν εγγραφή σε κάποια υπηρεσία. Δεν διερευνήθηκαν δεδομένα που είναι κλειστά και η πρόσβαση τους, τις περισσότερες φορές, καθίσταται δύσκολη ή αδύνατη. Τα δεδομένα που αναφέρονται παρακάτω χρησιμοποιούνται για την εκπαίδευση των μοντέλων και για την αξιολόγηση τους καθώς τα περισσότερα από αυτά θεωρούνται benchmarks.

Ένα σετ δεδομένων πρέπει να έχει κάποια χαρακτηριστικά, όπως για παράδειγμα αυτά που ορίζουν οι συγγραφείς που προτείνουν το VEDAI (Vehicle Detection in Aerial Imagery). Ένα dataset πρέπει να αποτελείται από εικόνες που δεν υπόκεινται σε πνευματικά δικαιώματα ή διατίθενται ελεύθερα προς χρήση από ερευνητές και, ευρύτερα, από την κοινότητα της μηχανικής όρασης. Ένα άλλο ζήτημα αφορά στην απεικόνιση αντικείμενων, ή αλλιώς «στόχων», διαφορετικών κατηγοριών όπου αυτές οι κατηγορίες να αντιπροσωπεύουν την πραγματικότητα και να καλύπτουν τις ανάγκες της κοινότητας. Τα αντικείμενα «στόχοι» πρέπει να είναι μικρά σε μέγεθος, να περιγράφονται από λίγα pixels και το background να είναι όσο γίνεται διαφορετικό από εικόνα σε εικόνα. Τέλος, το ground – truth συμπεριλαμβανομένων των ετικετών πρέπει να είναι πλήρες και όσο το δυνατόν πιο ακριβές ώστε να μπορεί η κοινότητα να αναπτύσσει και να αξιολογεί αλγορίθμους ανίχνευσης αντικειμένων [32].

VEDAI Dataset

Το σετ δεδομένων VEDAI (VEDAI dataset) περιέχει ορθοκανονικές δορυφορικές εικόνες που συλλέχθηκαν από το Utah AGRC και υπάρχουν ελεύθερες προς διάθεση για ερευνητικούς μη εμπορικούς σκοπούς. Οι εικόνες που αποτελούν το dataset είναι από το σετ HRO 2016 6in., έχουν χωρική ανάλυση 4.92 x 4.92 ίντσες/pixel περίπου δηλαδή 12.5 x 12.5 εκατοστά/pixel και η περίοδος λήψης τους είναι από το έτος 2012.

Οι εικόνες διαθέτουν τέσσερα φασματικά κανάλια ή μπάντες (Πίνακας 3.1): τρεις στο ορατό φάσμα (κόκκινο, πράσινο, μπλε) και μια στο εγγύς υπέρυθρο (NIR). Το αρχικό μέγεθος των εικόνων είναι ιδιαίτερα μεγάλο και για λόγους ευκολότερης διαχείρισης τους, χωρίστηκαν σε μικρότερα τμήματα. Με αυτό το τρόπο όχι μόνο καθίστανται πιο διαχειρίσιμες αλλά μειώνεται και η παρουσία επαναλαμβανόμενων μοτίβων, όπως για παράδειγμα βουνά και λίμνες τα οποία έχουν την πιθανότητα να προσθέσουν bias λόγω της υπερεκπροσώπησης τους στον αλγόριθμο. Το dataset που προέκυψε λοιπόν περιλαμβάνει τμήματα εικόνων που έχουν επιλεχθεί χειροκίνητα από τους συγγραφείς και σκοπός τους ήταν να δώσουν ποικιλία στο dataset (Εικόνα 3.2) ειδικά ως προς τα οχήματα και τα διαφορετικά background [32].

Πίνακας 3.1: Η βασική διάρθρωση του VEDAI dataset [32].

Ονομα	**Image size**	**Ανάλυση (cmpp)**	**#channels**	**Channel type**
Large-size color images (LCIs)	1024x1024	12.5 x 12.5	3	Colors
Small-size color images (SCIs)	512x512	25.0 x 25.0	3	Colors
Large-size infrared images (LIIs)	1024x1024	12.5 x 12.5	1	Near infrared
Small-size infrared images (SIIs)	512x512	25.0 x 25.0	1	Near infrared

Εικόνα 3.1: Δείγματα έγχρωμων (επάνω σειρά) και υπέρυθρων εικόνων (κάτω σειρά) από το VEDAI dataset [32].

Στο τελικό σύνολο επιλέχθηκαν 1210 εικόνες με ανάλυση 1024 x 1024, χωρισμένες σε τέσσερις κατηγορίες (Πίνακας 3.2) ανάλογα με το είδος των εικόνων (large – size color images, small – size color images, large – size infrared images, small – size infrared images). Μια τεχνική λεπτομέρεια είναι ότι εικόνες που ανήκουν στις μικρές κατηγορίες έχουν ανάλυση 512 x 512, ώστε τα αντικείμενα να απεικονίζονται μικρότερα και να αυξάνεται η δυσκολία εύρεσης τους. Οι έγχρωμες εικόνες αποτελούνται από τα τρία κλασικά κανάλια (R,G,B) και οι υπέρυθρες εικόνες (NIR) από ένα κανάλι 8 bit στη φασματική ζώνη του εγγύς υπέρυθρου. Αξίζει να σημειωθεί ότι όλες οι εικόνες έχουν ληφθεί από σταθερό ύψος, γεγονός που αποτελεί μεν μειονέκτημα καθώς ο αλγόριθμος εκπαιδεύεται αποκλειστικά σε αυτό το «είδος» λήψης εικόνων και δεν μπορεί να γενικεύσει αποτελεσματικά το μεταβαλλόμενο ύψος, αλλά ταυτόχρονα συνιστά πλεονέκτημα σε περιπτώσεις που η εφαρμογή είναι όμοια με το dataset. Τέλος, άλλο ένα πλεονέκτημα του VEDAI dataset αποτελεί η εμφάνιση οχημάτων σε ποικίλα background το οποίο ενισχύει την ικανότητα των αλγορίθμων να μαθαίνουν και να γενικεύουν καλύτερα. Παρακάτω (πίνακας 3.2) παρουσιάζεται ο πίνακας με τα στατιστικά.

Πίνακας 3.2: Τα στατιστικά του VEDAI dataset [32].

Class name	Tag	Targets per fold	Total	Orientation
Boat	Boa	17	170	[-π π]
Camping Car	Cam	39	390	[0 π]
Car	Car	134	1340	[-π π]
Others	Oth	20	200	[0 π]
Pickup	Pic	95	950	[-π π]
Plane	Pla	-	47	[-π π]
Tractor	Tra	19	190	[-π π]
Truck	Tru	30	300	[-π π]
Vans	Van	10	100	[-π π]
Small land vehicles	slv	295	2950	[-π π]
Large land vehicles	llv	69	690	[0 π]

Εικόνα 3.2: Εικόνα από το VEDAI dataset με επισημασμένα αντικείμενα [32].

Όσον αφορά στα οχήματα έχουμε εννιά κατηγορίες, plane, boat, camping car, pick up, tractor, truck van, other (Πίνακας 3.2 και Εικόνα 3.3). Κατά μέσο όρο υπάρχουν 5,5 οχήματα ανά εικόνα και καταλαμβάνουν το 0.7% των pixels της κάθε εικόνας. Η μικρή αναλογία του αντικειμένου ως προς την εικόνα δείχνει και την δυσκολία του προβλήματος της ανίχνευσης μικρών αντικειμένων. Οι ετικέτες (Εικόνα 3.3) των κατηγοριών καταχωρήθηκαν από άνθρωπο και η διαδικασία περιλάμβανε πληροφορίες σχετικά με τον προσανατολισμό του αντικειμένου, καθώς και αν αυτό εμφανιζόταν πλήρως ή μερικώς.

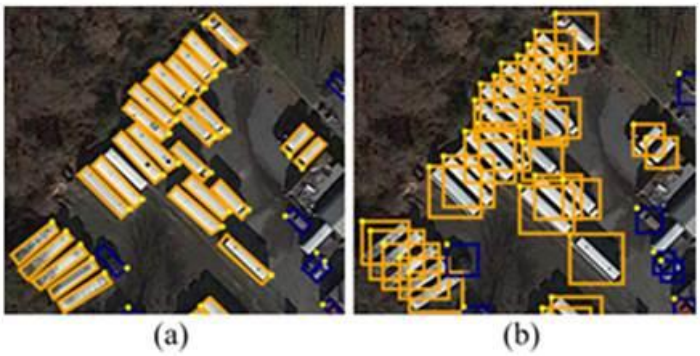
Για την αξιολόγηση του VEDAI dataset χρησιμοποιήθηκε η μέθοδος ten fold cross validation. Η μόνη διαφορά με την κλασική μέθοδο που διαχωρίζει το σετ τυχαία είναι, ότι εδώ τα folds ήταν προκαθορισμένα ώστε να υπάρχει περίπου ίδιος αριθμός οχημάτων. Οι εικόνες σε κάθε fold ήταν προκαθορισμένες, γεγονός που είχε ιδιαίτερη σημασία καθώς κάθε εικόνα χρησιμοποιήθηκε μόνο μια φορά στο testing. Από την αξιολόγηση του dataset και με τη χρήση αξιόπιστων αλγορίθμων ανίχνευσης αντικειμένων διαπιστώθηκε ότι κανένας δεν παρείχε καλά ή αξιοποιήσιμα αποτελέσματα. Με αυτό το τρόπο ενισχύεται το γεγονός ότι η ανίχνευση μικρών αντικειμένων είναι ένα απαιτητικό πρόβλημα και αναδεικνύει την ανάγκη ανάπτυξης πιο αποδοτικών και εξειδικευμένων προσεγγίσεων στην ανίχνευση μικρών αντικειμένων.

Εικόνα 3.3: Κατηγορίες αντικειμένων στο VEDAI dataset [32].

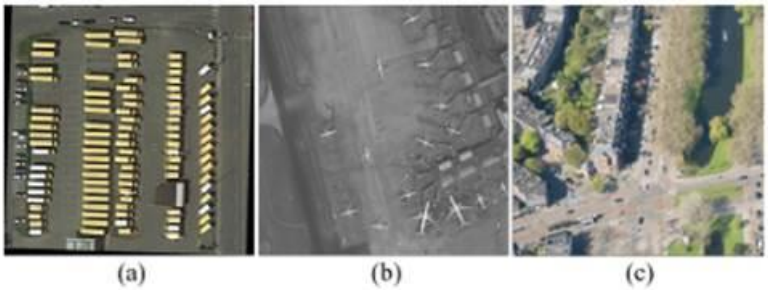
DOTA Dataset

Το DOTA dataset [33] περιέχει αποκλειστικά δορυφορικές εικόνες. Οι λήψεις προέρχονται από διάφορες πλατφόρμες και αισθητήρες παρέχοντας ένα μεγάλο εύρος σε ότι αφορά τα ποιοτικά χαρακτηριστικά των εικόνων. Η βασικά διαφορά σε σχέση με άλλα dataset είναι ότι χρησιμοποιούνται τα OBB (Oriented Bounding Boxes) αντί των οριζοντίων πλαισίων (Horizontal Bounding Boxes) (Εικόνα 3.4), που συνήθως έχουν τα υπόλοιπα dataset, επιτρέποντας καλύτερη και ακριβέστερη οριοθέτηση των αντικειμένων που υπάρχουν στις εικόνες.

Το dataset έχει δύο βασικές εκδόσεις, την DOTA v1.0 και την DOTA v2.0 με τη δεύτερη να αποτελεί ουσιαστικά προσθήκη στην πρώτη [33]. Στην δεύτερη έκδοση του περιέχονται περίπου 1.8 εκατομμύρια αντικείμενα ταξινομημένα σε 18 κατηγορίες. Οι εικόνες προέρχονται από τις δορυφορικές πλατφόρμες Gaofen-2 (GF-2), Jilin-1 (JL-1), από την υπηρεσία της Cyclomedia (*https://www.cyclomedia.com/*) (Εικόνα 3.5) που παρέχει εναέριες εικόνες της περιοχής του Rotterdam, και από την υπηρεσία του Google Earth που παρέχει παγκόσμια κάλυψη.



Εικόνα 3.4: Συγκριτική απεικόνιση HBB πλαισίων (a) και OBB πλαισίων (b) [33].



Εικόνα 3.5: Δείγματα εικόνων από Google Earth (a), δορυφόρους GF/JL (b) και την υπηρεσία Cyclomedia (c) [33].

Η ανάλυση των εικόνων έχει ένα εύρος από 800 x 800 έως 4000 x 4000 pixels για τις εικόνες από το Google Earth. Για το Cyclomedia η αντίστοιχη ανάλυση είναι 29200 x 27600 και για ένα μέρος των δορυφορικών εικόνων του GF-2, που συλλέχθηκαν αργότερα, είναι 7360 x 4912. Για όλες τις υπόλοιπες πλατφόρμες διατηρήθηκαν τα αρχικά μεγέθη. Αυτή η διαφορά στην ανάλυση των εικόνων ανάλογη των πηγών τους δίνει ετερογένεια που αυξάνει την δυσκολία αλλά ανταποκρίνεται καλύτερα στην πραγματικότητα.

Οι δεκαοχτώ κατηγορίες όπως αναγράφονται από τους συγγραφείς [33] είναι plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, swimming pool, container crane, airport, helipad. Οι εικόνες του dataset είναι RGB ή grayscale. Πιο συγκεκριμένα οι δορυφορικές εικόνες από τις υπηρεσίες Cyclomedia και Google Earth είναι έγχρωμες RGB ενώ αυτές των GF-2 και JL-1 είναι grayscale 8-bit (μετά την μετατροπή τους από τα 10 bit που ήταν αρχικά).

Ένα άλλο στοιχείο του dataset είναι η χωρική ανάλυση των εικόνων, που στη συγκεκριμένη περίπτωση, οι συγγραφείς την ονομάζουν Ground Sample Distance (GSD). Σε όλες τις εικόνες το κάθε τους pixel αντιστοιχεί σε πραγματικές διαστάσεις στο έδαφος. Η σημασία της GSD είναι πολύ σημαντική καθώς μας δείχνει την ποιότητα της ευκρίνειας της εικόνας. Μια εικόνα με μικρό GSD σημαίνει ότι έχει την ικανότητα να δείχνει μικρά αντικείμενα σε αντίθεση με μια εικόνα με μεγάλο GSD που αρχίζει να γενικεύει τα αντικείμενα στο έδαφος. Στο dataset οι εικόνες από τον GF-2 έχουν ανάλυση 0.81 meters/pixel, οι εικόνες από τον JL-1 έχουν ανάλυση 0.72 meters/pixel και οι εικόνες από την υπηρεσία Cyclomedia έχουν ανάλυση 0.10 meters/pixel. Η καλύτερη χωρική ανάλυση παρέχεται στις εικόνες από την υπηρεσία Cyclomedia η οποίες δεν χαρακτηρίζονται ως δορυφορικές. Οι δορυφορικές εικόνες του JL-1 είναι αυτές με την καλύτερη ανάλυση. Τέλος, οι εικόνες από το Google Earth προσφέρουν ανάλυση από 0.1 έως 4.5 meters/pixel.

Ως προς την αξιολόγηση χρησιμοποιήθηκαν δύο μεθοδολογίες, όπου η πρώτη βασίζεται στα HBB πλαίσια και η δεύτερη στα OBB πλαίσια. Για μετρική χρησιμοποιήθηκε η Mean Average Precision (mAP) και η Intersection Over Union (IoU). Στην αξιολόγηση με βάση τα HBB πλαίσια ακολουθήθηκαν οι εξής διαδικασίες: η μια αφορά σε απευθείας προβλέψεις με HBB πλαίσια και η άλλη σε προβλέψεις στα OBB πλαίσια που στην συνέχεια μετατράπηκαν σε HBB. Για τις απευθείας προβλέψεις έγινε χρήση των μοντέλων RetinaNet, Mask R-CNN, Cascade Mask R-CNN, Hybrid Task Cascade και ο Faster R-CNN. Για τις προβλέψεις με OBB πλαίσια η διαδικασία ήταν λίγο περισσότερο σύνθετη, καθώς οι πιο πολλοί αλγόριθμοι ανίχνευσης αντικειμένων δεν είναι φτιαγμένοι για να λειτουργούν με OBB πλαίσια το οποίο οδήγησε και στην μετατροπή τους. Αρχικά, το HBB head άλλαξε με ένα OBB head, στην συνέχεια το Mask head αξιολογεί τα OBB πλαίσια σε επίπεδο pixel με βάση τις περιοχές ενδιαφέροντος, ή αλλιώς Region of Interest (RoI). Η αξιολόγηση με OBB πλαίσια είναι και αυτή μια σύνθετη διαδικασία, καθώς δεν είναι τόσο διαδεδομένη, ωστόσο προσφέρει καλύτερη απόδοση ανίχνευσης αντικειμένων σε εικόνες που έχουν ληφθεί από αέρος.

Πραγματοποιήθηκε ένας πολύ μεγάλος αριθμός πειραμάτων, με τα αποτελέσματα εβδομήντα εξ' αυτών να παρουσιάζονται συνοπτικά παρακάτω. Οι δοκιμές έγιναν στις τρεις εκδόσεις του dataset, τις DOTA v1.0, DOTA v1.5 και DOTA v2.0, και με αλγορίθμους one-stage και two-stage. Για όλους τους αλγορίθμους καταγράφηκαν οι μετρικές mAP με HBB και OBB πλαίσια εκτός από τους αλγορίθμους RetinaNet και Faster R-CNN, καθώς αυτές αδυνατούν να υποστηρίξουν OBB head. Συγκεκριμένα για αυτούς τους δυο αλγορίθμους, τα HBB πλαίσια μετατράπηκαν σε OBB. Παρατηρήθηκε ότι για τους αλγορίθμους που υποστηρίζουν HBB πλαίσια και OBB πλαίσια η μετρική mAP με τα OBB είναι συνήθως λίγο μικρότερη σε σχέση με την αντίστοιχη της στα HBB πλαίσια. Αυτό οφείλεται στο γεγονός ότι τα OBB πλαίσια απαιτούν πολύ καλύτερο εντοπισμό σε σύγκριση με τα HBB και κάνουν την διαδικασία της πρόβλεψης αρκετά απαιτητική αυξάνοντας έτσι τον βαθμό της δυσκολίας, ειδικά στον υπολογισμό της IoU με τα ground truth πλαίσια.

Στην διαδικασία της αξιολόγησης χρησιμοποιήθηκε και επαύξηση δεδομένων (Data Augmentation). Πιο συγκεκριμένα, στα πειράματα με το μοντέλο Faster R-CNN OBB + RoI (Region of Interest) Transformer με backbone το ResNet-50 FPN εφαρμόστηκαν πέντε τεχνικές επαύξησης, οι patch overlap, multi – scale training, multi – scale testing, rotation training και rotation testing. Αξίζει να σημειωθεί ότι στα αποτελέσματα των πολύ μικρών αντικειμένων (κάτω από δέκα pixel), στον αλγόριθμο Faster R-CNN OBB RoI Transformer, το AP στα μικρά αντικείμενα από 77.45% στο dataset DOTA v1.0 έφτασε στο 52.05% στο dataset v1.5. Μια άλλη παρατήρηση είναι ότι τα OBB πλαίσια βοηθάνε στην αύξηση της απόδοσης του αλγορίθμου σε περιβάλλοντα με πολύ πυκνά αντικείμενα. Για παράδειγμα, κατά τη χρήση των αλγόριθμων Faster R-CNN OBB και Faster R-CNN, ο πρώτος σημείωσε AP υψηλότερο κατά 8 μονάδες από τον δεύτερο στην κλάση των μεγάλων οχημάτων του dataset DOTA v1.0, κάνοντας εμφανή την αποτελεσματικότητα των OBB πλαισίων.

Τα αποτελέσματα αναδεικνύουν ότι η επιλογή των υπερπαραμέτρων (hyperparameters) και ο σχεδιασμός του αλγορίθμου είναι πολύ διαφορετική για τον εντοπισμό μικρών αντικειμένων σε σύγκριση με άλλες εργασίες που αφορούν εικόνες και ακόμα ειδικότερα εναέριες εικόνες. Οι βέλτιστες ρυθμίσεις παραμέτρων διαφοροποιούνται σε σχέση με άλλες γενικές εφαρμογές μηχανικής όρασης ειδικά όταν πρόκειται για εικόνες φυσικού κόσμου όπου η κλίμακα, η πυκνότητα και η γωνία θέασης διαφοροποιούνται. Σαφές έγινε επίσης ότι το dataset DOTA μπορεί να χρησιμοποιηθεί βοηθητικά για εργασίες που περιλαμβάνουν εικόνες του πραγματικού κόσμου, και ιδιαίτερα του φυσικού, για την ανάπτυξη μεθοδολογιών και αλγορίθμων για τον εντοπισμό αντικειμένων [33].

FAIR1M Dataset

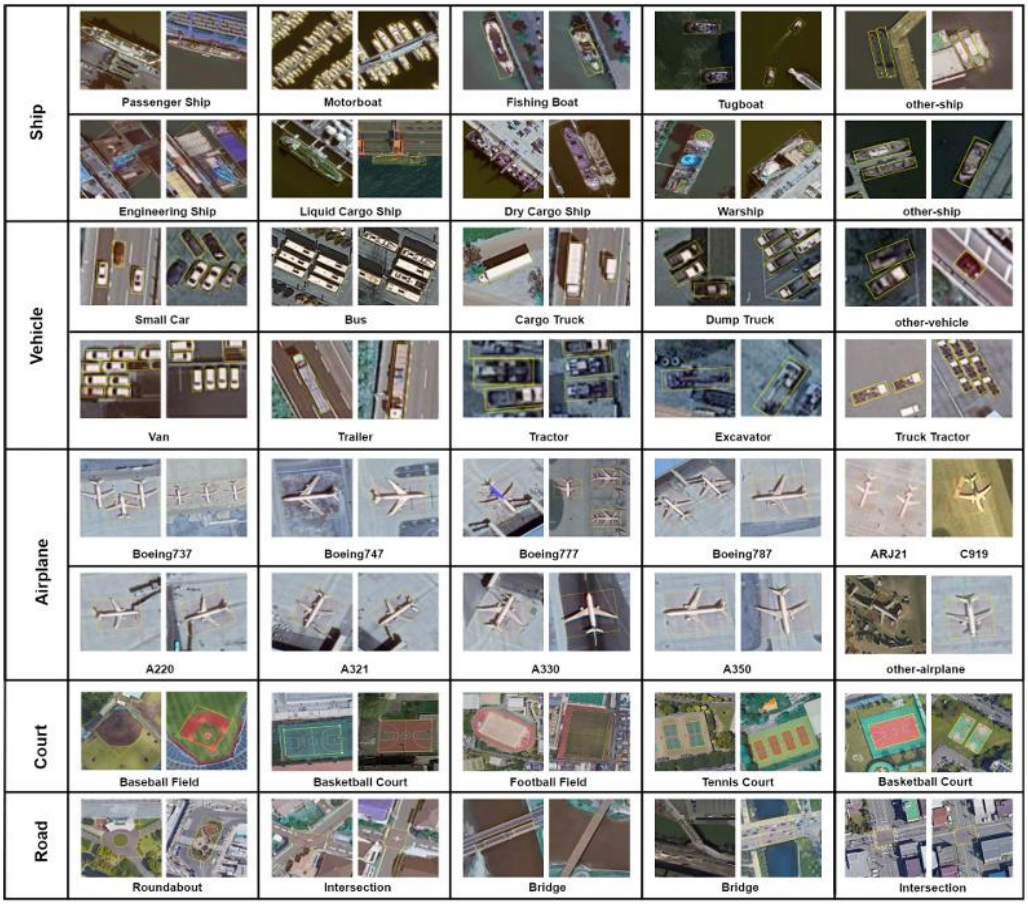
Το dataset FAIR1M [34] αποτελεί μια συλλογή δορυφορικών εικόνων. Ο αριθμός τους είναι περίπου 15000 με πάνω από ένα εκατομμύριο αντικείμενα οριοθετημένα πάνω σε αυτές. Η χωρική ανάλυση των εικόνων έχει εύρος από 0.3 μέτρα έως 0.8 μέτρα, παρέχοντας μεγάλη ανάλυση το οποίο διευκολύνει στη διάκριση πολύ μικρών αντικειμένων. Οι εικόνες προέρχονται από διάφορες πλατφόρμες και καλύπτουν αρκετές χώρες σε διαφορετικές ηπείρους. Τα αντικείμενα που έχουν οριοθετηθεί ανήκουν σε 5 κατηγορίες και σε 37 υποκατηγορίες. Τα κύρια χαρακτηριστικά του είναι ότι υπερτερεί σε μέγεθος από τα υπάρχοντα dataset, παρέχει πολύ υψηλότερη λεπτομέρεια αντικειμένων από δορυφορικές εικόνες, οι οποίες περιέχουν ενσωματωμένες πληροφορίες όπως γεωγραφικό μήκος, γεωγραφικό πλάτος και χωρική ανάλυση, ενώ το dataset παράχθηκε με τρόπο που εξασφαλίζει καλύτερη καθαρότητα και ποιότητα [34].

Η συλλογή των εικόνων βασίζεται στους δορυφόρους Gaofen ενώ στο Google Earth η ανάλυση εδάφους κυμαίνεται από 0.3 μέτρα έως 0.8 μέτρα. Οι εικόνες που περιέχονται στο dataset έχουν περάσει από αρκετά στάδια προεπεξεργασίας ώστε να έχουν υψηλή ποιότητα, δεδομένου ότι οι δορυφορικές εικόνες έχουν να αντιμετωπίσουν, εκτός των άλλων, προβλήματα νεφοκάλυψης, θορύβου και σημείων που εμφανίζουν μεγάλη φωτεινότητα (καμένα). Εικόνες που προβάλλουν τέτοιου είδους ζητήματα έχουν εξαιρεθεί από το dataset. Δεδομένου ότι οι συγγραφείς επιθυμούσαν να συμπεριλάβουν πληροφορίες θέσης, οι εικόνες ελέγχθηκαν και διορθώθηκαν ως προς τον εξωτερικό τους προσανατολισμό. Μετά τις διορθώσεις παρήχθησαν ορθογωνικές εικόνες, οι οποίες ήταν είτε παγχρωματικές είτε πολυφασματικές. Τέλος, οι πολυφασματικές εικόνες συνδυάστηκαν με τις παγχρωματικές για να ενισχυθεί η χωρική ανάλυση των πολυφασματικών μέσω αλγορίθμων Pan Sharpening, όπως επίσης χρησιμοποιήθηκε και ισοστάθμιση ιστογράμματος για να αποδοθεί καλύτερα ο χρωματικός τόνος των εικόνων (hue).

Τα ποιοτικά χαρακτηριστικά του dataset, ταξινομούνται σε πέντε βασικές κατηγορίες (Εικόνα 3.6) οι οποίες είναι, «αεροπλάνα», «πλοία», «δρόμοι», «γήπεδα», και «οχήματα». Οι πέντε αυτές κατηγορίες χωρίζονται σε 37 υποκατηγορίες ανάλογα, κυρίως, με το μέγεθος του αντικειμένου της βασικής κατηγορίας. Για παράδειγμα, στην κατηγορία «αεροπλάνα» περιλαμβάνονται μεταξύ άλλων, τα Airbus A320 και Airbus A330. Ομοίως, στην κατηγορία «πλοία» συναντάται το «επιβατηγό πλοίο» (passenger ship) και το «ρυμουλκό» (tug boat), ενώ στην κατηγορία «οχήματα» περιλαμβάνονται το «μικρό αυτοκίνητο» (small car) και το «φορτηγό» (truck). Σε όλες τις κατηγορίες υπάρχει η υποκατηγορία «άλλο» (“other”), για να συμπεριληφθούν αντικείμενα που δεν περιγράφονται στις οριζόμενες υποκατηγορίες.

Οι συγγραφείς εστίασαν σε πέντε σημεία με σκοπό την βέλτιστη παραγωγή ενός dataset ανίχνευσης μικρών αντικειμένων, καλύπτοντας κενά από προηγούμενες προσπάθειες. Στο πρώτο σημείο, δεδομένου ότι ο στόχος τους ήταν η δημιουργία ενός dataset από δορυφορικές εικόνες, προσκάλεσαν ειδικούς στην ανάλυση και ερμηνεία αυτών των εικόνων. Σκοπός ήταν να βοηθήσουν με την επιλογή των βέλτιστων κατηγοριών αντικειμένων που έπρεπε να συμπεριληφθούν, καθώς και στον καθορισμό των αντικειμένων που απαιτούσαν λεπτομερή περιγραφή. Στο δεύτερο σημείο, χρησιμοποίησαν εικόνες από διάφορους αισθητήρες με σκοπό την συλλογή εικόνων με διαφορετικές αναλύσεις και προσανατολισμούς. Στο τρίτο σημείο, εστίασαν στην απόκτηση μεγάλων within – class variation και between – class similarity. Αν και τα αντικείμενα έχουν κατηγοριοποιηθεί σε ξεκάθαρες και διακριτές κατηγορίες, ορισμένα παρουσιάζουν ομοιότητες στο μέγεθος ή σχήμα αλλά και στις περιβαλλοντικές τους σκηνές. Για να πετύχουν μεγάλη διακύμανση εντός της ίδιας κλάσης, συνέλεξαν σκηνές από διαφορετικές εποχές και περιβαλλοντικές συνθήκες, με αποτέλεσμα τα αντικείμενα να εμφανίζουν

διακυμάνσεις στη θέση, το χρώμα και το φόντο τους. Στο τέταρτο σημείο, επέλεξαν αρκετά σύνθετες σκηνές με αντικείμενα σε πολύ πυκνή διάταξη. Μετά την συλλογή εικόνων από τέτοια περιβάλλοντα, όπως για παράδειγμα, σταυροδρόμια και πάρκινγκ, τα αντικείμενα, που εμφανίστηκαν, σημάνθηκαν χειροκίνητα ένα προς ένα. Τέλος, στο πέμπτο σημείο απασχολήθηκαν με τη γεωγραφική πληροφορία. Το dataset τους περιλαμβάνει εκτός από την πληροφορία της ανάλυσης, το γεωγραφικό μήκος και το γεωγραφικό πλάτος. Η πληροφορία αυτή είναι χρήσιμη, καθώς επιτρέπει τη χρήση της χωρικής διάστασης και, δεδομένου ότι οι σκηνές έχουν ληφθεί επανειλημμένα σε διαφορετικούς χρόνους, καθιστά δυνατή και τη χρονική ανάλυση (ανάλυση χρονοσειράς - time series analysis) συγκεκριμένων αντικειμένων ή περιοχών.



Εικόνα 3.6: Οι βασικές κατηγορίες εικόνων του FAIR1M Dataset [34].

Ως προς την αξιολόγηση του dataset χρησιμοποιήθηκαν οι μετρικές Fine – grained Intersection – Over – Union (FIoU) και Fine – grained mean Average Precision (mAPF) [34]. Η μετρική FIoU είναι μια τροποποιημένη IoU που σκοπό έχει να «τιμωρεί» τα υπερβολικά καλά αποτελέσματα, ώστε να υπάρχει καλύτερη αντιπροσώπευση της πραγματικότητας ειδικά σε περιπτώσεις πολύ μικρών ή πυκνών αντικειμένων. Η μετρική mAPF [34] λαμβάνει υπόψη τον τύπο της ανίχνευσης. Για παράδειγμα, αν ένα πλαίσιο έχει σκορ μεγαλύτερο από 0.5, θεωρείται ότι είναι TP (True Positive), διαφορετικά χαρακτηρίζεται ως FP (False Positive). Με βάση αυτά τα στοιχεία, υπολογίζονται τα precision και recall. Στη συνέχεια, από όλες τις τιμές του precision υπολογίζεται η mAPF, η οποία παρουσιάζει μεγαλύτερη ευαισθησία στην λεπτομερή κατηγοριοποίηση σε σχέση με την μετρική mAP.

Συνεπώς, τα αποτελέσματα φανερώνουν ότι το FAIR1M dataset έχει αυξημένη δυσκολία και είναι αρκετά ανταγωνιστικό ως προς τα μοντέλα που δοκιμάστηκαν στις εικόνες του. Ένα άλλο πολύ βασικό

χαρακτηριστικό που το διαφοροποιεί από τα άλλα datasets, είναι η ύπαρξη αρκετά λεπτομερών κατηγοριών ως προς τον διαχωρισμό των αντικειμένων του, καθώς και η παροχή γεωγραφικών συντεταγμένων. Στις δοκιμές που έγιναν χρησιμοποιήθηκαν οι αλγόριθμοι Faster R-CNN, RetinaNet και Cascade R-CNN, όπως επίσης και οι Gliding Vertex και ROI Transformer. Το backbone δίκτυο ήταν το ResNet-101. Τα τρία πρώτα μοντέλα αντιπροσωπεύουν region based τεχνικές εύρεσης μικρών αντικειμένων με οριζόντια πλαίσια και τα δύο τελευταία μοντέλα αντιπροσωπεύουν τεχνικές με προσανατολισμένα πλαίσια. Οι αρχικές εικόνες του dataset είχαν πολύ μεγάλο μέγεθος, και για να μπορέσουν να χρησιμοποιηθούν «κόπηκαν» σε patches (τμήματα) μεγέθους 1024 x 1024 και stride (βήμα) 256. Κατόπιν σύγκρισης με το dataset DOTA, το FAIR1M παρουσιάζει αυξημένη δυσκολία στην εκπαίδευση μοντέλων ανίχνευσης μικρών αντικειμένων, καθώς η διαφορά της μέσης τιμής της μετρικής mAP ανέρχεται στα 27.92% με βάση το μοντέλο ROI Transformer-D και στο 6.77% με βάση το μοντέλο ROI Transformer-F.

xView Dataset

Το dataset xView [35] είναι και αυτό μια συλλογή από δορυφορικές εικόνες με στόχο τον εντοπισμό μικρών αντικειμένων. Οι εικόνες της συλλογής (εικόνα 3.7) προέρχονται από τον δορυφόρο WorldView – 3 με χωρική ανάλυση 0.3m, παρέχοντας εικόνες υψηλής ακρίβειας, κατάλληλες για χρήση σε ανάγκες εντοπισμού μικρών αντικειμένων. Ο σχεδιασμός με βάση τους συγγραφείς Lam et. al. [35], έγινε τέτοιο τρόπο ώστε το dataset να είναι κατανοητό τόσο από μελετητές τηλεπισκόπησης όσο και από μελετητές μηχανικής όρασης, και βασίστηκε σε τέσσερα σημεία.

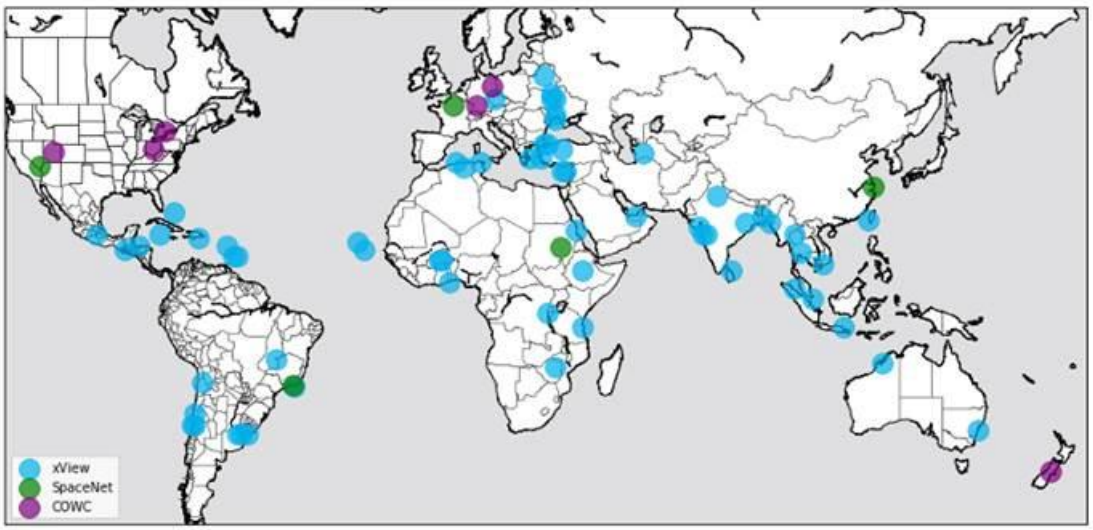
Εικόνα 3.7: Ενδεικτικά δείγματα δορυφορικών εικόνων του dataset xView [35].

Αρχικά, επιλέχθηκε ένας δορυφόρος υψηλής ακρίβειας για να καλυφθεί το κενό που προκύπτει από την απουσία ακριβούς ανάλυσης στα datasets. Δεύτερον, επέλεξαν να αποτυπώσουν μια ρεαλιστική κατανομή στο dataset, αντί για μια τεχνητά ιδανική με την ενίσχυση του learning efficiency στο dataset,

κάτι που σχετίζεται με τον αριθμό εμφανίσεων των αντικειμένων κάθε κατηγορίας και με το αν η κατανομή τους είναι balanced ή unbalanced. Τρίτον, όρισαν εξήντα κλάσεις για την κατηγοριοποίηση των αντικειμένων, αριθμός σημαντικά υψηλότερος σε σύγκριση με άλλα datasets δορυφορικών ή εναέριων εικόνων, οργανώνοντας το dataset όσο το δυνατόν καλύτερα από άποψη επιλογής κατηγοριών.

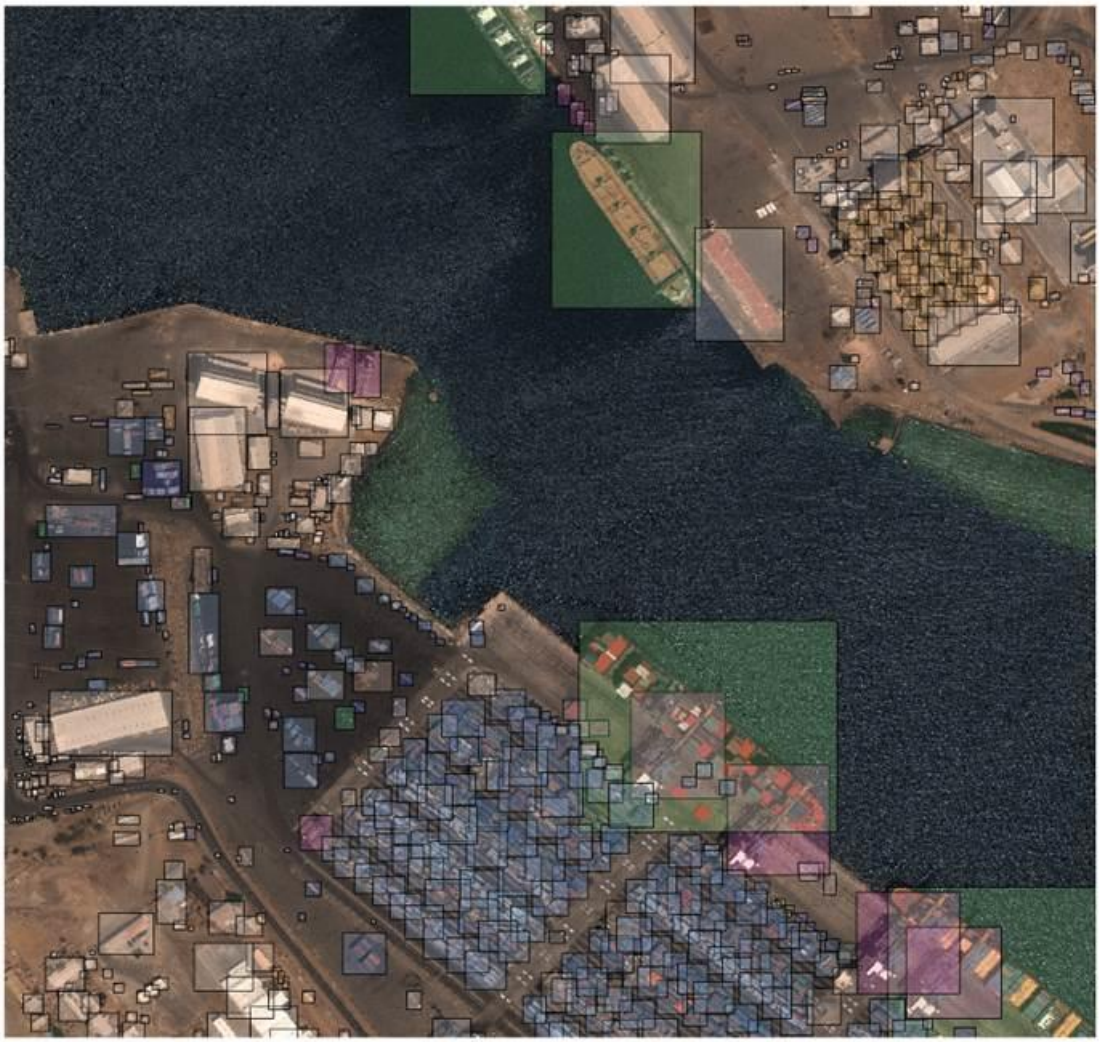
Τέλος, βελτίωσαν την ανίχνευση αντικειμένων στις πολύ ειδικές κλάσεις. Σε αντίθεση με άλλα datasets που περιορίζονται σε γενικές κατηγορίες, το συγκεκριμένο εισάγει 7 βασικές κλάσεις κάθε μια από τις οποίες χωρίζεται περαιτέρω σε αρκετές υποκατηγορίες. Θεώρησαν ότι έχει μεγάλη σημασία να διακρίνεται, για παράδειγμα, αν σε μια εικόνα εντοπιστεί ένα «πλοίο μεταφοράς κοντέινερ» (shipping container) ή ένα «ιστιοφόρο» (yacht), παρόλο που και τα δυο ανήκουν στην γενική κατηγορία «πλοία» (maritime vessels). Όσον αφορά τις ίδιες τις δορυφορικές εικόνες, αυτές υποβλήθηκαν σε διαδικασία προεπεξεργασίας, που περιλάμβανε την διόρθωση των ατμοσφαιρικών σφαλμάτων, την ορθοαναγωγή και το pan – sharpening, προκειμένου να αυξηθεί η χωρική τους ανάλυση. Για να μειωθεί το bias η συλλογή των εικόνων περιλαμβάνει σκηνές από διάφορες γεωγραφικές περιοχές (Εικόνα 3.8). Η κάθε μια με τις διαφορές και τις ιδιαιτερότητες της (διαφορετικό φόντο, όπως για παράδειγμα παράκτιες περιοχές, δάση ή πόλεις και δείγματα κτιρίων) για να συμβάλει στην ετερογένεια και σε ένα πιο πραγματικό δείγμα του κόσμου. Επιλέχθηκαν επίσης να μπουν σκηνές οι οποίες χρησιμοποιούνται στην πραγματικότητα. Το dataset έχει πάνω από ένα εκατομμύριο αντικείμενα και πάνω 1400 τετραγωνικά χιλιόμετρα επιφάνειας.

Οι περιοχές ενδιαφέροντος περιλαμβάνουν αστικές περιοχές και εξωαστικές περιοχές, εγκαταστάσεις υποδομών (αεροδρόμια, λιμάνια) όπως επίσης χερσαίες και νησιωτικές εκτάσεις. Για να μειωθεί το bias συλλέχθηκαν εικόνες από περιοχές διαφορετικών ηπείρων, ώστε να υπάρχει καλύτερη δειγματοληψία. Η συλλογή των περιοχών ενδιαφέροντος έγινε με βάση τις συντεταγμένες του (κάθε σημείου) που αντλήθηκαν από ανοιχτές βάσεις δεδομένων. Κατόπιν εντοπισμού των δορυφορικών εικόνων που καλύπτουν κάθε σημείο ενδιαφέροντος, σχεδιάστηκε περιμετρικά αυτών ένα πολύγωνο, το οποίο τελικά «κόπηκε» σε μια περιοχή ενός τετραγωνικού χιλιομέτρου γύρω από το σημείο. Η επιλογή των εικόνων έγινε με κριτήρια κάλυψης κλιματικών συνθηκών ή ποιοτικών χαρακτηριστικών που να μην επηρεάζουν την ευκρίνεια της εικόνας και των αντικειμένων, ώστε αυτά να είναι ορατά.



Εικόνα 3.8: Γεωγραφική διασπορά των dataset xView, SpaceNet και COWC. Το μέγεθος του κύκλου αντιστοιχεί στον αριθμό αντικειμένων.. Με γαλάζιο χρώμα είναι το dataset xView [35].

Σε ότι αφορά την διαδικασία της επισήμανσης - image annotation (Εικόνα 3.9), αυτή έγινε προσκαλώντας ειδικούς με εμπειρία στις κλάσεις οι οποίες έπρεπε να κατηγοριοποιηθούν. Εφαρμόστηκε ποιοτικός έλεγχος σε πολλά στάδια ώστε να εξασφαλιστεί η ποιότητα του dataset και να μειωθούν τα λάθη. Οι συμμετέχοντες εκπαιδεύτηκαν στην σωστή αναγνώριση και κατηγοριοποίηση των αντικειμένων σε όλες τις κλάσεις. Ο κάθε ένας συμμετέχοντας έπρεπε να κατηγοριοποιήσει όλα τα αντικείμενα που ανήκανε σε μια γενική κατηγορία και, αν αυτό ήταν δυνατό, σε μια ειδική. Αντικείμενα που ήταν ορατά σε ποσοστό μικρότερο του 20% εξαιρέθηκαν από τη σήμανση (label). Αντικείμενα που εμφανίζονταν σε ομάδες οριοθετήθηκαν ένα – ένα (για παράδειγμα, συγκροτήματα κατοικιών ή πυκνά παραταγμένα πλοία), αν και υπήρξαν περιπτώσεις όπου οριοθετήθηκαν ομαδικά λόγω σφάλματος του χειριστή.



Εικόνα 3.9: Διαδικασία επισημείωσης αντικειμένων (annotation) στο dataset xView [35].

Η αξιολόγηση έγινε χρησιμοποιώντας αλγορίθμους SSD (Single Shot Multibox Detector) σε πολλαπλά επίπεδα ανάλυσης για καλύτερη διαχείριση των προβλημάτων κλίμακας. Οι εκπαιδεύσεις πραγματοποιήθηκαν σε τρεις διαφορετικές εκδοχές του αρχικού dataset: η πρώτη ήταν η αρχική του μορφή, η δεύτερη ήταν σε muti – resolution, ενώ η τρίτη ήταν η επαυξημένη εκδοχή. Η καλύτερη επίδοση ήταν η επαυξημένη εκδοχή. Συνοπτικά, η muti – resolution έκδοση έδωσε mAP (Mean Average Precision) μεγέθους 0.2590 %, η αρχική έδωσε mAP 0.1456%, ενώ η επαυξημένη έδωσε mAP 0.1549%. Γίνεται κατανοητό ότι στο multi – resolution dataset η διαφορά είναι αρκετά μεγάλη σε σχέση

με τις υπόλοιπες δυο εκδοχές του dataset. Αυτό πιθανώς οφείλεται στο ότι η διαφορά κλίμακας του κάθε αντικειμένου έχει πολύ μεγάλη βαρύτητα. Ένα ακόμα συμπέρασμα είναι ότι το επαυξημένο dataset, δηλαδή αυτό με το μεγαλύτερο μέγεθος και τα περισσότερα δεδομένα, αύξησε σημαντικά το regularization (κανονικοποίηση), οπότε και τον χρόνο εκπαίδευσης [35].

VisDrone Dataset

Ένα άλλο πολύ σημαντικό dataset αποτελεί το VisDrone dataset [36]. Το VisDrone περιέχει εικόνες που έχουν ληφθεί υπό κλίση (Εικόνα 3.10) από μικρά εναέρια οχήματα. Αποτελεί προσπάθεια κάλυψης περιπτώσεων που απαιτούν ένα dataset με λήψεις από drone, και κυρίως αεροφωτογραφίες υπό κλίση. Το dataset προορίζεται επίσης για περιπτώσεις όπου η έρευνα εστιάζει όχι μόνο στον εντοπισμό αντικειμένων από στατικές εικόνες, αλλά και σε βίντεο. Οι εικόνες που λαμβάνονται από drone έχουν ορισμένες ιδιαιτερότητες, όπως το γεγονός ότι η γωνία θέασης τους δεν είναι ούτε σταθερή ούτε προκαθορισμένη. Η κλίμακα των αντικειμένων παρουσιάζει επίσης ζητήματα, καθώς μεταβάλλεται συνεχώς λόγω αλλαγών στο ύψος του drone. Επιπλέον, η κίνηση του drone, συχνά απότομη και απρόβλεπτη, προκαλεί motion blur, με αποτέλεσμα μια εικόνα ή συνηθέστερα ένα βίντεο να εμφανίζονται «θολά». Αυτό δημιουργεί την ανάγκη υψηλής απόδοσης μοντέλα, καθώς οι εικόνες εμφανίζουν χαρακτηριστικά που τα καθιστούν απαιτητικά στην εκπαίδευση.

Εικόνα 3.10: Παραδείγματα εικόνων από διαφορετικά datasets. Οι κατηγορίες αντικειμένων απεικονίζονται με διαφορετικά χρώματα. Τα διακεκομμένα πλαίσια υποδηλώνουν μειωμένη ορατότητα λόγω εμποδίων [36].

Το VisDrone [36] περιλαμβάνει 263 αποσπάσματα βίντεο με 179264 frame και 10209 εικόνες. Η ανάλυση των βίντεο είναι 3840 x 2610 και για τις εικόνες είναι 2000 x 1500. Η συλλογή έγινε σε 14 διαφορετικές πόλεις της Κίνας. Οι λήψεις μεταξύ τους έχουν διαφοροποιήσεις ως προς τις συνθήκες λήψης, όπως ο περιβάλλοντας φωτισμός και οι κλιματικές συνθήκες κατά την στιγμή της λήψης, γεγονός που καθιστά το VisDrone ένα dataset αντιπροσωπευτικό της πραγματικότητας.

Το dataset έρχεται ήδη χωρισμένο σε train, validation και test σετ: οι δύο πρώτοι φάκελοι χρησιμοποιούνται για την εκπαίδευση των μοντέλων (training) και τον έλεγχο της απόδοσης τους (validate), ενώ ο τρίτος φάκελος προορίζεται για χρήση στον διαγωνισμό που διοργανώθηκε τα έτη 2018, 2019, 2020 στα πλαίσια του IEEE International Conference on Computer Vision (ECCV).

