ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ & ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΜΣ: ΓΕΩΧΩΡΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

Μεταπτυχιακή Διπλωματική Εργασία

ΟΠΤΙΚΕΣ ΜΕΘΟΔΟΙ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΑΡΙΘΜΟΥ ΠΙΝΑΚΙΔΑΣ ΟΧΗΜΑΤΩΝ

ΒΛΑΜΙΔΗΣ Κωνσταντίνος
ΑΜ: 1805

Επιβλέποντες καθηγητές:
Γραμματικόπουλος Λάζαρος, Αναπληρωτής Καθηγητής ΠΑΔΑ
Πέτσα Έλλη, Καθηγήτρια ΠΑΔΑ
Τσάτσαρης Ανδρέας, Καθηγητής ΠΑΔΑ

Αθήνα, Σεπτέμβριος 2024

UNIVERSITY OF WEST ATTICA
FACULY OF ENGINEERING
DEPARTMENT OF SURVEYING AND GEOINFORMATICS ENGINEERING
MSC: GEOSPATIAL TECHNOLOGIES

Diploma thesis

VISUAL METHODS FOR THE DETECTION AND RECOGNITION OF VEHICLE LICENSE PLATES

VLAMIDIS Konstantinos
Registration Number: 1805

Superior name and surname:
Grammatikopoulos Lazaros, Associate Professor UNIWA
Petsa Elli, Professor UNIWA
Tsatsaris Andreas, Professor UNIWA

Athens, September 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΟΠΟΓΡΑΦΙΑΣ & ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΜΣ: ΓΕΩΧΩΡΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

ΟΠΤΙΚΕΣ ΜΕΘΟΔΟΙ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΑΡΙΘΜΟΥ ΠΙΝΑΚΙΔΑΣ ΟΧΗΜΑΤΩΝ

Μέλη Εξεταστικής Επιτροπής συμπεριλαμβανομένου και του Εισηγητή

Η μεταπτυχιακή διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή:

Α/α	ΟΝΟΜΑ ΕΠΩΝΥΜΟ	ΒΑΘΜΙΔΑ/ΙΔΙΟΤΗΤΑ	ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ
1	Λάζαρος Γραμματικόπουλος (επιβλέπων)	Αναπληρωτής Καθηγητής ΠΑΔΑ	LAZAROS GRAMMATIKOPOULOS Digitally signed by LAZAROS GRAMMATIKOPOULOS Date: 2024.09.27 13:52:23 +03'00'
2	Ελένη Πέτσα	Καθηγήτρια ΠΑΔΑ	ELENI PETSA Digitally signed by ELENI PETSA Date: 2024.09.27 14:10:13 +03'00'
3	Ανδρέας Τσάτσαρης	Καθηγητής ΠΑΔΑ	Andreas Tsatsaris Digitally signed by Andreas Tsatsaris