Οι συγγραφείς οργάνωσαν διάφορες δοκιμασίες, όπως είναι αυτή του DET (image object detection track), όπου οι συμμετέχοντες καλούνταν να εντοπίσουν μέσα σε εικόνες αντικείμενα που ανήκουν σε

προκαθορισμένες κλάσεις, όπως για παράδειγμα πεζούς ή ποδήλατα. Μια άλλη ήταν αυτή του VID (object detection track), παρόμοια με του DET, αλλά με εφαρμογή σε βίντεο. Ακολούθως, η δοκιμασία του SOT (single object tracking track) απαιτούσε την παρακολούθηση ενός αντικειμένου από το πρώτο frame έως τα επόμενα, ενώ του MOT (multi object tracking track) προέβλεπε τον εντοπισμό πολλαπλών αντικειμένων σε κάθε frame και την ανάκτηση της τροχιάς τους.

Η παρούσα εργασία απασχολείται αποκλειστικά με το DET όπου στο σύνολο του που περιλαμβάνει συνολικά 10209 εικόνες, εκ των οποίων οι 6471 προορίζονται για εκπαίδευση, 548 για validation και οι υπόλοιπες 1610 για τεστ. Επιπλέον, οι κατηγορίες στις οποίες ταξινομούνται είναι οι: «άνθρωπος» (person), «πεζός» (pedestrian), «αυτοκίνητο» (car), «λεωφορείο» (bus), «κλειστό φορτηγάκι» (van), «φορτηγό» (truck), «σκεπαστό» «τρίκυκλο» (awning – tricycle), «τρίκυκλο» (tricycle), «μοτοσυκλέτα» (motor) και «ποδήλατο» (bicycle). Παράλληλα παρέχονται, εκτός από τα όρια του πλαισίου bounding box, και τα ποσοστά του occlusion και του truncation. Το occlusion αναφέρεται στο ποσοστό του αντικειμένου που καλύπτεται από άλλο αντικείμενο, εμπόδιο ή φόντο, ενώ το truncation αφορά στο ποσοστό ενός αντικειμένου που βρίσκεται εκτός των ορίων του frame της εικόνας. Όταν αυτό το ποσοστό υπερβαίνει του 50%, το αντικείμενο εξαιρείται από την διαδικασία της αξιολόγησης. Η ύπαρξη αυτών των δύο ιδιοτήτων αυξάνει την δυσκολία της ανίχνευσης μικρών αντικειμένων και υποδηλώνει ότι τα μοντέλα πρέπει να είναι ικανά να εντοπίζουν και αντικείμενα που εμφανίζονται μερικώς κρυμμένα από άλλα εμπόδια ή είναι κατά κάποιο ποσοστό αποκομμένα. Τέλος, η ιδιότητα truncation έχει οριστεί στο 50% βοηθώντας το μοντέλο να μην χάνει βαθμολογία αν ένα αντικείμενο είναι κομμένο και δε μπορεί να εντοπιστεί.

Η αξιολόγηση του συνόλου δεδομένων έγινε σε γνωστά μοντέλα ανίχνευσης αντικειμένων χρησιμοποιώντας μετρικές όπως η mAP (Mean Average Precision), η IoU (Intersection Over Union) και το Recall. Τα μοντέλα που χρησιμοποιήθηκαν ήταν κυρίως anchor based two-stages (δυο βημάτων βασισμένα σε άγκυρες). Αρχικά, το Faster R-CNN λειτουργεί με ένα δίκτυο RPN για να παράξει τις προτάσεις των περιοχών για τα bounding boxes. Στη συνέχεια, το FPN αυξάνει την αποδοτικότητα του μοντέλου, καθώς χειρίζεται καλύτερα προβλήματα κλίμακας. Επίσης, χρησιμοποιήθηκε το Light RCNN, το οποίο είναι ένα σχετικά μικρό δίκτυο R-CNN και ένα υποδίκτυο που αποτελείται από επίπεδα pooling και fully connected. Τέλος, χρησιμοποιήθηκε και το Cascade R-CNN, το οποίο φημίζεται για την υψηλή του απόδοση στον εντοπισμό και τον διαχωρισμό των ψευδώς αρνητικών.

Δοκιμάστηκαν επίσης, και τέσσερα μοντέλα one-shot (ενός βήματος). Ο YOLOv3 είναι μια εξέλιξη του YOLO9000, χρησιμοποιεί το Darknet-53, και αποτελεί ένα μοντέλο που ισορροπεί ανάμεσα στην ακρίβεια και στην ταχύτητα. Ο SSD προβλέπει τις θέσεις και τις κλίμακες των αντικειμένων, βασισμένος σε ένα πλήθος προεπιλεγμένων anchor boxes. Το μοντέλο RetinaNet αντιμετωπίζει καλύτερα δεδομένα που παρουσιάζουν imbalance.

Το μοντέλο RefineDet έχει ικανοποιητικότερη απόδοση σε σχέση με τα μοντέλα two-stage, καθώς μέσω του υποσυστήματος refinement module διαχωρίζει τα θετικά από τα αρνητικά anchors και στην συνέχεια, με την περαιτέρω επεξεργασία από το κυρίως δίκτυο του μοντέλου, προβλέπει τις θέσεις και το μέγεθος των αντικειμένων.

Για τα μοντέλα anchor free οι συγγραφείς κάνουν αναφορά στον CornerNet και στον CenterNet. Το πρώτο παίρνει ως αναφορά το κέντρο, την επάνω αριστερή γωνία και την κάτω δεξιά γωνία του bounding box ως μια τριπλέτα, προσπαθώντας να ανιχνεύσει τα αντικείμενα. Το δεύτερο μοντέλο παίρνει το κάθε αντικείμενο ως σημείο και υπολογίζει τα χαρακτηριστικά του όπως οι διαστάσεις, το μέγεθος και ο προσανατολισμός του.

Υψηλότερο mAP πέτυχε το μοντέλο DroneEye2020 που βασίζεται στο Cascade R-CNN με βαθμολογία 34.57% από την δοκιμασία VisDrone2020 και, έπειτα το μοντέλο HAL-RetinaNet από παλαιότερη δοκιμασία VisDrone2018 με βαθμολογία 31.88%.

3.1.2 Επιλογή δεδομένων

Τα σύνολα δεδομένων επιλέχθηκαν με γνώμονα και στόχο να περιλαμβάνουν και να απεικονίζουν μικρά αντικείμενα, να παρουσιάζουν αυξημένο βαθμό δυσκολίας και να διαθέτουν την καλύτερη δυνατή ποιότητα. Όλα τα σύνολα δεδομένων που αναλύονται και συμμετέχουν στην εκπαίδευση των μοντέλων είναι χαρακτηρισμένα ως benchmarks ή προέρχονται από αξιόπιστους ερευνητικούς οργανισμούς και ερευνητές. Τα σύνολα δεδομένων που επιλέχθηκαν έπρεπε να πληρούν κάποια κριτήρια όπως: (1) τα αντικείμενα που παρουσιάζουν να είναι μικρά, (2) να αποτελούνται από πολύ λίγα pixels και να καταλαμβάνουν ένα πολύ μικρό μέρος της συνολικής εικόνας, (3) να περιλαμβάνουν εικόνες υψηλής ανάλυσης χωρίς την ύπαρξη αναλυτικής πληροφορίας για τα αντικείμενα, (4) τα αντικείμενα να εμφανίζουν αλληλοεπικάλυψη, καθώς αυτό είναι ένα σημείο που δοκιμάζει όλα τα μοντέλα εντοπισμού μικρών αντικειμένων και (5) να καταλαμβάνουν μέρος του φόντου της εικόνας. Τέλος, (6) όλα τα σύνολα δεδομένων να αφορούν εναέριες λήψεις είτε χαμηλού (π.χ. λήψεις από UAV) είτε μεγάλου ύψους (π.χ. δορυφορικές λήψεις) και (7) να έχουν αξιολογηθεί από ερευνητές σε δοκιμασίες.

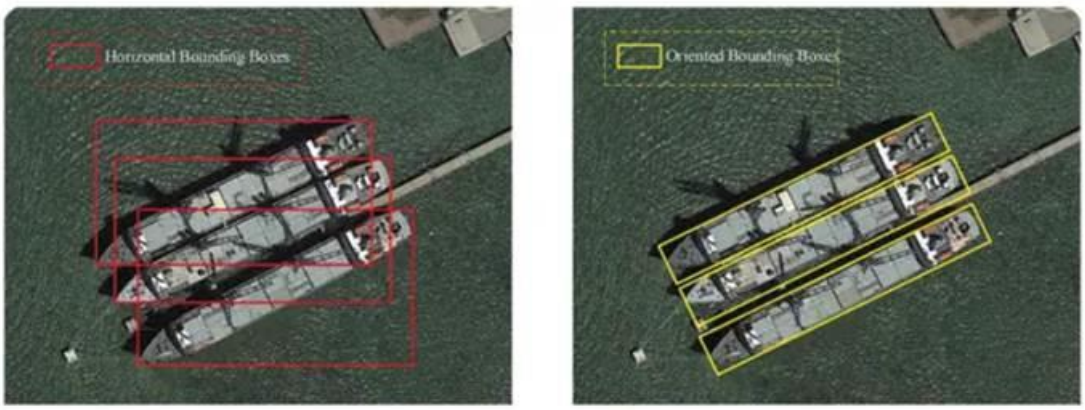
Επιπλέον, πρέπει να πληρούν και κάποια χαρακτηριστικά όπως: (1) να απεικονίζουν ποικιλομορφία περιβάλλοντος, δηλαδή αστικά και εξωαστικά περιβάλλοντα ή ορεινές και παραθαλάσσιες περιοχές, (2) να καλύπτουν είτε διαφορετικές χώρες είτε διαφορετικές πόλεις ως ελάχιστο ώστε να υπάρχουν αντικείμενα από διαφορετικές περιοχές και να αποφεύγεται το “locality bias”, (3) να εμφανίζουν ποικίλες κατηγορίες αντικειμένων, ώστε να παρέχεται καλύτερη αναπαράσταση του πραγματικού κόσμου και να υπάρχει αναλυτική κατηγοριοποίηση αυτών για την αξιολόγηση της ικανότητας του μοντέλου να διακρίνει διαφορετικές υποκατηγορίες ενός αντικειμένου, (4) να περιέχουν ήδη έτοιμα σημασμένα annotations, (5) να έχουν τεκμηριωθεί στη βιβλιογραφία και να θεωρούνται αξιόπιστα και αντικειμενικά, χωρίς να ευνοούν ένα μοντέλο έναντι άλλου. Παρακάτω αναλύονται τα κριτήρια και οι λόγοι που επιλέχθηκε το κάθε ένα από αυτά τα σύνολα δεδομένων.

VEDAI Dataset: Το VEDAI (Vehicle Detection in Aerial Imagery) είναι ένα σύνολο δεδομένων με εικόνες από εναέρια μέσα (UAV ή αεροσκάφη), περιλαμβάνει λήψεις χαμηλού ύψους και θεωρείται benchmark. Αποτελεί ένα multi-class σύνολο δεδομένων, με οχήματα διαφόρων τύπων σε πολλαπλά φόντα, όπου μέρος των αντικειμένων βρίσκεται σε περιοχές με μειωμένη ορατότητα και διαφορετικό προσανατολισμό.

DOTA Dataset: Το DOTA (Dataset for Object deTection in Aerial images) είναι και αυτό ένα multi-class σύνολο δεδομένων μικρών αντικειμένων που περιλαμβάνει τόσο δορυφορικές όσο και εναέριες λήψεις από UAV. Τα αντικείμενα στο συγκεκριμένο σύνολο, βρίσκονται σε προσανατολισμένα πλαίσια οριοθέτησης (Oriented Bounty Boxes - OBB), παρουσιάζουν μεγάλη αλληλοεπικάλυψη και πολύ μεγάλη γεωχωρική ποικιλομορφία.

FAIR1M Dataset: Το FAIR1M είναι ένα σύνολο δεδομένων εστιασμένο στο να διακρίνει λεπτές διαφορές ανάμεσα σε αντικείμενα της ίδιας κατηγορίας. Αποτελεί μέρος του ISPRS Benchmark και αποτελείται από 15000 εικόνες που απεικονίζουν ένα εκατομμύριο αντικείμενα. Οι εικόνες προέρχονται από έγχρωμες δορυφορικές λήψεις πολύ υψηλής ανάλυσης. Το συγκεκριμένο σύνολο δεδομένων καλύπτει πέντε βασικές κατηγορίες που αναλύονται σε 37 υποκατηγορίες και περιλαμβάνει τύπους οχημάτων και βασικές υποδομές. Επιλέχθηκε λόγω της λεπτομερούς διάκρισης που κάνει στα

αντικείμενα και στην ποικιλία των λήψεων ως προς το φόντο, συνθέτοντας πολύ χρήσιμες σκηνές, ενώ στη βελτιωμένη απόδοση συμβάλλει και η χρήση προσανατολισμένων πλαισίων οριοθέτησης (Oriented Bounty Boxes - OBB) (Εικόνα 3.11).



(a) Horizontal Bounding Boxes. (b) Oriented Bounding Boxes.

Εικόνα 3.11: Αριστερά (a) απεικονίζονται τρία πλοία με οριζόντια πλαίσια οριοθέτησης και δεξιά (b) απεικονίζεται η ίδια εικόνα με προσανατολισμένα πλαίσια οριοθέτησης [37].

Οι προκλήσεις που θέτει είναι η συμπερίληψη πολύ μικρών αντικείμενων, μέρος των οποίων βρίσκεται σε επικάλυψη ή σε πολύ πυκνή διάταξη, καθώς και η ανάγκη να αποτυπωθούν διαφορές σε επίπεδο υποκατηγοριών, γεγονός που καθιστά την διαδικασία υποκατηγοριοποίησης απαιτητική.

xView Dataset: Το xView είναι και αυτό ένα σύνολο δεδομένων δορυφορικών εικόνων πολύ υψηλής ανάλυσης. Η σύνθεση του έγινε από το U.S. National Geospatial - Intelligence Agency (NGA). Περιέχει ένα εκατομμύριο αντικείμενα σε 60 κλάσεις, καθιστώντας το στα μεγαλύτερα σύνολα δεδομένων που υπάρχουν διαθέσιμα ελεύθερα. Διαθέτει πολύ μεγάλο εύρος αντικειμένων, ενώ η κατανομή σε 60 κλάσεις δίνει αρκετές κατηγορίες σαφώς διαφοροποιημένες, αν και εννοιολογικά «κοντινές» ώστε να δυσκολεύει τα μοντέλα. Τα αντικείμενα έχουν διαφοροποιήσεις ως προς την κλίμακα τους. Οι σκηνές παρουσιάζουν διαφορετικά κλιματικά φαινόμενα και γεωγραφική διασπορά. Αξίζει να σημειωθεί ότι οι εικόνες έχουν διορθωθεί σύμφωνα με τους κανόνες της επιστήμης της τηλεπισκόπησης ώστε να απαλλαγούν από τυπικά σφάλματα. Αποτελεί ένα αρκετά απαιτητικό σύνολο δεδομένων, καθώς περιλαμβάνει υψηλό ποσοστό επικαλύψεων μεταξύ των αντικειμένων, τα οποία εμφανίζονται σε πολύ πυκνή διάταξη.

VisDrone Dataset: Το VisDrone είναι ένα σύνολο δεδομένων αποκλειστικά με εικόνες και βίντεο που λήφθηκαν από UAV. Κατασκευάστηκε στο πανεπιστήμιο Tianjin της Κίνας και είναι το πληρέστερο και πιο απαιτητικό σύνολο δεδομένων της κατηγορίας του. Περιέχει 288 βίντεο και 10209 εικόνες που ελήφθησαν με διαφορετικές κάμερες, σε διαφορετικές κλιματικές και χρονικές συνθήκες και σε 14 πόλεις της Κίνας. Τα 2.6 εκατομμύρια αντικείμενα που αναφέρονται δίνουν τη δυνατότητα να δοκιμαστούν μοντέλα σε υψηλές απαιτήσεις. Οι προκλήσεις του VisDrone είναι ότι τόσο η κάμερα όσο και τα αντικείμενα βρίσκονται σε κίνηση, ο φωτισμός και το ύψος της λήψης μεταβάλλονται, ενώ αρκετά αντικείμενα φαίνονται εν μέρει εκτός εικόνας και υπάρχει πολύ μεγάλο ποσοστό αλληλοεπικάλυψης μεταξύ τους.

Η επιλογή των παραπάνω συνόλων δεδομένων καλύπτει τα κριτήρια που θέτει η εργασία στην αρχή του κεφαλαίου και αντιπροσωπεύουν παραδείγματα του πραγματικού κόσμου με ανάλογες δυσκολίες και προκλήσεις. Καλύπτουν ένα ευρύ φάσμα εφαρμογών πραγματικού κόσμου με απαιτήσεις σε

ακρίβεια και αξιοπιστία. Από καταγραφές μέσω δορυφόρων έως λήψεις από UAV θεωρείται ότι καλύπτονται οι περισσότερες περιπτώσεις ανίχνευσης μικρών αντικειμένων από εναέριες λήψεις, όπως καλύπτονται επίσης και οι δύο τρόποι σήμανσης των πλαισίων οριοθέτησης. Τα σύνολα δεδομένων που επιλέχθηκαν (Πίνακας 3.3) προσφέρουν αντικειμενικά κριτήρια αξιολόγησης.

Πίνακας 3.3: Τα χαρακτηριστικά των συνόλων δεδομένων.

Σύνολο Δεδομένων	Αριθμός εικόνων	Αριθμός αντικειμένων	Διαστάσεις εικόνας	Είδος Bounding Box (HBB/OBB)	Ανάλυση επί εδάφους
VEDAI	1245	3750	1024x1024 / 512x512	HBB	12.5 - 25 cm/pixel
DOTA	11268	1793658	800x800 / 20000x20000	OBB	10 - 50 cm/pixel
FAIR1M	15000	1000000	1024x1024	HBB	30 - 80 cm/pixel
xView	1881	1000000	1024x1024 / 4096x4096	HBB	30 cm/pixel
VisDrone	10209	100000	720x1280	HBB	10 - 20 cm/pixel

Τέλος, τα σύνολα δεδομένων που επιλέχθηκαν δίνουν ποικιλομορφία ως προς τα ποιοτικά χαρακτηριστικά των εικόνων και τα σενάρια χρήσης, όπως και την αναγνώριση των περιορισμών και των δυνατοτήτων των σύγχρονων μοντέλων ανίχνευσης.

3.1.3 Είδη αντικειμένων

Τα μικρά αντικείμενα ορίζονται συνήθως με δυο τρόπους. Ο ένας είναι με το μέγεθος του αντικειμένου που συνήθως δεν πρέπει να υπερβαίνει τα 32 x 32 pixels ή/και ο δεύτερος να μην καταλαμβάνει περισσότερο από το 1% της συνολικής εικόνας. Βασικές κατηγορίες αντικειμένων με βάση την επιστήμη ή τις τεχνολογίες είναι οι παρακάτω: (1) Εναέριες και δορυφορικές και λήψεις, (2) Αυτόνομη οδήγηση, (3) Ιατρικές απεικονίσεις, (4) Βιομηχανικές εφαρμογές, (5) Αγροτικές εφαρμογές. Αυτοί είναι και οι πέντε πιο συνηθισμένοι τομείς που απασχολούν την ανίχνευση μικρών αντικειμένων.

Η παρούσα εργασία επικεντρώνεται στην πρώτη κατηγορία που αναφέρθηκε, τις εναέριες και δορυφορικές λήψεις. Η επιλογή αυτής της κατηγορίας έγινε με γνώμονα ότι αυτού του είδους οι εικόνες καλύπτουν πολύ μεγάλες γεωγραφικές περιοχές. Ειδικά οι εικόνες προερχόμενες από δορυφόρους σε πολλές περιπτώσεις παρέχεται η δυνατότητα ελεύθερης πρόσβασης σε αυτές. Αντίθετα οι εικόνες από UAV είναι συνήθως δαπανηρές και απαιτούν, στις περισσότερες χώρες, την ύπαρξη αδείας για την πραγματοποίηση πτήσεων, όμως καθώς η τεχνολογία των UAV σταθεροποιείται, γίνονται όλο και πιο προσιτές. Το πλεονέκτημα των UAV είναι ότι, αν και καλύπτουν μικρή περιοχή κάθε φορά, παρέχουν πολύ υψηλή ακρίβεια.

Οι εναέριες εικόνες είναι αρκετά σύνθετες και τα αντικείμενα περιγράφονται με πάρα πολύ λίγα pixel. Οι εφαρμογές τους όμως είναι ποικίλες και πολύ χρήσιμες για την επιστήμη, καθώς υποστηρίζουν πλήθος υπηρεσιών στην καθημερινότητα της κοινωνίας. Τα επιλεγμένα σύνολα δεδομένων καλύπτουν ένα πολύ μεγάλο εύρος αντικειμένων που συναντώνται στην καθημερινότητα και είναι ταυτόχρονα χρήσιμα σε διάφορες εφαρμογές. Παρακάτω αναλύονται συνοπτικά ορισμένα από αυτά.

Περιλαμβάνονται όλων των ειδών τα οχήματα με πολύ λεπτομερή κατηγοριοποίηση. Υπάρχουν κατηγορίες για «ποδήλατα», «τρίκυκλα», «αυτοκίνητα», «είδη φορτηγών οχημάτων» και αντίστοιχα για «πλοία» και «αεροσκάφη».

Καταγραφή ειδών αντικειμένων ανά σύνολο δεδομένων.

Το **VEDAI Dataset** (εναέριες και δορυφορικές λήψεις) προσφέρει τις παρακάτω κατηγορίες και είδη αντικειμένων: *Car, Truck, Camping car, Tractor, Plane, Boat, Other, Pick up, Van*

- Η κατηγορία Car αναφέρεται στα επιβατηγά αυτοκίνητα,
- Η κατηγορία Truck αναφέρεται σε φορτηγά οχήματα, συνήθως φορτηγά οχήματα έργου με ανοιχτή καρότσα,
- Η κατηγορία Camping car αναφέρεται σε αυτοκινούμενα οχήματα camping,
- Η κατηγορία Tractor αναφέρεται σε γεωργικούς ελκυστήρες,
- Η κατηγορία Plane αναφέρεται σε αεροπλάνα, συνήθως μικρά ελικοφόρα,
- Η κατηγορία Boat αναφέρεται σε σκάφη, συνήθως ταχύπλοα,
- Η κατηγορία Other αναφέρεται σε οχήματα έργου, συνήθως ισοπεδωτές γαιών,
- Η κατηγορία Pickup αναφέρεται σε ανοιχτά ημιφορτηγά οχήματα,
- Η κατηγορία Van αναφέρεται σε μικρά κλειστά φορτηγάκια.

Υπάρχουν επίσης δύο Meta-classes, η πρώτη που ονομάζεται Small land vehicles (μικρά οχήματα εδάφους) και περιέχει τα αντικείμενα από τις κλάσεις “Car”, “Pickup”, “Tractor”, “Van”, και η δεύτερη κατηγορία που ονομάζεται Large land vehicles και περιέχει τα αντικείμενα από τις κλάσεις “Truck” και “Camping car”. Κατά μέσο όρο υπάρχουν 5.5 αντικείμενα σε κάθε εικόνα και η έκταση που καταλαμβάνουν είναι περίπου το 0.7% της συνολικής επιφάνειας.

Για το σύνολο δεδομένων **DOTA**, που αποτελείται από δορυφορικές εικόνες υψηλής ανάλυσης, υπάρχουν τρεις εκδόσεις, όπου η κάθε μια μετά τη πρώτη εμπλουτίζει το σύνολο δεδομένων μόνο με την προσθήκη επιπλέον κατηγοριών.

Η έκδοση v1.0 περιλαμβάνει τις παρακάτω 15 κατηγορίες:

- Plane (αεροπλάνα)
- Ship (πλοία)
- Storage tank (δεξαμενή αποθήκευσης)
- Baseball diamond (γήπεδο Baseball)
- Tennis court (γήπεδο τένις)
- Basketball court (γήπεδο μπάσκετ)
- Ground track field (αθλητικός στίβος)
- Harbor (λιμάνι)
- Bridge (γέφυρα)
- Large vehicle (μεγάλο όχημα)
- Small vehicle (μικρό όχημα)
- Helicopter (ελικόπτερο)
- Roundabout (κυκλικός κόμβος)
- Soccer ball field (γήπεδο ποδοσφαίρου)
- Swimming pool (πισίνα)

Στην έκδοση v1.5 προστέθηκε η κατηγορία, Container crane (γερανός εμπορευματοκιβωτίων)

Στην έκδοση v2.0 προστέθηκαν οι επόμενες δύο κατηγορίες:

- Airport (αεροδρόμιο)
- Helipad (ελικοδρόμιο)

Η έκδοση v1.0 περιελάμβανε 2806 εικόνες με 188282 αντικείμενα σημασμένα, ενώ η έκδοση v1.5 403318 σημασμένα αντικείμενα στα οποία συμπεριλήφθηκαν και αντικείμενα με μέγεθος κάτω από 10 pixel. Η έκδοση v2.0 περιέχει 11268 εικόνες και 1793658 σημασμένα αντικείμενα.

Το **FAIR1M Dataset** είναι ένα Fine - grained σύνολο δεδομένων. Η κατηγοριοποίηση του είναι αρκετά λεπτομερής και με υποκατηγορίες. Αποτελείται από 15000 δορυφορικές εικόνες υψηλής ανάλυσης και περίπου ένα εκατομμύριο σημασμένα αντικείμενα.

Έχει πέντε βασικές κατηγορίες:

- Airplanes (αεροπλάνα)
- Ships (πλοία)
- Vehicles (οχήματα)
- Courts (γήπεδα)
- Roads (δρόμοι)

Οι πέντε παραπάνω αναλύονται σε 37 υποκατηγορίες, και αποτελούν το σημείο διαφοροποίησης από άλλα σύνολα δεδομένων καθώς εξιδανικεύει με μεγάλη λεπτομέρεια τα αντικείμενα που αναγνωρίζει.

Το **xView Dataset** είναι από τα μεγαλύτερα σύνολα δεδομένων που διατίθενται ελεύθερα και αποτελείται από δορυφορικές εικόνες υψηλής ανάλυσης. Είναι δομημένο με σκοπό να αποτελεί ένα Fine - grained σύνολο δεδομένων. Περιέχει 847 εικόνες με πάνω από ένα εκατομμύριο σημασμένα αντικείμενα. Τα αντικείμενα κατηγοριοποιούνται σε 60 κλάσεις συνολικά και τα μικρότερα από αυτά φτάνουν σε μέγεθος τα τρία μέτρα ή περίπου τα δέκα pixel.

Τέλος, το **VisDrone Dataset** είναι ένα σύνολο δεδομένων αποκλειστικά με εικόνες που έχουν ληφθεί από UAV. Αποτελείται από 8599 εικόνες με πάνω από 540000 σημασμένα αντικείμενα.

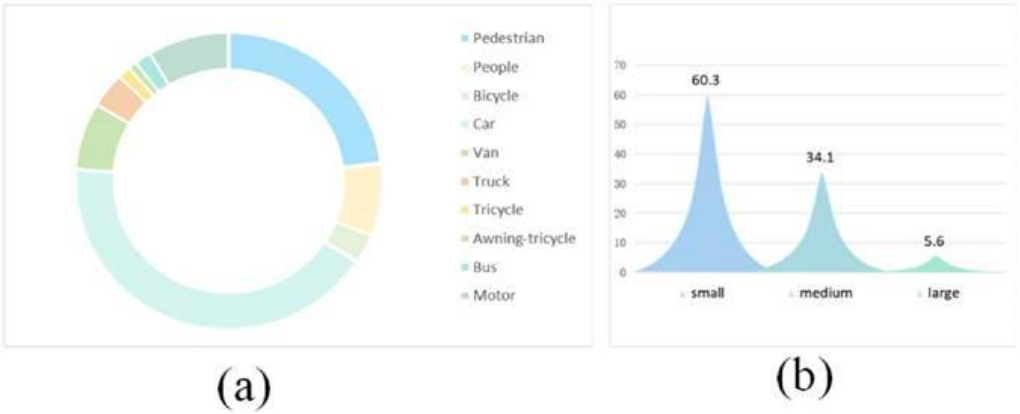
Τα αντικείμενα χωρίζονται σε δέκα κατηγορίες:

- Pedestrian (πεζός)
- Person (άνθρωπός)
- Bicycle (ποδήλατο)
- Car (αυτοκίνητο)
- Van (κλειστό φορτηγάκι)
- Truck (φορτηγό)
- Tricycle (τρίκυκλο)
- Awning tricycle (σκεπασμένο τρίκυκλο)
- Bus (λεωφορείο)
- Motor (μοτοσυκλέτα)

Ένας άνθρωπος αν έχει όρθια στάση ή φαίνεται ότι είναι σε κίνηση ή βηματισμό τότε καταχωρείται ως Pedestrian, αλλιώς καταχωρείται ως Person.

Η ανίχνευση μικρών αντικειμένων σε εναέριες λήψεις έχει πολλές και ποικίλες προκλήσεις εξαιτίας της αναπαράστασης των αντικειμένων από πολύ λίγα pixels, της μεγάλης συνήθως πυκνότητας τους και της μεταξύ τους αλληλοκάλυψης. Το φόντο είναι, επίσης, ένας άλλος παράγοντας που κάνει δύσκολη την ανίχνευση τους, όπως και η μερική απόκρυψη τους ή αποκοπή τους στα όρια της εικόνας. Η αναλυτική κατηγοριοποίηση τους όμως βοηθάει στον ορθότερο εντοπισμό και κατηγοριοποίηση

τους, καθώς κάθε αντικείμενο έχει τα δικά του λεπτά χαρακτηριστικά και πολλές φορές οι διαφορές μεταξύ κατηγοριών είναι δύσκολο να διακριθούν.



Εικόνα 3.12: Στατιστικά του συνόλου δεδομένων VisDrone. (a) Το ποσοστιαίο μέγεθος της κάθε κατηγορίας αντικειμένων, (b) Η κατανομή των αντικειμένων στις κλίμακες μικρό < 20x20 pixels (small), μεσαίο 20 – 32 pixels (medium), μεγάλο > 32x32 pixels (big) [38].

3.2 Προεπεξεργασία δεδομένων

Η προεπεξεργασία των δεδομένων είναι ένα βήμα που έχει πολύ μεγάλη σημασία στην ανάπτυξη των μοντέλων μηχανικής μάθησης, καθώς έτσι εξασφαλίζεται ότι τα δεδομένα είναι σε μορφή που μπορούν να χρησιμοποιηθούν στην εκπαίδευση. Στην ανίχνευση μικρών αντικειμένων η διαδικασία της προεπεξεργασίας περιλαμβάνει αρκετές τεχνικές όπως η κανονικοποίηση, η επαύξηση των δεδομένων, ο περιορισμός την ανισοκατανομής των δεδομένων στις κλάσεις, η αφαίρεση του θορύβου, ο μετασχηματισμός του μεγέθους των εικόνων και ο ποιοτικός έλεγχος των εικόνων ως προς την επάρκεια των χαρακτηριστικών τους (π.χ. μπορεί κάποιες εικόνες να έχουν πολύ μεγάλη κάλυψη από σύννεφα ή χιόνι). Στο τέλος της διαδικασίας τα δεδομένα είναι σε καλύτερη κατάσταση από ότι ήταν στην αρχική τους μορφή και θα προσφέρουν καλύτερα και πιο αξιόπιστα αποτελέσματα στην εκπαίδευση του μοντέλου.

3.2.1 Καθαρισμός και Κανονικοποίηση δεδομένων

Η κανονικοποίηση ανήκει στις βασικές μεθόδους της προεπεξεργασίας των δεδομένων. Είναι μια μέθοδος που αλλάζει την κατανομή ή το εύρος των αριθμητικών τιμών σε ένα συγκεκριμένο διάστημα, το οποίο κυμαίνεται συνήθως μεταξύ του [0,1] ή του [-1,1]. Αυτή η μεταβολή γίνεται χωρίς να αλλάζει η διαφορά των τιμών μεταξύ τους, κρατώντας δηλαδή σταθερή την αναλογικότητα. Το αποτέλεσμα είναι μια ίση συνεισφορά όλων των δεδομένων άσχετα αν κάποια από τα δεδομένα έχουν μεγαλύτερες τιμές. Η διαδικασία είναι εξαιρετικά χρήσιμη σε αλγορίθμους μηχανικής μάθησης ειδικά όταν υπάρχουν ζητήματα κλίμακας και εφαρμογή μεθόδων SGD [39].

Παρακάτω αναλύονται τρεις βασικές μεθόδους κανονικοποίησης.

Η πρώτη μέθοδος είναι η Batch Normalization (BN). Προτάθηκε από τους Ioffe et al. [40] και δίνει τη δυνατότητα της χρήσης μεγαλύτερου learning rate στην εκπαίδευση, παρέχοντας μεγαλύτερη ευελιξία στον ορισμό του learning rate κατά την εκκίνηση. Όπως αναφέρουν στην μελέτη τους σε ορισμένες περιπτώσεις καθίσταται περιττή η χρήση της τεχνικής του Dropout. Εφαρμόζεται πριν το μη γραμμικό στοιχείο, όπως για παράδειγμα η RELU, και αυτό διασφαλίζει μια σταθερή μέση τιμή και διασπορά περίπου στο 0 και στο 1 αντίστοιχα. Επιπροσθέτως, η εξάλειψη αρνητικών τιμών βελτιώνει τη

πιθανότητα τα inputs να είναι στο σωστό (μαθηματικά) χώρο, για παράδειγμα στην περίπτωση εισαγωγής αρνητικών τιμών στη RELU να μας δίνει στην εξαγωγή, τιμή μηδέν [40], [41].

Η μέθοδος Batch Normalization κάνει πιο ομαλές τις ενεργοποιήσεις με σταθερή μέση τιμή και διασπορά. Η έξοδος του επιπέδου είναι αμετάβλητη σε σχέση με το μέγεθος των βαρών. Επιτρέπει τη χρήση μεγαλύτερων learning rates χωρίς να επηρεάζεται σημαντικά η εκπαίδευση, προσφέροντας μεγαλύτερη σταθερότητα, ιδιαίτερα στα αρχικά στάδια της. Παράλληλα, συμβάλλει σε αποτελεσματικότερη εκπαίδευση βαθιών μοντέλων, καθώς παρέχει καλύτερο έλεγχο τους προβλήματος των "exploding gradients" μέσω της σταθεροποίησης τους [40].

Επιπλέον, οι Ioffe et al. [40], αναφέρουν ότι για να εκμεταλλευτούν στον μέγιστο βαθμό τη μέθοδο του Batch Normalization, έκριναν αναγκαία την τροποποίηση της αρχιτεκτονικής του δικτύου, η οποία στην συγκεκριμένη περίπτωση αφορούσε το μοντέλο Inception. Αυτό περιλαμβάνει: (1) Reduce L2 weight regularization, (2) Accelerate the learning rate decay, (3) Remove Local Response Normalization, (4) Shuffle training examples more thoroughly, (5) Reduce the photometric distortions [40]. Η παράμετρος L2 στη συνάρτηση απωλειών ελέγχει το overfit του μοντέλου [40]. Η ταχύτερη μείωση του Learning Rate, οφείλεται στο γεγονός ότι το μοντέλο μαθαίνει γρηγορότερα με τη χρήση του Batch Normalization. Η κατάργηση της εφαρμογής της (παλαιότερης) τεχνικής Local Response Normalization (LRN) γίνεται γιατί πλέον η νεότερη τεχνική του Batch Normalization καθώς έχει καλύτερα αποτελέσματα, και η χρήση της Local Response Normalization θεωρείται περιττή. Η «τυχαιοποίηση» των δεδομένων εκπαίδευσης βοηθάει το μοντέλο, καθώς σε κάθε mini-batch εμφανίζονται μοναδικά παραδείγματα, γεγονός που, μέσω της αποφυγής επαναλήψεων, οδηγεί σε βελτίωση της απόδοσης του μοντέλου [40]. Η μείωση των φωτομετρικών αλλοιώσεων οφείλεται στο γεγονός ότι το μοντέλο εκπαιδεύεται ταχύτερα, με αποτέλεσμα κάθε εικόνα να προσπελάσεται λιγότερες φορές. Συνεπώς, προτιμότερο είναι το μοντέλο να τροφοδοτείται με όσο το δυνατόν πιο ρεαλιστικά παραδείγματα [40].

Η χρήση της μεθόδου προσθέτει μόνο δυο παραμέτρους και έτσι δεν αλλοιώνει την αρχιτεκτονική του μοντέλου. Επίσης, η χρήση του Batch Normalization βελτιώνει την απόδοση και σύγκλιση του μοντέλου, επιταχύνοντας ελαφρώς την διαδικασία της εκπαίδευσης με μεγαλύτερα Learning Rates, και αφαίρεση του Dropout [40].

Τέλος, η τεχνική Layer Normalization (LN), που προτάθηκε από τους Lei Ba et al. [42], είναι μια εξέλιξη ή παραλλαγή της Batch Normalization όπου υπολογίζεται η μέση τιμή και η διακύμανση στα χαρακτηριστικά αλλά όχι στο batch (παρτίδα). Επίσης, οι δυο αυτές στατιστικές τιμές υπολογίζονται σε όλα τα νευρωνικά σήματα, σε όλες δηλαδή τις τιμές εξόδου των νευρώνων (ενεργοποιήσεις) για κάθε επίπεδο (layer) του νευρωνικού δικτύου. Ομοίως με την τεχνική Batch Normalization, η Layer Normalization προσφέρει βελτίωση σε ότι αφορά την ταχύτητα εκπαίδευσης σε σχέση με το βασικό μοντέλο αλλά και το μοντέλο που χρησιμοποιεί Batch Normalization. Επίσης, είναι μια τεχνική ιδιαίτερα χρήσιμη για αρχιτεκτονικές όπως οι Transformers και τα Recursive Neural Networks (RNN), καθώς η τεχνική Batch Normalization δε μπορεί να λειτουργήσει αξιόπιστα για περιπτώσεις όπου το batch size είναι μικρό [42].

Οι συγγραφείς παρατήρησαν πολύ καλά αποτελέσματα από την εφαρμογή της Layer Normalization σε δίκτυα RNN. Τα αποτελέσματα των δοκιμών έδειξαν ότι μπορεί να διαχειριστεί πολύ καλά τις κρυφές καταστάσεις (Hidden States), οι οποίες διατηρούν πληροφορία από προηγούμενα βήματα (χρόνους). Αν και τα επίπεδα είναι ασταθή κατά την εκπαίδευση, η Layer Normalization συμβάλλει στη σταθεροποίηση τους εκτελώντας κάθε βήμα ξεχωριστά, περιορίζοντας τις τιμές σε λογικά όρια και επιταχύνοντας τη σύγκλιση [42].

Παρατηρήθηκε επίσης ότι στα CNN δίκτυα, ενώ βελτιώθηκε η ταχύτητα της εκπαίδευσης, η γενική απόδοση δεν ήταν καλύτερη σε σχέση με αυτήν της μεθόδου Batch Normalization. Στα συνελικτικά νευρωνικά δίκτυα οι νευρώνες των κρυφών επιπέδων δεν έχουν όλοι την ίδια συνεισφορά σε αντίθεση με τα πλήρως συνδεδεμένα δίκτυα. Ιδιαίτερα στα όρια των εικόνων παρατηρούνται σπάνιες ενεργοποιήσεις, με αποτέλεσμα τα στατιστικά των νευρώνων να διαφέρουν από αυτά στο ίδιο επίπεδο. Αυτό οφείλεται στο ότι διαθέτουν πολύ λιγότερη πληροφορία από το πεδίο λήψης (reception field) σε σχέση με τους νευρώνες που βρίσκονται κοντά στο κέντρο [42].

Συνοψίζοντας, τα κύρια οφέλη της χρήσης τεχνικών κανονικοποίησης (Normalization) μπορούν να συνοψιστούν ως εξής: Αρχικά, δίνεται η δυνατότητα χρήσης μεγαλύτερων Learning Rates, επιταχύνοντας την εκπαίδευση, ενώ παράλληλα το μοντέλο γίνεται πιο ανθεκτικό στην αρχή της εκπαίδευσης ως προς την αρχικοποίηση των βαρών. Οι ενεργοποιήσεις γίνονται πιο σταθερές, με αποτέλεσμα να αποφεύγεται ως ένα βαθμό το φαινόμενο του “exploding gradients”. Παράλληλα, καθίσταται δυνατή η αποφυγή του Dropout, οδηγώντας σε αποτελεσματικότερη εκπαίδευση βαθύτερων δικτύων.

3.2.2 Επαύξηση δεδομένων

Η επαύξηση δεδομένων (data augmentation) είναι μια μεθοδολογία που στόχο έχει να δημιουργήσει πλασματικά δεδομένα από τα συνήθη διαθέσιμα πρωτογενή. Ωστόσο, με τις τελευταίες εξελίξεις, υπάρχουν προσπάθειες που χρησιμοποιούν δεδομένα επαύξησης παραγόμενα από μοντέλα τεχνητής νοημοσύνης. Σκοπός της διαδικασίας είναι η αύξηση του μεγέθους του dataset και της απόδοσης του μοντέλου.

Η επαύξηση δεδομένων αναλύεται σε δύο βασικές κατηγορίες, η πρώτη είναι η παραδοσιακή μέθοδος (traditional) ή βασική προσέγγιση (basic approach), και η δεύτερη είναι η ενισχυμένη (advanced) ή η προσέγγιση βαθιάς μάθησης (deep learning approach) [43]. Στην παρούσα εργασία εφαρμόζονται οι παραδοσιακές τεχνικές, οι οποίες χωρίζονται στις επιμέρους κατηγορίες της γεωμετρικής παραλλαγής, της φωτομετρικής παραλλαγής (photometric) και των τεχνικών αλλοίωσης, όπως η random erasing [43].

Όσον αφορά την επαύξηση δεδομένων εικόνων αυτό γίνεται με ποικίλους τρόπους, κάποιοι από τους οποίους αφορούν στη παραμετροποίηση των χρωμάτων με την διαφοροποίηση, για παράδειγμα, είτε του χρώματος ή της φωτεινότητας είτε με τον συνδυασμό πολλών διαφορετικών παραμέτρων [43], [44], [45]. Ένας άλλος τρόπος αφορά στη γεωμετρική μεταβολή της εικόνας αλλάζοντας τον προσανατολισμό της με περιστροφή (rotate). Αυτές οι δύο τεχνικές είναι οι πιο βασικές και χρησιμοποιούνται ευρέως σε μοντέλα που απευθύνονται σε διαφορετικούς τομείς επιστημών. Όταν όμως πρόκειται για πλαίσια οριοθέτησης (bounding boxes), αυτές οι τεχνικές μπορούν να εφαρμοστούν μόνο εντός αυτών των πλαισίων.

Οι γεωμετρικές τεχνικές είναι συνήθως οι πιο εύκολες να υλοποιηθούν και να εφαρμοστούν. Σε σύνολα δεδομένων όπως αυτά του εντοπισμού αντικειμένων είναι ασφαλείς να χρησιμοποιηθούν καθώς οι ετικέτες δεν αλλοιώνονται. Θέλει όμως προσοχή ποιες τεχνικές εφαρμόζονται, οι οποίες πρέπει να είναι ανάλογες με τη φύση των δεδομένων. Για παράδειγμα, αν εφαρμοστεί το flip ή το rotate, υπό προϋποθέσεις, σε μια εικόνα ενδιαφέροντος, οι χαρακτήρες εξάγουν εσφαλμένο αποτέλεσμα καθώς, για παράδειγμα, ο αριθμός “9” μετατρέπεται στον αριθμό “6” ή το λατινικό γράμμα “p” μετατρέπεται στο “q” [43]. Όλες οι τεχνικές αλλοιώνουν την αρχική εικόνα σε κάποιο βαθμό και δίνεται προσοχή στην επίδραση αυτών πάνω στις ετικέτες, καθώς η ανακατασκευή τους είναι μια κοστοβόρα διαδικασία και πρέπει πάντα να επιλέγεται ο τρόπος επαύξησης βάση του σύνολο δεδομένων και το είδος της

εργασίας. Επίσης, αν και οι τεχνικές είναι σχετικά απλές όταν συνδυάζονται αυξάνουν το υπολογιστικό κόστος και τον χρόνο εκτέλεσης της εκπαίδευσης του μοντέλου.

Η τεχνική Flipping μπορεί να εφαρμοστεί είτε στον οριζόντιο είτε στον κάθετο άξονα [43]. Το συνηθέστερο είναι να εφαρμόζεται στον οριζόντιο άξονα. Στην πράξη η τεχνική αυτή παράγει μια εικόνα που είναι ο αντικατοπτρισμός της αρχικής ως προς το επίπεδο που ορίστηκε.

Η τεχνική της περικοπής (Cropping) παράγει μια νέα εικόνα που είναι τμήμα της αρχικής. Η παραγόμενη εικόνα διαφέρει ως προς το μέγεθος και το τμήμα της εικόνας που επιλέγεται κάθε φορά. Αυτό σημαίνει ότι δημιουργούνται «νέες» εικόνες. Πρέπει να τονιστεί ότι η εικόνα που παράγεται έχει μικρότερες διαστάσεις από την αρχική και χρειάζεται προσοχή όταν χρησιμοποιείται σε σύνολα δεδομένων που χρησιμοποιούν ετικέτες [43].

Η τεχνική του Translation [43], [46] μοιάζει με την προηγούμενη τεχνική του Cropping αλλά με την διαφορά ότι κρατάει ίδιες τις διαστάσεις της εικόνας με αυτές της αρχικής. Αυτό βοηθάει το μοντέλο να αντιλαμβάνεται καλύτερα τα χαρακτηριστικά της εικόνας, επειδή διαφοροποιείται από τη συνηθισμένη πρακτική όπου το βασικό θέμα βρίσκεται στο κέντρο της, και το οποίο μπορεί να μην ισχύει στην πραγματική εφαρμογή του μοντέλου στον ανοιχτό κόσμο. Ο τρόπος με τον οποίο λειτουργεί η τεχνική είναι ότι διαλέγει ένα χώρο από την αρχική εικόνα, και το υπόλοιπο τμήμα «γεμίζει» με μια σταθερή τιμή στο διάστημα από 0 έως και 255 ή με τυχαίο θόρυβο [43].

Με την περιστροφή (Rotation) [43], [46] οι εικόνες γυρνάνε είτε προς τα δεξιά είτε προς αριστερά από 1 μοίρα έως 359 μοίρες σε σχέση με τον κάθετο άξονα. Προσοχή δίνεται σε σύνολα δεδομένων που περιέχουν αλφαριθμητικούς χαρακτήρες, επειδή όσο αυξάνεται η περιστροφή τόσο οι ετικέτες θα είναι λανθασμένες. Οι γεωμετρικοί μετασχηματισμοί επαύξησης δεδομένων αν και είναι πολύ απλοί έχει αποδειχθεί ότι είναι πολύ αποτελεσματικοί.

Οι μη αφηνικοί, και σε κάποιες περιπτώσεις μη γεωμετρικοί μετασχηματισμοί, τροποποιούν την εικόνα γεωμετρικά, περιστρέφοντας την, αλλάζοντας το μέγεθος της, αντικατοπτρίζοντας την (mirroring, reflection, flipping), ή μετατοπίζοντας την ολόκληρη (translation) [44].

Στους φωτομετρικούς μετασχηματισμούς επαύξησης δεδομένων υπάρχουν η εισαγωγή θορύβου (Noise injections), οι μετασχηματισμούς χρώματος (Color space) και τα φίλτρα εικόνας (image filters) [43], [44].

Με την εισαγωγή θορύβου (Noise injection) [43], [44], ενσωματώνεται τυχαίος θορύβος με την μορφή ενός πίνακα τυχαίων αριθμών (συνήθως της κανονικής κατανομής). Με αυτό τον τρόπο το μοντέλο ωθείται να μάθει νέα χαρακτηριστικά που αυξάνουν την απόδοση του.

Οι μετασχηματισμοί χρώματος (Color space) [43], [44] υλοποιούνται πρακτικά και εύκολα. Μια απλή υλοποίηση είναι να διατηρηθεί μόνο ένα χρώμα (από τα βασικά Red, Green, Blue - RGB) και να πάρουν τιμή μηδέν οι θέσεις των υπολοίπων δύο χρωμάτων (κάθε pixel αποτελείται από τον συνδυασμό διαφορετικών ποσοτήτων από τα βασικά χρώματα) [43]. Μπορεί να αλλάξει ακόμα και η ένταση της φωτεινότητας ή η αντίθεση της εικόνας. Άλλα στοιχεία που μπορούν να αλλάξουν είναι ο κορεσμός και ο χρωματικός τόνος [47]. Η τεχνική αυτή συμβάλει στην αποφυγή του lighting bias, καθώς συχνά επιδιώκεται η διατήρηση καλών φωτιστικών συνθηκών, με αποτέλεσμα οι περισσότερες εικόνες να έχουν όμοια χαρακτηριστικά φωτισμού [43]. Η χρησιμοποίηση της συγκεκριμένης τεχνικής πρέπει να γίνεται με προσοχή. Σε περιπτώσεις εφαρμογών που η ιδιότητα του χρώματος είναι αναγκαία, η αλλοίωση των εικόνων κατά αυτό τον τρόπο αποφεύγεται, όπως για παράδειγμα σε εφαρμογές εξαγωγής συναισθήματος εικόνας (Image Sentiment Analysis) ή σε περιπτώσεις που εντοπισμού αντικειμένων συγκεκριμένου χρώματος.

Τα **kernel filters** ανήκουν στα φίλτρα εικόνας [43], [46] και χρησιμοποιούνται είτε για την αύξηση της αντίθεσης (sharpen) είτε για την απόδοση «θολότητας» (blur). Το φίλτρο δουλεύει ως ένα κυλιόμενο παράθυρο (sliding window) με ένα πίνακα μεγέθους (n x n) που τρέχει πάνω στην εικόνα. Αυτό το «τρέξιμο, μπορεί να εφαρμόσει είτε με ένα Gaussian blur φίλτρο είτε με ένα φίλτρο υψηλής ανάλυσης. Αν η εικόνα υποστεί θόλωση, το μοντέλο δυσκολεύεται να διακρίνει χαρακτηριστικά στην εικόνα, ενώ αν ευκρίνεια αυξηθεί το μοντέλο διευκολύνεται να διακρίνει τα στοιχεία των αντικειμένων στην εικόνα.

Στη μελέτη τους οι Kumar et al. [46] προτείνουν τρεις τρόπους επαύξησης των δεδομένων εικόνας. Ο πρώτος είναι ο Pairwise channel transfer, ο οποίος αποτελεί μεθοδολογία μεταφοράς πληροφορίας από κάθε κανάλι (R,G,B) σε κάποιο άλλο με αντιμετάθεση. Για παράδειγμα, δίνει την ικανότητα μεταφοράς της πληροφορίας από το κόκκινο στο πράσινο κανάλι, μεταβάλλοντας τον χρωματισμό της εικόνας και το ίδιο μπορεί να εφαρμοστεί και στα υπόλοιπα κανάλια. Η αλλαγή χρώματος δίνει μια καλύτερη προσομοίωση της πραγματικότητας, καθώς το ίδιο αντικείμενο μπορεί να εμφανίζεται με διαφορετικά χρώματα ή σε φόντο διαφορετικού χρώματος [46].

Η δεύτερη μέθοδος που προτείνουν είναι η Random Object Occlusion, όπου πρώτα επιλέγεται μια τυχαία εικόνα από το σύνολο δεδομένων, η οποία παίρνει το ρόλο ενός «τυχαίου» αντικειμένου που εισάγεται στις εικόνες. Εν συνεχεία, αυτή η εικόνα επιλέγεται να μετατραπεί σε ανάλυση 100 x 100, και επανεισάγεται σε τυχαίες θέσεις των εικόνων του συνόλου δεδομένων [46].

Η τρίτη μέθοδος είναι η Novel masking augmentation. Η μέθοδος εφαρμόζεται είτε σε μορφή πλέγματος είτε σε κυκλική διάταξη. Αλλάζει το μέγεθος της εικόνας σε ανάλυση 300 x 300 και εφαρμόζεται Translation το οποίο μετακινεί τα περιεχόμενα της τόσο στο οριζόντιο όσο και στο κάθετο άξονα, κεντράροντας την διαφορετικά από την αρχική. Έπειτα, εφαρμόζεται ένα Gaussian blur φίλτρο και στην συνέχεια γίνεται ένας μετασχηματισμός χρώματος (Color jitter), ο οποίος θέτει στα τρία κανάλια χρώματος (R,G,B) τυχαίες τιμές, παίρνοντας διαφορετικό χρωματικό αποτέλεσμα. Τέλος, εφαρμόζεται διαγραφή τυχαίων τμημάτων της εικόνας (Random Erasing) [46].

Στη μελέτη τους οι E.D. Cubuk [48] παρουσιάζουν τον αλγόριθμο AutoAugment για την αυτόματη εύρεση και βελτίωση των στρατηγικών επαύξησης δεδομένων. Υλοποίησαν μια μέθοδο που βρίσκει τις κατάλληλες και πιο αποδοτικές πολιτικές -όπως ονομάζουν, τις τεχνικές επαύξησης- με μετρήσιμα κριτήρια. Η βασική δομή του αποτελείται από δύο τμήματα, το ένα είναι ο αλγόριθμος αναζήτησης (search algorithm) και το δεύτερο είναι ο χώρος αναζήτησης (search space) [48]. Χρησιμοποιώντας τον αλγόριθμο AutoAugment και τα σύνολα δεδομένων CIFAR-10, Reduced CIFAR-10, CIFAR-100, SVHN και το ImageNet εκτελούνται πειράματα για να βρεθεί ο καλύτερος συνδυασμός παραμέτρων. Για τα πειράματα χρησιμοποιήθηκαν ως backbone, δίκτυα της οικογένειας ResNet και πιο συγκεκριμένα τα Wide-ResNet και ResNet-50/200. Χρησιμοποιήθηκαν πάνω από 200 «πολιτικές» επαύξησης για να βρεθεί ο καλύτερος συνδυασμός. Από τις δοκιμές διαπιστώθηκε ότι μπόρεσαν να αυξήσουν την ακρίβεια των μοντέλων στα προαναφερθέντα σύνολα δεδομένων, ξεπερνώντας τα μέχρι σήμερα καλύτερα αποτελέσματα (State of the Art) [48].

Οι συγγραφείς [48] είχαν ως στόχο επίσης να υλοποιήσουν κανόνες που να μπορούν να μεταφερθούν σε διαφορετικά σύνολα δεδομένων και μοντέλα ανίχνευσης αντικειμένων (transferability). Διαπίστωσαν ότι οι κανόνες που εκπαιδεύτηκαν σε ένα συγκεκριμένο σύνολο δεδομένων με ένα συγκεκριμένο μοντέλο, επιτυγχάνουν καλύτερη απόδοση όταν μεταφέρονται σε ένα άλλο μοντέλο με το ίδιο σύνολο δεδομένων, σε σχέση με τη μεταφορά τους σε ένα μοντέλο με διαφορετικό σύνολο δεδομένων [48].

Οι DeVries et al. [49] αναλύουν την τεχνική **Cutout**, στην οποία μια εικόνα εξάγεται από το αρχικό σύνολο δεδομένων, και σε τυχαίες θέσεις της αποκόπτουμε τετράγωνες περιοχές. Με την αφαίρεση πληροφορίας, στην ίδια φιλοσοφία με το dropout, εξαναγκάζεται ο αλγόριθμος να λάβει υπόψη του και την γύρω πληροφορία πριν δώσει αποτέλεσμα [49]. Η αξιοποίηση της χωρικής πληροφορίας βοηθάει το δίκτυο να μην εκπαιδευτεί σε ένα μόνο μικρό τμήμα [49]. Η αφαίρεση τμήματος της πληροφορίας έχει παρόμοια χαρακτηριστικά με αντικείμενα που είναι εμφανή εν μέρη στα όρια της εικόνας ή επικαλύπτονται εν μέρη από άλλα αντικείμενα.

Στην έρευνα τους οι DeVries et al. [49] δοκιμάζουν δυο σενάρια εφαρμογής που δίνουν παρόμοιες αποδόσεις. Στο πρώτο εφαρμόζεται Cutout μόνο στα χαρακτηριστικά εκείνα που έχουν υψηλή συνεισφορά ενισχύοντας την απόδοση του αλγορίθμου στα αντικείμενα που έχουν μικρότερη επίδραση. Στο δεύτερο σενάριο πραγματοποιείται τυχαία εφαρμογή στις εικόνες.

Τα τετράγωνα που εφαρμόζονται στις εικόνες είναι μια μηδενική μάσκα σταθερού μεγέθους (fixed-size zero mask). Στην υλοποίηση του το Cutout, από άποψη υπερπαραμέτρων, είναι σημαντικότερο το μέγεθος της καλυπτόμενης περιοχής από την μάσκα, παρά το σχήμα της. Για το λόγο αυτό, το σχήμα της μάσκας κρατείται στο τετράγωνο. Σημαντικό επίσης είναι, η εφαρμογή του Cutout να γίνει σε ποσοστό 50% ώστε το μοντέλο να έχει αρκετές ανεπεξέργαστες εικόνες στην αρχική τους μορφή κατά την εκπαίδευση [49].

Η μέθοδος μπορεί να τρέξει παράλληλα με άλλες διεργασίες επαύξησης δεδομένων στην CPU κατά τη φόρτωση των δεδομένων, ενώ στη GPU να εκτελέσει την εκπαίδευση του μοντέλου. Με την εφαρμογή του Cutout παρατηρείται μεγαλύτερη ενεργοποίηση των πρώτων ρηχών επιπέδων (shallow layers activation), ενώ στα βαθύτερα επίπεδα (deeper layers) οι ενεργοποιήσεις συγκεντρώνονται περσότερο στην ουρά. Αυτό υποδηλώνει ότι το Cutout βοηθάει το μοντέλο να λάβει υπόψη του περισσότερα και γενικότερα χαρακτηριστικά της εικόνας όταν κάνει μια πρόβλεψη, μειώνοντας τον κίνδυνο του overfitting. Στα πειράματα που έγιναν, στο σύνολο δεδομένων CIFAR-10 και CIFAR-100, η εφαρμογή του Cutout έδωσε μια βελτίωση σε εύρος από 0.3% έως 0.6%.

Το **CutMix** αποτελεί μια τεχνική επαύξησης των Yun et al. [50], η οποία δεν αφαιρεί πληροφορία σε σύγκριση με την τεχνική CutOut. Αναφέρουν ότι οι τεχνικές αφαίρεσης πληροφορίας είναι εσφαλμένες καθώς τα συνελικτικά νευρωνικά δίκτυα θέλουν όσο το δυνατόν περισσότερη πληροφορία.

Η παρούσα τεχνική, σε αντίθεση με την αφαίρεση πληροφορίας από την εικόνα, αντικαθιστά τμήματα της με τμήματα άλλης εικόνας. Μαζί με το τμήμα της εικόνας αντικαθιστούνται και οι ετικέτες (labels) του αντίστοιχου τμήματος. Αυτό οδηγεί το μοντέλο να αναγνωρίζει αντικείμενα που εμφανίζονται μερικώς. Οι συγγραφείς [50] για να αυξήσουν την αποδοτικότητα της εκπαίδευσης, έχουν ορίσει το CutMix να λαμβάνει τα πλήρη όρια των αντικειμένων ως ένδειξη για ταξινόμηση, ένα χαρακτηριστικό κοινό με το Cutout, και παράλληλα να αναγνωρίζει δύο αντικείμενα από εν μέρη εμφανίσεις μέσα σε μια μόνο εικόνα. Η έκθεση του μοντέλου σε συνθετικές εικόνες, που αποτελούνται από τμήματα ή αποσπάσματα από δύο διαφορετικά αντικείμενα του, επιτρέπει να αναγνωρίζει αντικείμενα από μερική πληροφορία.

Στις δοκιμές τους, η τεχνική CutMix, με χρήση του συνόλου δεδομένων του ImageNet, έδωσε μια αύξηση 2.28% για το μοντέλο ResNet-50 και 1.70% για το μοντέλο ResNet-101. Αυτό υποδεικνύει ότι με τη χρήση του CutMix αξιοποιούνται αποδοτικότερα τα δεδομένα, ενώ απαιτείται ελάχιστο υπολογιστικό κόστος.

Στον ίδιο τρόπο σκέψης με τις δύο προηγούμενες τεχνικές είναι και η μέθοδος **Mixup** που προτείνουν οι Zhang et al. [51]. Η τεχνική πετυχαίνει την επαύξηση των δεδομένων συνδυάζοντας δύο

διαφορετικές εικόνες και παράγοντας μια καινούργια. Η τεχνική Mixup εκπαιδεύει το μοντέλο συνδυάζοντας ζεύγη εικόνων με τις αντίστοιχες ετικέτες τους σε γραμμική παρεμβολή (linear interpolation) αντί να χρησιμοποιεί αυτόνομες μεμονωμένες εικόνες. Ο τρόπος με τον οποίο λειτουργεί είναι ο εξής: Έστω x_i με label y_i (εξίσωση 3.1) και x_j με label y_j (εξίσωση 3.2), λ είναι ο συντελεστής ανάμειξης, x_i και y_i είναι το διάνυσμα χαρακτηριστικών του πρώτου τυχαίου δείγματος και x_j y_j είναι το διάνυσμα χαρακτηριστικών του δεύτερου τυχαίου δείγματος, η παραγόμενη συνδυαστική τεχνητή εικόνα $\tilde{x}$ (new data) (εξίσωση 3.1) είναι η αντίστοιχη ετικέτα είναι η $\tilde{y}$ (εξίσωση 3.2):

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j \tag{3.1}$$

και η αντίστοιχη ετικέτα - label (new label) είναι η παρακάτω εξίσωση:

$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j \tag{3.2}$$

Το $\lambda \in [0,1]$ ανήκει στην Beta distribution $\lambda \sim \text{Beta}(\alpha,\alpha)$ και το $\alpha > 0$ ορίζει την επήρεια της παρεμβολής με τυπικές τιμές του $\alpha \in [0.1,0.4]$.

Η χρήση Beta κατανομής δίνει την δυνατότητα ρύθμισης της ποσότητας της ανάμειξης. Μια μικρή τιμή του α οδηγεί τις τιμές του λ κοντά στο 0 και στο 1, οπότε και η παραγόμενη εικόνα πλησιάζει στις αρχικές τιμές. Αντίθετα, οι μεγάλες τιμές του α δίνουν μια παραγόμενη εικόνα με ισορροπημένη συγχώνευση [51]. Η χρησιμοποίηση της βοηθάει ώστε το μοντέλο να μην έχει μεγάλο overfitting, να αυξήσει την γενίκευση και να έχει ανεκτικότητα σε θόρυβο που είναι πιθανό να βρει. Η χρησιμότητα της είναι μεγάλη σε σύνολα δεδομένων που παρουσιάζουν μεγάλη μεταβλητότητα. Επίσης, δίνει έναν απλό αλλά ισχυρό μαθηματικά μηχανισμό ενίσχυσης των μοντέλων ανίχνευσης. Ένα ακόμα σημαντικό χαρακτηριστικό είναι, ότι με τη χρήση του Mixup μειώνεται η αποτύπωση των λανθασμένων ετικετών (corrupt labels) στην «μνήμη» του μοντέλου. Η τεχνική Mixup είναι ευέλικτη και μπορεί να χρησιμοποιηθεί σε τομείς της υπολογιστικής όρασης σε θέματα που σχετίζονται με ποικίλες επιστήμες αλλά και με σύνολα δεδομένων και προβλήματα, όπως για παράδειγμα αυτά της φυσικής γλώσσας (Natural Language Problems - NLP).

Στις δοκιμές που έγιναν, στο σύνολο δεδομένων CIFAR-10 με παραλλαγές των μοντέλων ResNet και DenseNet, η τεχνική οδήγησε σε βελτίωση του σφάλματος κατά 1.0 έως 1.4 μονάδες, ενώ αντίστοιχα στο σύνολο δεδομένων CIFAR-100 σημειώθηκε βελτίωση από 1.9 έως 4.5 μονάδες. Η τεχνική Mixup έχει πολύ μικρό υπολογιστικό κόστος, ενώ η υλοποίηση και εκπαίδευση των δεδομένων με τη χρήση της είναι αρκετά απλή και σαφής. Επιπλέον, βοηθάει το μοντέλο να συμπεριφέρεται γραμμικά ανάμεσα στα παραδείγματα του συνόλου δεδομένων [51].

3.2.3 Ανισοκατανομή δεδομένων

Η ανισοκατανομή σε ένα σύνολο δεδομένων είναι πολύ συχνό πρόβλημα και υπάρχει σχεδόν σε όλα τα σύνολα δεδομένων. Στην ανισοκατανομή οι εμφανίσεις των αντικειμένων δεν κατανέμονται ομοιόμορφα, με αποτέλεσμα κάποια αντικείμενα να εμφανίζονται πιο συχνά σε σχέση με κάποια άλλα. Παρατηρείται μεταξύ των κλάσεων (between class imbalance), όπως σε ένα σύνολο δεδομένων με οχήματα, όπου υπάρχουν παρά πολλές εμφανίσεις αυτοκίνητων, λιγότερες μοτοσυκλετών, και ακόμα λιγότερες ποδηλάτων. Αναλύεται σε δύο υποκατηγορίες. Η πρώτη περίπτωση είναι η Foreground - Background imbalance, στην οποία τα αντικείμενα που εμφανίζονται στο φόντο σε σχέση με αυτά που εμφανίζονται στο προσκήνιο δεν είναι ίσα. Ενώ στην περίπτωση της Foreground - Foreground ανισοκατανομής, η ανισοκατανομή παρατηρείται μεταξύ των αντικειμένων που βρίσκονται στο προσκήνιο [52].

Κάποιες πιο συγκεκριμένες περιπτώσεις ανισοκατανομής σε σύνολα δεδομένων εικόνων και ειδικότερα σε περιπτώσεις που αφορούν τον εντοπισμό αντικειμένων, αφορά το μέγεθος και η τοποθεσία των πλαισίων οριοθέτησης (bounding box) [53].

Η ύπαρξη μεγάλων ανισοκατανομών οδηγεί το μοντέλο σε bias ως προς τα αντικείμενα που εμφανίζονται σε κυρίαρχες κλάσεις. Ένα άλλο πρόβλημα που προκύπτει είναι ότι γίνεται πολύ δύσκολο το μοντέλο να μάθει τα αντικείμενα που βρίσκονται σε κλάσεις που υποεκπροσωπούνται. Τέλος, το μοντέλο έχει μειωμένη αποτελεσματικότητα και μειωμένη ικανότητα γενίκευσης [53].

Η μελέτη του Crasto [53] ερεύνησε τρεις τρόπους για τον έλεγχο της ανισοκατανομής μεταξύ κλάσεων. Ειδικότερα, την περίπτωση της Foreground – Foreground, εξετάστηκαν η Sampling (Class - aware sampling και Repeat Factor Sampling), η loss weighing και η μέθοδος της επαύξησης (augmentation techniques).

Η μέθοδος Class - Aware sampling βελτιστοποιεί την σύνθεση των mini - batches κατά την διάρκεια της εκπαίδευσης με σκοπό την καλύτερη και πιο καλά κατανεμημένη εκπροσώπηση των κλάσεων [53]. Ο τρόπος λειτουργίας της είναι ο εξής. Δημιουργείται μια λίστα των κλάσεων και μια όμοια λίστα με εικόνες που αντιστοιχούν στην κάθε κλάση. Στην συνέχεια, κατά την διάρκεια της εκπαίδευσης, και επιλέγοντας μια κλάση και μια εικόνα που να περιέχει αυτή την κλάση, δημιουργούνται τα mini batches. Αυτό βοηθάει στην βελτίωση της εκπροσώπησης των μικρών κλάσεων, αλλά οδηγεί στο πρόβλημα υποεκπροσώπησης μεγάλων κλάσεων [53].

Η μέθοδος Repeat Factor sampling [53] λειτουργεί κάνοντας oversampling κλάσεις αντικειμένων που είναι υποεκπροσωπούμενες. Ο αλγόριθμος προσαρμόζει συνεχώς το sampling probability κατά την εκπαίδευση. Η μέθοδος αυξάνει την πιθανότητα κλάσεις που δεν εμφανίζονται συχνά, να αυξάνουν την συχνότητα συμμετοχής τους στην εκπαίδευση, πετυχαίνοντας μια ισορροπία της εμφάνισης των κλάσεων.

Η μέθοδος loss reweighting [53] τροποποιεί την loss function με σκοπό να συνυπολογίζει τις συχνότητες των κλάσεων. Η συνάρτηση με αυτό τον τρόπο προστατεύει το αλγόριθμο από το να επικρατήσει κάποια από τις κλάσεις έναντι των υπολοίπων.

Τέλος, για την μέθοδο της επαύξησης προτείνονται οι Mosaic και Mixup [53]. Στην πρώτη μέθοδο, δημιουργείται μια νέα εικόνα από την ένωση τεσσάρων εικόνων από το σύνολο δεδομένων. Η Mosaic [53] μπορεί να διαχειριστεί τα labels των αρχικών εικόνων, χωρίς απώλεια πληροφορίας. Βοηθάει να προβληθούν στο μοντέλο εικόνες με διαφορετικές αναλογίες και κλίμακες, ενώ το περιεχόμενο μεταβάλλεται, καθώς οι τέσσερις εικόνες που συρράπτονται σε μια περιέχουν αντικείμενα και φόντα που ποικίλλουν.

Η δεύτερη μέθοδος δημιουργεί μια καινούργια εικόνα με τον γραμμικό συνδυασμό ζευγαριών εικόνων. Οι συγγραφείς [53] τροφοδοτούν το Mixup με πιο ποικίλες εικόνες προερχόμενες από την τεχνική του Mosaic για καλύτερη γενίκευση, με αποτέλεσμα να παράγουν ακόμα πιο διαφοροποιημένες εικόνες.

Ένας επιπλέον τρόπος είναι η αφαίρεση κλάσεων που υπεραντιπροσωπεύονται σημαντικά σε σύγκριση με τις υπόλοιπες του συνόλου δεδομένων. Η τεχνική αυτή μπορεί να λειτουργήσει μόνο με σύνολα δεδομένων που έχουν πάρα πολύ μεγάλο πλήθος εικόνων. Μια διαφορετική προσέγγιση είναι η αντιγραφή εικόνων που περιέχουν αντικείμενα με περιορισμένο αριθμό εμφανίσεων. Η μέθοδος αποδίδει καλύτερα αν υπάρχει αρκετά μεγάλο πλήθος εικόνων προς αντιγραφή, με αντικείμενα που υποεκπροσωπούνται. Η βέλτιστη επιλογή, εφόσον υπάρχει ο χρόνος και οι διαθέσιμοι πόροι, είναι η συλλογή δεδομένων κλάσεων που εμφανίζονται λίγες φορές. Αυτό μπορεί να δώσει δεδομένα από τον πραγματικό κόσμο σε αντίθεση με όλες τις υπόλοιπες μεθόδους όπου τα δεδομένα που δημιουργούνται

είναι συνθετικά, και αναλύονται στο παρών κεφαλαίο. Τα περισσότερα σύνολα δεδομένων έχουν το χαρακτηριστικό της ανισοκατανομής των αντικειμένων μεταξύ των κλάσεων, γι' αυτό κρίσιμο βήμα στην αρχή της επεξεργασίας είναι η στατιστική ανάλυση της συχνότητας εμφάνισης τους, ώστε να επιλεγεί η καταλληλότερη μεθοδολογία για κάθε υπό εξέταση περίπτωση.

Κεφάλαιο 4ο: Μεθοδολογία και Αλγόριθμοι Ανίχνευσης

Το πρόβλημα της αναγνώρισης μικρών αντικειμένων σε εικόνες ανήκει στην μεγαλύτερη οικογένεια της αναγνώρισης αντικειμένων με μηχανική όραση (Computer Vision). Τα τελευταία χρόνια η εξέλιξη στον τομέα του Computer Vision γίνεται με γρηγορότερο ρυθμό και με καινοτόμες εφαρμογές εξαιτίας της ανάπτυξης των συνελικτικών νευρωνικών δικτύων και των αλγορίθμων τους που δίνουν μεγαλύτερες δυνατότητες σε σχέση με το παρελθόν. Καίρια είναι και η συμβολή της τεχνολογίας της παράλληλης επεξεργασίας στις κάρτες γραφικών που παρέχουν αρκετή ισχύ εκτέλεσης βαριών υπολογιστικών εργασιών σε σύντομο σχετικά χρόνο. Επιπροσθέτως, η ανάπτυξη που υπάρχει τα τελευταία χρόνια οφείλεται εν μέρει στην μεγάλη ζήτηση τεχνολογιών που βασίζονται στην μηχανική όραση, όπως, στην αυτόνομη οδήγηση, στις εφαρμογές έρευνας και διάσωσης, στην ιατρική, στην παραγωγή και στον ποιοτικό έλεγχο στην βιομηχανία.

Όλες οι εφαρμογές της μηχανικής όρασης απαιτούν υψηλή ακρίβεια και αξιοπιστία στα αποτελέσματα τους. Ερευνητικές ομάδες προσπαθούν να λύσουν προβλήματα απόδοσης των αλγορίθμων, αξιοποιώντας με τον καλύτερο δυνατό τρόπο την υπάρχουσα τεχνολογία, και επεκτείνοντας την μέσω της ανάπτυξης νέων τεχνολογιών.

Στο παρόν κεφάλαιο αναλύεται η μεθοδολογία και οι αλγόριθμοι που εφαρμόζονται σε προβλήματα ανίχνευσης μικρών αντικειμένων. Υιοθετούνται αλγόριθμοι State of the Art, οι οποίοι χρησιμοποιούνται ευρέως και έχουν αποδείξει μέσα από δοκιμές σε μεγάλα σύνολα δεδομένων γενικού περιεχομένου ότι αποδίδουν καλά.

Ως μικρά αντικείμενα θεωρούνται εκείνα με μέγεθος έως 32 x 32 pixels και καταλαμβάνουν λιγότερο από το 1% της εικόνας. Έχουν μικρή ανάλυση και πολλές φορές παρουσιάζουν απόκρυψη των χαρακτηριστικών τους λόγω της επικάλυψης και μεγάλης ομοιότητας μεταξύ τους, ενώ το φόντο του της εικόνας περιπλέκει τον διαχωρισμό των αντικειμένων.

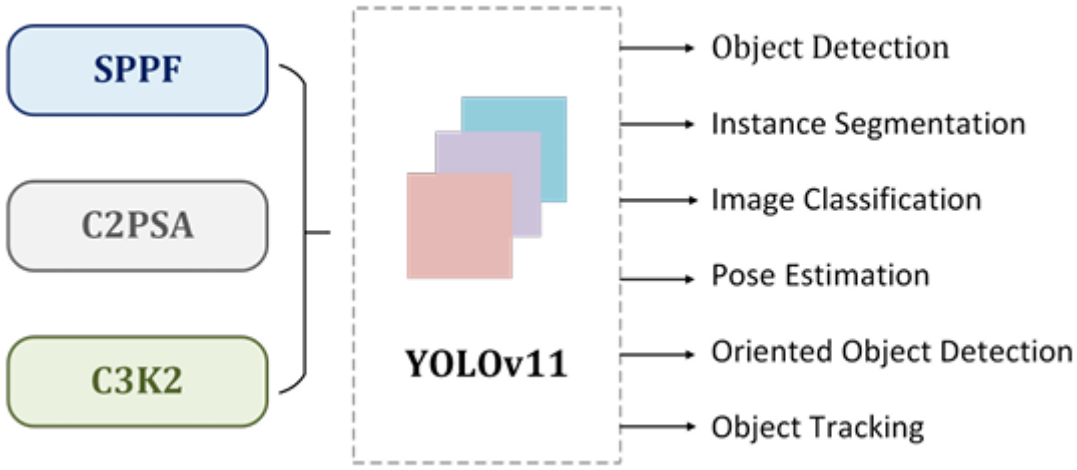
4.1 Τεχνολογικό υπόβαθρο

Τα βασικά μοντέλα με τα οποία απασχολείται η εργασία είναι τα YOLOv11 και RT-DETR μέσα από το framework των Ultralytics.

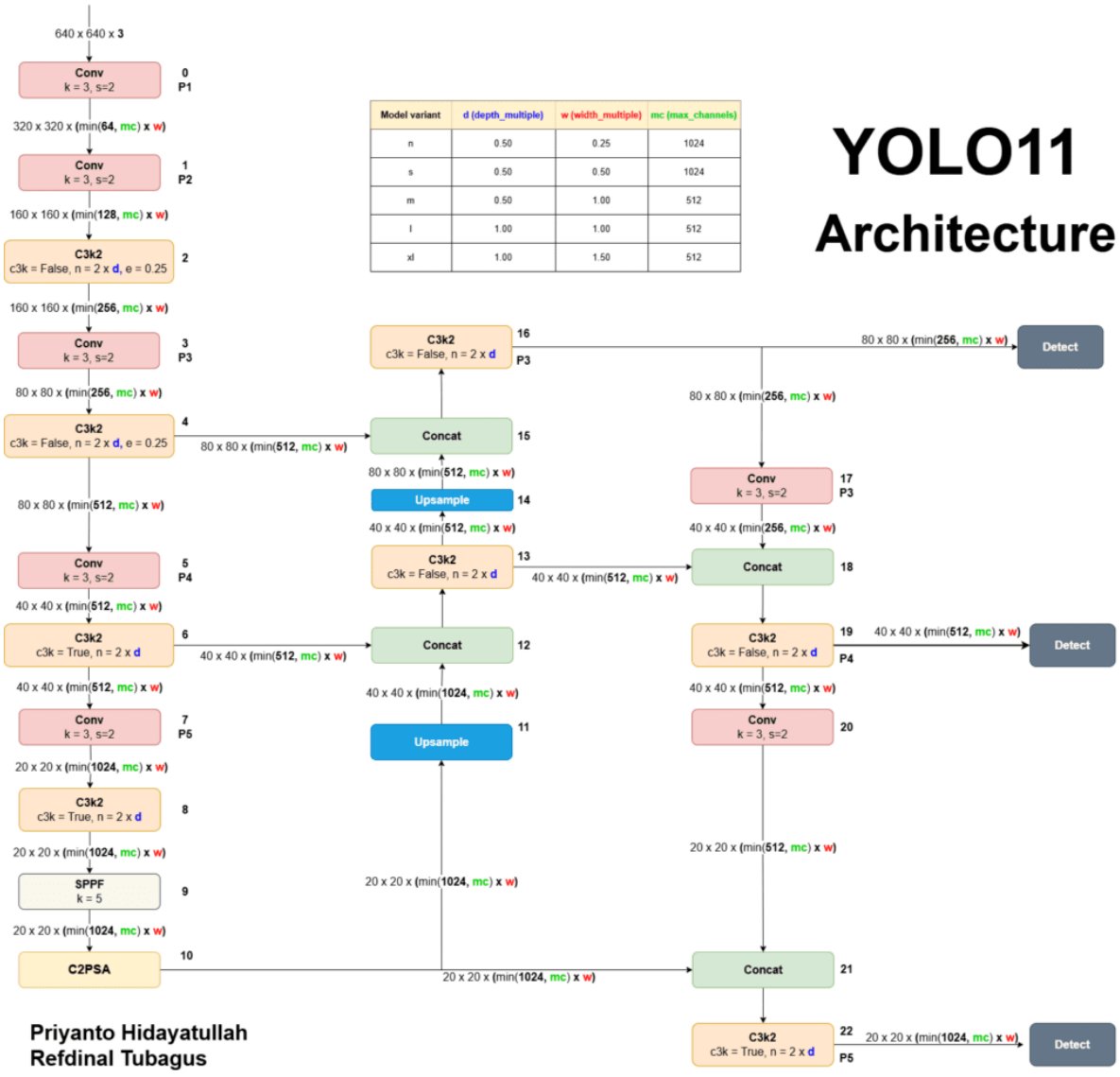
4.1.1 Το μοντέλο YOLOv11

Το YOLOv11 είναι ένα μοντέλο ενός βήματος (One Stage) με πολλές δυνατότητες (εικόνα 4.1), το οποίο σε ένα του πέρασμα σαρώνει ολόκληρη την εικόνα και εντοπίζει τα αντικείμενα μαζί με τις συντεταγμένες τους. Το YOLOv11 αποτελεί εξέλιξη του πρώτου αλγορίθμου YOLO που δημιουργήθηκε από τον Redmon et al. το 2015 [54]. Χρησιμοποιεί ένα ενιαίο συνελικτικό δίκτυο (εικόνα 4.2) τόσο για να προβλέψει τα πλαίσια οριοθέτησης (bounding boxes) όσο και για να προβλέψει τις κατηγορίες των αντικειμένων [54].

Το backbone δίκτυο είναι ο βασικός μηχανισμός εξαγωγής χαρακτηριστικών, και βασίζεται σε συνελικτικά νευρωνικά δίκτυα για να μετατρέψει τις εισόδους (εικόνες) σε multi-scale feature maps [54]. Το δεύτερο μέρος της αρχιτεκτονικής του είναι ο λαιμός (neck) ο οποίος αποτελεί ένα ενδιάμεσο στάδιο στο οποίο γίνεται η επεξεργασία, και όπου με τη βοήθεια συγκεκριμένων επιπέδων ενισχύονται τα χαρακτηριστικά σε ένα μεγάλο εύρος κλιμάκων [54]. Στο τρίτο μέρος είναι η κεφαλή όπου γίνονται οι τελικές προβλέψεις των κατηγοριών και των ορίων του κάθε αντικειμένου [54].



Εικόνα 4.1: Οι δυνατότητες του μοντέλου YOLOv11 [54].



Εικόνα 4.2: Η αναλυτική αρχιτεκτονική του μοντέλου YOLOv11 [55].

Μια αλλαγή στην έκδοση YOLOv11 είναι ότι στο backbone αντικαταστάθηκε το C2f block με το C2k2 που είναι πιο γρήγορο και αποδοτικό υπολογιστικά, και αποτελείται από δύο μικρότερα συνελικτικά δίκτυα αντί για ένα μεγάλο σε σχέση με την προηγούμενη έκδοση του [54]. Μια ακόμα αλλαγή είναι ότι το C2PSA module δίνει αυξημένο spatial attention, το οποίο βοηθάει το μοντέλο να επικεντρωθεί μόνο στις περιοχές που έχουν σημασία για τον εντοπισμό αντικειμένων. Τέλος, μια ακόμα αλλαγή είναι στην κεφαλή του μοντέλου YOLOv11 που ενσωματώνονται τα CBS Blocks (Convolution-BatchNorm-Silu) [54]. Αυτά τα CBS επίπεδα τοποθετούνται μετά από τα C3k2 Blocks για ακόμα καλύτερη ακρίβεια μέσα από την εξαγωγή ορθότερων χαρακτηριστικών, την βελτίωση της ροής των δεδομένων μέσα από την διαδικασία του batch normalization, και της απόδοσης μέσα από την χρήση της συνάρτησης ενεργοποίησης (SiLU) [54].

Στο μοντέλο εφαρμόζεται η διαδικασία fine tuning με τα σύνολα δεδομένων VisDrone και DOTA. Το VisDrone αποτελείται από κεκλιμένες εικόνες που έχουν ληφθεί από UAV με πλαίσια οριοθέτησης HBB και το DOTA αποτελείται από δορυφορικές λήψεις με πλαίσια οριοθέτησης OBB. Τα δυο αυτά σύνολα δεδομένων είναι αξιόπιστα και χρησιμοποιούνται για την αξιολόγηση αλγορίθμων ανίχνευσης μικρών αντικειμένων.

4.1.2 Το μοντέλο RT-DETR

Το RT-DETR (εικόνα 4.3) είναι ένα μοντέλο βασισμένο σε transformers που σχεδιάστηκε για εφαρμογές πραγματικού χρόνου και έχει ενιαία δομή από άκρη σε άκρη [56], [57]. Σε αντίθεση με τα περισσότερα μοντέλα που βασίζονται σε transformers και διαχωρίζουν τον εντοπισμό αντικειμένων από την κατηγοριοποίηση, στο RT-DETR γίνονται και οι δύο λειτουργίες σε ένα δίκτυο με σκοπό την αύξηση της αποδοτικότητας [58].

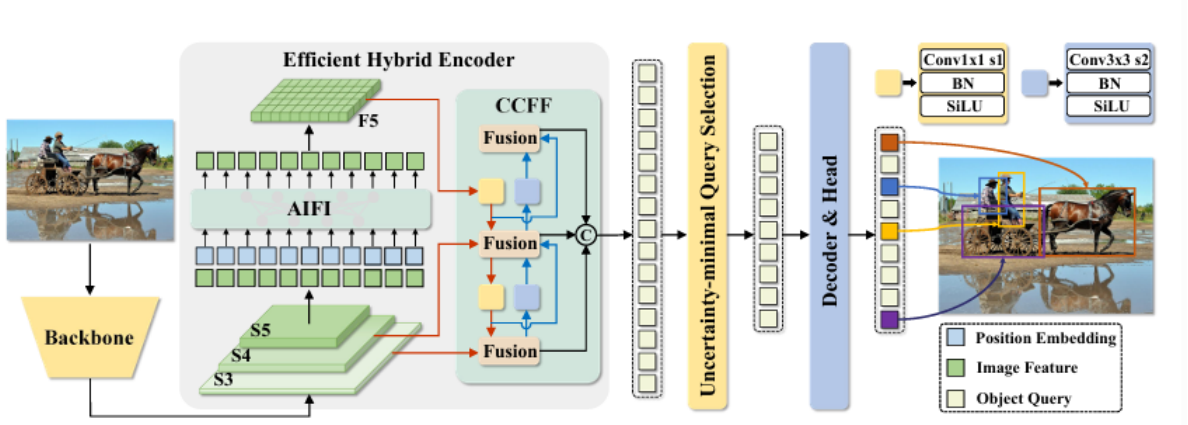
Για την διαχείριση δεδομένων πολλαπλών κλιμάκων σχεδιάστηκε ένας υβριδικός encoder που δίνει καλύτερη ταχύτητα στο inference αντικαθιστώντας τον κλασσικό transformer encoder. Αντικείμενα με χαμηλό σκορ στο localization confidence αποφεύγονται από το να επιλεγούν μέσω του uncertainty-minimal query selection που τροφοδοτεί τον decoder με αρχικά queries υψηλής εμπιστοσύνης παρέχοντας καλύτερη ακρίβεια στις προβλέψεις [58]. Μια ακόμα αλλαγή του αλγορίθμου RT-DETR είναι ότι δεν εξαρτάται από άγκυρες (anchor free) ούτε στο στοιχείο NMS, αλλά αξιοποιεί bipartite matching για να πραγματοποιεί τις προβλέψεις απευθείας [58], [59].

Ο encoder είναι υβριδικός και αποτελείται από δύο στοιχεία, το μεν πρώτο είναι το Intra-scale Feature Interaction (AIFI) και το δεύτερο είναι το CNN-based Cross-scale Feature Fusion (CCFF) [58], [59]. Το πρώτο στοιχείο μειώνει το υπολογιστικό κόστος κάνοντας καλύτερη διαχείριση των intra-scale interactions [58]. Το πετυχαίνει κάνοντας αυτούς τους υπολογισμούς σε high level επίπεδα οπότε και δίνει αρκετή πληροφορία που μπορεί να χρησιμοποιηθεί για την εξαγωγή των συνδέσεων μεταξύ των αντικειμένων, των τοποθεσιών και την περαιτέρω αναγνώριση αντικειμένων από τα επόμενα επίπεδα του δικτύου [58]. Το στοιχείο CCFF εισάγει fusion blocks και σκοπό έχει να συνδυάσει αντικείμενα κοντινών κλιμάκων σε νέα [58].

Το στοιχείο uncertainty-minimal Query selection είναι υπεύθυνο για την δημιουργία και την παροχή ερωτημάτων στον decoder [58]. Η βελτίωση, σε σχέση με τα απλά μοντέλα DETR, είναι ότι η επιλογή των queries γίνεται με βάση την επιστημική ή γνωστική αβεβαιότητα (epistemic uncertainty) έναντι της ταξινόμησης [58].

Ο αλγόριθμος RT-DETR μπορεί να δώσει αποτελέσματα με υψηλή ακρίβεια όπως αυτά των μοντέλων που βασίζονται σε transformers, με το πλεονέκτημα ότι μπορούν να εφαρμοστούν σε πραγματικό χρόνο [58]. Ωστόσο, έχει μεγαλύτερη πολυπλοκότητα σε σχέση με τα μοντέλα YOLO και χρειάζεται

περισσότερους υπολογιστικούς πόρους και χρόνο για εκπαίδευση συγκλίνοντας πιο αργά με αυτά τα μοντέλα [58].



Εικόνα 4.3: Η αρχιτεκτονική του μοντέλου RT-DETR [58].

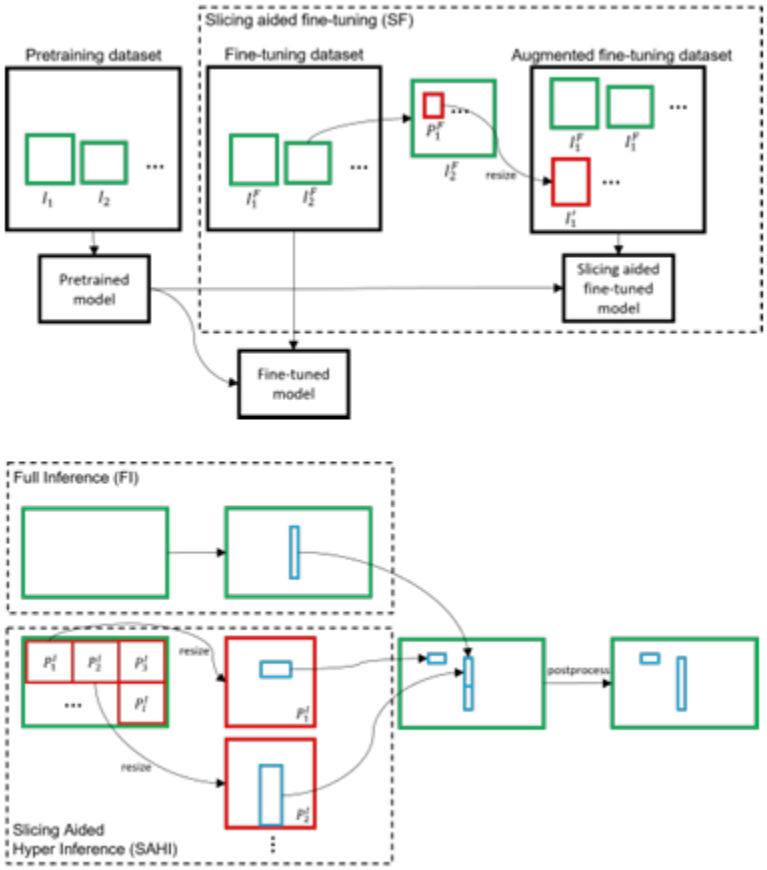
Το μοντέλο RT-DETR, όπως και το μοντέλο YOLOv11, περνάει μέσα από την διαδικασία του fine tuning με τα σύνολα δεδομένων VisDrone και DOTA. Αυτό οφείλεται στο γεγονός ότι το σύνολο δεδομένων VisDrone είναι σύνθετο και απαιτείται η ύπαρξη κοινής βάσης σύγκρισης των αποτελεσμάτων, η οποία και θα αναλυθεί σε 5ο κεφάλαιο.

4.1.3 Τεχνικές Ενίσχυσης Αναγνώρισης Μικρών Αντικειμένων

Αν και τα σύγχρονα μοντέλα έχουν πολύ υψηλές δυνατότητες και απόδοση σε σχέση με το παρελθόν, ακόμα δυσκολεύονται στον εντοπισμό μικρών αντικειμένων. Έχουν αναπτυχθεί τεχνικές που βοηθούν τα μοντέλα να αυξήσουν την ακρίβεια τους. Μια τεχνική είναι η SAHI (Slicing Aided Hyper Inference). Το SAHI είναι ένα framework που αυξάνει την απόδοση των μοντέλων σε εργασίες ανίχνευσης μικρών αντικειμένων τεμαχίζοντας την εικόνα σε τμήματα (εικόνα 4.4) και εκτελώντας την ανίχνευση σε κάθε επιμέρους τμήμα αυτής [60]. Τα τμήματα της αρχικής εικόνας έχουν αλληλοεπικάλυψη ώστε όσα αντικείμενα είναι στα όρια των τμημάτων να μην χαθούν ως πληροφορία [60], [61]. Όταν εκτελεστεί η ανίχνευση σε όλα τα τμήματα αυτά επανενώνονται ξανά στην αρχική ενιαία εικόνα. Το τεμάχισμα της εικόνας σε τμήματα έχει ως αποτέλεσμα τα αντικείμενα του κάθε τμήματος να αυξάνονται σε μέγεθος, διευκολύνοντας τον εντοπισμό τους [60].

Ο τρόπος λειτουργίας είναι αρκετά εύκολος και κατανοητός. Αρχικά, η εικόνα εισόδου χωρίζεται σε τμήματα (π.χ. 512 x 512 pixels) με αλληλοεπικαλύψεις στα όρια των τμημάτων [61]. Το ποσοστό της αλληλοεπικάλυψης μπορεί να οριστεί με τον ίδιο τρόπο που ορίζεται και το μέγεθος των τμημάτων [61]. Στην συνέχεια, πάνω στο κάθε τμήμα εκτελείται ο αλγόριθμος ανίχνευσης (π.χ. YOLOv11). Εν τέλει, αφού ελεγχθούν όλα τα τμήματα από τον αλγόριθμο ανίχνευσης, επανενώνονται στην ενιαία εικόνα και εφαρμόζεται NMS (Non Maximum Suppression) για την διαγραφή των διπλότυπων [60], [61]. Μια προαιρετική λειτουργία είναι η εκτέλεση του SAHI σε ολόκληρη την εικόνα (Full Inference - FI) χωρίς τον κατακερματισμό της, με σκοπό την εύρεση ευκολότερων αντικειμένων μεγαλύτερου μεγέθους [61]. Αν χρησιμοποιηθεί και η προαιρετική λειτουργία FI, τότε τα αποτελέσματα των παραθύρων συγχωνεύονται μαζί με αυτά της FI και εκτελώντας NMS αποκλείονται τα διπλότυπα, παράγοντας τις τελικές βελτιωμένες προβλέψεις [61]. Ένα επιπλέον χαρακτηριστικό χρήσης του SAHI, μέσα από το framework των Ultralytics, είναι η δυνατότητα προβλέψεων σε δέσμες (Batch Prediction) χάρη στην οποία οι προβλέψεις εκτελούνται πολύ ταχύτερα και το inference time μειώνεται σημαντικά [60].

Η μέθοδος μπορεί να εφαρμοστεί στα περισσότερα μοντέλα ανίχνευσης αντικειμένων και είναι ιδιαίτερα αποδοτική σε μικρά αντικείμενα. Μπορεί να εφαρμοστεί τόσο σε πλαίσια HBB όσο και σε πλαίσια OBB, χωρίς να χρειάζονται αλλαγές στις διαδικασίες εκπαίδευσης ή επανεκπαίδευσης κάποιου μοντέλου για τη χρήση του SAHI.



Εικόνα 4.4: Η αρχή λειτουργίας του SAHI [61].

4.1.4 Επαύξηση δεδομένων (Data Augmentation)

Στα δεδομένα έγινε επαύξηση (data augmentation) για την ενίσχυση των σύνολα δεδομένων. Η επαύξηση πραγματοποιήθηκε εντός του framework των Ultralytics μετά από την φάση των dataloaders (οι εικόνες δηλαδή, δεν είναι κάπου τοπικά αποθηκευμένες στην επαυξημένη τους μορφή). Χρησιμοποιήθηκαν αρκετές παράμετροι για επαύξηση, όχι όμως όλες ταυτόχρονα καθώς ορισμένες αλληλοκαλύπτονται ως προς το αποτέλεσμα. Η επιλογή των παραμέτρων έγινε με κριτήριο οι παραγόμενες εικόνες να παραμείνουν ρεαλιστικές ως προς τα χαρακτηριστικά τους. Μερικές από τις παραμέτρους που χρησιμοποιήθηκαν, μεμονωμένα, ήταν οι παρακάτω:

- Degrees
- Translate
- Scale
- Perspective
- Mosaic
- Mixup
- Cutmix

4.1.4.1 Weight Decay

Το weight decay [62] είναι μια τεχνική κανονικοποίησης στη μηχανική μάθηση. Ο τρόπος λειτουργίας της βασίζεται στο ότι εισάγει μια παράμετρο στη συνάρτηση κόστους, η οποία «τιμωρεί» τα μεγάλα βάρη ώστε να μην γίνουν πολύ μεγάλα, με αποτέλεσμα το μοντέλο να γενικεύει καλύτερα σε καινούργια δεδομένα [62]. Βοηθάει στη μείωση του overfitting και αυξάνει τη σταθερότητα καθώς έχοντας μικρότερα βάρη, το μοντέλο συγκλίνει ομαλότερα και δίνει καλύτερη γενίκευση [62]. Μπορεί να γίνει χρήση των optimizers SGD και AdamW με πολύ καλά αποτελέσματα. Ωστόσο, απαιτείται διερεύνηση της βέλτιστης τιμής, καθώς αν είναι πολύ μικρή δεν επιφέρει κάποιο αποτέλεσμα και ενδέχεται να συμβάλλει στο overfitting, ενώ αν είναι πολύ μεγάλη τα βάρη τείνουν προς το μηδέν, καθιστώντας το μοντέλο αδύναμο και οδηγώντας σε underfitting [63]. Στο πλαίσιο του SGD, η μέθοδος (weight decay) είναι ισοδύναμη της L2 κανονικοποίησης και συχνά αναφέρεται και ως τέτοια [62],[64].

4.1.4.2 Multi-scale training

Στην διαδικασία της εκπαίδευσης εντάχθηκε και η παράμετρος του multi-scale. Το multi-scale training συμβάλει στην ενίσχυση της απόδοσης του μοντέλου και την ικανότητα του να γενικεύσει καλύτερα. Εφαρμόζοντας την τεχνική του multi-scale επιτρέπεται στο μοντέλο να «δει» εικόνες σε διαφορετικές κλίμακες και να εκτεθεί σε ποικίλες αναπαραστάσεις των ίδιων αντικειμένων. Η εφαρμογή της μεθόδου μεταβάλει τις εικόνες εισόδου κατά ένα ποσοστό το οποίο ορίζεται από τον ερευνητή, με αποτέλεσμα ορισμένες να μικραίνουν και κάποιες άλλες να μεγεθύνονται. Αυτό σημαίνει ότι ορισμένα αντικείμενα μικραίνουν σε μέγεθος και ο εντοπισμός τους γίνεται πιο δύσκολος, ενώ η απώλεια μέρους «ευκρίνειας» τους δυσκολεύει τον αλγόριθμο, καθιστώντας τον πιο ανθεκτικό σε απαιτητικές εικόνες. Το multi-scale training είναι αρκετά κοστοβόρο υπολογιστικά και για αυτό τον λόγο εφαρμόστηκε μόνο στις μικρότερες εκδόσεις του μοντέλου YOLOv11. Η παράμετρος ορίστηκε στο 0.5, το οποίο σημαίνει ότι οι εικόνες μεταβάλλονται με έναν παράγοντα από 0.5% έως 1.5%. Επειδή, τα αντικείμενα είναι ήδη μικρά, δεν επιλέχθηκε μεγαλύτερη τιμή, καθώς περαιτέρω σμίκρυνση καθιστά τον εντοπισμό ιδιαίτερα απαιτητικό.

4.2 Μεθοδολογία Υλοποίησης και Πειραματικής Δοκιμής

Στις επόμενες παραγράφους αναλύονται η μεθοδολογία υλοποίησης και το πλάνο τον πειραμάτων που ακολουθείται. Παρουσιάζονται τεχνικές fine-tuning που εφαρμόζονται με σκοπό την καλύτερη προσαρμογή του μοντέλου σε ένα συγκεκριμένο πρόβλημα όπως επίσης και οι παραλλαγές που έγιναν ως αποτέλεσμα της εφαρμογής των υπερπαραμέτρων.

4.2.1 Προεπεξεργασία δεδομένων και annotations

Αρχικά, ορίστηκαν τα σύνολα δεδομένων που χρησιμοποιήθηκαν στα πειράματα. Επιλέχθηκαν δυο σύνολα δεδομένων, με το πρώτο να είναι το VisDrone και το δεύτερο το DOTA. Το μεν πρώτο για τις λήψεις υπό κλίση από UAV και το δεύτερο για τις λήψεις από δορυφόρο με επιπλέον χαρακτηριστικό ότι τα πλαίσια οριοθέτησης ήταν προσανατολισμένα. Τα δυο αυτά σύνολα δεδομένων δίνουν μεγάλη ποικιλομορφία και παραδείγματα πραγματικού κόσμου.

Στα σύνολα δεδομένων δεν έγινε κάποια αλλαγή ή κανονικοποίηση στην ανάλυση τους πριν εισαχθούν στο μοντέλο. Αυτό έγινε για να κρατηθούν, όσο γίνεται καλύτερα, τα σύνολα δεδομένων κοντά σε πραγματικές συνθήκες. Τα σύνολα δεδομένων παρέχονται ήδη χωρισμένα σε train, validate, test χωρίς να πραγματοποιηθεί κάποια τροποποίηση στη δομή τους, καθώς βάση βιβλιογραφίας χρησιμοποιούνται με τον υφιστάμενο διαχωρισμό εικόνων και αυτός αποτελεί μέσο αξιολόγησης και σύγκρισης

αποτελεσμάτων σε δημοσιευμένες εργασίες. Τέλος, οι ετικέτες (labels) των συνόλων δεδομένων μετασχηματίστηκαν σε μορφή που να είναι συμβατή με το YOLO, και γενικότερα με το framework των Ultralytics. Το script για τις μετασχηματισμένες ετικέτες είναι διαθέσιμο στο παράρτημα.

Για τις δοκιμές χρησιμοποιήθηκαν δυο σύγχρονα μοντέλα: το πρώτο βασίζεται στο YOLO όπου η αρχιτεκτονική του είναι One-Stage, και το δεύτερο βασίζεται στο RT-DETR, ένα μοντέλο αρχιτεκτονικής transformer προσαρμοσμένο για εφαρμογές πραγματικού χρόνου. Η επιλογή των συγκεκριμένων έγινε λόγω της απόδοσης, της ταχύτητας και της διερεύνησης της δυνατότητας να εφαρμοστούν σε edge devices. Τα μοντέλα χρησιμοποιήθηκαν στην προεκπαιδευμένη τους μορφή με τα βάρη τους να προέρχονται από εκπαίδευση με το σύνολο δεδομένων MS COCO, που είναι ένα μεγάλο σύνολο εικόνων γενικού ενδιαφέροντος. Έπειτα, πραγματοποιήθηκε fine-tuning πάνω στα δεδομένα του συνόλου δεδομένων VisDrone. Το fine-tuning έγινε χωρίς την τροποποίηση κάποιας υπερπαράμετρου (hyperparameter), προκειμένου να υπάρχει μια βάση αναφοράς και να είναι δυνατή η αξιολόγηση των επιπτώσεων από τυχόν αλλαγές στην ρύθμιση του μοντέλου.

4.2.2 Εκπαίδευση μοντέλων – Fine-tuning

Τα μοντέλα δοκιμάστηκαν με τρεις optimizers: τον Adam, τον AdamW και τον SGD. Ο Adam είναι ο πιο διαδεδομένος optimizer, καθώς έχει το πλεονέκτημα της ταχύτητας και είναι αρκετά κατανοητός στη λειτουργία του. Διατηρεί ένα learning rate για κάθε βάρος το οποίο ανανεώνεται ξεχωριστά. Επιλέγει να δώσει ένα μικρότερο learning rate για τις παραμέτρους που ανανεώνονται συχνά και ένα μεγαλύτερο για στοιχεία που ανανεώνονται πιο αραιά [65]. Αυτή η δυνατότητα είναι χρήσιμη για σύνολα δεδομένων που έχουν διαφορετικές κλίμακες και η συμπεριφορά του αλγορίθμου αλλάζει στην διάρκεια της εκπαίδευσης. Ο AdamW είναι μια παραλλαγή του αλγορίθμου Adam και βελτιώνει τη συμπεριφορά του αλγορίθμου όταν γίνεται χρήση της παραμέτρου weight decay. Ο AdamW αποσυνδέει το weight decay από την δυναμική του learning rate και το εφαρμόζει ανεξάρτητα στις παραμέτρους του μοντέλου (βάρη), δίνοντας καλύτερη κανονικοποίηση και περιορισμό του overfitting [66]. Ο AdamW είναι επίσης πιο αποδοτικός ως προς τη χρήση της μνήμης ακόμη και σε μεγάλα μοντέλα με πολλές παραμέτρους [67]. Ο SGD (Stochastic Gradient Descent) είναι αρκετά αποδοτικός και ευέλικτος αλγόριθμος βελτιστοποίησης με εφαρμογή σε πολλές κατηγορίες προβλημάτων [68]. Αντί να υπολογίζει το gradient της συνάρτησης κόστους για ολόκληρο το σύνολο δεδομένων, ο αλγόριθμος υπολογίζει μόνο για ένα τυχαίο υποσύνολο, με αποτέλεσμα η εκτέλεση του να είναι ταχύτερη, ειδικά σε μεγάλα σύνολα δεδομένων [69]. Μπορεί να εκπαιδευτεί γρήγορα και αποδίδει καλά σε μεγάλα σύνολα δεδομένων, με μικρότερες απαιτήσεις μνήμης, δίνοντας γενικευμένες λύσεις [68].

4.2.3 Learning Rate Schedulers

Στις δοκιμές που έγιναν στη παρούσα εργασία, συμπεριλήφθηκε και η παράμετρος του Learning Rate Scheduler, με σκοπό να διευκολυνθεί η εκπαίδευση του μοντέλου και να βελτιωθεί η ικανότητα του να γενικεύει γρηγορότερα. Περιγράφηκαν αρκετοί αλγόριθμοι στην συγκεκριμένη εργασία, ωστόσο για τις ανάγκες των δοκιμών επιλέχθηκε ο αλγόριθμος Cosine Annealing LR.

Από τον κώδικα του YOLOv11, προκύπτει ότι ο προεπιλεγμένος αλγόριθμος Learning Rate Scheduler είναι ο CyclicLR. Σε αυτόν τον αλγόριθμο [70] ορίζεται ένα εύρος για το Learning Rate και οι τιμές επιλέγονται μέσα από αυτό. Αντί να διατηρείται σταθερό ή να μειώνεται συνεχώς το Learning Rate, ο CyclicLR επιτρέπει στις τιμές να αυξάνονται μεταξύ των επαναλήψεων. Ο αλγόριθμος CyclicLR έχει αποδειχθεί ότι είναι αποδοτικός, και βοηθάει στην συντομότερη εκπαίδευση του μοντέλου μας.

Ο αλγόριθμος Cosine Annealing [71] μειώνει ομαλά το learning rate, βασισμένος σε μια συνάρτηση συνημίτονου. Αρχικά, ξεκινάει από ένα προκαθορισμένο learning rate, και στη συνέχεια ανάλογα με τον αριθμό των εποχών (epochs), φτάνει ομαλά στο χαμηλότερο σημείο [72], [73]. Το Cosine Annealing βοηθάει στην ομαλή μείωση, επιφέροντας καλύτερη σύγκλιση, αποφεύγοντας τοπικά ελάχιστα και καθιστώντας το μοντέλο πιο ανθεκτικό, βελτιώνοντας συνολικά στην απόδοση του [72], [73].

4.2.4 Μετρικές αξιολόγησης απόδοσης

Η αξιολόγηση της απόδοσης του μοντέλου βασίστηκε σε κοινά αποδεκτές και καθιερωμένες μετρικές. Η επιλογή όμως των παραμέτρων προσαρμόστηκε και στη φύση του προβλήματος, δηλαδή το πρόβλημα της ανίχνευση μικρών αντικειμένων το οποίο απαιτεί την ύπαρξη στοχευμένων μετρικών.

Η λίστα των μετρικών που παρελήφθησαν ως βασικό μέσο αξιολόγησης είναι οι παρακάτω:

- mAP50%
  Είναι η μέση ακρίβεια (Mean Average Precision) με κατώφλι IoU = 0.5 και χρησιμοποιείται ευρέως στην αξιολόγηση μοντέλων. Δίνει την ικανότητα του μοντέλου να μπορεί να εντοπίζει σωστά τα αντικείμενα αλλά με «χαλαρή» απαίτηση επικάλυψης λόγω του ορισμού 50%.
- mAP50-95%
  Είναι η ίδια μετρική με την mAP@0.5 αλλά πιο αυστηρή. Υπολογίζει το mAP για το εύρος από 0.5 έως 0.95 IoU με βήμα 0.05, και δίνει καλύτερη εικόνα για την συνολική ακρίβεια του μοντέλου.
- Precision
  Υπολογίζει το ποσοστό των σωστών ανιχνεύσεων (True Positives) σε σχέση με το σύνολο των ανιχνεύσεων. Υψηλό precision σημαίνει ότι το μοντέλο αποφεύγει τα ψευδώς θετικά αποτελέσματα. Εκφράζεται με τον παρακάτω μαθηματικό τύπο (συνάρτηση 4.1) όπου TP είναι τα True Positives και FP τα False Positives:

$$Precision = \frac{TP}{TP+FP} \tag{4.1}$$

- Recall
  Υπολογίζει το ποσοστό των σωστών ανιχνεύσεων σε σχέση με όλα τα αντικείμενα. Υψηλό recall σημαίνει ότι το μοντέλο εντοπίζει το μεγαλύτερο μέρος των αντικειμένων. Εκφράζεται με τον παρακάτω μαθηματικό τύπο (συνάρτηση 4.2) όπου TP είναι τα True Positives και FN είναι τα False Negatives:

$$Recall = \frac{TP}{TP+FN} \tag{4.2}$$

- IoU (Intersection of Union)
  Είναι από τις βασικότερες μετρικές στην αξιολόγηση μοντέλων στην μηχανική όραση και χρησιμοποιείται για την εύρεση του κατά πόσο καλά μια προβλεπόμενη περιοχή (predicted bounding box) ταυτίζεται με την πραγματική περιοχή (ground truth bounding box) (εικόνα 4.5 και εικόνα 4.6). Εκφράζεται με τον παρακάτω μαθηματικό τύπο (συνάρτηση 4.3):

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{4.3}$$

- Inference time
  Ο χρόνος που χρειάζεται για την επεξεργασία μιας εικόνας. Δίνει ένα μέτρο αξιολόγησης για την χρήση που κάνει το μοντέλο πάνω στους υπολογιστικούς πόρους που εφαρμόζεται. Είναι

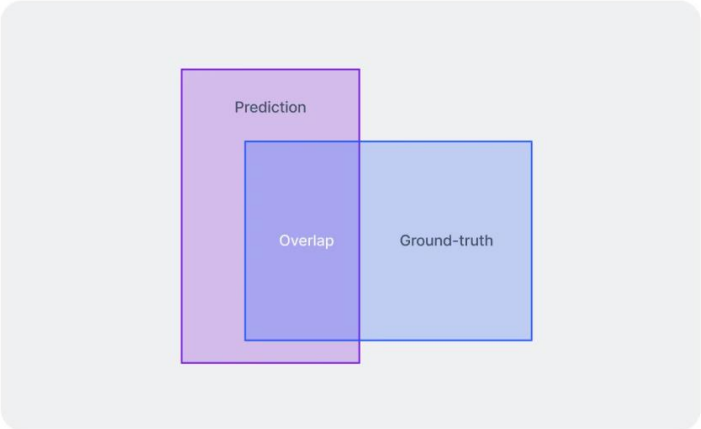
χρήσιμο στην κατανόηση του αν ένα μοντέλο κάνει έντονη χρήση των πόρων και αν πρέπει να εξεταστούν αλλαγές στη δομή ή την βελτίωση των διαθέσιμων πόρων, εφόσον απαιτείται. Η εξίσωση για το inference time προκύπτει από τη διαίρεση του συνολικού χρόνου με τον αριθμό συνολικών δειγμάτων. δίνοντας τον μέσο χρόνο inference ανά εικόνα. Σε frameworks όπως το Ultralytics, αυτός αναλύεται και στα στάδια επεξεργασίας. Εκφράζεται με τον παρακάτω μαθηματικό τύπο (συνάρτηση 4.4) όπου t_i είναι το inference time προς υπολογισμό, TP είναι τα True Positives και FP είναι τα False Positives:

$$t_i = \frac{Συνολικός\ χρόνος}{Αριθμός\ δειγμάτων} \tag{4.4}$$

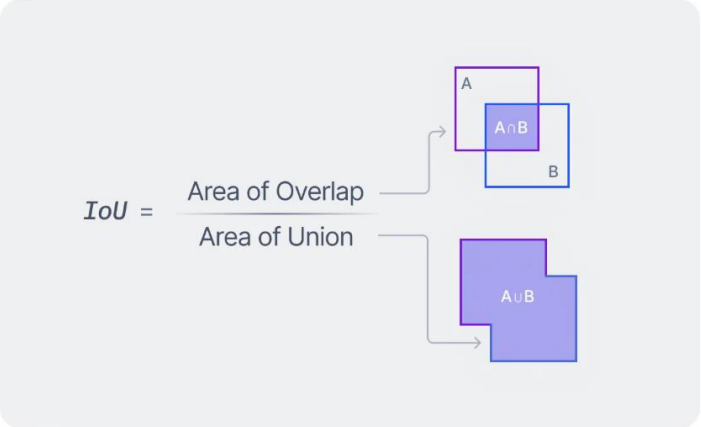
- FPS

  Η μετρική FPS (Frames Per Second) δίνει τον αριθμό καρέ που ένα μοντέλο μπορεί να επεξεργαστεί ανά δευτερόλεπτο. Η συγκεκριμένη μετρική ορίζει την δυνατότητα ένα μοντέλο να εφαρμοστεί σε εργασίες πραγματικού χρόνου. Ορίζεται από τον παρακάτω τύπο (συνάρτηση 4.4) και έχει άμεση με το inference time ανά εικόνα σε δευτερόλεπτα.

$$FPS = \frac{1}{inference\ time\ per\ image\ (seconds)} \tag{4.5}$$



Εικόνα 4.5: Η σχηματική αναπαράσταση της έννοιας της IoU σε high level [74].



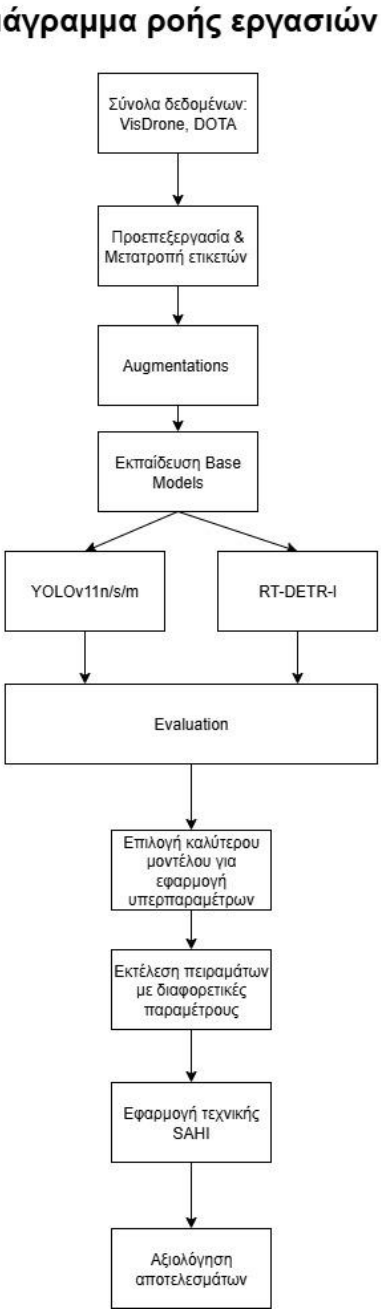
Εικόνα 4.6: Η σχηματική αναπαράσταση του μαθηματικού τύπου της μετρική IoU [74].

4.3 Δομή πειραμάτων

Αρχικά, εκτελέστηκαν από τον αλγόριθμο YOLO των Ultralytics οι εκδοχές YOLOv11n, YOLOv11s, YOLOv11m. Τα πρώτα πειράματα έγιναν σε 50 εποχές χωρίς να την χρήση κάποιας υπερπαραμέτρου,

προκειμένου να αξιολογηθεί ποιο μοντέλο βελτιωνόταν καλύτερα και παρέμενε σταθερό σε αυτό το εύρος. Η ροή εργασιών περιγράφεται στο διάγραμμα (Γράφημα 4.1). Τα επόμενα βήματα είχαν ως εξής:

- Πρώτα δοκιμάστηκε ο optimizer AdamW με τις default τιμές του μοντέλου δηλαδή, learning rate = 0.01, momentum = 0.937, image size = 640 και batch size = 8. Το batch size ορίστηκε στο οκτώ μόνο στο μοντέλο YOLOv11n, ενώ στα μεγαλύτερα μοντέλα YOLOv11s/m ορίστηκε σε τέσσερα καθώς είναι μεγαλύτερα και πιο απαιτητικά σε υπολογιστικούς πόρους.
- Στα επόμενα βήματα, εφαρμόστηκε μικρότερο learning rate με cosine annealing. Επίσης, καθώς από την βιβλιογραφία προκύπτει ότι ο AdamW και ο SGD λειτουργούν καλά με το weight decay, εφαρμόστηκε και αυτό στο πείραμα για να αξιολογηθεί η απόδοση του.
- Στα επόμενα πειράματα, διατηρήθηκαν οι παράμετροι του προηγούμενου βήματος και προστέθηκε η επαύξηση δεδομένων.



Γράφημα 4.1: Η ροή εργασιών των πειραμάτων της εργασίας

Για το μοντέλο RT-DETR-l, λόγω του υψηλού υπολογιστικού κόστους ακόμα και σε περιβάλλον Google Colab, το πρώτο πείραμα εκτελέστηκε με τις αρχικές του παραμέτρους. Εν συνεχεία, πραγματοποιήθηκε ακόμα ένα πείραμα 73 εποχών σε τοπικό υπολογιστή, μιας και το κόστος σε υπολογιστικές μονάδες δεν επέτρεπε την εκτέλεση του πειράματος σε Google Colab.

Κεφάλαιο 5ο: Πειραματική Αξιολόγηση και Αποτελέσματα

Στο παρόν κεφάλαιο παρατίθεται μια παρουσίαση των πειραματικών αποτελεσμάτων με μοντέλα ανίχνευσης για μικρά αντικείμενα. Στόχος είναι η αξιολόγηση της αποτελεσματικότητας του κάθε μοντέλου τόσο σε σύγκριση με τις μεθόδους που χρησιμοποιήθηκαν στην εργασία όσο και με τα αποτελέσματα που έχουν προκύψει από δημοσιεύσεις και καθιερωμένες προσεγγίσεις. Η αξιολόγηση έγινε και σε εφαρμογή με πραγματικές συνθήκες εκτέλεσης (inference) σε πραγματικό χρόνο.

5.1 Πειραματικό περιβάλλον

Τα πειράματα έγιναν σε δύο διαφορετικά περιβάλλοντα. Το πρώτο και βασικό ήταν ο τοπικός υπολογιστής ο οποίος φέρει το ακόλουθο υλικό:

- CPU: Intel I5 11400H 4.5 GHz
- GPU: Nvidia RTX 3050 Ti 4Gb CUDA Capable
- RAM: 24 Gb

Το δεύτερο πειραματικό περιβάλλον είναι το workspace που παρέχει το Google Colab με συνδρομή και δίνει το ακόλουθο υλικό:

- CPU: Intel XEON
- GPU: Nvidia T4 vRAM 15 Gb CUDA Capable, Nvidia L4 vRAM 22.5 Gb CUDA Capable, Nvidia A100 vRAM 40 Gb CUDA Capable
- RAM: 51Gb, 53Gb, 83.5 Gb

Η αντιστοίχιση είναι ανάμεσα σε GPU και RAM ένα προς ένα.

Από άποψη λογισμικού, χρησιμοποιήθηκαν τα παρακάτω:

- Anaconda Navigator
- Jupyter Notebooks, το οποίο είναι και το βασικό μέσο για την εκτέλεση των πειραμάτων, καθώς προσφέρει ένα οργανωμένο περιβάλλον εργασίας και είναι εύκολο προς τον διαμοιρασμό. Επίσης, είναι συμβατό με το Google Colab.
- Python 3.13
- Microsoft Visual Studio Community Edition
- Pytorch with CUDA 12.6 & torchvision & torchaudio
- Ultralytics API
- Ultralytcis YOLOv11
- Ultralytics RT-DETR-l
- SAHI

Σύνολα δεδομένων που χρησιμοποιήθηκαν:

- VisDrone 2019
  Χρησιμοποιήθηκαν 6471 εικόνες για εκπαίδευση, 548 εικόνες για validation και 1610 εικόνες για testing.
- DOTA v2
  Χρησιμοποιήθηκαν εικόνες για εκπαίδευση, εικόνες για validation και εικόνες για testing.

5.1.1 Σχεδιασμός πειραμάτων

Για τις εκτελέσεις των πειραμάτων ακολουθήθηκε η εξής διαδικασία. Τα πειράματα έγιναν με τους αλγορίθμους YOLOv11 στις εκδόσεις n, s, m και στο μοντέλο RT-DETR στην έκδοση l. Κάθε έκδοση του YOLOv11 εκπαιδεύεται αρχικά με τις default τιμές για να υπάρχει μια βάση σύγκρισης για τις μετέπειτα εκτελέσεις που διαφοροποιούνται ως προς τις υπερπαραμέτρους (fine-tuning) και το στοιχείο της επαύξησης δεδομένων (data augmentation). Αντίστοιχα, στη φάση του inference η αξιολόγηση της ποιότητας των προβλέψεων πραγματοποιείται τόσο με τη χρήση SAHI όσο και χωρίς αυτήν, ώστε να αξιολογηθεί σε ποιες καταστάσεις πλεονεκτεί.

5.2 Μετρικές Απόδοσης

Για την αξιολόγηση των μοντέλων χρησιμοποιήθηκαν γνωστές και αξιόπιστες μετρικές. Σκοπός είναι η αντικειμενική και ακριβής αξιολόγηση των αποτελεσμάτων. Απαιτείται η δυνατότητα αξιολόγησης των υλοποιήσεων και προτάσεων της εργασίας με ένα κοινό τρόπο σε σχέση με την υπόλοιπη κοινότητα.

Οι βασικές μετρικές αξιολόγησης είναι οι παρακάτω:

- mAP@50 και mAP@75-90
- Precision - Recall Curve
- F1-Score
- IoU (Intersection over Union)
- FPS

5.2.1 Η μετρική mAP

Καθώς η μετρική mAP (Mean Average Precision) έχει προαναφερθεί σε προηγούμενο κεφάλαιο, στο παρόν γίνεται μνεία στον περιορισμό της IoU (Intersection over Union). Με τον συγκεκριμένο περιορισμό μια πρόβλεψη για να θεωρηθεί σωστή, πρέπει η μετρική της IoU να έχει φτάσει το κατώφλι του 0.5 για την μετρική mAP@50, ενώ για την μετρική mAP@50-95 είναι απαραίτητο η τιμή της μετρικής IoU να κυμαίνεται (ξεπερνώντας το κατώφλι) από 0.50 έως το 0.95 με βήμα 0.05.

5.2.2 Η καμπύλη Precision - Recall

Η καμπύλη Precision - Recall (PR-Curve) αποτελεί τη σχέση ανάμεσα στο Precision και στο Recall. Είναι ιδιαίτερα χρήσιμη σε περιπτώσεις όπου το σύνολο δεδομένων παρουσιάζει ανισοκατανομή ανάμεσα στις κλάσεις του. Στο αντίστοιχο διάγραμμα, ο άξονας y αντιστοιχεί στο Recall, ενώ ο άξονας x στο Precision. Αυτό σημαίνει ότι ο συγκεκριμένος τύπος καμπύλης επικεντρώνεται στην θετική κλάση. Όσο μεγαλύτερο είναι το εμβαδό που παρουσιάζεται κάτω από την καμπύλη (Area Under the Curve - AUC), τόσο καλύτερη θεωρείται η απόδοση του μοντέλου, καθώς συνδυάζει υψηλές τιμές precision και recall. Το σχήμα της καμπύλης αναπαριστά και την σχέση ανάμεσα στο precision και στο recall με τον ακόλουθο τρόπο:

- Όταν το precision είναι υψηλό και το recall χαμηλό, η καμπύλη εντοπίζεται στο πάνω αριστερό τμήμα του διαγράμματος και το μοντέλο προβλέπει μόνο αντικείμενα για τα οποία έχει υψηλή βεβαιότητα.
- Όταν το recall είναι υψηλό και το precision χαμηλό, η καμπύλη εντοπίζεται στο κάτω δεξιό τμήμα του διαγράμματος και το μοντέλο προβλέπει τα περισσότερα αντικείμενα, περιλαμβάνοντας και πολλές λάθος προβλέψεις.

- Όταν το recall είναι υψηλό και το precision υψηλό, η καμπύλη μένει στις ανώτερες τιμές του άξονα y πλησιάζοντας την πάνω δεξιά γωνία του διαγράμματος, το οποίο αντιστοιχεί σε μέγιστο εμβαδό.

Στην ανίχνευση αντικειμένων συνήθως μετριέται η καμπύλη ανά κλάση και συνολικά στο mAP@50.

5.2.3 Η μετρική F1-Score

Η μετρική του F1-score είναι ένας αρμονικός μέσος ανάμεσα στο precision και στο recall. Η σχέση μεταξύ των δυο μετρικών έχει συμβιβασμούς, και συνήθως ένα μοντέλο εμφανίζει καλύτερη απόδοση σε μία από τις δύο μετρικές εις βάρος της άλλης. Το F1-score δίνει μια ισορροπημένη τιμή, καθώς «τιμωρεί» τις ακραίες τιμές. Εάν ένα από τα precision ή recall είναι "1" και το άλλο είναι "0", τότε ο μέσος όρος είναι 0.5, ενώ με το F1-score είναι "0". Αυτό σημαίνει ότι το F1-score γίνεται πολύ μικρό ή μηδέν, αν μια από τις δύο τιμές (precision ή recall) είναι πολύ μικρή ή μηδέν. Ο μαθηματικός τύπος της μετρικής του F1-score είναι ο παρακάτω:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.1}$$

Αποτελεί σημαντική μετρική για περιπτώσεις που τα σύνολα δεδομένων που χρησιμοποιήθηκαν έχουν ανισοκατανομή ανάμεσα στις κλάσεις του και δίνει μια εικόνα για τα false positives και false negatives των προβλέψεων.

5.2.4 Intersection over Union (IoU)

Για την μετρική IoU έχει γίνει αναφορά σε προηγούμενο κεφάλαιο. Η συγκεκριμένη μετρική είναι σημαντική σε προβλήματα μηχανικής όρασης ειδικά σε εργασίες ανίχνευσης αντικειμένων. Υπολογίζεται διαιρώντας την ένωση των δειγμάτων με την τομή αυτών, δίνοντας το ποσοστό της επικάλυψης των bounding boxes από τις προβλέψεις σε σχέση με τα αληθή (ground truth). Η μετρική IoU παρέχει σημαντική πληροφορία σχετικά με τα true positives και false positives ως προς τα bounding boxes.

5.2.5 Μετρική FPS (Frames Per Second)

Η μετρική FPS αν και δεν είναι αναγκαία για την αξιολόγηση της απόδοσης ενός μοντέλου, χρησιμοποιείται για την αξιολόγηση της εφαρμογής ενός μοντέλου σε πραγματικό χρόνο. Αντιπροσωπεύει τον αριθμό των εικόνων (frames) που μπορεί να επεξεργαστεί ένα μοντέλο ανά δευτερόλεπτο. Μια συνήθης ροή βίντεο έχει 25 fps, επομένως μια εφαρμογή πραγματικού χρόνου πρέπει βρίσκεται όσο πιο κοντά γίνεται σε αυτή. Η συγκεκριμένη μετρική επηρεάζεται, εκτός από την δομή του μοντέλου που χρησιμοποιείται και από το υλικό πάνω στο οποίο εκτελείται το μοντέλο ανίχνευσης αντικειμένων.

5.3 Σύγκριση μεθόδων και βελτιώσεις

Στην παρούσα ενότητα παρουσιάζονται τα αποτελέσματα των πειραματικών δοκιμών που πραγματοποιήθηκαν στο πλαίσιο της εργασίας. Η ανάλυση περιλαμβάνει τόσο τις ποσοτικές μετρήσεις με βάση τυποποιημένες μετρικές αξιολόγησης, όσο και την ποιοτική παρουσίαση και ερμηνεία των αποτελεσμάτων. Τα αποτελέσματα οργανώνονται ανά μοντέλο ανίχνευσης αντικειμένων με στόχο την αποτίμηση της αποτελεσματικότητας τους. Ακολουθούν τα αποτελέσματα των εκπαιδεύσεων, τα οποία παρουσιάζονται με την εξής σειρά: αρχικά τα αποτελέσματα με βάση τον AdamW optimizer,

ακολουθούν τα αποτελέσματα με βάση τον SGD optimizer και, τέλος, τα αποτελέσματα με βάση τον Adam optimizer.

5.3.1 Σύγκριση απόδοσης YOLOv11 και RT-DETR

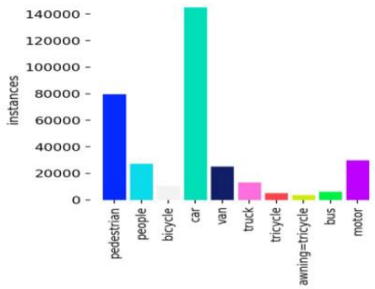
Αρχικά, όλα τα μοντέλα εκπαιδεύτηκαν με τις default παραμέτρους τους για 50 εποχές ώστε να υπάρχει σύγκριση με τις διαφοροποιημένες εκπαιδεύσεις ως προς τις υπερπαραμέτρους. Το μοντέλο YOLOv11 εκπαιδεύτηκε στις εκδόσεις YOLOv11n/s/m. Η δομή των εκδόσεων είναι όμοια μεταξύ τους αλλά διαφοροποιούνται ως προς τον αριθμό των παραμέτρων. Στο παρακάτω πίνακα (πίνακας 5.1) παρατίθενται οι τρεις εκδόσεις του YOLOv11 με τον αριθμό των παραμέτρων τους.

Πίνακας 5.1: Τα μοντέλα YOLOv11 με τον αντίστοιχο αριθμό παραμέτρων

Μοντέλο	Παράμετροι (εκατομμύρια)
YOLOv11n	2.6
YOLOv11s	9.4
YOLOv11m	20.1

5.3.1.1 Το μοντέλο YOLOv11n (AdamW optimizer)

Το μοντέλο YOLOv11n είναι το μικρότερο και αναμένεται να έχει την χαμηλότερη απόδοση από τις υπόλοιπες εκδόσεις του YOLOv11 που συμμετέχουν. Αν και ο αριθμός των παραμέτρων σε σχέση με το σύνολο δεδομένων είναι μικρός, επιλέχθηκε λόγω του χαμηλού υπολογιστικού κόστους και της δυνατότητας χρήσης από συσκευές με περιορισμένους πόρους. Το μοντέλο χωρίς καμία παραμετροποίηση και εκπαίδευση για 50 εποχές, image size = 640 και batch size = 8, δίνει mAP50 = 0.31 και mAP50-95 = 0.18 για όλες τις κλάσεις. Παρατηρείται εύκολα ότι, η κλάση με την καλύτερη ανίχνευση είναι η “car”, ενώ η κλάση με τη χαμηλότερη είναι η “bicycle”. Αυτό έχει σχέση και με την εικόνα 5.1 που δείχνει ότι η κλάση “car” υπερεκπροσωπείται σε σχέση με τη κλάση “bicycle” που εμφανίζεται ελάχιστα. Παρακάτω, και συγκεκριμένα στον πίνακα 5.2, καταγράφονται τα αποτελέσματα της αξιολόγησης. Η μετρική F1 έχει μέση τιμή 0.348 και η mAP75 0.183. Τέλος, παρατίθενται και οι μετρικές στο σύνολο test των εικόνων (πίνακας 5.3).



Εικόνα 5.1: Το διάγραμμα κατανομής των κλάσεων για το σύνολο δεδομένων VisDrone 2019.

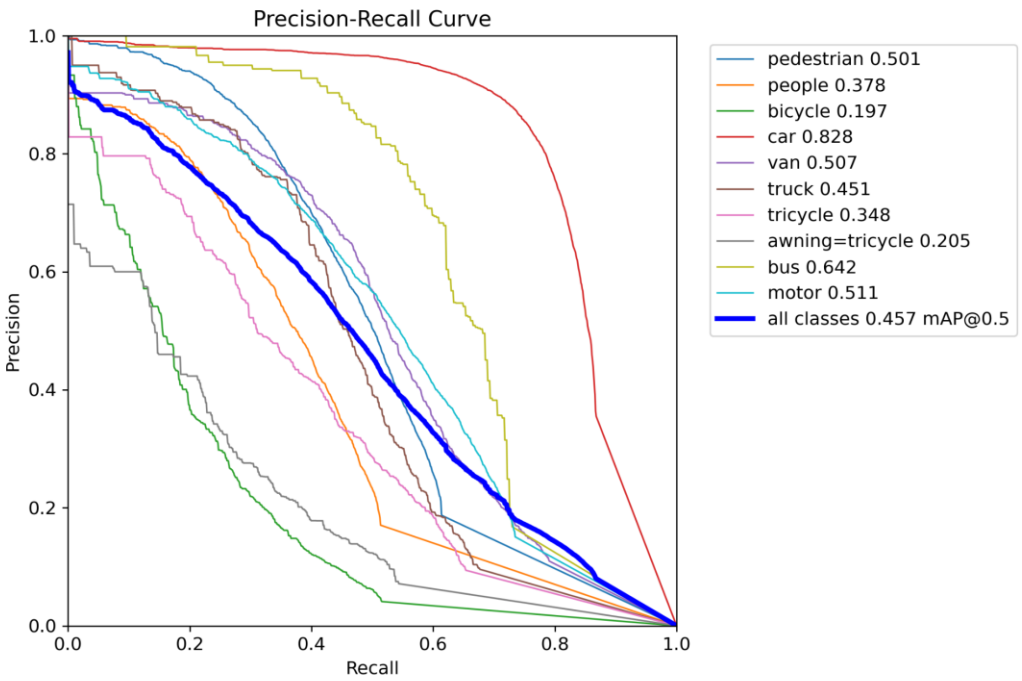
Πίνακας 5.2: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11n (AdamW)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.375	0.356	0.365	0.325	0.139
people	482	5125	0.495	0.221	0.305	0.26	0.094
bicycle	364	1287	0.199	0.096	0.13	0.067	0.025
car	515	14064	0.604	0.75	0.669	0.744	0.51
van	421	1975	0.408	0.387	0.397	0.362	0.252
truck	266	750	0.413	0.277	0.332	0.279	0.185
tricycle	337	1045	0.434	0.196	0.27	0.202	0.111
awning-tricycle	220	532	0.231	0.133	0.169	0.115	0.074
bus	131	251	0.525	0.406	0.458	0.441	0.31
motor	485	4886	0.46	0.344	0.393	0.339	0.137

Πίνακας 5.3: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11n (AdamW)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.3841	0.2192	0.2791	0.2094	0.08
people	797	6376	0.3547	0.0816	0.1326	0.1044	0.0353
bicycle	377	1302	0.1778	0.063	0.093	0.0521	0.0201
car	1530	28074	0.5814	0.6774	0.6258	0.6569	0.4054
van	1168	5771	0.303	0.371	0.3336	0.3063	0.198
truck	750	2659	0.367	0.378	0.3724	0.3186	0.1971
tricycle	245	530	0.2293	0.1585	0.1874	0.1145	0.0596
awning-tricycle	233	599	0.3147	0.1436	0.1972	0.1298	0.0687
bus	838	2940	0.5157	0.5078	0.5118	0.4998	0.3375
motor	794	5845	0.3763	0.2392	0.2925	0.2129	0.0794

Η καμπύλη Precision-Recall (Γράφημα 5.1) αναδεικνύει ότι η κλάση “car” αποδίδει καλύτερα από τις υπόλοιπες, και επιβεβαιώνεται και από τον πίνακα 5.2 καθώς έχει το μεγαλύτερο mAP.



Γράφημα 5.1: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11n

5.3.1.2 Το μοντέλο YOLOv11s (AdamW optimizer)

Το μοντέλο YOLOv11s είναι το δεύτερο που δοκιμάστηκε, είναι ελαφρώς μεγαλύτερο από το YOLOv11n και αναμένεται να αποδώσει λίγο καλύτερα αποτελέσματα. Με εννέα εκατομμύρια παραμέτρους παραμένει μικρό αλλά εμφανίζει σταδιακά καλύτερες δυνατότητες ανίχνευσης. Με εκπαίδευση για 50 εποχές, image size = 640 και batch size = 8, η βασική μετρική mAP50 φτάνει το 0.38, ενώ η mAP50-95 το 0.23. Παρακάτω (πίνακας 5.4) παρουσιάζονται τα αποτελέσματα στο σύνολο του validation και στον πίνακα 5.5 τα αποτελέσματα από το σύνολο δεδομένων test. Η μετρική F1 έχει μέση τιμή 0.419 και είναι βελτιωμένη κατά 20.40% σε σχέση με το μοντέλο YOLOv11n.

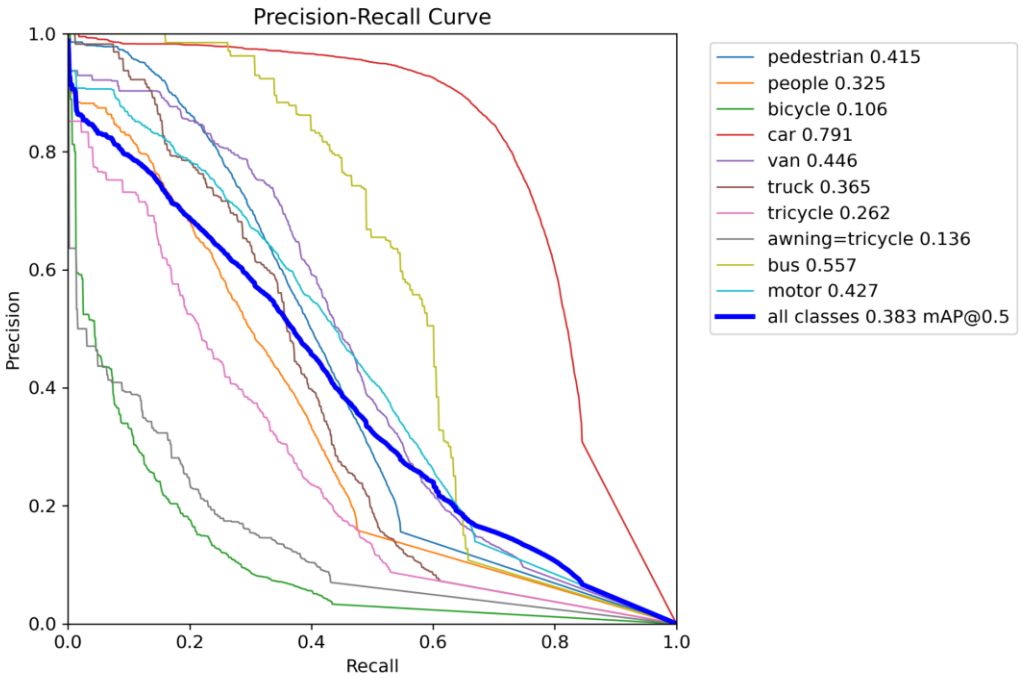
Πίνακας 5.4: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11s (AdamW)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.531	0.385	0.4464	0.415	0.1905
people	482	5125	0.561	0.2646	0.3596	0.3254	0.1239
bicycle	364	1287	0.2104	0.1655	0.1853	0.1062	0.0443
car	515	14064	0.7129	0.7665	0.7387	0.7914	0.5629
van	421	1975	0.483	0.4511	0.4665	0.4457	0.3198
truck	266	750	0.4746	0.3707	0.4163	0.3647	0.2473
tricycle	337	1045	0.4258	0.2565	0.3201	0.2615	0.1541
awning-tricycle	220	532	0.3006	0.1692	0.2165	0.1362	0.0851
bus	131	251	0.6162	0.5458	0.5789	0.5575	0.4027
motor	485	4886	0.5414	0.4102	0.4667	0.4274	0.1855

Πίνακας 5.5: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11s (AdamW)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.4979	0.2482	0.3312	0.2651	0.1088
people	797	6376	0.4543	0.0988	0.1623	0.1363	0.048
bicycle	377	1302	0.2057	0.1152	0.1477	0.0854	0.035
car	1530	28074	0.6751	0.714	0.694	0.7183	0.4603
van	1168	5771	0.4045	0.4349	0.4192	0.3879	0.2601
truck	750	2659	0.4537	0.4396	0.4466	0.4129	0.2685
tricycle	245	530	0.2749	0.266	0.2704	0.1756	0.0931
awning-tricycle	233	599	0.4048	0.187	0.2558	0.1743	0.0959
bus	838	2940	0.5782	0.559	0.5684	0.5645	0.3991
motor	794	5845	0.4242	0.3136	0.3606	0.2731	0.1059

Η καμπύλη Precision-Recall (Γράφημα 5.2) αναδεικνύει ότι η κλάση “car” αποδίδει καλύτερα από τις υπόλοιπες, και επιβεβαιώνεται και από τον πίνακα 5.4 καθώς έχει το μεγαλύτερο mAP, κάτι που παρατηρήθηκε και στο προηγούμενο μοντέλο.



Γράφημα 5.2: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11s

5.3.1.3 Το μοντέλο YOLOv11m (AdamW optimizer)

Το μοντέλο YOLOv11m είναι το μεγαλύτερο που δοκιμάστηκε από την οικογένεια των μοντέλων YOLO με περίπου 20 εκατομμύρια παραμέτρους. Δείχνει μεγαλύτερη σταθερότητα, αλλά αναλογικά παρουσιάζει το ίδιο μοτίβο ανάμεσα απόδοση των κλάσεων. Στη συγκεκριμένη έκδοση του μοντέλου έχουμε (εκπαίδευση για 50 εποχές, image size = 640 και batch size = 8) mAP50 = 0.466 και mAP50-

95 = 0.223 στο σύνολο του validation (πίνακας 5.6), τα οποία είναι αρκετά βελτιωμένα σε σύγκριση με τις άλλες δυο εκδόσεις του μοντέλου που εκτελέστηκαν. Ο πίνακας 5.7 δίνει τα αποτελέσματα με τις εικόνες του συνόλου test.

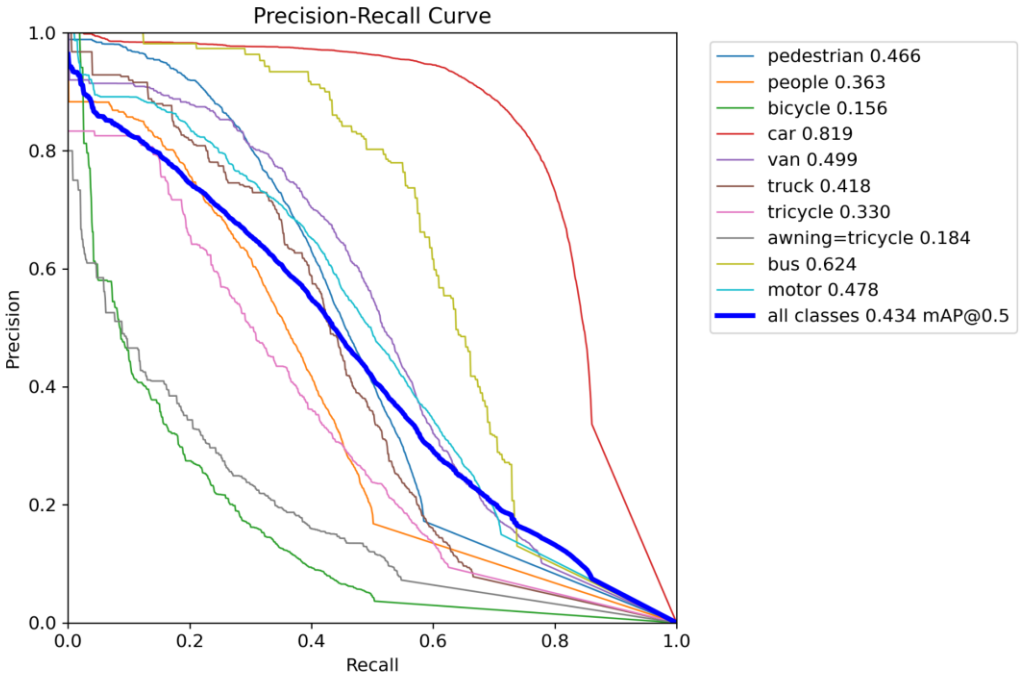
Πίνακας 5.6: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (AdamW)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.606	0.411	0.49	0.466	0.223
people	482	5125	0.628	0.295	0.401	0.363	0.144
bicycle	364	1287	0.268	0.211	0.236	0.156	0.071
car	515	14064	0.783	0.779	0.781	0.819	0.599
van	421	1975	0.539	0.505	0.521	0.499	0.362
truck	266	750	0.531	0.421	0.47	0.418	0.289
tricycle	337	1045	0.487	0.309	0.378	0.33	0.191
awning-tricycle	220	532	0.341	0.201	0.253	0.184	0.115
bus	131	251	0.661	0.583	0.62	0.624	0.45
motor	485	4886	0.598	0.435	0.503	0.478	0.221

Πίνακας 5.7: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11m (AdamW)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.5744	0.2658	0.3634	0.3029	0.1277
people	797	6376	0.484	0.1089	0.1777	0.1489	0.0534
bicycle	377	1302	0.2295	0.1621	0.19	0.1145	0.0464
car	1530	28074	0.7143	0.7412	0.7275	0.7546	0.4981
van	1168	5771	0.4374	0.4781	0.4569	0.4339	0.2999
truck	750	2659	0.4983	0.4889	0.4935	0.4688	0.3137
tricycle	245	530	0.2968	0.3113	0.3039	0.2203	0.1234
awning-tricycle	233	599	0.3985	0.212	0.2768	0.2066	0.1191
bus	838	2940	0.6505	0.5684	0.6067	0.6013	0.4374
motor	794	5845	0.4857	0.3413	0.4009	0.3254	0.1342

Όπως και στις μικρότερες εκδόσεις του μοντέλου YOLOv11m, είναι εμφανές ότι η κλάση με τις περισσότερες εμφανίσεις παρουσιάζει την καλύτερη απόδοση, ενώ η κλάση “bicycle”, αν και δεν έχει τις λιγότερες εμφανίσεις από όλες έχει σταθερά την μικρότερη απόδοση. Η καμπύλη Precision-Recall (Γράφημα 5.3) είναι βελτιωμένη σε σχέση με τις μικρότερες εκδόσεις του μοντέλου.



Γράφημα 5.3: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m (AdamW)

5.3.1.4 Το μοντέλο YOLOv11n (SGD optimizer)

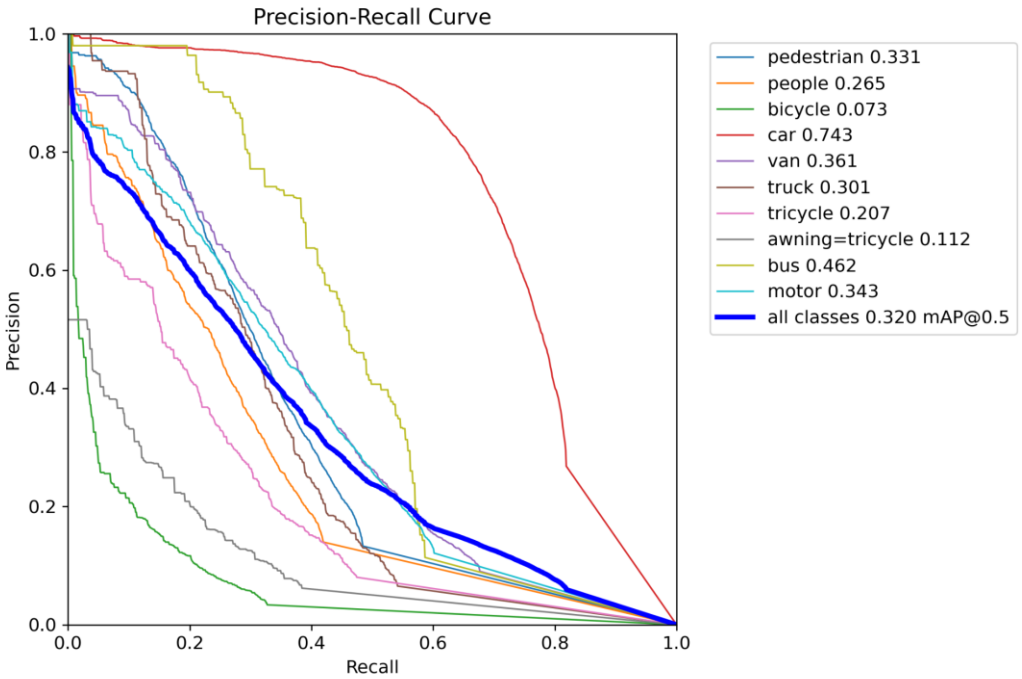
Η εκπαίδευση του μοντέλου έγινε και με την χρήση του SGD optimizer, όπως αναφέρθηκε στην αρχή, και παρακάτω παρέχονται τα αποτελέσματα της αξιολόγησης. Η εκπαίδευση έγινε για περίοδο 50 εποχών με image size = 640 και batch size 8. Ο πίνακας 5.8 εμφανίζει τα αποτελέσματα των μετρικών στο σύνολο του validation set, με τη μετρική F1 να έχει μέση τιμή 0.350, παρόμοια με την απόδοση του με τον optimizer AdamW, ενώ η μέση τιμή mAP75 είναι 0.184. Το γράφημα 5.4 απεικονίζει την καμπύλη Precision-Recall και ο πίνακας 5.9 παρουσιάζει τα αποτελέσματα αξιολόγησης για το σύνολο test.

Πίνακας 5.8: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11n (SGD)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.4259	0.3333	0.374	0.3314	0.1415
people	482	5125	0.4938	0.2338	0.3173	0.265	0.0962
bicycle	364	1287	0.2054	0.1052	0.1392	0.073	0.0284
car	515	14064	0.6086	0.7419	0.6686	0.7431	0.5095
van	421	1975	0.4463	0.3742	0.4071	0.3607	0.2483
truck	266	750	0.4276	0.3187	0.3652	0.3012	0.1972
tricycle	337	1045	0.3667	0.2211	0.2758	0.2069	0.1163
awning-tricycle	220	532	0.2601	0.156	0.195	0.1123	0.0719
bus	131	251	0.5398	0.4462	0.4886	0.4624	0.3101
motor	485	4886	0.4456	0.3596	0.398	0.3432	0.1387

Πίνακας 5.9: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11n (SGD)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.4373	0.2079	0.2818	0.2162	0.0853
people	797	6376	0.4207	0.0781	0.1318	0.1092	0.0374
bicycle	377	1302	0.2181	0.0676	0.1032	0.0621	0.0244
car	1530	28074	0.6006	0.6733	0.6349	0.6594	0.4056
van	1168	5771	0.3398	0.3537	0.3466	0.2927	0.186
truck	750	2659	0.347	0.387	0.3659	0.3229	0.197
tricycle	245	530	0.2246	0.2019	0.2126	0.1319	0.0683
awning-tricycle	233	599	0.3438	0.1586	0.2171	0.1344	0.0713
bus	838	2940	0.5961	0.4959	0.5414	0.5112	0.3425
motor	794	5845	0.3626	0.265	0.3062	0.2217	0.0845



Γράφημα 5.4: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m (SGD)

5.3.1.5 Το μοντέλο YOLOv11s (SGD optimizer)

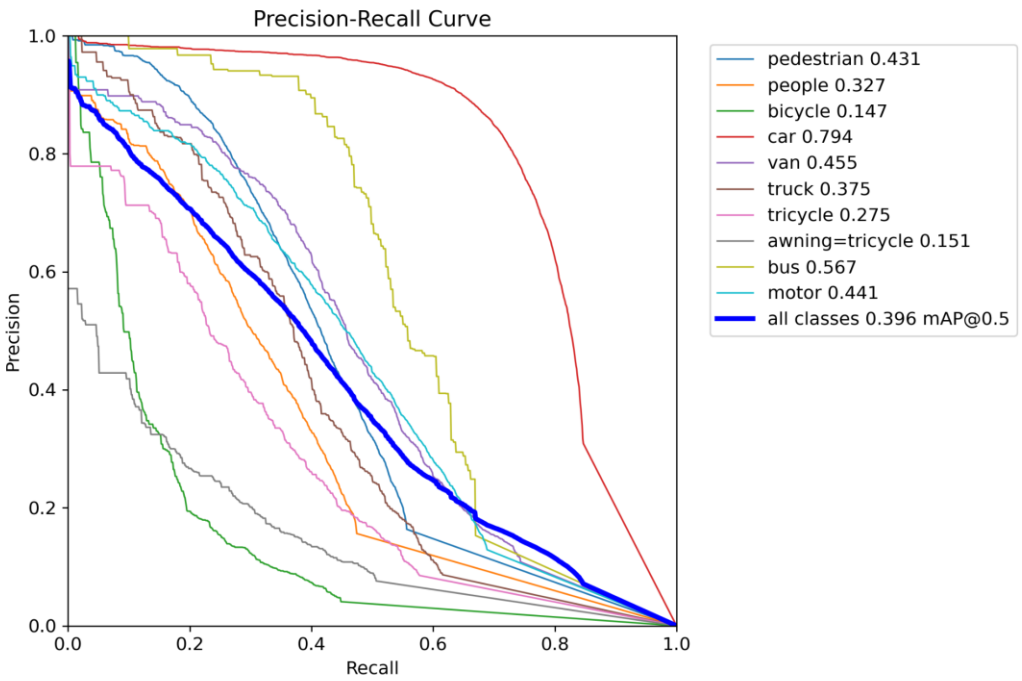
Το μοντέλο εκπαιδεύτηκε για 50 εποχές με image size = 640 και batch size = 8. Ο πίνακας 5.10 παρουσιάζει τα αποτελέσματα των μετρικών αξιολόγησης στο validation set, όπως προέκυψαν από την εκπαίδευση, και το διάγραμμα 5.5 τη καμπύλη Precision-Recall. Η βασική εκπαίδευση έδωσε μέση τιμή για την μετρική F1 = 0.430 και μέση τιμή για την μετρική mAP75 = 0.242. Ο πίνακας 5.11 δίνει τις τιμές των μετρικών από το σύνολο test.

Πίνακας 5.10: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11s (SGD)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.5732	0.3832	0.4593	0.4311	0.1977
people	482	5125	0.5798	0.2658	0.3645	0.3274	0.1259
bicycle	364	1287	0.2952	0.1647	0.2115	0.1473	0.0631
car	515	14064	0.7311	0.7649	0.7476	0.7943	0.5645
van	421	1975	0.5327	0.4451	0.485	0.4549	0.3237
truck	266	750	0.5419	0.3565	0.4301	0.375	0.2519
tricycle	337	1045	0.4191	0.2842	0.3387	0.275	0.1558
awning-tricycle	220	532	0.2996	0.1748	0.2208	0.1515	0.0922
bus	131	251	0.6567	0.51	0.5741	0.5673	0.4163
motor	485	4886	0.5112	0.4482	0.4777	0.4413	0.1954

Πίνακας 5.11: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11s (SGD)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.51522	0.25488	0.34104	0.27575	0.11283
people	797	6376	0.47238	0.11277	0.18207	0.14612	0.05129
bicycle	377	1302	0.2424	0.12366	0.16377	0.09803	0.03782
car	1530	28074	0.67613	0.72305	0.6988	0.72473	0.46161
van	1168	5771	0.3948	0.42991	0.41161	0.36647	0.24291
truck	750	2659	0.40702	0.43513	0.4206	0.39129	0.2499
tricycle	245	530	0.23555	0.30115	0.26434	0.16607	0.09218
awning-tricycle	233	599	0.37314	0.23706	0.28993	0.19281	0.10878
bus	838	2940	0.65157	0.5551	0.59948	0.57879	0.40735
motor	794	5845	0.39609	0.34286	0.36756	0.28501	0.11464



Γράφημα 5.5: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11s (SGD)

5.3.1.6 Το μοντέλο YOLOv11m (SGD optimizer)

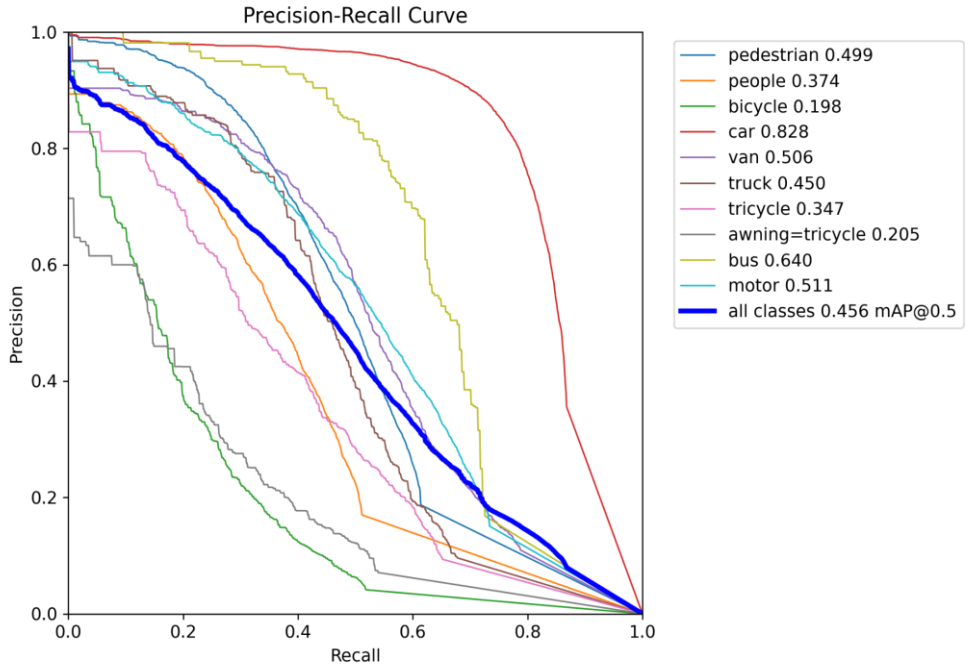
Η εκπαίδευση του μοντέλου YOLOv11m έγινε για 50 εποχές με image size = 640 και batch size = 8. Ο πίνακας 5.12 παρουσιάζει τα αποτελέσματα των μετρικών αξιολόγησης στο validation set όπως προέκυψαν από την εκπαίδευση. Το διάγραμμα 5.6 αποτυπώνει την καμπύλη Precision-Recall και ο πίνακας 5.13 παρουσιάζει τα αποτελέσματα όπως προέκυψαν από την αξιολόγηση με το σύνολο δεδομένων test. Η μέση τιμή της μετρικής F1 είναι 0.494 και η αντίστοιχη της μετρικής mAP75 είναι 0.296.

Πίνακας 5.12: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (SGD)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.6325	0.4324	0.5137	0.4985	0.2457
people	482	5125	0.6258	0.3208	0.4241	0.3843	0.1569
bicycle	364	1287	0.3016	0.2424	0.2688	0.203	0.0957
car	515	14064	0.7837	0.7905	0.7871	0.8269	0.6067
van	421	1975	0.5412	0.5058	0.5229	0.4974	0.3595
truck	266	750	0.5804	0.452	0.5082	0.4645	0.3211
tricycle	337	1045	0.48	0.3818	0.4253	0.3608	0.2069
awning-tricycle	220	532	0.3884	0.2406	0.2971	0.2124	0.1342
bus	131	251	0.7249	0.609	0.6619	0.6532	0.4957
motor	485	4886	0.57	0.4969	0.531	0.5155	0.2469

Πίνακας 5.13: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11m (SGD)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.57827	0.2983	0.39357	0.32907	0.13921
people	797	6376	0.51932	0.12908	0.20676	0.17269	0.06283
bicycle	377	1302	0.26761	0.1874	0.22044	0.13185	0.05594
car	1530	28074	0.73902	0.74621	0.74259	0.7625	0.50073
van	1168	5771	0.49425	0.46026	0.47665	0.43985	0.30033
truck	750	2659	0.51565	0.49688	0.50609	0.48641	0.32533
tricycle	245	530	0.29883	0.34906	0.322	0.24474	0.13388
awning-tricycle	233	599	0.40004	0.25714	0.31305	0.21962	0.13287
bus	838	2940	0.73364	0.55929	0.63471	0.62331	0.45476
motor	794	5845	0.4715	0.39453	0.42959	0.35371	0.1478



Γράφημα 5.6: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m (SGD)

5.3.1.7 Το μοντέλο YOLOv11n (Adam optimizer)

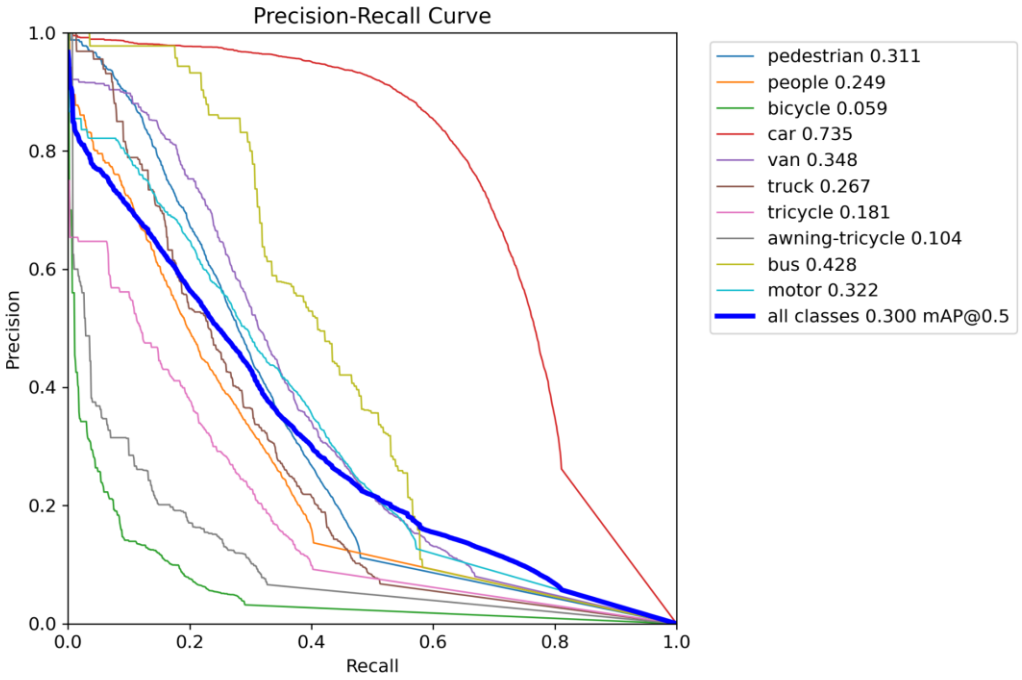
Το μοντέλο YOLOv11n χρησιμοποιήθηκε με τον Adam optimizer για εκπαίδευση σε 50 εποχές με image size = 640 και batch size =8. Στον πίνακα 5.14 παρατίθεται τα αποτελέσματα από το validation set του μοντέλου και στο γράφημα 5.7 η καμπύλη Precision-Recall. Η εκπαίδευση καταγράφει μέση τιμή F1 = 0.335 και μέση τιμή στο mAP75 = 0.171. Ο πίνακας 5.15 εμφανίζει τα αποτελέσματα αξιολόγησης με τη χρήση του test set.

Πίνακας 5.14: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11n (Adam)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.3484	0.3515	0.3499	0.3123	0.132
people	482	5125	0.4682	0.2152	0.2949	0.2511	0.0889
bicycle	364	1287	0.1598	0.0894	0.1146	0.0595	0.0217
car	515	14064	0.586	0.7415	0.6547	0.7351	0.5021
van	421	1975	0.3707	0.3742	0.3725	0.3484	0.2423
truck	266	750	0.3895	0.284	0.3285	0.2661	0.1757
tricycle	337	1045	0.4183	0.17	0.2417	0.1813	0.1002
awning-tricycle	220	532	0.2549	0.1297	0.1719	0.1045	0.0686
bus	131	251	0.4655	0.4303	0.4472	0.4249	0.2813
motor	485	4886	0.4361	0.3391	0.3816	0.3226	0.1296

Πίνακας 5.15: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11n (Adam)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.47122	0.24969	0.32642	0.26391	0.10852
people	797	6376	0.4808	0.09379	0.15696	0.13115	0.04695
bicycle	377	1302	0.20206	0.11905	0.14982	0.08515	0.03137
car	1530	28074	0.66214	0.71465	0.68739	0.71665	0.46857
van	1168	5771	0.4109	0.43545	0.42282	0.40007	0.27302
truck	750	2659	0.36204	0.41707	0.38761	0.34399	0.22684
tricycle	245	530	0.26778	0.23585	0.2508	0.15198	0.08426
awning-tricycle	233	599	0.35842	0.19032	0.24862	0.16147	0.09557
bus	838	2940	0.6018	0.52279	0.55952	0.54095	0.38325
motor	794	5845	0.40863	0.28862	0.3383	0.25386	0.09867



Γράφημα 5.7: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11n (Adam)

5.3.1.8 Το μοντέλο YOLOv11s (Adam optimizer)

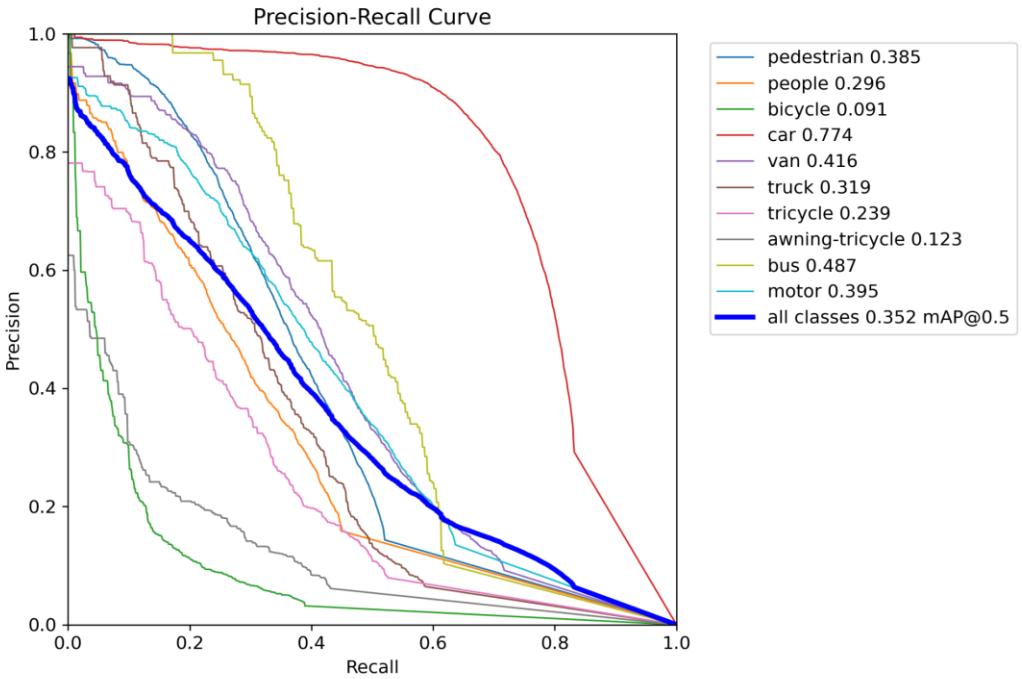
Η εκπαίδευση του μοντέλου YOLOv11s έγινε με τη χρήση του Adam και για 50 εποχές με image size = 640 και batch size = 4 (για την αποδοτικότερη χρονικά και υπολογιστικά εκπαίδευση) και η οποία έδωσε τα παρακάτω αποτελέσματα όπως αυτά δίνονται στο πίνακα 5.16. Επιπροσθέτως, δίνεται και το διάγραμμα της καμπύλης Precision-Recall (Γράφημα 5.8). Σε αυτή την εκπαίδευση η μέση τιμή F1 φτάνει το 0.386 ενώ η μέση τιμή mAP75 το 0.210. Στον πίνακα 5.17 παρουσιάζονται τα αποτελέσματα από την αξιολόγηση με την χρήση του συνόλου test.

Πίνακας 5.16: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11s (Adam)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.4755	0.3707	0.4166	0.3842	0.1723
people	482	5125	0.5481	0.2372	0.3311	0.2994	0.1096
bicycle	364	1287	0.1549	0.1414	0.1478	0.0909	0.0368
car	515	14064	0.6852	0.7566	0.7191	0.7747	0.5458
van	421	1975	0.4549	0.4305	0.4424	0.4163	0.2955
truck	266	750	0.3985	0.3507	0.3731	0.3189	0.2137
tricycle	337	1045	0.4239	0.2412	0.3074	0.2387	0.1307
awning-tricycle	220	532	0.2371	0.1466	0.1812	0.1228	0.0763
bus	131	251	0.6	0.4343	0.5039	0.4851	0.3442
motor	485	4886	0.5092	0.3842	0.4379	0.3959	0.1689

Πίνακας 5.17: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11s (Adam)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.49069	0.22113	0.30487	0.24577	0.10041
people	797	6376	0.44468	0.08548	0.14339	0.12425	0.04365
bicycle	377	1302	0.17307	0.11367	0.13722	0.07178	0.02717
car	1530	28074	0.60425	0.70783	0.65195	0.6946	0.44448
van	1168	5771	0.39076	0.40392	0.39723	0.36935	0.24733
truck	750	2659	0.38816	0.40015	0.39406	0.35226	0.22862
tricycle	245	530	0.24575	0.22075	0.23258	0.1413	0.07397
awning-tricycle	233	599	0.35164	0.17028	0.22945	0.14309	0.07579
bus	838	2940	0.5291	0.52517	0.52713	0.5194	0.35846
motor	794	5845	0.37985	0.28811	0.32768	0.2424	0.0957



Γράφημα 5.8: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11s (Adam)

5.3.1.9 Το μοντέλο YOLOv11m (Adam optimizer)

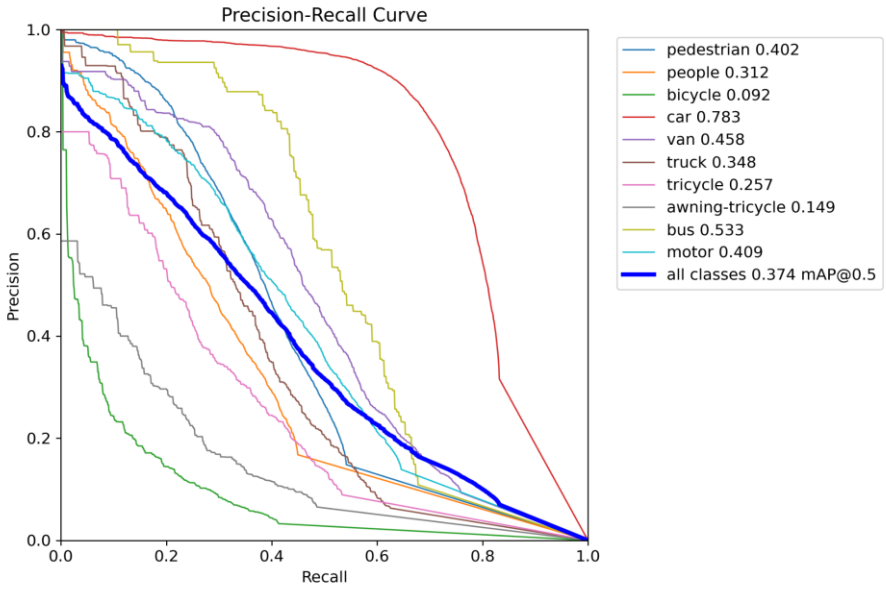
Τέλος, το μοντέλο YOLOv11m εκπαιδεύτηκε και αυτό με τις default παραμέτρους του για 50 εποχές με image size = 640 και batch size = 2 (για αποδοτικότερη χρονικά και υπολογιστικά εκπαίδευση). Στον πίνακα 5.18 παρουσιάζονται τα αποτελέσματα της εκπαίδευσης με τη χρήση του συνόλου validation, και η καμπύλη Precision-Recall (Γράφημα 5.9). Η τιμή F1 έχει μέση τιμή 0.406 και η τιμή mAP75 έχει μέση τιμή 0.235. Στον πίνακα 5.19 δίνονται τα αποτελέσματα με τη χρήση του συνόλου test.

Πίνακας 5.18: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (Adam)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.4985	0.3917	0.4387	0.4046	0.1869
people	482	5125	0.5994	0.2238	0.3259	0.3129	0.1168
bicycle	364	1287	0.1829	0.15	0.1648	0.0916	0.0373
car	515	14064	0.7264	0.7511	0.7385	0.7836	0.5631
van	421	1975	0.4548	0.4911	0.4723	0.4581	0.3273
truck	266	750	0.4164	0.3707	0.3922	0.3474	0.2387
tricycle	337	1045	0.4705	0.2287	0.3078	0.2573	0.152
awning-tricycle	220	532	0.2938	0.2025	0.2397	0.1486	0.0974
bus	131	251	0.6207	0.4781	0.5401	0.5333	0.3862
motor	485	4886	0.5414	0.3762	0.4439	0.4108	0.1754

Πίνακας 5.19: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11m (Adam)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.47122	0.24969	0.32642	0.26391	0.10852
people	797	6376	0.4808	0.09379	0.15696	0.13115	0.04695
bicycle	377	1302	0.20206	0.11905	0.14982	0.08515	0.03137
car	1530	28074	0.66214	0.71465	0.68739	0.71665	0.46857
van	1168	5771	0.4109	0.43545	0.42282	0.40007	0.27302
truck	750	2659	0.36204	0.41707	0.38761	0.34399	0.22684
tricycle	245	530	0.26778	0.23585	0.2508	0.15198	0.08426
awning-tricycle	233	599	0.35842	0.19032	0.24862	0.16147	0.09557
bus	838	2940	0.6018	0.52279	0.55952	0.54095	0.38325
motor	794	5845	0.40863	0.28862	0.3383	0.25386	0.09867



Γράφημα 5.9: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m (Adam)

5.3.1.10 Το μοντέλο RT-DETR (AdamW optimizer)

Το μοντέλο RT-DETR είναι ένα μοντέλο βασισμένο σε transformer και σχεδιασμένο για να λειτουργεί σε εφαρμογές πραγματικού χρόνου. Στην εργασία χρησιμοποιείται ο RT-DETR-l με περίπου 33 εκατομμύρια παραμέτρους. Εφαρμόζεται και εδώ το API της Ultralytics για την φόρτωση του προεκπαιδευμένου μοντέλου.

Πίνακας 5.20: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου RT-DETR-l

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.717	0.627	0.669	0.666	0.348
people	482	5125	0.671	0.553	0.606	0.562	0.256
bicycle	364	1287	0.56	0.355	0.435	0.361	0.182
car	515	14064	0.836	0.853	0.844	0.876	0.656
van	421	1975	0.664	0.542	0.597	0.553	0.429
truck	266	750	0.638	0.508	0.566	0.502	0.355
tricycle	337	1045	0.538	0.467	0.5	0.428	0.27
awning-tricycle	220	532	0.412	0.203	0.272	0.203	0.139
bus	131	251	0.83	0.665	0.739	0.672	0.517
motor	485	4886	0.7	0.658	0.678	0.667	0.349

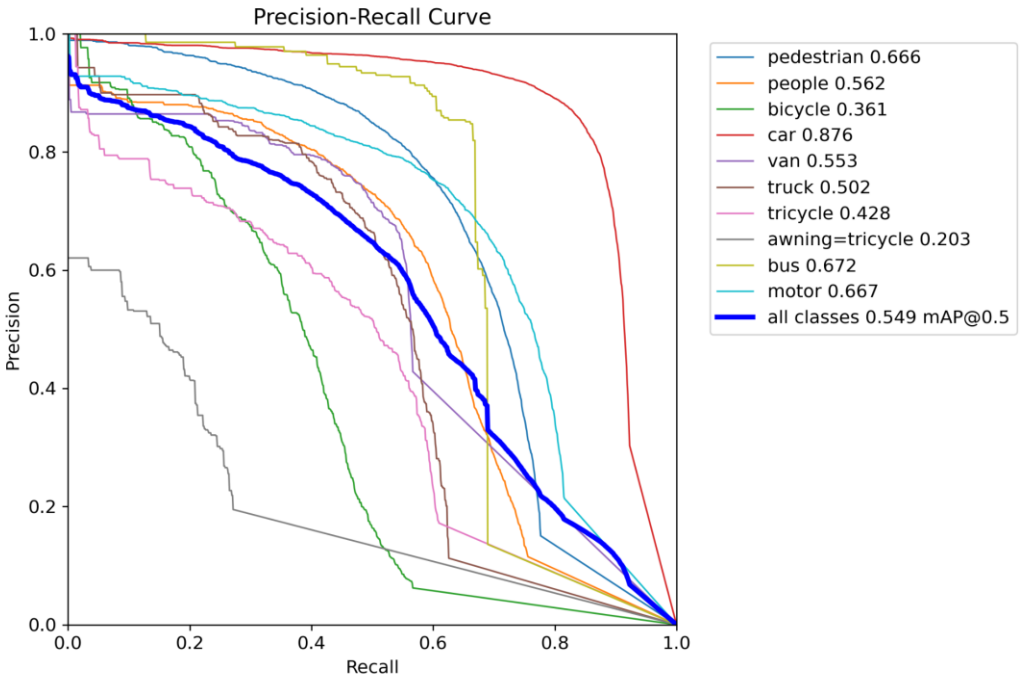
Το μοντέλο εκπαιδεύτηκε σε περιβάλλον Google Colab με την χρήση GPU A100, ανάλυση εικόνων 1024 x 1024 και batch size = 16. Οι αλλαγές αυτές βελτίωσαν τα αποτελέσματα, ωστόσο κατέστησαν την εκπαίδευση ιδιαίτερα δαπανηρή, καθώς χρειάστηκαν περισσότερες από πέντε ώρες συνεχούς εκπαίδευσης και η απαίτηση σε vRAM ήταν κατά διαστήματα περισσότερο από 32 Gb. Σε εκπαίδευση 50 εποχών, η mAP50 φτάνει το 0.54 ενώ η mAP50-95 το 0.34, αποτελώντας μια από τις καλύτερες επιδόσεις μεταξύ των μοντέλων πριν την παραμετροποίηση τους σε επίπεδο υπερπαραμέτρων. Στον

πίνακα 5.20 αναφέρονται αναλυτικά ανά κλάση για το σύνολο validation όπως και το γράφημα Precision-Recall (Γράφημα 5.10) και στον πίνακα 5.21 για το σύνολο test.

Πίνακας 5.21: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου RT-DETR-l

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.53837	0.30772	0.39161	0.31382	0.11774
people	797	6376	0.48987	0.23557	0.31815	0.22914	0.075
bicycle	377	1302	0.3404	0.17281	0.22924	0.13405	0.05444
car	1530	28074	0.76109	0.73965	0.75022	0.74822	0.46484
van	1168	5771	0.56785	0.39473	0.46572	0.37323	0.25499
truck	750	2659	0.55804	0.46785	0.50898	0.43658	0.27727
tricycle	245	530	0.38718	0.30943	0.34397	0.22957	0.12248
awning-tricycle	233	599	0.4499	0.19662	0.27364	0.19268	0.11961
bus	838	2940	0.75399	0.52143	0.61651	0.55287	0.37882
motor	794	5845	0.51541	0.37605	0.43484	0.35044	0.13303

Στο μοντέλο RT-DETR-l, για πρώτη φορά, το μοτίβο αναλογικότητας παρατηρείται στα αποτελέσματα του YOLOv11n/s/, δεν φαίνεται να υφίσταται. Η καμπύλη της κλάσης “car”, και σε αυτή τη περίπτωση, είναι πολύ καλύτερη σε σχέση με τις υπόλοιπες κλάσεις. Ωστόσο η κλάση “bicycle” δεν εμφανίζει την χαμηλότερη βαθμολογία καθώς η κλάση “awning-tricycle” έχει την χαμηλότερη απόδοση. Σε ότι αφορά την γενικότερη εικόνα, όλες οι κλάσεις είναι βελτιωμένες ακόμα και εκείνες με τις λιγότερες εμφανίσεις αντικειμένων στο σύνολο δεδομένων.



Γράφημα 5.10: Η καμπύλη Precision-Recall για το μοντέλο RT-DETR-l

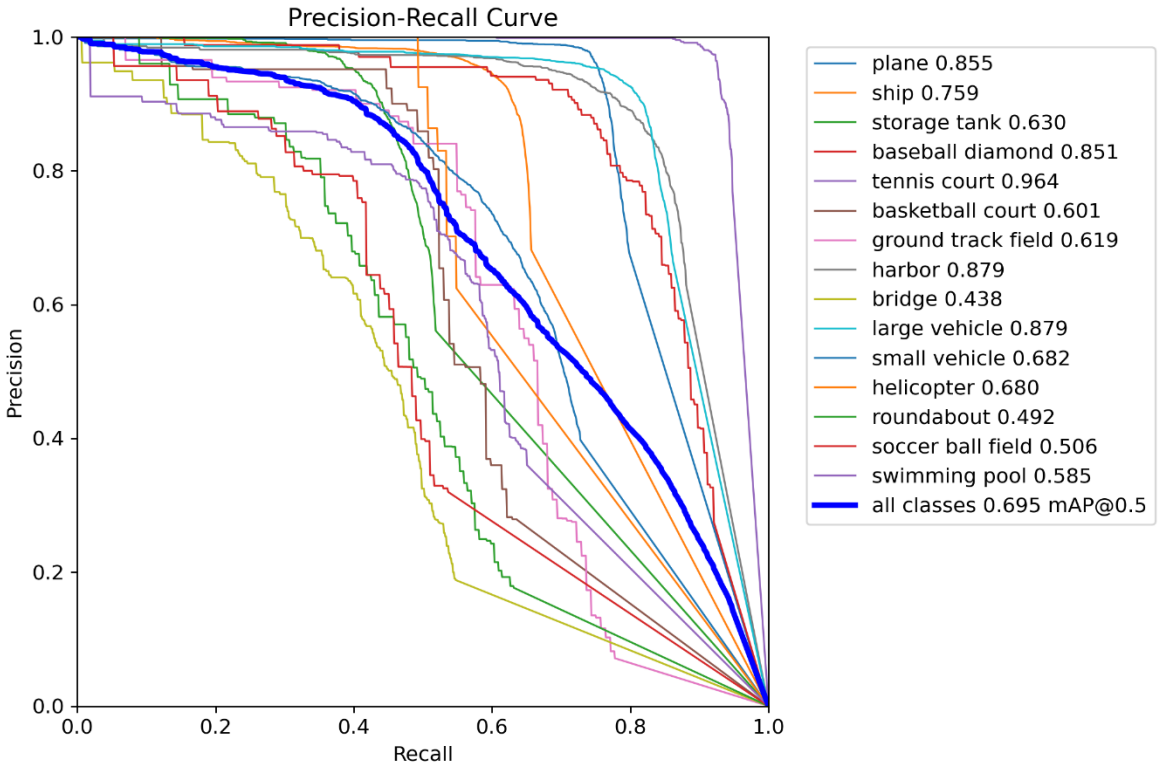
5.3.1.11 Το μοντέλο YOLOv11n-OBB

Το YOLOv11m-OBB είναι ένα μοντέλο που εστιάζει σε σύνολα δεδομένων με προσανατολισμένα πλαίσια οριοθέτησης (oriented bounding boxes). Χρησιμοποιήθηκε με το σύνολο δεδομένων DOTA το οποίο είναι ένα σύνολο δεδομένων με προσανατολισμένα πλαίσια. Η αρχική του εκτέλεση ήταν με τις default παραμέτρους για να υπάρχει μια βάση σύγκρισης με τις τροποποιήσεις που γίνονται στις υπερ-παραμέτρους σε επόμενη εκτέλεση.

Το μοντέλο εκτελέστηκε για περίοδο εκπαίδευσης 50 εποχών, batch size = 16, image size = 1024. Τα αποτελέσματα ήταν εξαρχής ικανοποιητικά, με mAP75 = 0.437 και F1-Score = 0.531, ακόμα και χωρίς τη χρήση υπερπαραμέτρων. Παρακάτω, παρατίθεται ο πίνακας 5.22 της αξιολόγησης με σύνολο του validation και η καμπύλη Precision-Recall (Γράφημα 5.11).

Πίνακας 5.22: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11n-OBB (με default τιμές)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
plane	70	2531	0.817	0.682	0.743	0.752	0.582
ship	108	8960	0.778	0.515	0.62	0.635	0.413
storage tank	55	2888	0.784	0.256	0.386	0.396	0.261
baseball diamond	53	214	0.737	0.421	0.536	0.518	0.382
tennis court	94	760	0.909	0.896	0.902	0.932	0.879
basketball court	35	132	0.509	0.394	0.444	0.424	0.377
ground track field	70	144	0.594	0.417	0.49	0.429	0.359
harbor	114	2090	0.73	0.724	0.727	0.754	0.422
bridge	75	464	0.514	0.129	0.207	0.178	0.076
large vehicle	110	4387	0.803	0.808	0.805	0.849	0.656
small vehicle	136	5438	0.61	0.554	0.581	0.594	0.41
helicopter	14	73	0.943	0.225	0.364	0.493	0.354
roundabout	61	179	0.619	0.127	0.211	0.192	0.15
soccer ball field	63	153	0.618	0.381	0.471	0.409	0.336
swimming pool	41	440	0.62	0.389	0.478	0.42	0.22



Γράφημα 5.11: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11n-OBB (με default τιμές)

5.4 Πλεονεκτήματα και μειονεκτήματα της κάθε μεθόδου

Η εφαρμογή κάθε μεθόδου εκπαίδευσης φέρει πλεονεκτήματα και μειονεκτήματα. Τον σημαντικότερο ρόλο στο αποτέλεσμα το έχει το ίδιο το μοντέλο με την αρχιτεκτονική του η οποία καθορίζει σε μεγάλο βαθμό την ακρίβεια και την ταχύτητα της ανίχνευσης του. Η αποτελεσματικότητα καθορίζεται και από τις υπερπαραμέτρους που εφαρμόζονται κατά την διαδικασία της εκπαίδευσης. Η επιλογή του optimizer μπορεί να βελτιώσει ή να περιορίσει την ικανότητα γενίκευσης του μοντέλου. Παρακάτω (πίνακας 5.22) συνοψίζονται τα βασικά πλεονεκτήματα και μειονεκτήματα του κάθε optimizer με τα μοντέλα που χρησιμοποιούνται στην εργασία.

Από την συγκριτική ανάλυση και την παρουσίαση των αποτελεσμάτων προκύπτει ότι καμία μέθοδος δεν υπερτερεί ταυτόχρονα σε όλα τα κριτήρια. Είναι αναμενόμενο ότι το μοντέλο YOLOv11m υπερτερεί των δυο μικρότερων εκδόσεων σε όλες σχεδόν τις μετρικές, αλλά απαιτεί περισσότερους πόρους, γεγονός που περιορίζει την εφαρμογή του σε συστήματα με περιορισμένους πόρους ή περιορισμένη πρόσβαση σε υψηλή υπολογιστική ισχύ. Τα μοντέλα YOLOv11n και YOLOv11s πλεονεκτούν σε ταχύτητα και σε απαιτήσεις πόρων, καθιστώντας τα καταλληλότερα τόσο σε συσκευές περιορισμένων υπολογιστικών πόρων, όπως embedded συσκευές, όσο και για συστήματα που διαθέτουν σύγχρονο υλικό σε επίπεδο επεξεργαστή, μνήμης και GPU. Από τους optimizers ξεχωρίζουν οι AdamW και SGD για την καλύτερη απόδοση και σταθερότητα στην εκπαίδευση τους. Τέλος, το μοντέλο RT-DETR-l παρουσιάζει ιδιαίτερα υψηλή ακρίβεια, αλλά έχει και τις μεγαλύτερες ανάγκες σε υπολογιστική ισχύ. Τα μοντέλα YOLOv11n/m-OBB παρουσιάζουν πολύ καλή ακρίβεια, με το δεύτερο να είναι υπολογιστικά πιο απαιτητικό.

Πίνακας 5.23: Συγκριτική παρουσίαση των μοντέλων με τους optimizers μετά τα πειράματα.

Μοντέλο	Optimizer	Πλεονεκτήματα	Μειονεκτήματα
YOLOv11n	AdamW	Ταχύτητα	Σχετικά χαμηλή απόδοση σε σχέση με τους άλλους δυο optimizers
	SGD	Πολύ καλή απόδοση, καλύτερη από τους άλλους δυο optimizers για την συγκεκριμένη έκδοση	Αυξημένη χρήση της μνήμης
	Adam	Ταχύτητα	Χαμηλότερη απόδοση, δυσκολία σύγκλισης του μοντέλου
YOLOv11s	AdamW	Καλή απόδοση	Περισσότερες απαιτήσεις σε πόρους
	SGD	Καλή απόδοση, καλύτερη από τους άλλους δυο optimizers για την συγκεκριμένη έκδοση	Αυξημένη χρήση της μνήμης
	Adam	Ταχύτητα	Πολύ χαμηλή απόδοση
YOLOv11m	AdamW	Χαμηλή σχετικά απόδοση	Χαμηλή ταχύτητα
	SGD	Πολύ υψηλή σε όλες τις μετρικές	Αυξημένη χρήση της μνήμης
	Adam	Ταχύτητα	Πολύ χαμηλή απόδοση, αργή σύγκλιση, δε βοηθάει το μοντέλο όσο οι άλλοι δυο optimizers
RT-DETR-l	AdamW	Πολύ καλή απόδοση	Αργή εκπαίδευση, ανάγκη για υψηλή υπολογιστική ισχύ

5.5 Παραμετροποιημένα μοντέλα

Παρακάτω παρουσιάζονται ορισμένα παραμετροποιημένα μοντέλα, με το YOLOv11m/-OBB (optimizer SGD και AdamW) και το RT-DETR-l να επιλέγονται λόγω της σταθερότητας και καλής ανάπτυξης που σημειώνουν κατά την εκπαίδευση.

5.5.1 YOLOv11m - SGD v1

Έγινε τροποποίηση του YOLOv11m - SGD ως προς τις υπερπαραμέτρους, και πιο συγκεκριμένα ορίστηκαν τα παρακάτω:

- Multi_scale = True, για να λάβει υπόψη του το μοντέλο τη διάσταση του multi-scale και να «μάθει» καλύτερα
- Scale = 0.37, ο λόγος τροποποίησης της κλίμακας
- Freeze = 10, "πάγωμα" των δέκα πρώτων επιπέδων, προκειμένου να διατηρηθεί καλύτερα η γνώση του προεκπαιδευμένου μοντέλου και να εκπαιδευτεί στα νέα δεδομένα.
- Cos_lr = True, (cosine annealing) αποτελεί μια τεχνική learning rate scheduler που προσαρμόζει το βήμα του learning rate.

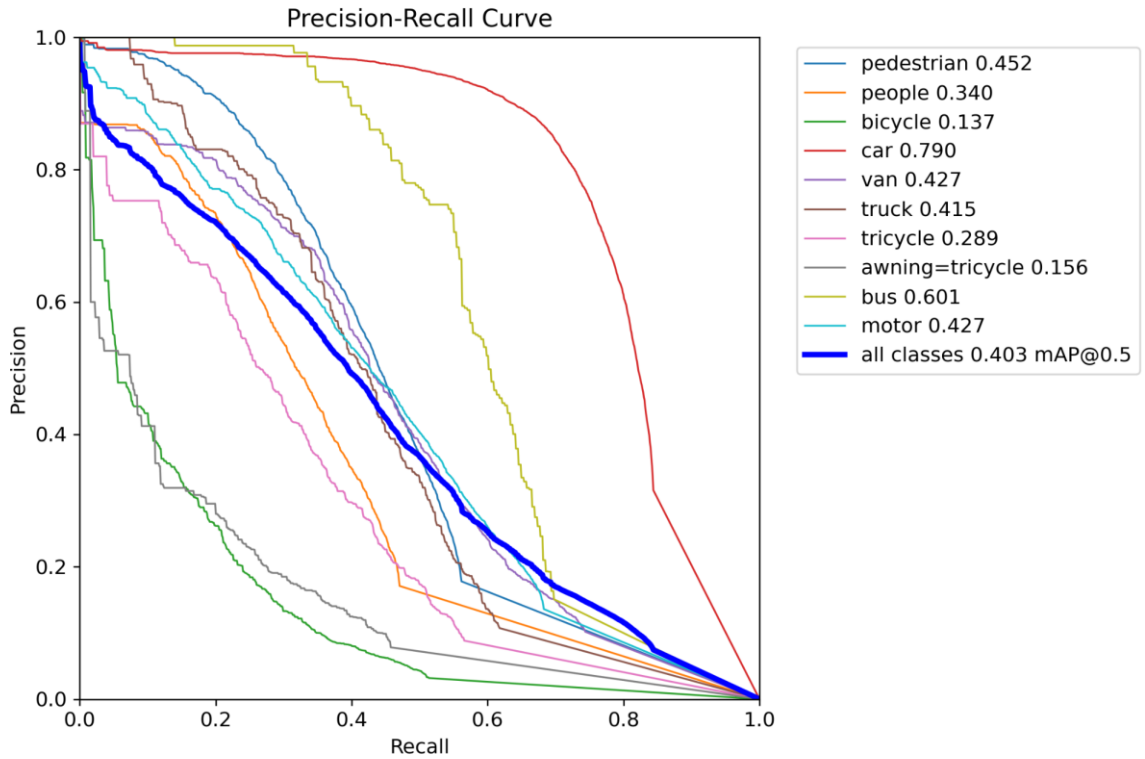
Στις υπόλοιπες παραμέτρους παραμένει το ίδιο. Στον πίνακα 5.24 και στο γράφημα 5.12 παρουσιάζονται τα αποτελέσματα και η καμπύλη Precision-Recall με βάση το σύνολο validation ενώ στον πίνακα 5.25 με βάση το σύνολο test. Η συγκεκριμένη εκπαίδευση έχει μέση τιμή F1 = 0.435 και μέση τιμή mAP75 = 0.245.

Πίνακας 5.24: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.60144	0.39552	0.47722	0.45183	0.21245
people	482	5125	0.61611	0.26205	0.3677	0.3395	0.1328
bicycle	364	1287	0.17949	0.25796	0.21169	0.13675	0.06189
car	515	14064	0.73152	0.75917	0.74509	0.79008	0.56223
van	421	1975	0.46879	0.44658	0.45742	0.42662	0.30459
truck	266	750	0.57312	0.37467	0.45312	0.41531	0.2803
tricycle	337	1045	0.4383	0.30335	0.35855	0.28895	0.16331
awning-tricycle	220	532	0.30986	0.15789	0.20919	0.1558	0.10146
bus	131	251	0.69666	0.55378	0.61706	0.60052	0.42821
motor	485	4886	0.53043	0.40196	0.45735	0.42735	0.19406

Πίνακας 5.25: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.54766	0.26483	0.35702	0.28726	0.12186
people	797	6376	0.49953	0.1032	0.17106	0.14543	0.0528
bicycle	377	1302	0.1905	0.20661	0.19823	0.12414	0.05468
car	1530	28074	0.68973	0.71322	0.70128	0.72261	0.46401
van	1168	5771	0.39432	0.41813	0.40587	0.35063	0.2334
truck	750	2659	0.45115	0.40542	0.42706	0.38566	0.25302
tricycle	245	530	0.2794	0.33396	0.30425	0.20147	0.10501
awning-tricycle	233	599	0.35755	0.19032	0.24841	0.18055	0.10803
bus	838	2940	0.66119	0.51939	0.58177	0.5638	0.40728
motor	794	5845	0.4116	0.3284	0.36533	0.28514	0.11339



Γράφημα 5.12: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m – SGD v1 (με υπερπαραμέτρους)

5.5.2 YOLOv11m - SGD v2

Η δεύτερη τροποποίηση του YOLOv11m - SGD αφορά τις υπερπαραμέτρους, και συγκεκριμένα ορίστηκαν τα παρακάτω:

- Multi_scale = True, για να λάβει υπόψη του το μοντέλο τη διάσταση του multi-scale και να «μάθει» καλύτερα.
- Batch size = 16
- Mosaic = True
- Scale = 0.37, ο λόγος τροποποίησης της κλίμακας
- Mossaic = True
- Cos_lr = True, (cosine annealing) αποτελεί μια τεχνική learning rate scheduler που προσαρμόζει το βήμα του learning rate.

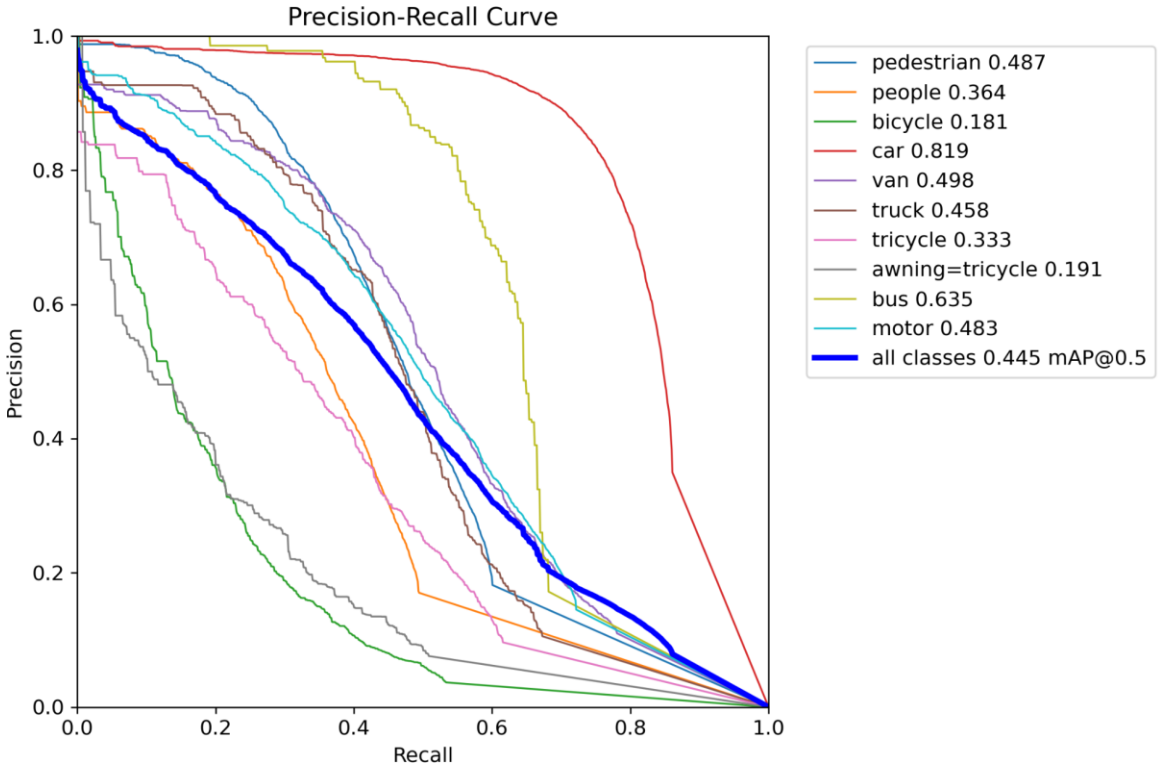
Στις υπόλοιπες παραμέτρους παραμένει το ίδιο. Στον πίνακα 5.26 και στο γράφημα 5.13 παρουσιάζονται τα αποτελέσματα και η καμπύλη Precision-Recall με βάση το σύνολο validation, ενώ στον πίνακα 5.27 με βάση το σύνολο test. Η εκπαίδευση στη συγκεκριμένη περίπτωση, έχει μέση τιμή F1 = 0.476 και μέση τιμή mAP75 = 0.289.

Πίνακας 5.26: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.60826	0.42888	0.50306	0.48712	0.23728
people	482	5125	0.64474	0.29215	0.4021	0.36449	0.1478
bicycle	364	1287	0.25534	0.24631	0.25074	0.18101	0.08446
car	515	14064	0.77348	0.78029	0.77687	0.81863	0.59906
van	421	1975	0.55916	0.48709	0.52064	0.49812	0.36148
truck	266	750	0.61362	0.42667	0.50335	0.45803	0.31487
tricycle	337	1045	0.48415	0.3378	0.39795	0.3328	0.18966
awning-tricycle	220	532	0.39118	0.19445	0.25977	0.19127	0.1236
bus	131	251	0.7522	0.56839	0.6475	0.63511	0.47318
motor	485	4886	0.5741	0.45025	0.50469	0.48287	0.2278

Πίνακας 5.27: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.54393	0.29901	0.38589	0.3206	0.13328
people	797	6376	0.57972	0.1192	0.19774	0.17329	0.06169
bicycle	377	1302	0.23894	0.20584	0.22116	0.12744	0.05502
car	1530	28074	0.73116	0.74318	0.73712	0.76078	0.49872
van	1168	5771	0.46656	0.45902	0.46276	0.42677	0.29135
truck	750	2659	0.55863	0.47537	0.51364	0.48685	0.3253
tricycle	245	530	0.33745	0.34691	0.34211	0.24913	0.13798
awning-tricycle	233	599	0.4031	0.21421	0.27975	0.20626	0.12463
bus	838	2940	0.69942	0.54286	0.61127	0.58891	0.42997
motor	794	5845	0.46756	0.36177	0.40792	0.33582	0.14009



Γράφημα 5.13: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m - SGD v2 (με υπερπαραμέτρους)

5.5.3 YOLOv11m - AdamW v1

Η τρίτη τροποποίηση του YOLOv11m περιλαμβάνει τον optimizer AdamW ως προς τις υπερπαραμέτρους. Πιο συγκεκριμένα ορίστηκαν τα παρακάτω:

- Multi_scale = True, για να λάβει υπόψη του το μοντέλο τη διάσταση του multi-scale και να μάθει καλύτερα
- Batch size = 16
- Scale = 0.37, ο λόγος τροποποίησης της κλίμακας
- Multi scale = True
- Cos_lr = True, (cosine annealing) αποτελεί μια τεχνική learning rate scheduler που προσαρμόζει το βήμα του learning rate.

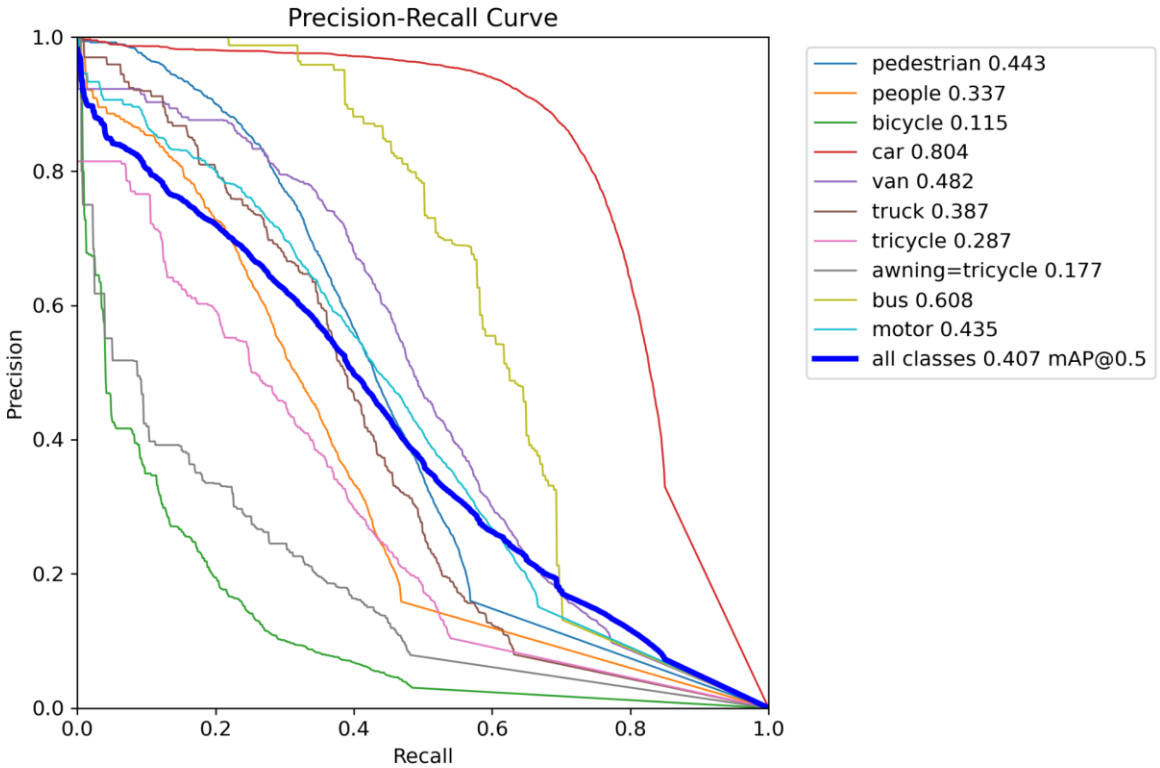
Στις υπόλοιπες παραμέτρους παραμένει το ίδιο. Στον πίνακα 5.28 και στο γράφημα 5.14 παρουσιάζονται τα αποτελέσματα και η καμπύλη Precision-Recall με βάση το σύνολο validation, ενώ στον πίνακα 5.29 με βάση το σύνολο test. Η εκπαίδευση έχει μέση τιμή F1 = 0.435 και μέση τιμή mAP75 = 0.259.

Πίνακας 5.28: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.53521	0.41384	0.46676	0.44274	0.20982
people	482	5125	0.59446	0.27395	0.37506	0.33669	0.13203
bicycle	364	1287	0.17581	0.21601	0.19385	0.11481	0.05254
car	515	14064	0.74554	0.76785	0.75653	0.80379	0.58243
van	421	1975	0.47414	0.49772	0.48565	0.48249	0.34584
truck	266	750	0.45978	0.40133	0.42857	0.38685	0.26488
tricycle	337	1045	0.50549	0.25072	0.33519	0.28747	0.1643
awning-tricycle	220	532	0.34681	0.18045	0.23739	0.17712	0.11027
bus	131	251	0.68896	0.54582	0.60909	0.60779	0.43947
motor	485	4886	0.56886	0.39049	0.46309	0.43487	0.19591

Πίνακας 5.29: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.51656	0.25269	0.33937	0.27564	0.11312
people	797	6376	0.4443	0.10445	0.16914	0.13395	0.04756
bicycle	377	1302	0.16588	0.17358	0.16964	0.08745	0.03579
car	1530	28074	0.69238	0.7227	0.70722	0.73465	0.48231
van	1168	5771	0.42304	0.47028	0.44541	0.42435	0.29082
truck	750	2659	0.45341	0.46576	0.4595	0.43707	0.29452
tricycle	245	530	0.33207	0.2566	0.2895	0.19856	0.10346
awning-tricycle	233	599	0.42116	0.18865	0.26058	0.19237	0.11066
bus	838	2940	0.62875	0.54495	0.58386	0.56953	0.41131
motor	794	5845	0.46174	0.31001	0.37096	0.29015	0.11668



Γράφημα 5.14: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m - AdamW (με υπερπαραμέτρους)

5.5.4 YOLOv11m - SGD v3

Η τρίτη τροποποίηση του YOLOv11m περιλαμβάνει τον optimizer SGD ως προς τις υπερπαραμέτρους. Πιο συγκεκριμένα ορίστηκαν τα παρακάτω:

- Multi_scale = True, για να λάβει υπόψη του το μοντέλο τη διάσταση του multi-scale και να μάθει καλύτερα
- Batch size = 16
- Mosaic = True
- Scale = 0.37, ο λόγος τροποποίησης της κλίμακας
- Multi scale = True
- Translate = 0.20
- Shear = 5
- Cos_lr = True, (cosine annealing) αποτελεί μια τεχνική learning rate scheduler που προσαρμόζει το βήμα του learning rate.

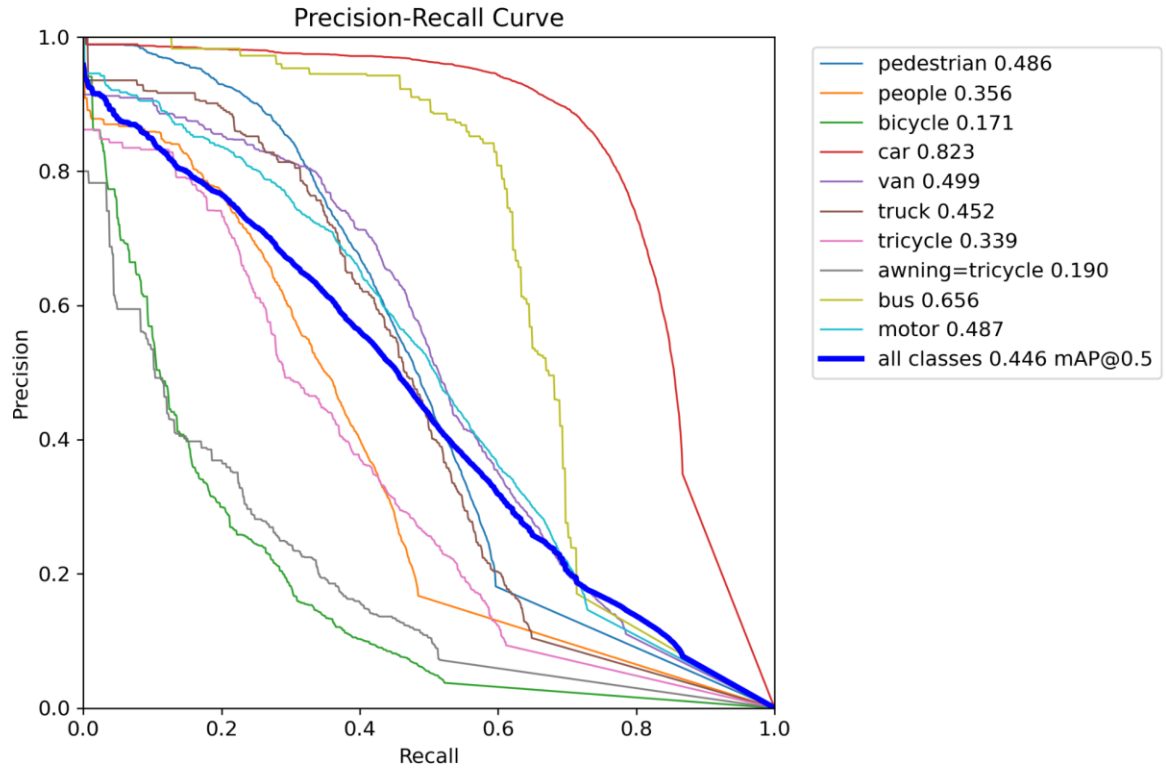
Στις υπόλοιπες παραμέτρους παραμένει το ίδιο. Στον πίνακα 5.30 και στο γράφημα 5.15 παρουσιάζονται τα αποτελέσματα και η καμπύλη Precision-Recall με βάση το σύνολο validation, ενώ στον πίνακα 5.31 με βάση το σύνολο test. Η εκπαίδευση έχει μέση τιμή F1 = 0.435 και μέση τιμή mAP75 = 0.259.

Πίνακας 5.30: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.59521	0.43634	0.50354	0.48595	0.2224
people	482	5125	0.613	0.29034	0.39405	0.35637	0.13913
bicycle	364	1287	0.2456	0.24631	0.24596	0.17128	0.07812
car	515	14064	0.77569	0.78434	0.77999	0.82322	0.58515
van	421	1975	0.54563	0.4957	0.51947	0.49928	0.3478
truck	266	750	0.59566	0.424	0.49538	0.45227	0.30144
tricycle	337	1045	0.47516	0.31866	0.38148	0.33865	0.18681
awning-tricycle	220	532	0.34888	0.21992	0.26979	0.18959	0.11969
bus	131	251	0.80182	0.60558	0.69002	0.65586	0.46351
motor	485	4886	0.56015	0.46521	0.50828	0.48672	0.22217

Πίνακας 5.31: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου YOLOv11m (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.56096	0.29269	0.38467	0.31843	0.12875
people	797	6376	0.55885	0.11722	0.1938	0.16901	0.05861
bicycle	377	1302	0.23409	0.20507	0.21862	0.12926	0.05455
car	1530	28074	0.72852	0.73969	0.73406	0.75775	0.48012
van	1168	5771	0.47972	0.45951	0.4694	0.42892	0.28504
truck	750	2659	0.54697	0.47236	0.50693	0.47618	0.29831
tricycle	245	530	0.32677	0.34528	0.33577	0.23187	0.12533
awning-tricycle	233	599	0.36607	0.21536	0.27118	0.20513	0.11565
bus	838	2940	0.70485	0.5499	0.61781	0.59756	0.41042
motor	794	5845	0.47327	0.35398	0.40502	0.3306	0.1344



Γράφημα 5.15: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m – SGD v3 (με υπερπαραμέτρους)

5.5.5 RT-DETR-l v1

Έγινε τροποποίηση του RT-DETR - AdamW ως προς τις υπερπαραμέτρους και πιο συγκεκριμένα ορίστηκαν τα παρακάτω:

- Multi_scale = True, για να λάβει υπόψη του το μοντέλο τη διάσταση του multi-scale και να μάθει καλύτερα
- Batch size = 16
- Learning rate = 0.0015
- Cos_lr – True, τέθηκε το cosine annealing = True ώστε να βρεθεί το κατάλληλο βήμα, με βάση την καμπύλη, για την μείωση του learning rate.
- hsv_h = 0.017
- Degrees = 3, τέθηκε μια τυχαία περιστροφή τριών μοιρών.
- Translate = 0.3, με το translate αποκρύπτουμε μέρος της εικόνας μετακινώντας την κατά 30% κατά τον άξονα x και κατά 30% για τον άξονα y.
- Scale = 0.37
  Ο λόγος τροποποίησης της κλίμακας

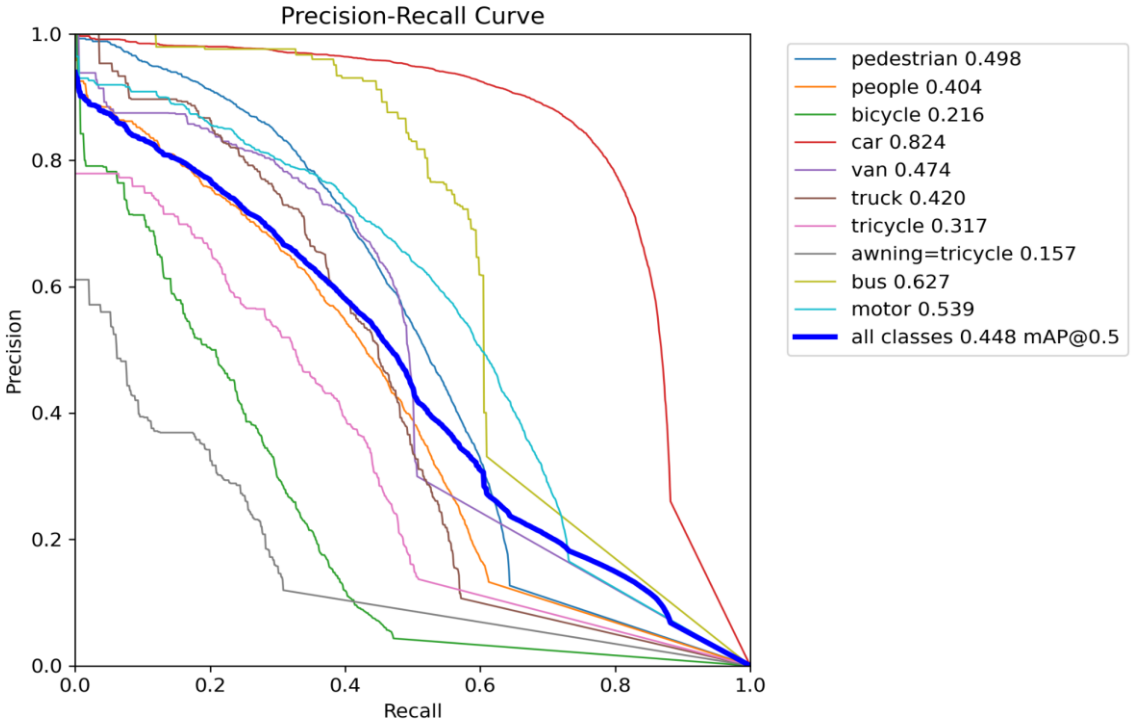
Στις υπόλοιπες παραμέτρους παραμένει το ίδιο. Στον πίνακα 5.32 και στο γράφημα 5.16 παρουσιάζονται τα αποτελέσματα και η καμπύλη Precision-Recall με βάση το σύνολο validation, ενώ στον πίνακα 5.33 με βάση το σύνολο test. Σε αυτή τη περίπτωση, η μέση τιμή F1 είναι 0.478 και μέση τιμή mAP75 είναι 0.264

Πίνακας 5.32: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου RT-DETR-l (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	520	8844	0.57925	0.48226	0.52632	0.4991	0.22504
people	482	5125	0.52209	0.4227	0.46717	0.4059	0.1695
bicycle	364	1287	0.37886	0.2634	0.31075	0.21575	0.09483
car	515	14064	0.7666	0.80734	0.78644	0.82463	0.56913
van	421	1975	0.59912	0.47646	0.53079	0.47514	0.3384
truck	266	750	0.55125	0.42258	0.47842	0.42001	0.26919
tricycle	337	1045	0.45936	0.35311	0.39929	0.3166	0.17564
awning-tricycle	220	532	0.30601	0.20865	0.24812	0.1553	0.09308
bus	131	251	0.74635	0.55777	0.63843	0.62818	0.44474
motor	485	4886	0.5881	0.54605	0.5663	0.53906	0.24669

Πίνακας 5.33: Τα αποτελέσματα από την αξιολόγηση (test set) του μοντέλου RT-DETR-l (με υπερπαραμέτρους)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
pedestrian	1197	21006	0.48647	0.33276	0.3952	0.32585	0.1271
people	797	6376	0.48778	0.22135	0.30451	0.23179	0.08681
bicycle	377	1302	0.34377	0.18356	0.23933	0.13016	0.0548
car	1530	28074	0.72083	0.75689	0.73842	0.75471	0.47092
van	1168	5771	0.51477	0.41639	0.46038	0.36795	0.25309
truck	750	2659	0.50437	0.46935	0.48623	0.41925	0.25525
tricycle	245	530	0.31575	0.35472	0.3341	0.23605	0.12449
awning-tricycle	233	599	0.43244	0.19589	0.26964	0.19033	0.10963
bus	838	2940	0.72644	0.5017	0.59351	0.54841	0.37959
motor	794	5845	0.49892	0.39367	0.44009	0.356	0.1436



Γράφημα 5.16: Η καμπύλη Precision-Recall για το μοντέλο RT-DETR-l (με υπερπαραμέτρους)

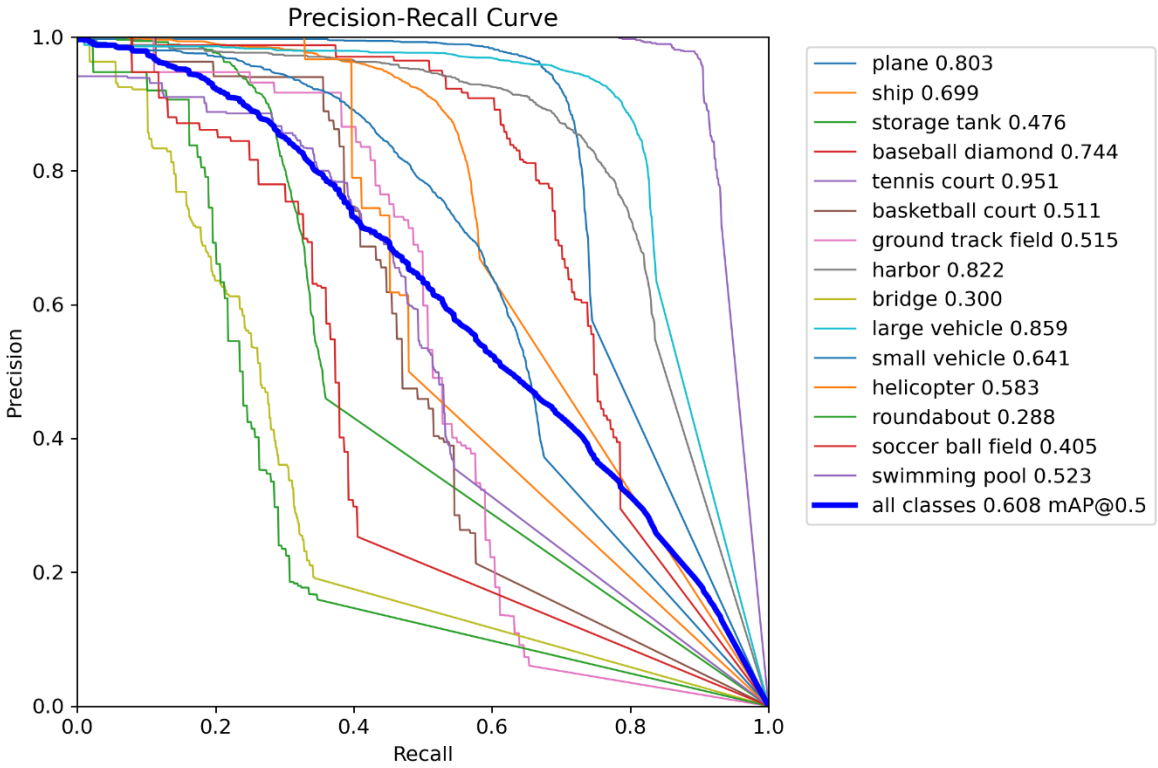
5.5.6 YOLOv11m-OBB με χρήση υπερπαραμέτρων

Η εκτέλεση του μοντέλου YOLOv11m-OBB με την χρήση ορισμένων παραμέτρων έδωσε καλύτερα αποτελέσματα. Πιο συγκεκριμένα με mAP75 = 0.520 και F1-Score = 0.596 παρουσίασε σημαντική

βελτίωση, παρ ‘όλους τους περιορισμούς που υπάρχουν στη χρήση υπερπαραμέτρων. Η χρήση των παραμέτρων multi-scale = True και scale = 0.37 έδωσαν άμεσα OutOfMemory σφάλμα στο περιβάλλον Google Colab με την χρήση της GPU A100. Εν συνεχεία, έγινε πιο ελαφριά προσέγγιση στην ενίσχυση του μοντέλου χρησιμοποιώντας τις παρακάτω υπερπαραμέτρους: hsv_v = 0.15, mosaic = True, cos_lr = True, seed = 42, ενώ οι υπόλοιπες παράμετροι διατηρήθηκαν όπως στην προηγούμενη εκτέλεση. Ακολουθεί ο πίνακας αποτελεσμάτων με τις μετρικές από το validation set και η καμπύλη Precision-Recall.

Πίνακας 5.34: Τα αποτελέσματα από την αξιολόγηση (validation set) του μοντέλου YOLOv11m-OBB (με την χρήση υπερπαραμέτρων)

Class	Images	Instances	Box-P	Box-R	Box-F1	mAP50	mAP50-95
plane	70	2531	0.921	0.7	0.795	0.803	0.66
ship	108	8960	0.883	0.535	0.666	0.699	0.501
storage tank	55	2888	0.928	0.254	0.399	0.476	0.338
baseball diamond	53	214	0.858	0.617	0.718	0.744	0.575
tennis court	94	760	0.946	0.905	0.925	0.951	0.917
basketball court	35	132	0.642	0.447	0.527	0.511	0.442
ground track field	70	144	0.564	0.507	0.534	0.515	0.42
harbor	114	2090	0.857	0.719	0.782	0.822	0.519
bridge	75	464	0.678	0.186	0.292	0.3	0.141
large vehicle	110	4387	0.885	0.798	0.839	0.859	0.694
small vehicle	136	5438	0.694	0.574	0.628	0.641	0.471
helicopter	14	73	0.914	0.397	0.554	0.583	0.437
roundabout	61	179	0.633	0.212	0.317	0.288	0.21
soccer ball field	63	153	0.658	0.34	0.448	0.405	0.372
swimming pool	41	440	0.763	0.391	0.517	0.523	0.295



Γράφημα 5.17: Η καμπύλη Precision-Recall για το μοντέλο YOLOv11m-OBB (με την χρήση υπερπαραμέτρων)

5.6 Ανάλυση False Positives και False Negatives

Η αξιολόγηση ενός μοντέλου δεν περιορίζεται μόνο στην εξέταση των σωστών ταξινομήσεων των δειγμάτων. Εξίσου σημαντική είναι και η αξιολόγηση των False Positives και των False Negatives (FN). Τα FP αναφέρονται στις περιπτώσεις όπου το μοντέλο κατατάσσει ένα αντικείμενο σε μια κατηγορία που στην πραγματικότητα δεν υπάρχει, ενώ τα FN αντιστοιχούν σε περιπτώσεις όπου το αντικείμενο στην πραγματικότητα υπάρχει αλλά δεν ταξινομείται.

Η ανάλυση των σφαλμάτων FN και FP είναι σημαντική για την κατανόηση των αδυναμιών του μοντέλου, καθώς αποκαλύπτει μοτίβα λανθασμένων προβλέψεων και προβλήματα που σχετίζονται με όμοιες ή ιδιαίτερα απαιτητικές κατηγορίες.

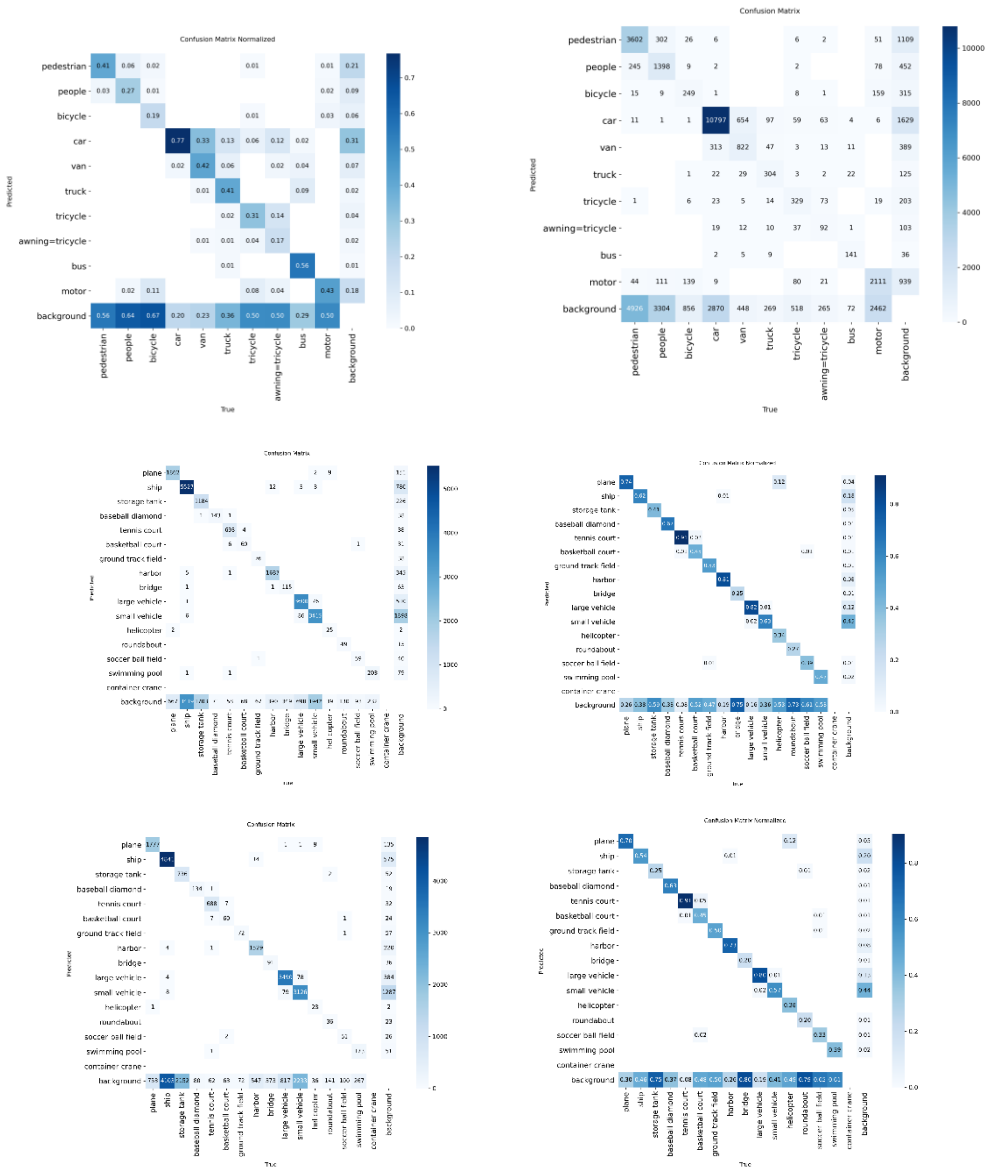
Σημαντικά εργαλεία αξιολόγησης, αυτού του είδους προβλημάτων, είναι η μετρική F1-Score και ο πίνακας σύγχυσης (confusion matrix). Για την μετρική F1 έχει γίνει αναφορά στην αρχή αυτού του κεφαλαίου. Ο πίνακας σύγχυσης παρουσιάζει τις πραγματικές τιμές στον οριζόντιο άξονα και τις προβλέψεις του μοντέλου στον κάθετο. Κάθε κελί του πίνακα περιέχει τον αριθμό των σωστών προβλέψεων, ο οποίος μπορεί να εκφραστεί είτε ως απόλυτη τιμή είτε ως ποσοστό κανονικοποιημένο στο διάστημα (0,1). Η διαγώνιος (αριστερά προς δεξιά) αποτυπώνει τον αριθμό των σωστών προβλέψεων ανά κλάση, ενώ τα στοιχεία εκτός αυτής αντιστοιχούν σε λανθασμένες κατηγοριοποιήσεις. Παρακάτω δίνονται ο πίνακας 5.35 με την μετρική F1 για τα μοντέλα με την αντίστοιχη μεταβολή τους και παραδείγματα πινάκων (εικόνα 5.2) confusion matrix.

Πίνακας 5.35: Οι τιμές F1 για κάθε μοντέλο και η αντίστοιχη μεταβολή τους ως προς την μικρότερη έκδοση.

Μοντέλο	F1-Score	Μεταβολή
YOLOv11n - AdamW	0.348	-
YOLOv11s - AdamW	0.419	+20.46%
YOLOv11m - AdamW	0.465	+10.98%
YOLOv11n - SGD	0.373	-
YOLOv11s - SGD	0.459	+23.06%
YOLOv11m - SGD	0.494	+7.63%
YOLOv11n - Adam	0.335	-
YOLOv11s - Adam	0.386	+15.22%
YOLOv11m - Adam	0.406	+5.18%
RT-DETR-l	0.596	-
RT-DETR-l v1	0.495	
YOLOv11n-OBB	0.596	-
YOLOv11m-OBB		-

Παρατηρείται στην εικόνα 5.2 ότι η μεγαλύτερη απώλεια είναι ως προς το φόντο. Σε αυτή, τη περίπτωση το μοντέλο αποτυγχάνει να ανιχνεύει τα αντικείμενα, τα αφήνει να θεωρηθούν μη σχετικά με τις κλάσεις που έχει εκπαιδευτεί, και τα κατατάσσει στο φόντο. Οι πρώτες τρεις κλάσεις (YOLOv11m – SGD) παρουσιάζουν απώλειες στην ανίχνευση αντικειμένων μεταξύ τους, γεγονός που οφείλεται τόσο στο μικρό τους μέγεθος όσο και στην περιορισμένη εκπροσώπηση τους στο σύνολο των δεδομένων, η οποία δεν επαρκεί για να εκπαιδευτεί ο αλγόριθμος σε περισσότερα παραδείγματα. Αντίθετα, κατηγορίες αντικειμένων με μεγαλύτερο μέγεθος εμφανίζουν καλύτερα ποσοστά ορθής κατηγοριοποίησης. Στα μοντέλα YOLOv11n/m-OBB παρατηρείται ότι οι απώλειες αφορούν κυρίως αντικείμενα που κατατάσσονται στο φόντο.

Για την βελτίωση των αποτελεσμάτων μπορούν να εφαρμοστούν κάποιες στρατηγικές που να βοηθήσουν το μοντέλο να μαθαίνει καλύτερα κάποιες κατηγορίες αντικειμένων με δυσκολότερη διάκριση χαρακτηριστικών. Οι τεχνικές επαύξησης μπορούν να ενισχύσουν το μοντέλο, αν και σε ορισμένες περιπτώσεις προσθέτουν σημαντικό υπολογιστικό κόστος. Η τεχνική SAHI, που αναφέρθηκε σε προηγούμενο κεφάλαιο, μπορεί να αυξήσει την ικανότητα του μοντέλου να διακρίνει μικρότερα αντικείμενα χωρίς να αλλάζει κάτι στη δομή της εκπαίδευσης, επεμβαίνοντας μόνο στην διαδικασία του inference, όπου «τεμαχίζεται» η εικόνα σε μικρότερα τμήματα.



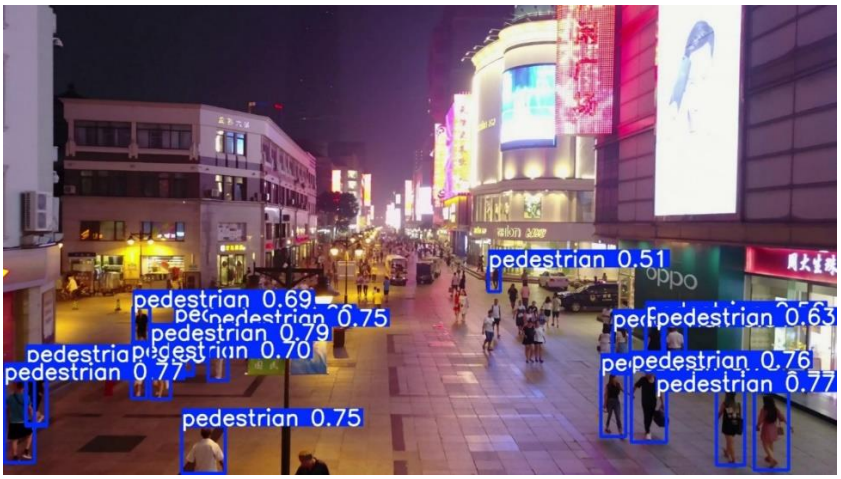
Εικόνα 5.2: : Ο πίνακας σύγχυσης επάνω αριστερά για το μοντέλο YOLOv11m - SGD, με τις απόλυτες τιμές ανά κλάση και δεξιά με τις τιμές κανονικοποιημένες. Αντίστοιχα στη μέση για το μοντέλο YOLOv11n-OBB και κάτω για το μοντέλο YOLOv11m-OBB.

5.7 Οπτική ανάλυση αποτελεσμάτων: Σύγκριση με και χωρίς χρήση SAHI

Παρακάτω παρουσιάζονται μερικά ενδεικτικά παραδείγματα προβλέψεων από το σύνολο δεδομένων VisDrone και DOTA, με τα δεύτερα να είναι πιο απαιτητικά λόγω του είδους των εικόνων και των συνθηκών που αποτυπώνουν. Αξίζει να σημειωθεί ότι τα μοντέλα πέτυχαν μικρότερη βαθμολογία στο σύνολο δεδομένων VisDrone λόγω αυξημένης δυσκολίας, γι' αυτό και γίνεται χρήση παραδειγμάτων από αυτό. Στην πρώτη ενότητα παρουσιάζονται παραδείγματα προβλέψεων χωρίς την χρήση της τεχνικής SAHI ενώ στη δεύτερη παρουσιάζονται παραδείγματα προβλέψεων με την χρήση της.

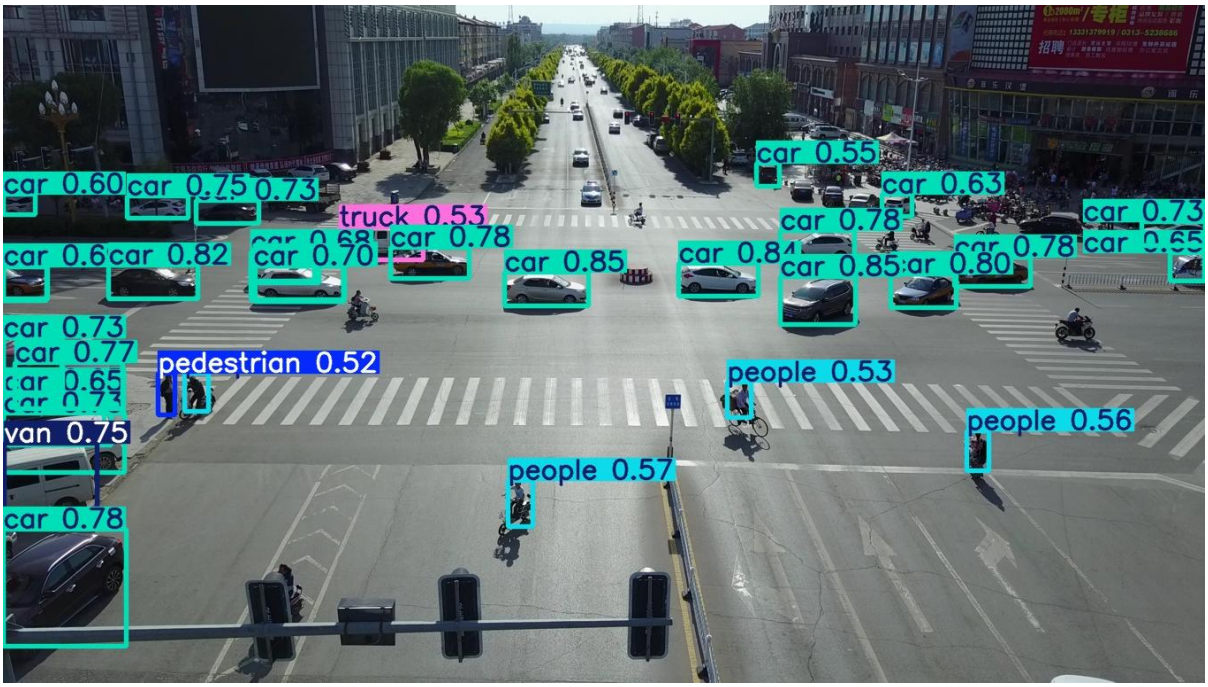
5.7.1 Αποτελέσματα προβλέψεων χωρίς την χρήση της τεχνικής SAHI

Εικόνα 5.3: Παράδειγμα πρόβλεψης χωρίς τη χρήση SAHI (α)

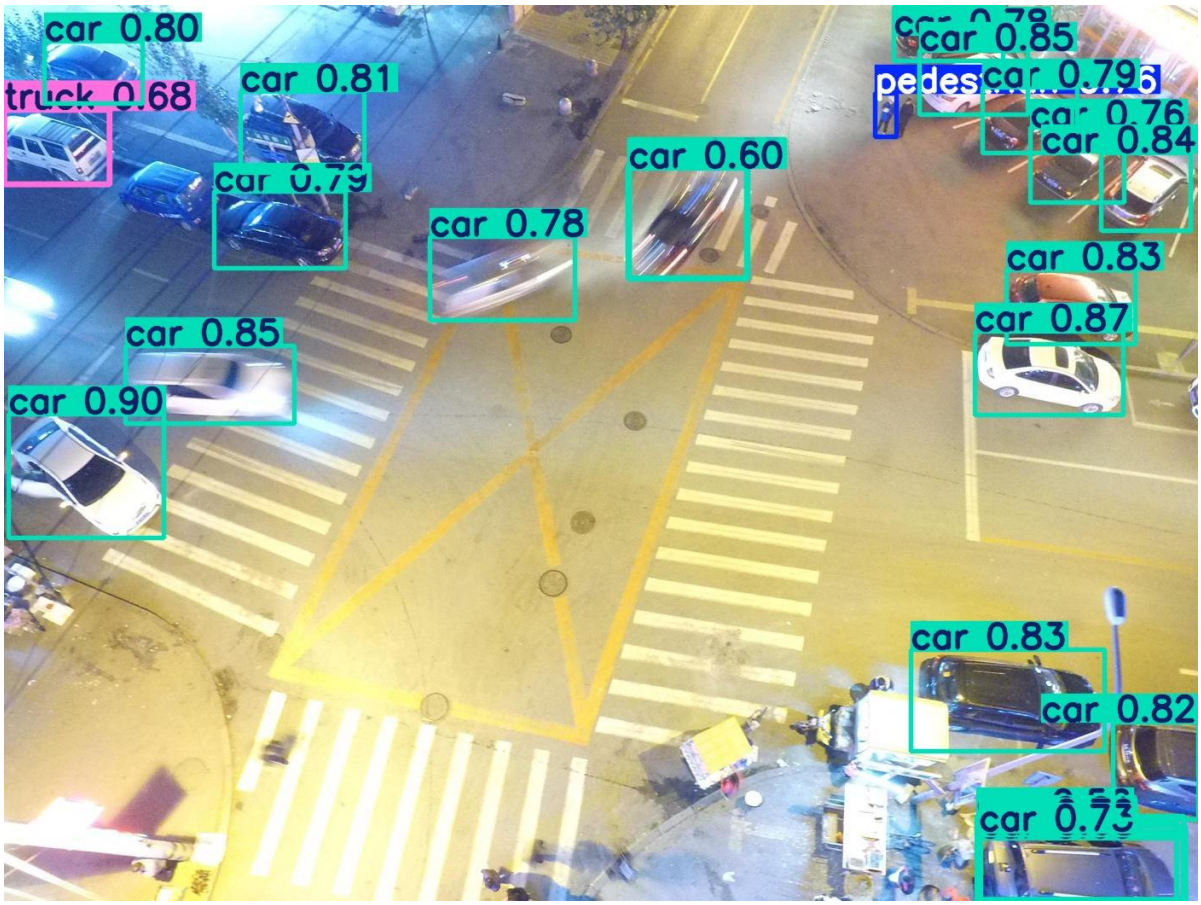


Εικόνα 5.4: Παράδειγμα πρόβλεψης χωρίς τη χρήση SAHI (β)

Εικόνα 5.5: Παράδειγμα πρόβλεψης χωρίς τη χρήση SAHI (γ)

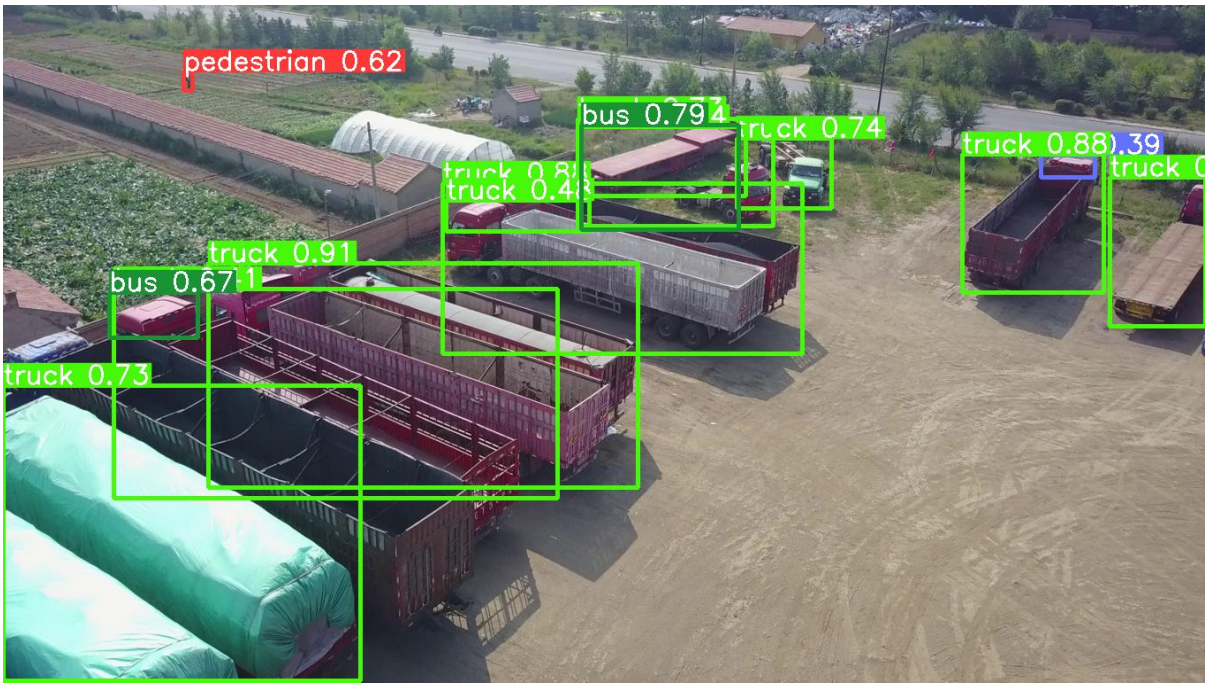


Εικόνα 5.6: Παράδειγμα πρόβλεψης χωρίς τη χρήση SAHI (δ)



Εικόνα 5.7: Παράδειγμα πρόβλεψης χωρίς τη χρήση SAHI (ε)

5.7.2 Αποτελέσματα προβλέψεων με την χρήση της τεχνικής SAHI



Εικόνα 5.8: Παράδειγμα πρόβλεψης με τη χρήση SAHI (α)

Εικόνα 5.9: Παράδειγμα πρόβλεψης με τη χρήση SAHI (β)

Εικόνα 5.10: Παράδειγμα πρόβλεψης με τη χρήση SAHI (γ)

Εικόνα 5.11: Παράδειγμα πρόβλεψης με τη χρήση SAHI (δ)

Εικόνα 5.12: Παράδειγμα πρόβλεψης με τη χρήση SAHI (ε)

5.8 Συζήτηση – Αξιολόγηση Αποτελεσμάτων

Η συγκριτική αξιολόγηση ανέδειξε σημαντικές διαφοροποιήσεις και δυσκολίες στις εφαρμογές της κάθε μεθόδου. Το YOLOv11 έδειξε ικανοποιητική απόδοση στην ανίχνευση μικρών αντικειμένων, ακόμη και όταν αυτά βρίσκονται κοντά μεταξύ τους, ενώ διατήρησε υψηλή ταχύτητα τόσο κατά την εκπαίδευση όσο και κατά την διαδικασία του inference, επιδεικνύοντας ισορροπία ανάμεσα στα δύο αυτά ζητούμενα. Το μοντέλο RT-DETR-l που βασίζεται σε transformers έδειξε επίσης υψηλές μετρικές αλλά με το μειονέκτημα των αυξημένων απαιτήσεων σε υπολογιστικούς πόρους. Σημειώνεται ότι η εκπαίδευση του πραγματοποιήθηκε σε περιβάλλον Google Colab με την χρήση της GPU A100, μια από τις ισχυρότερες κάρτες γραφικών κατά την συγγραφή της εργασίας, με κατανάλωση μνήμης που έφτασε τα 39/40 Gb. Αντίστοιχα, ο χρόνος εκτέλεσης για τις 50 εποχές ανήλθε περίπου στις πέντε ώρες. Σημαντικό είναι να αναφερθεί ότι επιλεγμένα σύνολα δεδομένων είναι από τα πιο απαιτητικά, ειδικά αυτό του VisDrone. Οι εκτελέσεις έγιναν αρχικά σε τοπικό υπολογιστή, με στόχο τη διερεύνηση της δυνατότητας εκπαίδευσης μοντέλων ανίχνευσης μικρών αντικειμένων σε τοπικό περιβάλλον. Οι χρόνοι εκτέλεσης σε GPU Nvidia RTX 3050 Ti 4Gb είναι ικανοποιητικοί για τα μοντέλα YOLOv11 στις εκδόσεις n και s με μέγιστο batch size 8. Αντίθετα, στις εκδόσεις m η εκπαίδευση απαιτούσε μείωση του batch size σε 4 ή 2 για να ολοκληρωθεί σε λογικό χρόνο, ειδικά όταν δοκιμάστηκαν τεχνικές επαύξησης, ενώ για το μοντέλο RT-DETR το batch size περιοριζόταν στο 2. Στη περίπτωση που το batch size υπερέβαινε τα όρια, που αναφέρονται παραπάνω, ο χρόνος εκπαίδευσης ανά εποχή αυξανόταν σημαντικά: για τα μοντέλα YOLOv11s/m ξεπερνούσε τα 40 λεπτά, ενώ για το μοντέλο RT-

DETR-l ξεπερνούσε τη μία ώρα. Η περίπτωση του μοντέλου YOLOv11m-OBB είναι επίσης παρόμοια, καθώς ακόμα και με ελάχιστη παραμετροποίηση προκαλούνταν σφάλμα OutOfMemory κατά την εκτέλεση σε GPU A100 στο περιβάλλον Google Colab, ενώ στην αρχική του μορφή, χωρίς την προσθήκη παραμέτρων, εκτελούνταν αρκετά γρήγορα.

Αναφορά πρέπει να γίνει και στη μετρική F1-Score, η οποία δίνει μια πληρέστερη εικόνα της σχέσης μεταξύ Precision και Recall, συμβάλλοντας στη πιο αξιόπιστη εκτίμηση των μοντέλων. Η ανάλυση αυτής της μετρικής δείχνει ότι τα πειράματα που έγιναν με το YOLOv11m έχουν πάντα καλύτερη απόδοση, στη συγκεκριμένη μετρική, ανεξάρτητα από τον optimizer που χρησιμοποιήθηκε. Αυτό υποδηλώνει ότι οι συγκεκριμένες παράμετροι της έκδοσης YOLOv11m βοηθούν το μοντέλο να γενικεύει καλύτερα, και να μειώνει τα σφάλματα κατηγοριοποίησης, υστερώντας όμως σε χρόνο εκτέλεσης.

Η ενσωμάτωση augmentation κατά την διαδικασία της εκπαίδευσης, απέφερε θετικά αποτελέσματα σε ορισμένες περιπτώσεις, ενώ σε άλλες τα οφέλη του δεν υπερτερούν του πρόσθετου υπολογιστικού κόστους. Ο παράγοντας που συμβάλλει σημαντικά στην αύξηση της απόδοσης είναι η τεχνική SAHI, η οποία γενικά έχει θετική συνεισφορά στο αποτέλεσμα ενώ επιβαρύνει με μικρό υπολογιστικό κόστος. Η τεχνική SAHI επειδή εφαρμόζεται μόνο κατά το inference, έχει «χώρο» να προσθέσει κόστος, μιας και η διαδικασία του inference είναι υπολογιστικά ελαφρύτερη από αυτή της εκπαίδευσης και ενσωμάτωσης επιπλέον τεχνικών.

Η μέτρηση της απόδοσης με τις συνήθεις μετρικές είναι πολύ σημαντική καθώς, πρωτίστως, παρέχει πληροφορίες για την πραγματική απόδοση του μοντέλου, ενώ εξίσου σημαντική αναδεικνύεται και η μετρική της ταχύτητας. Με ενσωμάτωση της μηχανικής όρασης σε όλο και περισσότερα αντικείμενα, διαδικασίες και τομείς δραστηριότητας, χρειάζονται μοντέλα που να μπορούν να προβλέπουν ορθά και με υψηλή ταχύτητα.

5.9 Υλοποίηση του συστήματος σε Raspberry Pi

Ο τομέας του edge computing και των embedded συστημάτων αναπτύσσεται συνεχώς, και τα τελευταία χρόνια εμφανίζει όλο και περισσότερο ενδιαφέρον, λόγω κάποιων ωφέλιμων χαρακτηριστικών του, όπως η υψηλή φορητότητα, η ευχρηστία και η προσαρμοστικότητα.

5.9.1 Περιγραφή του υλικού και λογισμικού

Raspberry Pi Zero 2W

Το Raspberry Pi Zero 2W είναι από τα μικρότερα συστήματα της σειράς Raspberry Pi. Το μέγεθος του είναι ιδιαίτερα μικρό (65mm x 30mm) και διαθέτει έναν επεξεργαστή Arm Cortex-A53 @ 1GHz, μνήμη 512 MB LPDDR2 και συνδεσιμότητα με:

- IEEE 802.11b/g/n,
- Bluetooth 4.2
- HAT-compatible 40-pin header
- έξοδο mini-HDMI
- θέση για κάρτα μνήμης microSD
- θύρα CSI-2 για την ενσωμάτωση κάμερας
- Raspberry Pi camera module V2

Η κάμερα Raspberry Pi camera module V2 είναι η επίσημη κάμερα του Raspberry Pi Foundation. Έχει έναν υψηλής ποιότητας αισθητήρα 8 MP από την Sony (Sony IMX 219), ο φακός της είναι σταθερής εστίασης και μπορεί να παρέχει σταθερές εικόνες ανάλυσης 3280 x 2464 pixel και βίντεο 1080p 30, 720p 60, 640 x 480p 90.

Το λειτουργικό σύστημα που χρησιμοποιείται είναι κατά βάση το Rasbian, το οποίο είναι μια έκδοση βασισμένη σε Linux Debian, και προσαρμοσμένο για το hardware των Raspberry Pi ώστε να εκμεταλλεύεται καλύτερα τις δυνατότητες του και να μη το επιβαρύνει. Επιπλέον, χρησιμοποιήθηκε το λογισμικό VNC για την απομακρυσμένη διαχείριση του Raspberry Pi, και φορτώθηκαν οι απαραίτητες βιβλιοθήκες για την εκτέλεση των προβλέψεων.

5.9.2 Η εφαρμογή

Η εφαρμογή του μοντέλου YOLOv11n υλοποιήθηκε σε headless περιβάλλον, ώστε να μην απαιτούνται επιπλέον συνδέσεις με περιφερειακά. Με τον τρόπο αυτό το σύστημα γίνεται πιο ευέλικτο και μπορεί να εγκατασταθεί σε οποιοδήποτε σημείο, διευκολύνοντας την δοκιμή της ανίχνευσης σε πραγματικό χρόνο.

Όπως αναφέρθηκε στην παρούσα εργασία, επιχειρείται η διερεύνηση και εφαρμογή της ανίχνευσης μικρών αντικειμένων σε πραγματικό χρόνο (real time application) σε φορητή συσκευή. Η καταλληλότερη έκδοση από τα μοντέλα που δοκιμάστηκαν είναι η YOLOv11n, καθώς είναι η ελαφρύτερη συμβιβάζοντας μέρος της ακρίβειας.

Στο μοντέλο YOLOv11n έγιναν κάποιες διαφοροποιήσεις σε σχέση με την έκδοση που εκτελέστηκε, τόσο τοπικά στο υπολογιστή όσο και στο Google Colab. Για να βοηθηθεί το Raspberry Pi Zero 2W, καθώς έχει πολύ περιορισμένο υλικό, περάστηκε το μοντέλο από την διαδικασία του quantization, μειώνοντας την ακρίβεια στα βάρη και βελτιστοποιώντας την χρήση της μνήμης.

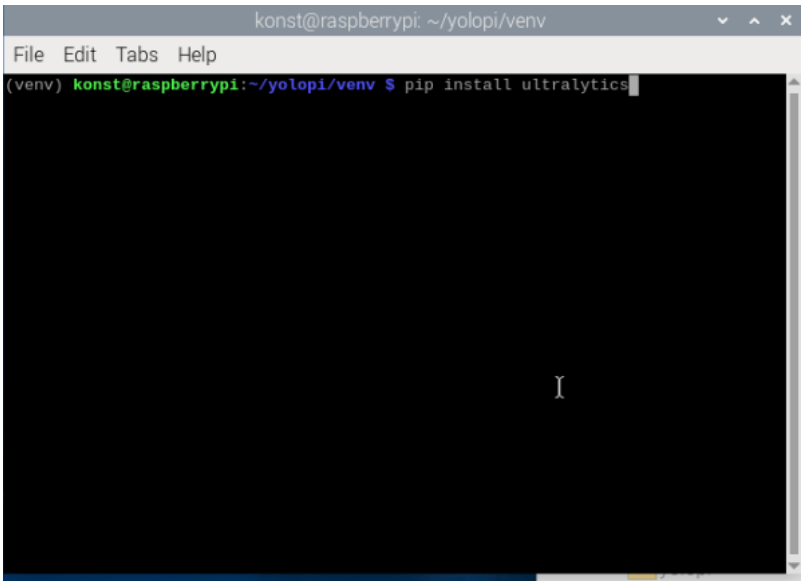
Στην συγκεκριμένη περίπτωση εφαρμογής, το αρχικό μοντέλο εκπαιδεύτηκε ξανά με εκτεταμένη επαύξηση ώστε να ενισχυθεί όσο καλύτερα γίνεται και ορίστηκε η παράμετρος int8 = True για να μειωθεί το αποτύπωμα στην μνήμη μέσω της μείωσης της ακρίβειας στα βάρη. Στη συνέχεια, μετά την ολοκλήρωση της εκπαίδευσης το μοντέλο εξήχθη σε μορφή “NCNN” με σκοπό την περαιτέρω βελτιστοποίηση και χρήση του μέσα από το Raspberry Pi για την πραγματοποίηση προβλέψεων.

Οι δοκιμές έγιναν με δυο τρόπους. Ο πρώτος, με την αυτόνομη ανίχνευση σε πραγματικό χρόνο χρησιμοποιώντας μόνο τους πόρους του Raspberry Pi, και ο δεύτερος χρησιμοποιώντας το Raspberry Pi ως server που μεταδίδει τις εικόνες στον τοπικό υπολογιστή κάνοντας χρήση των πόρων του δεύτερου πραγματοποιώντας την ανίχνευση. Στην περίπτωση που χρησιμοποιούνται οι πόροι του Raspberry Pi, τα αποτελέσματα εξάγονται κάθε 2 – 3 δευτερόλεπτα με καλή ακρίβεια. Όταν χρησιμοποιείται η επιλογή του μοντέλου σε μορφή ncnn, τότε ο συνολικός χρόνος κυμαίνεται από 500 – 700 ms επιτυγχάνοντας μείωση περίπου έξι φορές σε σχέση με την αρχική υλοποίηση.

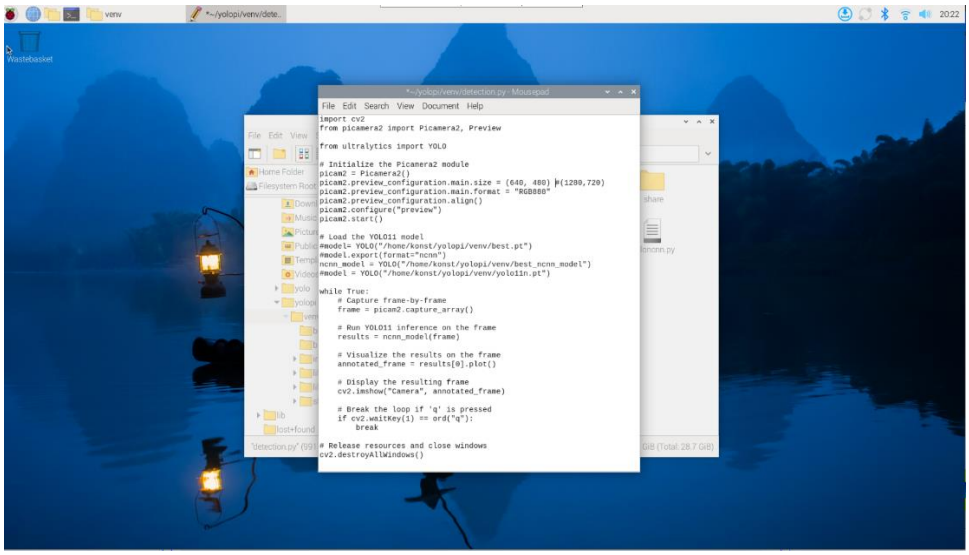
Στο δεύτερο τρόπο, με το Raspberry Pi να λειτουργεί ως server, οι ανιχνεύσεις γίνονται από τον τοπικό υπολογιστή σε πολύ μικρό χρόνο, της τάξης των 15 – 20 ms. Η συγκεκριμένη αρχιτεκτονική εκμεταλλεύεται την υπολογιστική ισχύ του τοπικού υπολογιστή μειώνοντας σημαντικά τους χρόνους εξαγωγής αποτελεσμάτων.

Παρόλο που η κάμερα Camera Module 2 δεν παρέχει στην πράξη την υψηλότερη δυνατή ποιότητα εικόνας λόγω μικρού μεγέθους αισθητήρα, αποτελεί πολύ καλή επιλογή, καθώς κρατάει το συνολικό μέγεθος του συστήματος πολύ μικρό.

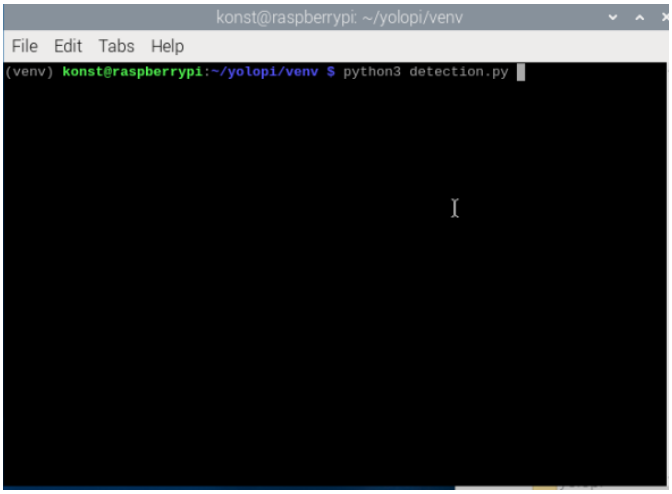
Τα αποτελέσματα είναι πολύ ικανοποιητικά όταν το Raspberry Pi χρησιμοποιείται ως server για την μετάδοση της εικόνας, και η λειτουργία του είναι απρόσκοπτη και ομαλή. Όταν όμως χρησιμοποιείται ως αυτοτελές σύστημα, λόγω των πολύ μικρών πόρων υπάρχει καθυστέρηση στην λήψη αποτελεσμάτων. Το σημαντικό πλεονέκτημα είναι ότι είναι ένα σύστημα με πολύ χαμηλό κόστος και μικρό μέγεθος, το οποίο μπορεί να αξιοποιηθεί σε εφαρμογές πραγματικού χρόνου που οι σκηνές δε μεταβάλλονται με ταχύτητα.



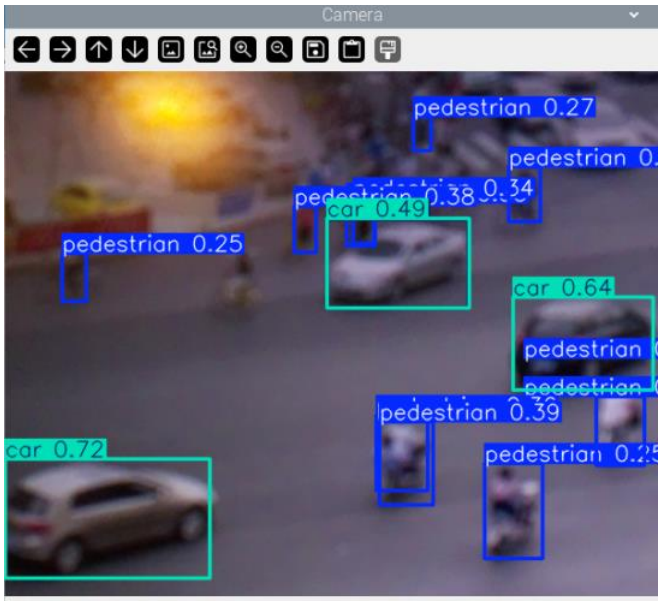
Εικόνα 5.13: Η εντολή εγκατάστασης της βιβλιοθήκης Ultralytics στο Raspberry Pi



Εικόνα 5.14: Ο κώδικας εκτέλεσης για την πραγματοποίηση της ανίχνευσης σε πραγματικό χρόνο



Εικόνα 5.15: Η εντολή εκτέλεσης του κώδικα για την ανίχνευση σε πραγματικό χρόνο



Εικόνα 5.16: Παράδειγμα ανίχνευσης πραγματικού χρόνου

Κεφάλαιο 6ο: Συμπεράσματα και Μελλοντική Εργασία

Στο παρόν κεφάλαιο παρουσιάζονται συνολικά, το πρόβλημα της ανίχνευσης μικρών αντικειμένων, η αναφορά σε παραδείγματα περιπτώσεων που προέκυψαν μέσα από τα πειράματα, οι δυσκολίες που ανέκυψαν κατά την διαδικασία της δοκιμής των μοντέλων, καθώς και οι περιορισμοί του υλικού και οι δυνατότητες για μελλοντικές εργασίες.

6.1 Κύρια ευρήματα της εργασίας

Σκοπός της παρούσας διπλωματικής εργασίας είναι η διερεύνηση της ανίχνευσης μικρών αντικειμένων. Η ανίχνευση μικρών αντικειμένων παραμένει ένα δυσεπίλυτο πρόβλημα, καθώς τα αντικείμενα αναπαρίστανται από πολύ μικρό αριθμό εικονοστοιχείων, και καταλαμβάνουν πολύ μικρό μέρος από την συνολική εικόνα. Επιπλέον οι λήψεις στις οποίες αναζητούνται τα αντικείμενα δεν διαθέτουν πάντα ιδανικές συνθήκες, καθώς συχνά μπορεί να έχουν χαμηλό φωτισμό, διαφορετικές κλίσεις κατά τη λήψη, μερική κάλυψη από άλλα αντικείμενα ή αποκοπή στα όρια της εικόνας. Η χαμηλή ανάλυση, λοιπόν, ακόμα και όταν αφορά «τοπικά» το αντικείμενο, καθιστά την ανίχνευση εξ' ορισμού δύσκολη. Αν η εικόνα είναι υψηλής ανάλυσης, το αντικείμενο συχνά περιγράφεται με λιγότερο από δέκα εικονοστοιχεία. Αυτά τα χαρακτηριστικά κάνουν την διαδικασία του εντοπισμού αρκετά δύσκολη και επίπονη για τα μοντέλα ανίχνευσης. Δυσκολία, επίσης, προσθέτει το γεγονός ότι τα αντικείμενα δεν κατανέμονται όμοια στην εικόνα αλλά πολλές φορές σχηματίζουν συστάδες (clusters of objects) που δυσχεραίνουν την διαδικασία ιδιαίτερα όταν αυτά είναι ετερογενή, ανήκουν δηλαδή σε διαφορετικές κατηγορίες. Κατά την εξέταση των εικόνων από τα διαθέσιμα σύνολα δεδομένων, τα οποία είναι σχετικά με την ανίχνευση μικρών αντικειμένων από εναέριες λήψεις, γινόταν πολύ γρήγορα αντιληπτό ότι οι εικόνες παρουσιάζουν πολύ μεγάλη γωνία λήψης και επιφάνεια στο φόντο - χαρακτηριστικά που διαφοροποιούνται σχεδόν σε όλα τα σύνολα δεδομένων γενικού εντοπισμού αντικειμένων, και επηρεάζουν αρνητικά την απόδοση των μοντέλων. Τα παραπάνω οδηγούν στο συμπέρασμα ότι δεν υπάρχουν πολλά χαρακτηριστικά για να εξαχθούν με ένα επίπεδο δικτύου, αλλά χρειάζονται αρχιτεκτονικές που να μπορούν να εξάγουν χαρακτηριστικά σε πολλαπλές κλίμακες μιας και τα αντικείμενα που βρίσκονται πιο κοντά στον φακό της κάμερας φαίνονται μεγαλύτερα και καταλαμβάνουν περισσότερα εικονοστοιχεία σε σχέση με τα αντικείμενα που είναι μακρύτερα και άρα έχουν λιγότερα εικονοστοιχεία για να εξαχθούν χαρακτηριστικά (feature extraction).

Για τα δεδομένα χρησιμοποιήθηκαν δύο πολύ γνωστά σύνολα δεδομένων, το VisDrone Dataset και το DOTA Dataset. Τα δύο αυτά σύνολα δεδομένων είναι από τα δυσκολότερα για την ανίχνευση μικρών αντικειμένων, και θεωρούνται σύνολα δεδομένων benchmark καθώς χρησιμοποιούνται ευρέως για την αξιολόγηση των προτεινόμενων μοντέλων ανίχνευσης μικρών αντικειμένων από την επιστημονική κοινότητα. Το σύνολο δεδομένων VisDrone παρέχει εικόνες που έχουν ληφθεί από μικρό ύψος και υπό κλίση από οχήματα UAV και είναι αρκετά τεχνικά δύσκολο. Το σύνολο δεδομένων DOTA παρέχει δορυφορικές εικόνες υψηλής ανάλυσης όπου τα αντικείμενα εμφανίζονται πυκνά μεταξύ τους. Ένα χαρακτηριστικό που το διαφοροποιεί από τα άλλα σύνολα δεδομένων είναι η χρήση προσανατολισμένων οριοθετημένων κουτιών (OBB). Τα δύο αυτά σύνολα δυσχεραίνουν την επίτευξη καλών αποτελεσμάτων, στοιχείο που υποδηλώνει ότι η επιλογή τους είναι σωστή, καθώς ενθαρρύνει την αναζήτηση βέλτιστων πρακτικών.

Στα δεδομένα εφαρμόστηκαν τεχνικές επαύξησης δεδομένων κατά την διάρκεια της εκπαίδευσης διαφοροποιώντας ένα ποσοστό αυτών ώστε ο αλγόριθμος να «μάθει» σε συνθετικά χαρακτηριστικά. Δημιουργήθηκαν εικόνες που δεν υπήρχαν στο αρχικό σύνολο δεδομένων, κάτι που όπως

παρατηρήθηκε, σε ορισμένες περιπτώσεις βελτίωσε την απόδοση, ενώ σε άλλες δεν προσέφερε επιπλέον οφέλη. Το βασικότερο χαρακτηριστικό augmentation ήταν η χρήση της υπερπαραμέτρου multi_scale = True η οποία μικραίνει ή μεγεθύνει ένα αντικείμενο κατά ένα ποσοστό σε ένα τυχαίο δείγμα του συνόλου εκπαίδευσης. Σημαντική ήταν, επίσης, η υπερπαράμετρος cos_lr (cosine annealing) που βοήθησε με τη διαχείριση του learning rate, τον περιορισμό του overfitting και την γρηγορότερη εκπαίδευση του μοντέλου.

Χρησιμοποιήθηκαν γνωστά μοντέλα μέσω της open source πλατφόρμας Ultralytics, η οποία παρέχει εύκολη εκτέλεση εκπαιδεύσεων και δυνατότητα παραμετροποίησης των μοντέλων, καθώς αυτά είναι ελεύθερα διαθέσιμα υπό την άδεια χρήσης AGPL 3.0. Η χρήση του Ultralytics API έδωσε την δυνατότητα ενός περιβάλλοντος χωρίς ασυμβατότητες με τις βιβλιοθήκες της γλώσσας Python και διευκόλυνε την επικοινωνία με την Pytorch.

Αρχικά, η εργασία εκτελέστηκε σε τοπικό υπολογιστή με την χρήση της κάρτας γραφικών Nvidia RTX 3050 Ti με σκοπό να διερευνηθεί και η δυνατότητα εκτέλεσης εκπαίδευσης μοντέλων μηχανικής όρασης σε τοπικό υπολογιστή χωρίς την χρήση υπηρεσίας cloud computing. Αυτό δίνει αφενός μια εικόνα των δυνατοτήτων και των δυσκολιών που έχει μια τέτοια προσπάθεια, αφετέρου εξασφαλίζει ένα κοινό υπολογιστικό σύστημα για την αξιολόγηση των μοντέλων, ειδικά με την μετρική FPS.

Στη συνέχεια, λόγω βλάβης στον τοπικό υπολογιστή, τα πειράματα μεταφέρθηκαν στο περιβάλλον του Google Colab. Αυτό είχε ως αποτέλεσμα να χρειαστεί να γίνουν ξανά κάποιες εκτελέσεις εκπαιδεύσεων ώστε εξασφαλιστεί ένα μέτρο σύγκρισης. Το συμπέρασμα που εξήχθη από την εμπειρία με το περιβάλλον Google Colab είναι ότι η χρήση του είναι αρκετά κοστοβόρα, όσον αφορά την κατανάλωση των υπολογιστικών μονάδων (compute units) που «φορτώνονται» στην πλατφόρμα. Επίσης, αξίζει να σημειωθεί ότι στα μικρά μοντέλα η χρήση της πλατφόρμας δεν συμφέρει, καθώς η χρονική επιτάχυνση που προσφέρει δεν είναι ανάλογη του κόστους. Η εκπαίδευση ανά εποχή στα μικρά μοντέλα, στο τοπικό σύστημα, ήταν ελαφρώς πιο αργή κατά ένα με ενάμιση λεπτό. Η χρήση της πλατφόρμας προσφέρει ουσιαστικό όφελος στην εκπαίδευση του μοντέλου YOLOv11m, καθώς παρατηρείται σημαντική διαφορά στην απόδοση. Σε ότι αφορά την εκπαίδευση του μοντέλου RT-DETR-l στο τοπικό σύστημα, όπως ήδη αναφέρθηκε, ήταν ιδιαίτερα δύσκολη, καθώς για ένα διάστημα εκπαίδευσης 73 εποχών χρειάστηκαν συνολικά 21 ώρες εκπαίδευσης που εκτελέστηκαν σε τρία τμήματα. Αντιθέτως, η ίδια εκπαίδευση σε περιβάλλον Google Colab ολοκληρώθηκε σε περισσότερο από επτά ώρες. Επιπροσθέτως, ο αλγόριθμος YOLOv11m-OBB, αν και στην default ρύθμιση του φαίνεται ελαφρύς, παρουσιάζει περιορισμούς. Με την προσθήκη ελάχιστων παραμέτρων, και ειδικότερα αυτών που σχετίζονται με την κλίμακα, ήδη από την πρώτη εποχή εκπαίδευσης στο περιβάλλον Google Colab με την GPU A100 40 Gb, εμφανίζεται σφάλμα OutOfMemoryError. Τέλος, για την εργασία πραγματοποιήθηκαν περισσότερες από 120 εκπαιδεύσεις μοντέλων με θετικά και αρνητικά αποτελέσματα, και έγινε σαφές ότι η σωστή ρύθμιση μιας εκπαίδευσης είναι δύσκολη διαδικασία, η οποία για να αποδώσει αποτελέσματα απαιτεί τόσο θεωρητική γνώση όσο και επαρκείς πόρους.

Στη παρούσα εργασία επιχειρήθηκε η εφαρμογή του μοντέλου YOLOv11n σε πραγματικό χρόνο σε περιβάλλον Raspberry Pi. Το συγκεκριμένο μοντέλο επιλέχθηκε λόγω του μικρού μεγέθους και της ανάγκης του για υπολογιστικούς πόρους. Προτιμήθηκε το Raspberry Pi Zero 2W λόγω του μικρού μεγέθους και της προσαρμοστικότητας του. Ενσωματώθηκε η Camera Module 2 για την καλύτερη συμβατότητα της. Τα αποτελέσματα από άποψη ακρίβειας είναι ιδιαίτερα ικανοποιητικά ειδικά αν ληφθεί υπόψη η ποιότητα του αισθητήρα που χρησιμοποιείται. Οι χρόνοι λήψης των αποτελεσμάτων με τη χρήση του Raspberry Pi ως server, είναι σχεδόν εφάμιλλοι με αυτούς του τοπικού υπολογιστή. Όταν αξιοποιούνται οι υπολογιστικοί πόροι του Raspberry Pi και χρησιμοποιείται το μοντέλο στην

αρχική του μορφή, τα αποτελέσματα δείχνουν ότι η εφαρμογή σε πραγματικό χρόνο είναι δυνατή, ωστόσο δεν είναι κατάλληλη για σκηνές με γρήγορες μεταβολές. Με τη χρήση του μοντέλου σε μορφή ncnn, οι χρόνοι λήψης αποτελέσματος μειώνονται μέχρι και έξι φορές, καθιστώντας δυνατή την εφαρμογή του σε πιο δυναμικά περιβάλλοντα. Τέλος, η υλοποίηση σε Raspberry Pi επιβεβαιώνει την πρακτική βιωσιμότητα της μεθόδου και ανοίγει προοπτικές για την ενσωμάτωση σε συστήματα χαμηλού κόστους και real-time παρακολούθησης.

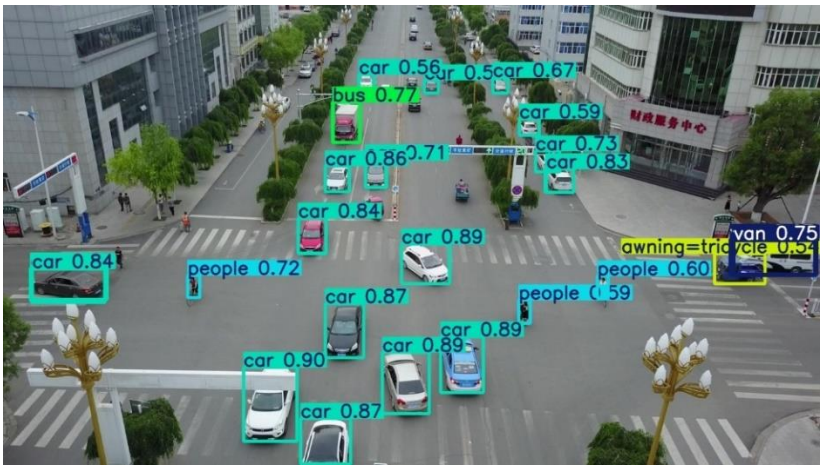
Οι αρχικές εκπαιδεύσεις των μοντέλων έγιναν χωρίς την χρήση υπερπαραμέτρων, για να εξαχθεί μια βάση απόδοσης και να παρθούν αποτελέσματα τα οποία να συγκρίνονται στη συνέχεια με τις αλλαγές που εφαρμόστηκαν. Παρατηρήθηκε ότι τα μοντέλα με τις default παραμέτρους αποδίδουν αρκετά καλά ειδικά η έκδοση m του μοντέλου YOLOv11. Τα τροποποιημένα μοντέλα εμφανίζουν τιμές mAP και F1 πολύ κοντά στα αρχικά, γεγονός που υποδηλώνει ότι χρειάζεται περαιτέρω διερεύνηση ως προς την χρήση υπερπαραμέτρων εκπαίδευσης και τεχνικών επαύξησης.

Τα βέλτιστα αποτελέσματα προκύπτουν με τον optimizer SGD σε όλες τις εκδόσεις του μοντέλου YOLOv11, ενώ οι υψηλότερες επιδόσεις καταγράφονται με το μοντέλο YOLOv11m - SGD με τη χρήση υπερπαραμέτρων, batch size = 16, img_size = 640, epochs = 50, mosaic = True, mutli_scale = True, scale = 0.37, cos_lr = True. Σε αυτές τις ρυθμίσεις παρατηρείται mAP75 = 0.289 και F1-Score = 0.476. Ένα ακόμα συμπέρασμα που βγαίνει από τα πειράματα είναι ότι ο καλύτερος optimizer είναι ο SGD και έπειτα ο AdamW, το οποίο πιθανότατα οφείλεται στην καλύτερη διαχείριση που κάνουν στα βάρη. Ο optimizer Adam είχε χαμηλότερη απόδοση σε όλες τις μετρικές στα πειράματα που πραγματοποιήθηκαν. Το μοντέλο RT-DETR-l παρουσιάζει επίσης παρόμοιες μετρικές και αποδίδει ικανοποιητικά σε δύσκολα περιβάλλοντα, καθώς χρησιμοποιώντας τους attention μηχανισμούς του, μπορεί να διατηρεί ικανοποιητική απόδοση, αν και με υψηλό υπολογιστικό κόστος σε όλα τα στάδια του. Αν και κάποια από τα πειράματα (περισσότερα από 120) δεν έδειξαν καλύτερα αποτελέσματα, ήταν μια διαδικασία που δίδαξε τεχνικές και έδειξε τις δυσκολίες που συναντά ένας ερευνητής που ξεκινάει την τριβή του με το αντικείμενο της ανίχνευσης μικρών αντικειμένων.

Η επίδραση της τεχνικής SAHI είναι γενικά ευεργετική, καθώς αυξάνει την απόδοση χωρίς να επιβαρύνει στην εκτέλεση ή την εκπαίδευση, δεδομένου ότι εφαρμόζεται μόνο κατά την διάρκεια του inference, το οποίο σε πολλές περιπτώσεις διαθέτει περιθώρια αύξησης κόστους.

Η μηχανική όραση εφαρμόζεται σε τομείς όπου, τις περισσότερες φορές, ζητείται να λειτουργήσει σε πραγματικό χρόνο, κρατώντας υψηλή την ακρίβεια των προβλέψεων. Αυτό εξελίσσεται στο σημαντικότερο κριτήριο, καθώς υπάρχει ζήτηση για εφαρμογές σε διάφορους τομείς οι οποίοι πολλές φορές είναι κρίσιμοι για τον άνθρωπο, όπως για παράδειγμα η ιατρική και εφαρμογές αυτόνομης οδήγησης.

Ένα ακόμα συμπέρασμα είναι ότι δεν υπάρχει μια καθολική λύση για κάθε πρόβλημα ή εφαρμογή αλλά όλες οι υλοποιήσεις φέρουν θετικά και αρνητικά στοιχεία τα οποία πρέπει να λαμβάνονται υπόψη λόγω των επιπτώσεων που έχουν. Στο πλαίσιο όμως της παρούσας εργασίας, προτείνεται το YOLOv11, καθώς η οικογένεια αυτών των μοντέλων συνδυάζει υψηλή ταχύτητα με ικανοποιητική ποιότητα αποτελεσμάτων. Η επιλογή των μοντέλων YOLOv11 αιτιολογείται από το γεγονός ότι οι σύγχρονες εφαρμογές απαιτούν λειτουργία σε πραγματικό χρόνο, κριτήριο το οποίο αποκτά πλέον ιδιαίτερη σημασία. Προτείνεται, επίσης, και η χρήση της τεχνικής SAHI ή τουλάχιστον η διερεύνηση της καθώς βελτιώνει τα αποτελέσματα με πολύ μικρό υπολογιστικό κόστος (εικόνα 6.1).



Εικόνα 6.1: Παράδειγμα αποτελέσματος πρόβλεψης με την χρήση της τεχνικής SAHI

6.2 Προτάσεις για βελτίωση και εφαρμογή σε προβλήματα πραγματικού κόσμου.

Το ζήτημα της ανίχνευσης μικρών αντικειμένων παραμένει ανοιχτό και πολυδιάστατο μιας και εφαρμόζεται σε πλήθος εφαρμογών από διαφορετικά επιστημονικά πεδία. Μια πρώτη βελτίωση μπορεί να είναι η διερεύνηση της εκπαίδευσης του μοντέλου σε πολλαπλά σύνολα δεδομένων για την μέτρηση της μεταβολής των αποτελεσμάτων και την εκτίμηση της γενίκευσης σε εφαρμογές πραγματικού χρόνου. Η εκπαίδευση σε πολλαπλά σύνολα δεδομένων ίσως να έχει θετικά αποτελέσματα σε προβλήματα που προκύπτουν είτε από την ανισοκατανομή των κλάσεων είτε από τα ραδιομετρικά χαρακτηριστικά των εικόνων κατά την στιγμή της λήψης λόγω κλιματικών συνθηκών ή συνθηκών φωτισμού.

Η ανάπτυξη αποδοτικότερων μοντέλων πρέπει να είναι η κύρια κατεύθυνση καθώς η ενσωμάτωση της μηχανικής όρασης σε όλο και περισσότερα αντικείμενα και διαδικασίες, απαιτεί την διάθεση ελαφριών μοντέλων ώστε να μην έχουν απαιτήσεις σε υπολογιστικούς πόρους. Αυτό μπορεί να επιτευχθεί είτε μέσω της ανάπτυξης μιας νέας αρχιτεκτονικής με στόχο την βελτίωση των αποτελεσμάτων σε συνάρτηση πάντα τον χρόνο, δεδομένου ότι εξετάζονται προβλήματα πραγματικού κόσμου που χρειάζονται απόκριση σε πραγματικό χρόνο, είτε με την τροποποίηση υφιστάμενου μοντέλου YOLOv11, το οποίο αποτελεί μια καλή βάση ειδικά ως προς τον χρόνο προσφέροντας δυνατότητες περαιτέρω έρευνας στην βελτίωση της απόδοσης.

Μια ακόμα πρόταση για έρευνα και βελτίωση αφορά την ανάπτυξη εφαρμογών ανίχνευσης μικρών αντικειμένων σε συσκευές edge σε πραγματικό χρόνο, ανοίγοντας έναν καινούργιο δρόμο για εφαρμογές πιο κοντά στην καθημερινότητα, καθιστώντας τα οφέλη της προσιτά σε περισσότερο κόσμο.

6.3 Ενσωμάτωση SAHI, Transformers και Hardware Acceleration

Η αποτελεσματική ανίχνευση μικρών αντικειμένων αποτελεί πρόκληση λόγω της μικρής χωρικής κλίμακας, των περιορισμένων χαρακτηριστικών, της ύπαρξης θορύβου και των περιορισμένων υπολογιστικών πόρων. Η χρήση της τεχνικής ενίσχυσης SAHI στο στάδιο των προβλέψεων βελτίωσε τα αποτελέσματα, καθώς είναι tile based, διασπά την εικόνα σε τμήματα και τα εξετάζει μεμονωμένα, ανιχνεύοντας αντικείμενα σε κάθε τμήμα της πριν την ενώσει ξανά. Έχει όμως την δυνατότητα, όπως προαναφέρθηκε και στο αντίστοιχο κεφάλαιο, να λειτουργήσει σε ολόκληρη την εικόνα.

Η χρήση Transformers, όπως το μοντέλο RT-DETR-l, ενσωματώνοντας λειτουργίες attention δίνει ικανοποιητικά αποτελέσματα, και είναι μια τεχνολογία που σίγουρα πρέπει να ερευνηθεί παραπάνω. Επιπροσθέτως, δίνει καλά αποτελέσματα σε πυκνές σκηνές με αντικείμενα σε κοντινή απόσταση, ενώ

αντιμετωπίζει αποτελεσματικά το πρόβλημα των διαφορετικών κλιμάκων. Βελτίωση χρειάζεται κυρίως σε ότι αφορά τον χρόνο και το πολύ υψηλό υπολογιστικό του κόστος.

Το Hardware Acceleration αποτελεί καθοριστικό παράγοντα, καθώς η εκπαίδευση τόσο μεγάλων δομών δεν είναι εφικτή μόνο με την χρήση της CPU - η αξιοποίηση GPU ή TPU καθίσταται απαραίτητη. Από τα πειράματα που διεξήχθησαν στην παρούσα εργασία, είναι εμφανές ότι ακόμα και η χρήση μόνο μιας GPU, όπως αυτή της A100, είναι πολύ περιοριστική. Πολλές φορές η χρήση της έφτασε τα όρια της με τα μοντέλα της οικογένειας YOLOv11, καθώς και με το μοντέλο RT-DETR-l, με εξαίρεση την εκπαίδευση το μοντέλου με τις default παραμέτρους. Σε όλες τις υπόλοιπες περιπτώσεις προσθήκης υπερπαραμέτρων ή augmentation, το περιβάλλον του Google Colab υπερέβαινε τη διαθέσιμη μνήμη και η εκπαίδευση σταματούσε στις πρώτες εποχές. Αυτό κάνει σαφές ότι απαιτείται ένα σύστημα cluster με πολλαπλές GPU, καθώς οι απαιτήσεις είναι αυξημένες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] A. Georgouli, 'Τεχνητή νοημοσύνη', p. 180, 2016, doi: 10.57713/KALLIPOS-666.

[2] 'One Hidden Layer (Shallow) Neural Network Architecture', One Hidden Layer (Shallow) Neural Network Architecture. Accessed: July 25, 2025. [Online]. Available: https://medium.com/data-science-365/one-hidden-layer-shallow-neural-network-architecture-d45097f649e6

[3] Λ. Χασάπη, 'Deep Learning', Accessed: July 25, 2025. [Online]. Available: https://eclass.upatras.gr/modules/document/file.php/PT187/11.DEEP%20LEARNING%2C%20MACHINE%20LEARNING.pdf

[4] 'What is a neural network?', What is a neural network? Accessed: July 25, 2025. [Online]. Available: https://www.ibm.com/think/topics/neural-networks

[5] 'What is computer vision?', What is computer vision? Accessed: July 25, 2025. [Online]. Available: https://www.ibm.com/think/topics/computer-vision

[6] T. S. Huang, 'Computer Vision: Evolution and Promise'.

[7] 'Neural Mechanisms of High-Level Vision', in *Comprehensive Physiology*, 1st edn, Wiley, 2018, pp. 903–953. doi: 10.1002/cphy.c160035.

[8] 'A history of vision models', A history of vision models. Accessed: July 25, 2025. [Online]. Available: https://www.ultralytics.com/blog/a-history-of-vision-models

[9] 'A Brief History of Computer Vision', A Brief History of Computer Vision. Accessed: July 25, 2025. [Online]. Available: https://opencv.courses/blog/a-brief-history-of-computer-vision/

[10] 'Computer Vision Market Replicate Human Vision', Computer Vision Market Replicate Human Vision. Accessed: July 25, 2025. [Online]. Available: https://www.kbvresearch.com/blog/computer-vision-market-replicate-human-vision/

[11] M. Tan, R. Pang, and Q. V. Le, 'EfficientDet: Scalable and Efficient Object Detection', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, June 2020. doi: 10.1109/cvpr42600.2020.01079.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, 'Focal Loss for Dense Object Detection', in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice: IEEE, Oct. 2017. doi: 10.1109/iccv.2017.324.

[13] H. Faraji and B. Chen, 'Drone-YOLO: Improved YOLO for Small Object Detection in UAV', in *2023 8th International Conference on Image, Vision and Computing (ICIVC)*, Dalian, China: IEEE, July 2023, pp. 93–100. doi: 10.1109/icivc58118.2023.10270571.

[14] X. Xie, J. Ren, Y. Zeng, S. Wei, Y. Wang, and W. Luan, 'HATSC-YOLOv10:Improved YOLOv10 for Satellite Remote Sensing Images of Small Object Detection', in *2024 China Automation Congress (CAC)*, Qingdao, China: IEEE, Nov. 2024, pp. 3795–3799. doi: 10.1109/cac63892.2024.10865623.

[15] S. Ren, K. He, R. Girshick, and J. Sun, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', Jan. 06, 2016, *arXiv*: arXiv:1506.01497. doi: 10.48550/arXiv.1506.01497.

[16] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, 'An Evaluation of Deep Learning Methods for Small Object Detection', *Journal of Electrical and Computer Engineering*, vol. 2020, pp. 1–18, Apr. 2020, doi: 10.1155/2020/3189691.

[17] H. Liu, X. Ma, Y. Yu, L. Wang, and L. Hao, 'Application of Deep Learning-Based Object Detection Techniques in Fish Aquaculture: A Review', *JMSE*, vol. 11, no. 4, p. 867, Apr. 2023, doi: 10.3390/jmse11040867.

[18] A. Vaswani *et al.*, 'Attention is All you Need'.

[19] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, 'End-to-End Object Detection with Transformers', May 28, 2020, *arXiv*: arXiv:2005.12872. doi: 10.48550/arXiv.2005.12872.

[20] 'Introduction to DETR - Part 2: The Crucial Role of the Hungarian Algorithm', Introduction to DETR - Part 2: The Crucial Role of the Hungarian Algorithm. [Online]. Available: Introduction to DETR - Part 2: The Crucial Role of the Hungarian Algorithm

[21] ‘DETR Breakdown Part 2: Methodologies and Algorithms’, DETR Breakdown Part 2: Methodologies and Algorithms. Accessed: July 15, 2025. [Online]. Available: https://pyimagesearch.com/2023/06/12/detr-breakdown-part-2-methodologies-and-algorithms/
[22] Y. Kong, X. Shang, and S. Jia, ‘Drone-DETR: Efficient Small Object Detection for Remote Sensing Image Using Enhanced RT-DETR Model’, *Sensors*, vol. 24, no. 17, p. 5496, Aug. 2024, doi: 10.3390/s24175496.
[23] Z. Yao, J. Ai, B. Li, and C. Zhang, ‘Efficient DETR: Improving End-to-End Object Detector with Dense Prior’, Apr. 03, 2021, *arXiv*: arXiv:2104.01318. doi: 10.48550/arXiv.2104.01318.
[24] A. G. Howard *et al.*, ‘MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications’, Apr. 17, 2017, *arXiv*: arXiv:1704.04861. doi: 10.48550/arXiv.1704.04861.
[25] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, ‘SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size’, Nov. 04, 2016, *arXiv*: arXiv:1602.07360. doi: 10.48550/arXiv.1602.07360.
[26] J. Redmon and A. Farhadi, ‘YOLOv3: An Incremental Improvement’, Apr. 08, 2018, *arXiv*: arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767.
[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, ‘Rectifier Nonlinearities Improve Neural Network Acoustic Models’.
[28] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, ‘Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)’, Feb. 22, 2016, *arXiv*: arXiv:1511.07289. doi: 10.48550/arXiv.1511.07289.
[29] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, ‘Feature Pyramid Networks for Object Detection’, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, July 2017. doi: 10.1109/cvpr.2017.106.
[30] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, ‘A survey and performance evaluation of deep learning methods for small object detection’, *Expert Systems with Applications*, vol. 172, p. 114602, June 2021, doi: 10.1016/j.eswa.2021.114602.
[31] W. Hua and Q. Chen, ‘A survey of small object detection based on deep learning in aerial images’, *Artif Intell Rev*, vol. 58, no. 6, Mar. 2025, doi: 10.1007/s10462-025-11150-9.
[32] S. Razakarivony and F. Jurie, ‘Vehicle detection in aerial imagery : A small target detection benchmark’, *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, Jan. 2016, doi: 10.1016/j.jvcir.2015.11.002.
[33] J. Ding *et al.*, ‘Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7778–7796, Nov. 2022, doi: 10.1109/tpami.2021.3117983.
[34] X. Sun *et al.*, ‘FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery’, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 184, pp. 116–130, Feb. 2022, doi: 10.1016/j.isprsjprs.2021.12.004.
[35] D. Lam *et al.*, ‘xView: Objects in Context in Overhead Imagery’, Feb. 22, 2018, *arXiv*: arXiv:1802.07856. doi: 10.48550/arXiv.1802.07856.
[36] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, ‘Vision Meets Drones: A Challenge’, Apr. 23, 2018, *arXiv*: arXiv:1804.07437. doi: 10.48550/arXiv.1804.07437.
[37] ‘What is oriented bounding box (OBB) detection?’, What is oriented bounding box (OBB) detection? Accessed: July 17, 2025. [Online]. Available: https://www.ultralytics.com/blog/what-is-oriented-bounding-box-obb-detection-a-quick-guide
[38] C. Wang *et al.*, ‘CF-YOLO for small target detection in drone imagery based on YOLOv11 algorithm’, *Sci Rep*, vol. 15, no. 1, May 2025, doi: 10.1038/s41598-025-99634-0.
[39] ‘Normalization’, Normalization. Accessed: July 17, 2025. [Online]. Available: https://www.ultralytics.com/glossary/normalization
[40] S. Ioffe and C. Szegedy, ‘Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift’.
[41] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, ‘Micro-Batch Training with Batch-Channel Normalization and Weight Standardization’, Aug. 09, 2020, *arXiv*: arXiv:1903.10520. doi: 10.48550/arXiv.1903.10520.
[42] J. L. Ba, J. R. Kiros, and G. E. Hinton, ‘Layer Normalization’, July 21, 2016, *arXiv*: arXiv:1607.06450. doi: 10.48550/arXiv.1607.06450.

[43] C. Shorten and T. M. Khoshgoftaar, ‘A survey on Image Data Augmentation for Deep Learning’, *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
[44] A. Mumuni and F. Mumuni, ‘Data augmentation: A comprehensive survey of modern approaches’, *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/j.array.2022.100258.
[45] R. Takahashi, T. Matsubara, and K. Uehara, ‘Data Augmentation using Random Image Cropping and Patching for Deep CNNs’, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 2917–2931, Sept. 2020, doi: 10.1109/TCSVT.2019.2935128.
[46] S. Kumar, P. Asiamah, O. Jolaoso, and U. Esiowu, ‘Enhancing Image Classification with Augmentation: Data Augmentation Techniques for Improved Image Classification’, Feb. 25, 2025, *arXiv*: arXiv:2502.18691. doi: 10.48550/arXiv.2502.18691.
[47] ‘Transforming and augmenting images’, Transforming and augmenting images. Accessed: July 17, 2025. [Online]. Available: https://docs.pytorch.org/vision/stable/transforms.html?spm=a2c6h.13046898.publish-article.32.15d16ffaf39NzQ
[48] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, ‘AutoAugment: Learning Augmentation Policies from Data’, Apr. 11, 2019, *arXiv*: arXiv:1805.09501. doi: 10.48550/arXiv.1805.09501.
[49] T. DeVries and G. W. Taylor, ‘Improved Regularization of Convolutional Neural Networks with Cutout’, Nov. 29, 2017, *arXiv*: arXiv:1708.04552. doi: 10.48550/arXiv.1708.04552.
[50] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, ‘CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features’, Aug. 07, 2019, *arXiv*: arXiv:1905.04899. doi: 10.48550/arXiv.1905.04899.
[51] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, ‘mixup: Beyond Empirical Risk Minimization’, Apr. 27, 2018, *arXiv*: arXiv:1710.09412. doi: 10.48550/arXiv.1710.09412.
[52] R. Kaur and S. Singh, ‘A comprehensive review of object detection with deep learning’, *Digital Signal Processing*, vol. 132, p. 103812, Jan. 2023, doi: 10.1016/j.dsp.2022.103812.
[53] N. Crasto, ‘Class Imbalance in Object Detection: An Experimental Diagnosis and Study of Mitigation Strategies’, Mar. 11, 2024, *arXiv*: arXiv:2403.07113. doi: 10.48550/arXiv.2403.07113.
[54] R. Khanam and M. Hussain, ‘YOLOv11: An Overview of the Key Architectural Enhancements’, Oct. 23, 2024, *arXiv*: arXiv:2410.17725. doi: 10.48550/arXiv.2410.17725.
[55] ‘YOLO11: Redefining Real-Time Object Detection’, YOLO11: Redefining Real-Time Object Detection. Accessed: Aug. 14, 2025. [Online]. Available: https://learnopencv.com/yolo11/
[56] ‘Baidu’s RT-DETR: A Vision Transformer-Based Real-Time Object Detector’, Baidu’s RT-DETR: A Vision Transformer-Based Real-Time Object Detector. Accessed: Aug. 14, 2025. [Online]. Available: https://docs.ultralytics.com/models/rtdetr/
[57] ‘RT-DETR’, RT-DETR. [Online]. Available: https://roboflow.com/model/rt-detr
[58] Y. Zhao *et al.*, ‘DETRs Beat YOLOs on Real-time Object Detection’, Apr. 03, 2024, *arXiv*: arXiv:2304.08069. doi: 10.48550/arXiv.2304.08069.
[59] ‘Real-Time Object Detection: A Comprehensive Guide to Implementing Baidu’s RT-DETR’, Real-Time Object Detection: A Comprehensive Guide to Implementing Baidu’s RT-DETR. Accessed: Aug. 14, 2025. [Online]. Available: https://www.digitalocean.com/community/tutorials/rt-detr-realtime-detection-transformer
[60] ‘Ultralytics Docs: Using YOLO11 with SAHI for Sliced Inference’, Ultralytics Docs: Using YOLO11 with SAHI for Sliced Inference. Accessed: Aug. 14, 2025. [Online]. Available: https://docs.ultralytics.com/guides/sahi-tiled-inference/
[61] F. C. Akyon, S. O. Altinuc, and A. Temizel, ‘Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection’, in *2022 IEEE International Conference on Image Processing (ICIP)*, Oct. 2022, pp. 966–970. doi: 10.1109/ICIP46576.2022.9897990.
[62] F. D’Angelo, M. Andriushchenko, A. Varre, and N. Flammarion, ‘Why Do We Need Weight Decay in Modern Deep Learning?’, Nov. 04, 2024, *arXiv*: arXiv:2310.04415. doi: 10.48550/arXiv.2310.04415.
[63] Z. Xie, Z. Xu, J. Zhang, I. Sato, and M. Sugiyama, ‘On the Overlooked Pitfalls of Weight Decay and How to Mitigate Them: A Gradient-Norm Perspective’.

[64] T. Sun, Y. Huang, L. Shen, and K. Xu, ‘Investigating the Role of Weight Decay in Enhancing Nonconvex SGD’.
[65] O. Hospodarskyy, V. Martsenyuk, N. Kukharska, and S. Sverstiuk, ‘Understanding the Adam Optimization Algorithm in Machine Learning’.
[66] I. Loshchilov and F. Hutter, ‘Decoupled Weight Decay Regularization’, Jan. 04, 2019, *arXiv*: arXiv:1711.05101. doi: 10.48550/arXiv.1711.05101.
[67] ‘AdamW’, AdamW. Accessed: Aug. 15, 2025. [Online]. Available: https://keras.io/2/api/optimizers/adamw/
[68] ‘Stochastic Gradient Descent (SGD)’, Stochastic Gradient Descent (SGD). Accessed: Aug. 15, 2025. [Online]. Available: https://www.ultralytics.com/glossary/stochastic-gradient-descent-sgd
[69] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, ‘The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study’, *Multimed Tools Appl*, vol. 82, no. 11, pp. 16591–16633, May 2023, doi: 10.1007/s11042-022-13820-0.
[70] L. N. Smith, ‘Cyclical Learning Rates for Training Neural Networks’, Apr. 04, 2017, *arXiv*: arXiv:1506.01186. doi: 10.48550/arXiv.1506.01186.
[71] I. Loshchilov and F. Hutter, ‘SGDR: Stochastic Gradient Descent with Warm Restarts’, May 03, 2017, *arXiv*: arXiv:1608.03983. doi: 10.48550/arXiv.1608.03983.
[72] ‘Cosine Annealing In Machine Learning Simplified: Understand How It Works’, Cosine Annealing In Machine Learning Simplified: Understand How It Works. Accessed: Aug. 15, 2025. [Online]. Available: https://spotintelligence.com/2024/04/29/cosine-annealing-in-machine-learning/?utm_source=chatgpt.com
[73] ‘Mastering Cosine Annealing in Deep Learning’, Mastering Cosine Annealing in Deep Learning. Accessed: Aug. 15, 2025. [Online]. Available: https://www.numberanalytics.com/blog/cosine-annealing-deep-learning-ultimate-guide
[74] ‘Intersection over Union (IoU): Definition, Calculation, Code’, Intersection over Union (IoU): Definition, Calculation, Code. Accessed: Aug. 15, 2025. [Online]. Available: https://www.v7labs.com/blog/intersection-over-union-guide

ΠΑΡΑΡΤΗΜΑ Α : Snippets κώδικα

Τυπικός κώδικας εκτέλεσης εκπαιδεύσεων χωρίς την χρήση υπερπαραμέτρων εκτός των απαραίτητων όπως οι παρακάτω: 50 εποχές, batch size = 8 και η παράμετρος workers που κάνει πιο γρήγορη την φόρτωση εικόνων.

A1: Για το μοντέλο YOLOv11n (AdamW) default

```
!uv pip install ultralytics
!pip install ultralytics --upgrade -q
from ultralytics import YOLO
import ultralytics
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO('yolo11n.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='AdamW',
    workers = 16
)
```

Ακολουθεί το τμήμα του validation

```
# Load the model
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
metrics.box.maps
```

A2: Για το μοντέλο YOLOv11s (AdamW) default

```
!uv pip install ultralytics
!pip install ultralytics --upgrade -q
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')


torch.cuda.empty_cache()

# Load model
model = YOLO('yolo11s.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='AdamW',
    workers = 16
)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A3: Για το μοντέλο YOLOv11m (AdamW) – default

```
!uv pip install ultralytics
!pip install ultralytics --upgrade -q
from ultralytics import YOLO
import ultralytics
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO('yolo11m.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='AdamW',
    workers = 16
)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A4: Για το μοντέλο YOLOv11m (SGD) – default

```
!uv pip install ultralytics
!pip install ultralytics --upgrade -q
from ultralytics import YOLO
import ultralytics
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO('yolo11n.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='SGD',
    workers = 16
)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A5: Το μοντέλο YOLOv11s (SGD) default

```
!uv pip install ultralytics
!pip install ultralytics --upgrade -q
from ultralytics import YOLO
import ultralytics
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO('yolo11s.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='SGD',
    workers = 16
)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A6: Το μοντέλο YOLOv11m (SGD) default

```
!uv pip install ultralytics
!pip install ultralytics --upgrade -q
from ultralytics import YOLO
import ultralytics
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO('yolo11m.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 4,
    imgsz=640,
    optimizer='SGD',
    workers = 16
)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A7:Το μοντέλο YOLOv11n (Adam) default

```
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()
model = YOLO("yolo11n.pt")
!nvidia-smi
torch.cuda.empty_cache()
results = model.train(data="visdrone.yaml",
                      epochs=50,
                      batch=8,
                      optimizer="adam",
                      imgsz=640,
                      device=0,
                      plots=True,
                      seed=42)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A8: Το μοντέλο YOLOv11s (Adam) default

```
import ultralytics
import torch
ultralytics.checks()
from ultralytics import YOLO
model = YOLO("yolo11s.pt")
!nvidia-smi
torch.cuda.empty_cache()

results = model.train(data="visdrone.yaml",
                      epochs=50,
                      batch=4,
                      optimizer="adam",
                      imgsz=640,
                      device=0,
                      plots=True,
                      seed=42)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A9: Το μοντέλο YOLOv11m (Adam) default

```
!pip install ultralytics --upgrade -q
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

ultralytics.checks()

# Load model
model = YOLO('yolo11m.pt')

# Train
results = model.train(data="visdrone.yaml",
                      epochs=50,
                      batch=4,
                      optimizer="adam",
                      imgsz=640,
                      device=0,
                      plots=True,
                      seed=42)
```

```
from ultralytics import YOLO
model = YOLO("/content/runs/detect/train2/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:", metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A10: Το μοντέλο YOLOv11-OBB – default

```
!uv pip install ultralytics
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO("yolo11n-obb.pt")

results = model.train(data="/content/drive/MyDrive/DOTA.yaml", epochs=50, imgsz=1024)
```

```
import ultralytics
from ultralytics import YOLO

model = YOLO("/content/drive/MyDrive/yolo11n_val/best.pt")

metrics = model.val(data="/content/drive/MyDrive/DOTA.yaml", imgsz = 640, plots=True)
metrics.box.map50
metrics.box.map75
print(metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
```

Ακολουθεί τμήμα με τα παραμετροποιημένα μοντέλα

A11: YOLOv11m – SGD v1

```
!pip install ultralytics --upgrade -q
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')
torch.cuda.empty_cache()


# Load model
model = YOLO('yolo11m.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='SGD',
    multi_scale = True,
    scale = 0.37,
    freeze = 10,
    cos_lr = True,
    workers = 16
)
```

```
from ultralytics import YOLO

# Load a model
#model = YOLO("yolo11n.pt")  # load an official model
model = YOLO("/content/runs/detect/train/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:",metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A12: Το μοντέλο YOLOv11m – SGDv2

```
!pip install ultralytics --upgrade -q
import ultralytics
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

# Load model
model = YOLO('yolo11m.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 16,
    imgsz=640,
    optimizer='SGD',
    multi_scale = True,
    scale = 0.37,
    mosaic = True,
    cos_lr = True,
    workers = 16
)
```

```
from ultralytics import YOLO

model = YOLO("/content/runs/detect/train/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:",metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A13: Το μοντέλο YOLOv11m – AdamW v1

```
!pip install ultralytics --upgrade -q
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

# Load model
model = YOLO('yolo11m.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 8,
    imgsz=640,
    optimizer='AdamW',
    momentum=0.9,
    lr0 = 0.001,
    weight_decay=0.0005,
    workers = 16,
    mosaic=0.32,
    copy_paste=0.15,
    multi_scale = True,
    scale = 0.5,
    verbose = True,
    save = True,
    val = True,
    plots = True,
    save_period = 1,
)
```

```
from ultralytics import YOLO

model = YOLO("/content/runs/detect/train/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:",metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A14: Το μοντέλο YOLOv11 – SGD v3

```
!pip install ultralytics --upgrade -q
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

torch.cuda.empty_cache()

model = YOLO('yolo11m.pt')

# Train
model.train(
    data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 16,
    imgsz=640,
    optimizer='SGD',
    multi_scale = True,
    scale = 0.37,
    mosaic = True,
    translate = 0.20,
    shear = 5,
    cos_lr = True,
    workers = 16
)
```

```
from ultralytics import YOLO

model = YOLO("/content/runs/detect/train/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:",metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
metrics.to_csv('metrics.csv')
```

A15: Το μοντέλο RT-DETR-l v1

```
!pip install ultralytics --upgrade -q
import ultralytics
from ultralytics import RTDETR
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

model = RTDETR("rtdetr-l.pt")
model.info()

results = model.train(data='/content/drive/MyDrive/Model/datasets/visdrone.yaml',
    epochs=50,
    batch = 16,
    imgsz=640,
    lr0=0.0015,
    cos_lr = True,
    hsv_h = 0.017,
    degrees = 3,
    translate = 0.3,
    #multi_scale = True,
    #scale = 0.35,
    optimizer='AdamW',
    workers = 16)
```

```
from ultralytics import YOLO

model = YOLO("/content/runs/detect/train/weights/best.pt")

# Validate the model
metrics = model.val()
metrics.box.map
metrics.box.map50
metrics.box.map75
print("mAP75 is:",metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
results = model.val()
detection_summary = results.summary()
print(detection_summary)
metrics.to_csv('metrics.csv')
```

Α16: Το μοντέλο YOLOv11m-OBB με χρήση υπερπαραμέτρων

```
!uv pip install ultralytics
import ultralytics
from ultralytics import YOLO
import torch
ultralytics.checks()

from google.colab import drive
drive.mount('/content/drive')

torch.cuda.empty_cache()

model = YOLO("yolo11m-obb.pt")

results = model.train(data="/content/drive/MyDrive/DOTA.yaml",
    epochs=50,
    batch = 16,
    imgsz=1024,
    optimizer='SGD',
    hsv_v = 0.15,
    mosaic = True,
    cos_lr = True,
    seed = 42,
    workers = 16)
```

```
import ultralytics
from ultralytics import YOLO

model = YOLO("/content/drive/MyDrive/yolo11n_val/best.pt")

metrics = model.val(data="/content/drive/MyDrive/DOTA.yaml", imgsz = 640, plots=True)
metrics.box.map50
metrics.box.map75
print(metrics.box.map75)
metrics.box.maps
metrics.box.f1
metrics.results_dict
metrics.summary
```

Το απόσπασμα από τα αρχείο yaml για το VisDrone (visdrone.yaml)

```
# Example usage: yolo train data=VisDrone.yaml
# parent
# ├── ultralytics
# └── datasets
#     └── VisDrone  ← downloads here (2.3 GB)

# Train/val/test sets as dir: path/to/imgs
path: 'C:\\Users\\...\\datasets\\VisDrone' # dataset root dir
train: 'C:\\Users\\...\\datasets\\VisDrone\\VisDrone2019-DET-train\\images' # train images
val: 'C:\\Users\\...\\datasets\\VisDrone\\VisDrone2019-DET-val\\images' # val images
test: 'C:\\Users\\...\\datasets\\VisDrone\\VisDrone2019-DET-test-dev\\images' # test
cache: images

nc: 10

# Classes
names:
  0: pedestrian
  1: people
  2: bicycle
  3: car
  4: van
  5: truck
  6: tricycle
  7: awning-tricycle
  8: bus
  9: motor
```

Ο κώδικα μετατροπής των labels σε μορφή συμβατή με το format του YOLO

```python
import os
from pathlib import Path
import shutil

from ultralytics.utils.downloads import download
from ultralytics.utils import TQDM


def visdrone2yolo(dir, split, source_name=None):
    """Convert VisDrone annotations to YOLO format with images/{split} and labels/{split} structure."""
    from PIL import Image

    source_dir = dir / (source_name or f"VisDrone2019-DET-{split}")
    images_dir = dir / "images" / split
    labels_dir = dir / "labels" / split
    labels_dir.mkdir(parents=True, exist_ok=True)

    # Move images to new structure
    if (source_images_dir := source_dir / "images").exists():
        images_dir.mkdir(parents=True, exist_ok=True)
        for img in source_images_dir.glob("*.jpg"):
            img.rename(images_dir / img.name)

    for f in TQDM((source_dir / "annotations").glob("*.txt"), desc=f"Converting {split}"):
        img_size = Image.open(images_dir / f.with_suffix(".jpg").name).size
        dw, dh = 1.0 / img_size[0], 1.0 / img_size[1]
        lines = []

        with open(f, encoding="utf-8") as file:
            for row in [x.split(",") for x in file.read().strip().splitlines()]:
                if row[4] != "0":  # Skip ignored regions
                    x, y, w, h = map(int, row[:4])
                    cls = int(row[5]) - 1
                    # Convert to YOLO format
                    x_center, y_center = (x + w / 2) * dw, (y + h / 2) * dh
                    w_norm, h_norm = w * dw, h * dh
                    lines.append(f"{cls} {x_center:.6f} {y_center:.6f} {w_norm:.6f} {h_norm:.6f}\n")

        (labels_dir / f.name).write_text("".join(lines), encoding="utf-8")


# Convert
splits = {"VisDrone2019-DET-train": "train", "VisDrone2019-DET-val": "val", "VisDrone2019-DET-test-dev": "test"}
for folder, split in splits.items():
    visdrone2yolo(dir, split, folder)
    shutil.rmtree(dir / folder)
```

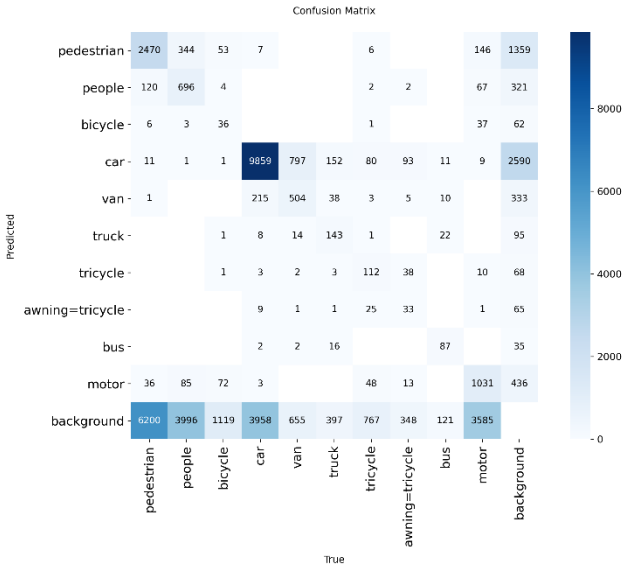
Το απόσπασμα από τα αρχείο yaml για το DOTA (dota.yaml)

```yaml
# Train/val/test sets as dir: path/to/imgs
path: "/content/drive/MyDrive/DOTA_Dataset/" # dataset root dir
train: "/content/drive/MyDrive/DOTA_Dataset/images/train/" # train images
val: "/content/drive/MyDrive/DOTA_Dataset/images/val/" # val images
test: "/content/drive/MyDrive/DOTA_Dataset/images/test/" # test images

nc: 16

# Classes for DOTA 1.5
names:
  0: plane
  1: ship
  2: storage tank
  3: baseball diamond
  4: tennis court
  5: basketball court
  6: ground track field
  7: harbor
  8: bridge
  9: large vehicle
  10: small vehicle
  11: helicopter
  12: roundabout
  13: soccer ball field
  14: swimming pool
  15: container crane
```

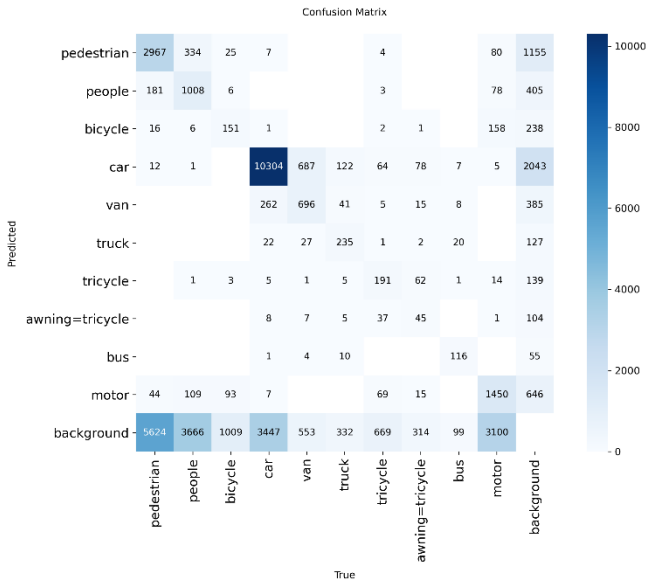
ΠΑΡΑΡΤΗΜΑ Β : Αποτελέσματα εκπαιδεύσεων

Παρατίθενται οι πίνακες σύγχυσης (confusion matrix) όπως προέκυψαν από την διαδικασία του validation

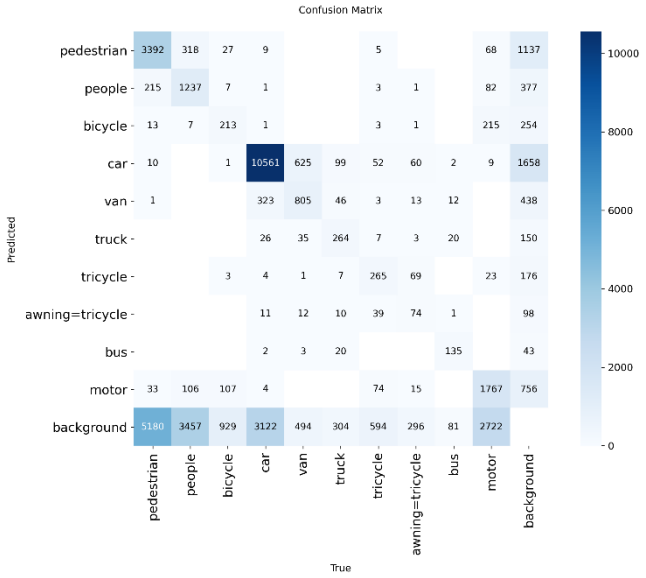
B1: Το μοντέλο YOLOv11n (AdamW)



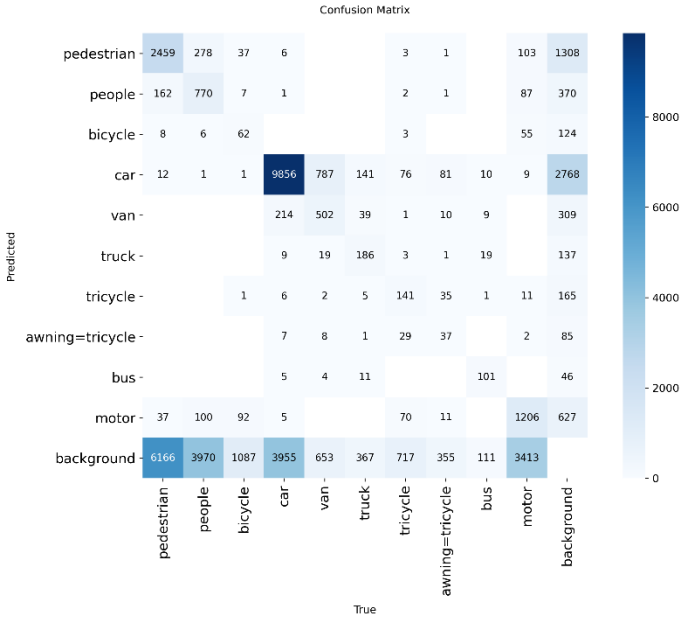
B2: Το μοντέλο YOLOv11s (AdamW)



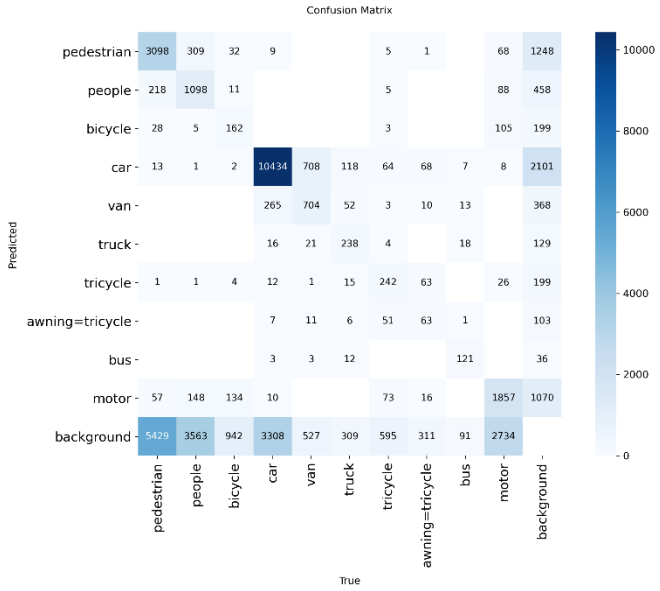
B3: Το μοντέλο YOLOv11m (AdamW)



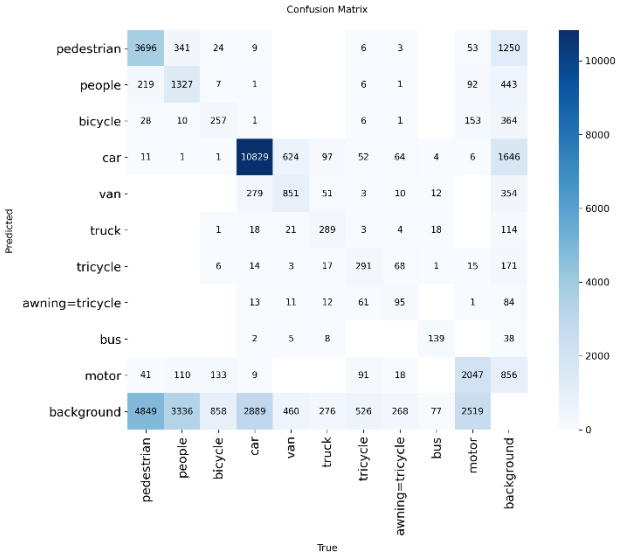
B4: Το μοντέλο YOLOv11n (SGD)



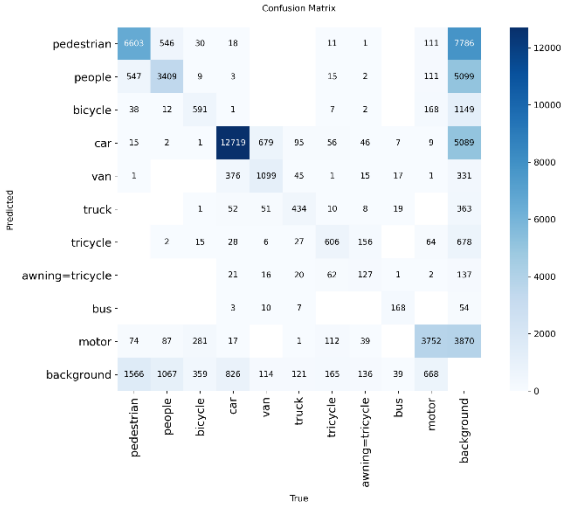
B5: Το μοντέλο YOLOv11s (SGD)



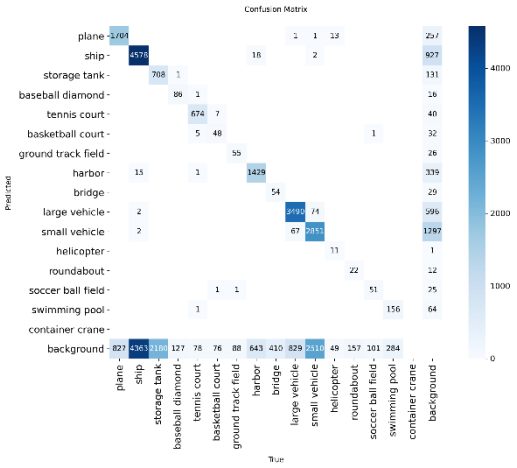
B6: Το μοντέλο YOLOv11m (SGD)



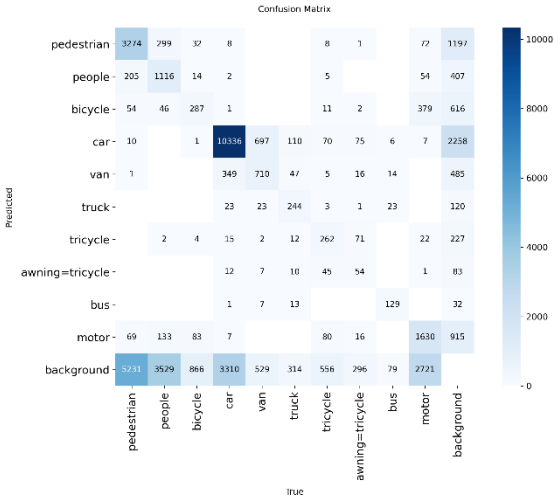
B7: Το μοντέλο RT-DETR-l



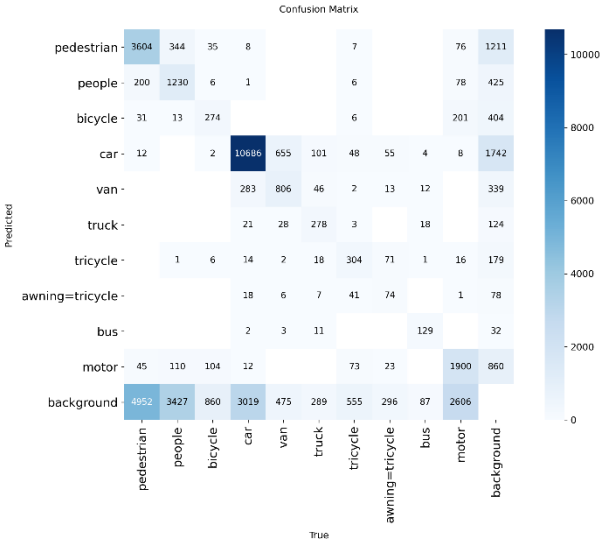
B8: Το μοντέλο YOLOv11n-OBB



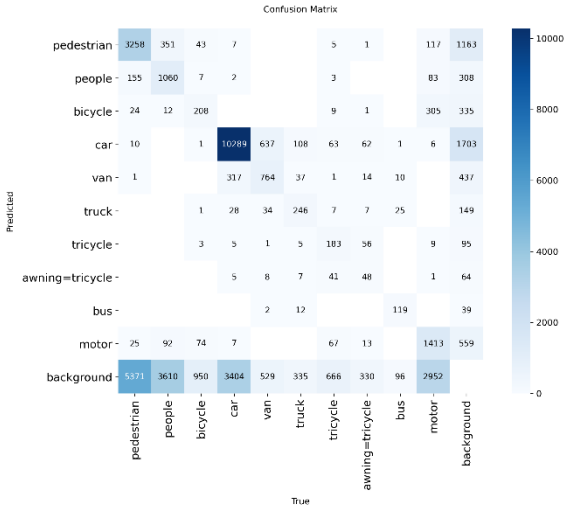
B9 Το μοντέλο YOLOv11m SGD v1



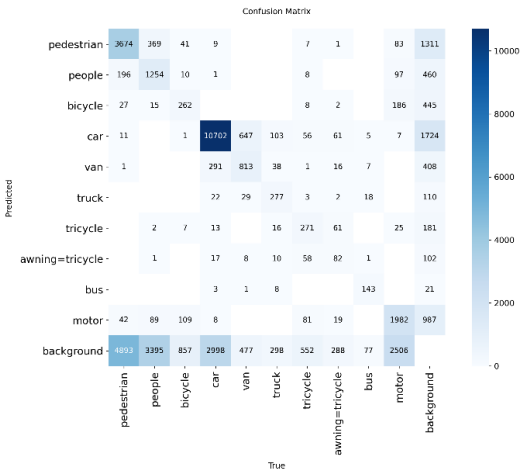
B10: Το μοντέλο YOLOv11m SGD v2



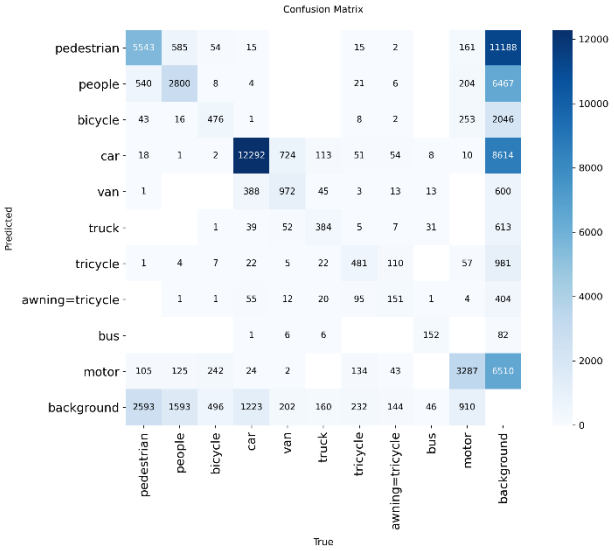
B11: Το μοντέλο YOLOv11m AdamW v1



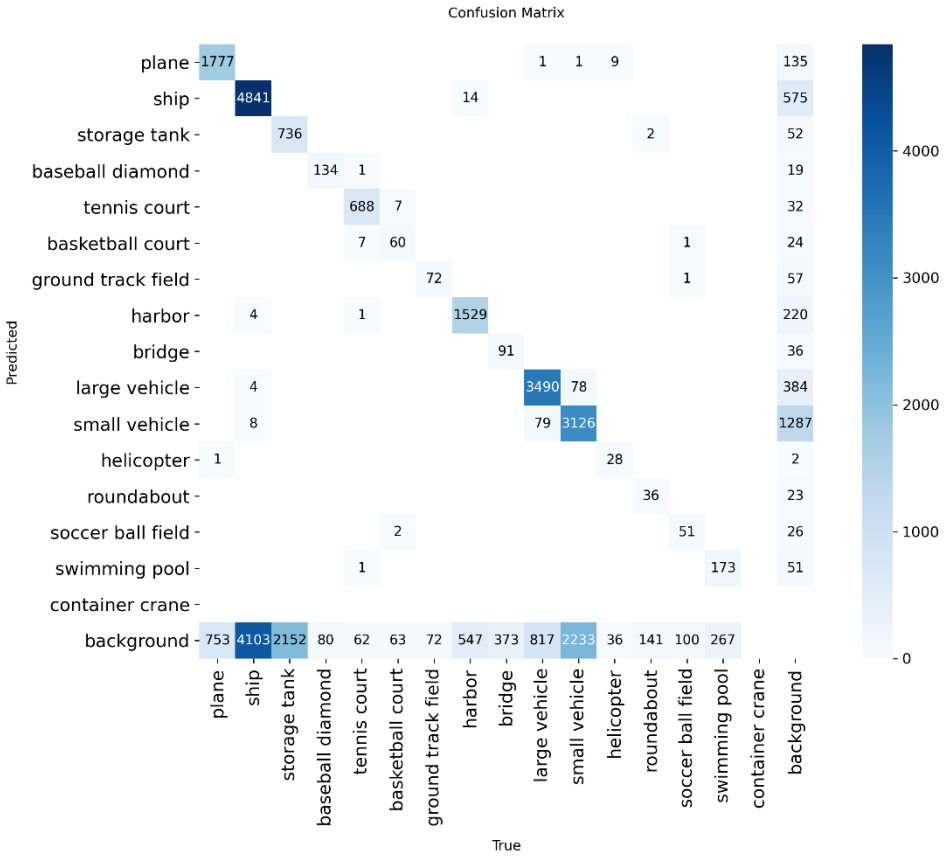
B12: Το μοντέλο YOLOv11m SGD v3



B13: Το μοντέλο RT-DETR-l v1



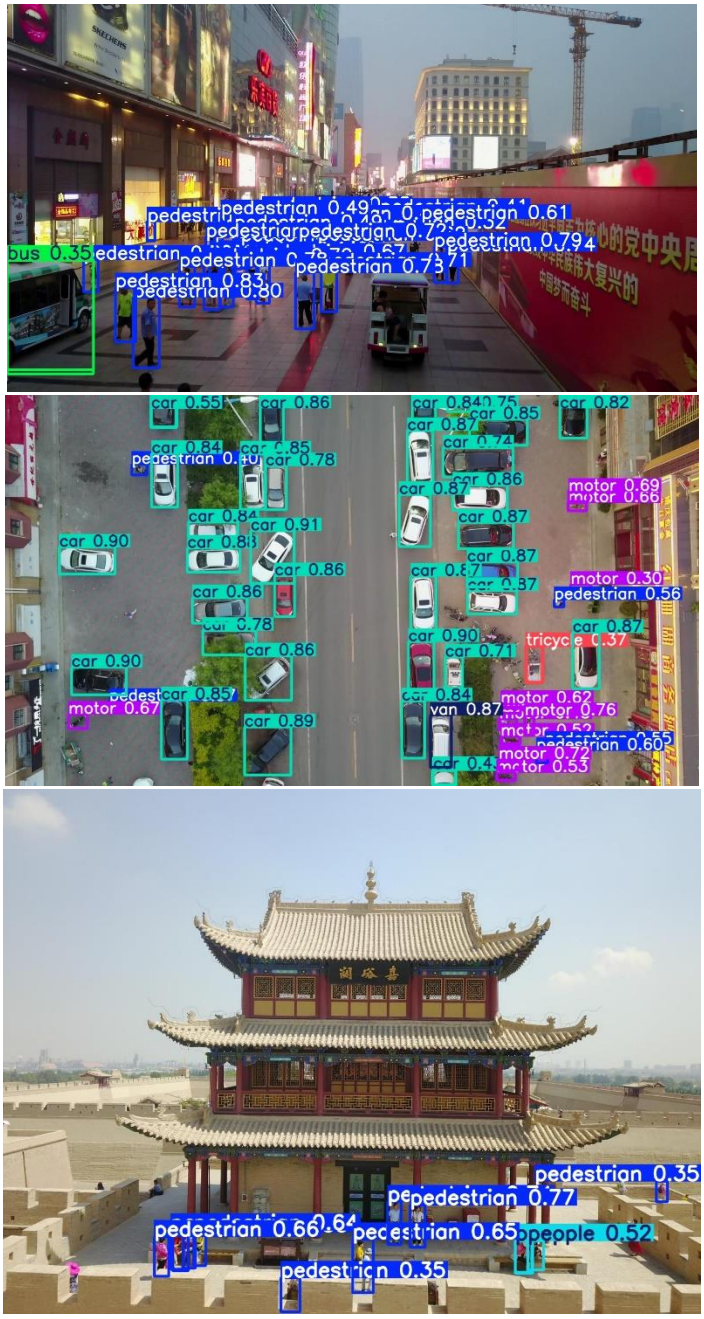
B14: Το μοντέλο YOLOv11m-OBB v1



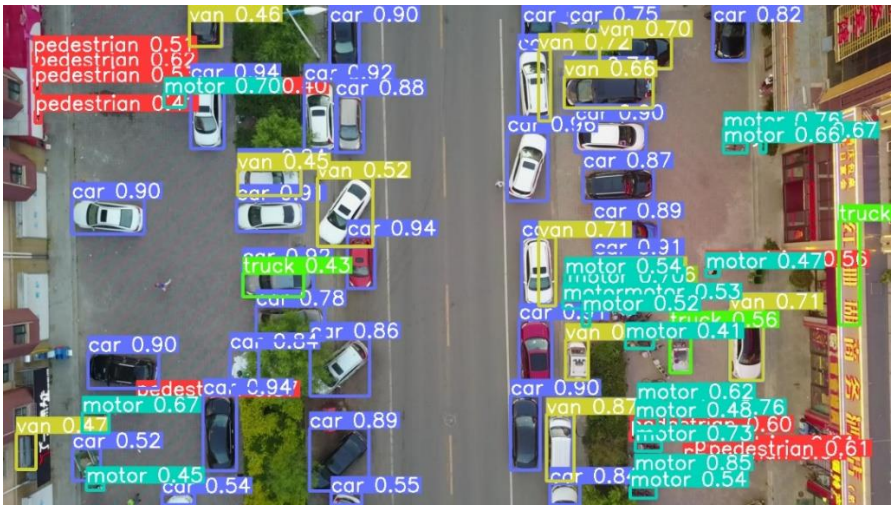
ΠΑΡΑΡΤΗΜΑ C : Παραδείγματα inference με τα μοντέλα YOLOv11m, YOLOv11m-SGDv2, YOLOv11m-OBB

Παράδειγμα προβλέψεων με το μοντέλο YOLOv11s

Χωρίς την χρήση SAHI

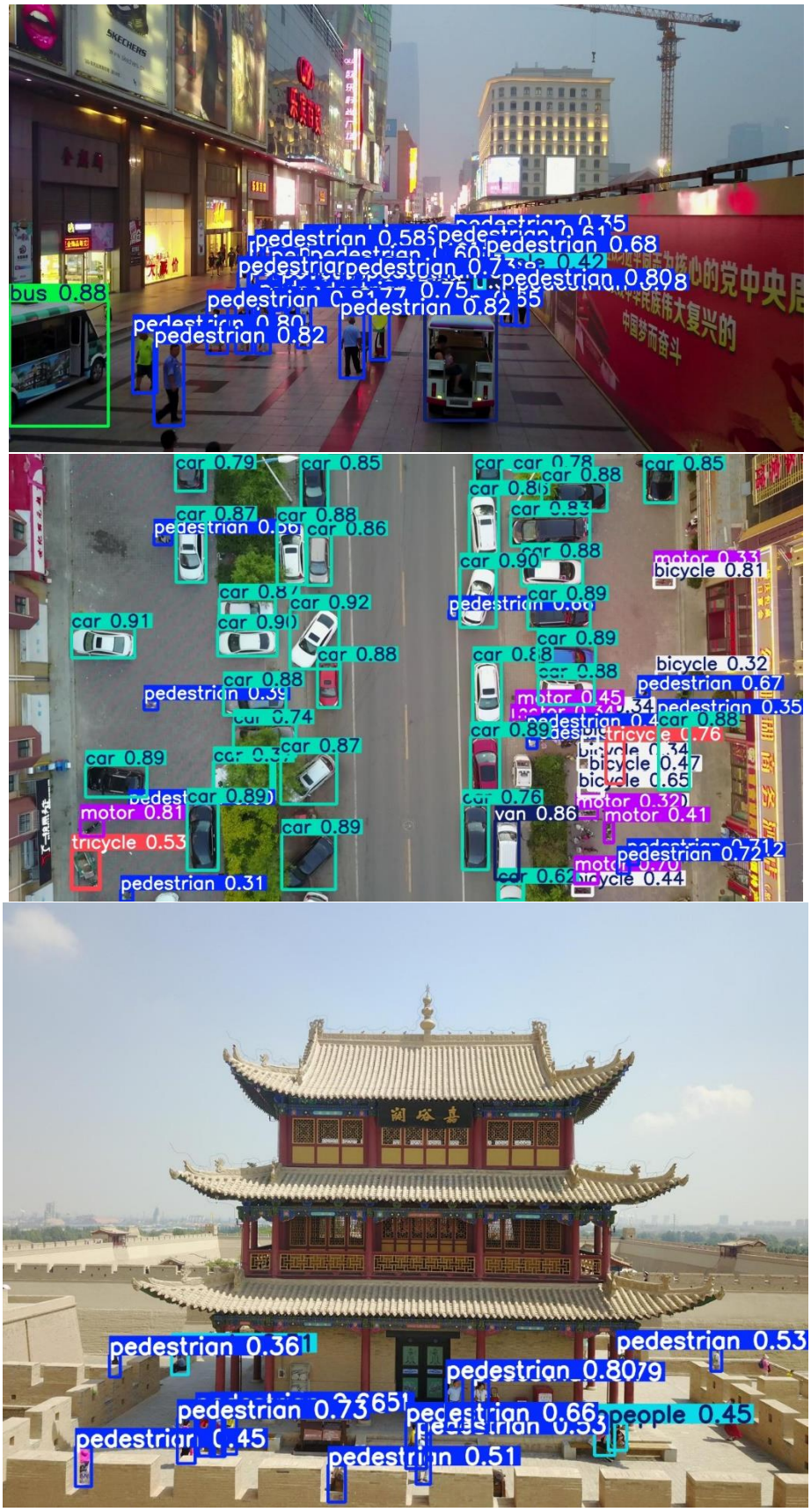


Με την χρήση SAHI

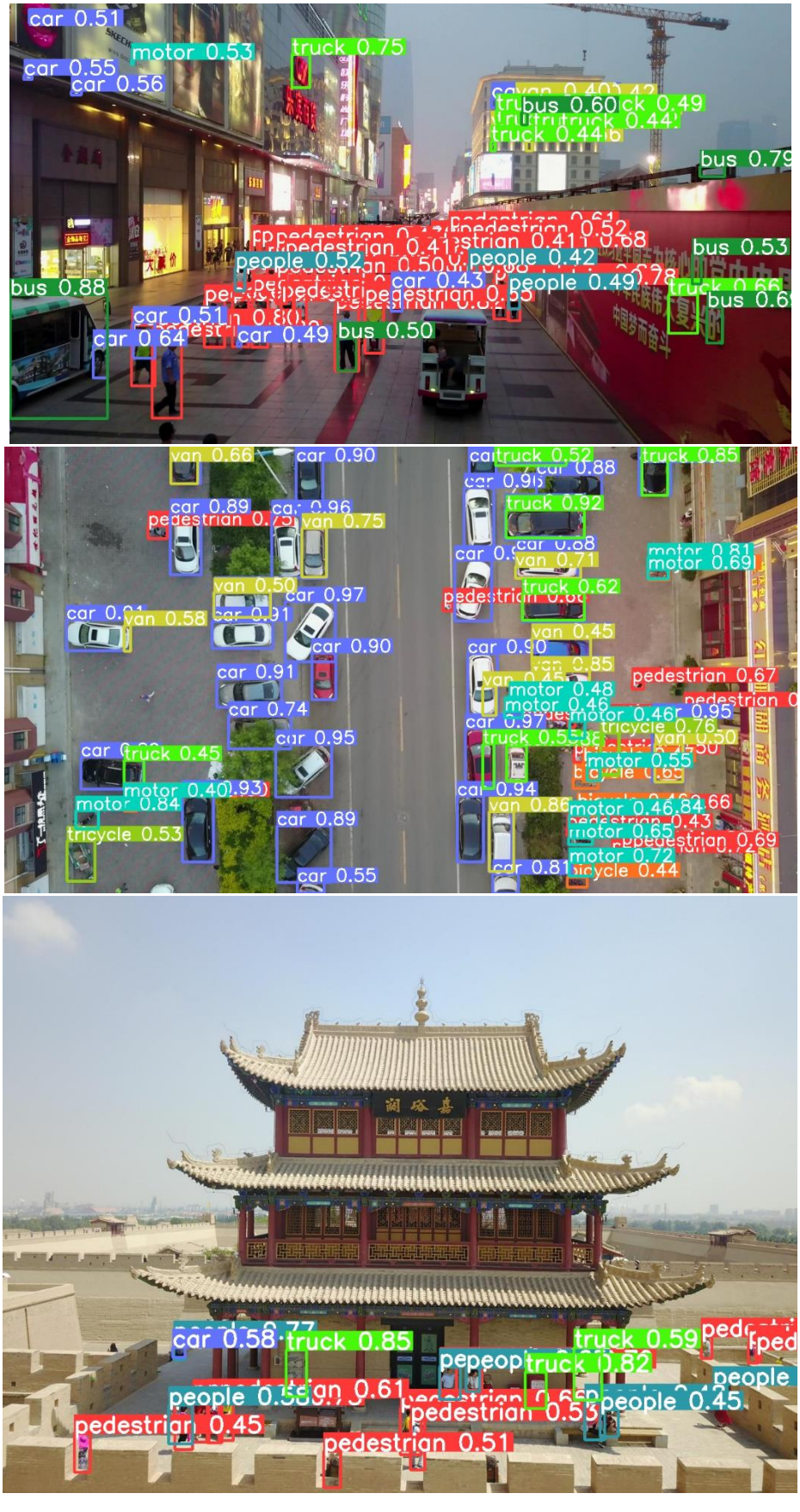


Παράδειγμα προβλέψεων με το μοντέλο YOLOv11m-SGDv2

Χωρίς την χρήση SAHI



Με την χρήση SAHI



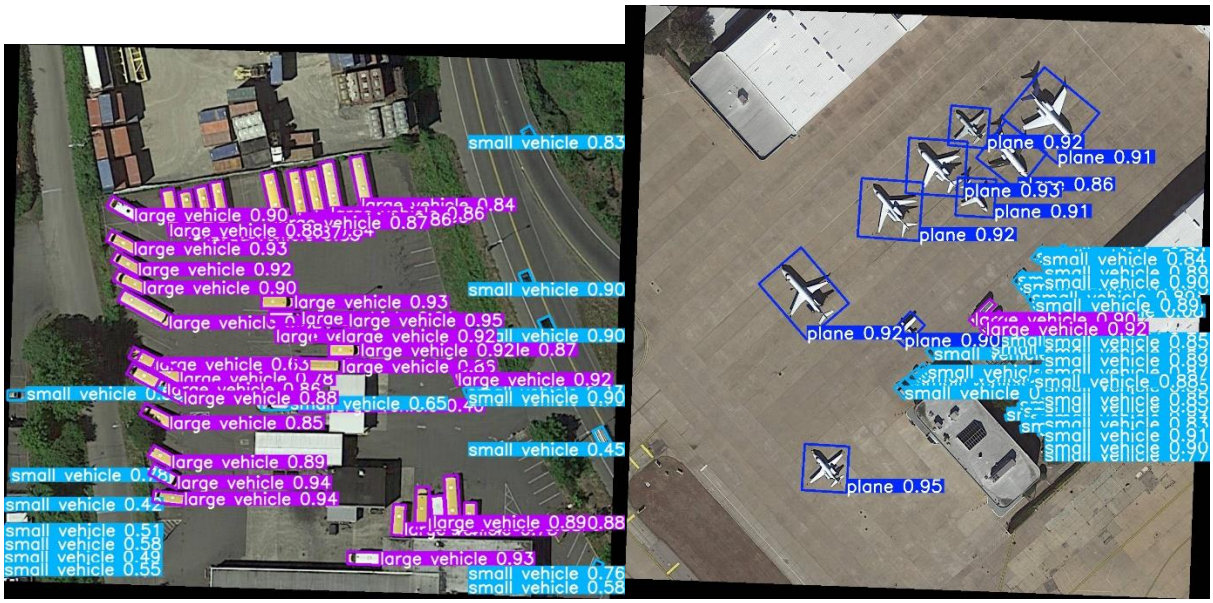
Παράδειγμα προβλέψεων με το μοντέλο RT-DETR-l

Χωρίς την χρήση SAHI

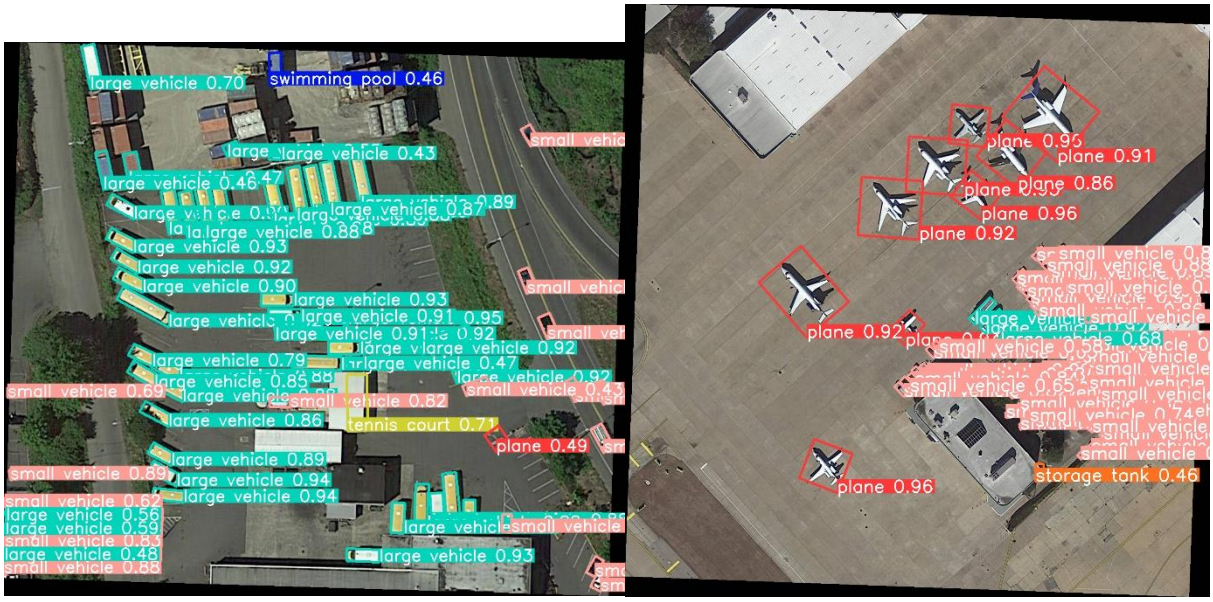
Με την χρήση SAHI

Παράδειγμα προβλέψεων με το μοντέλο YOLOv11n-OBB

Χωρίς την χρήση SAHI

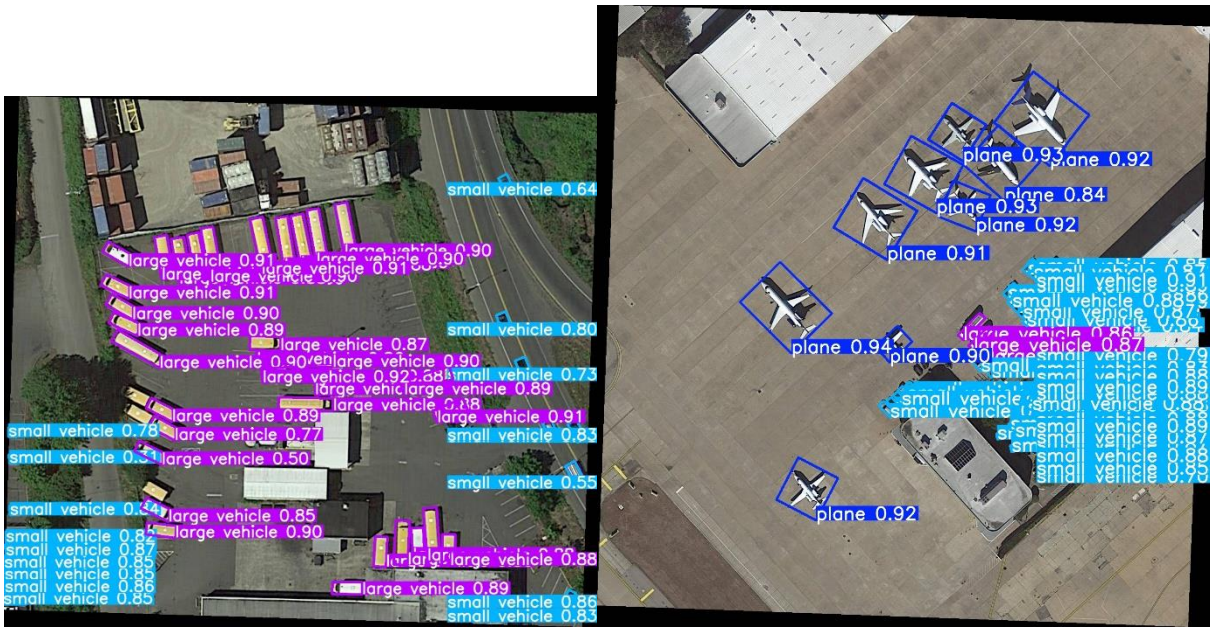


Με την χρήση SAHI



Παράδειγμα προβλέψεων με το μοντέλο YOLOv11m-OBBv1

Χωρίς την χρήση SAHI



Με την χρήση SAHI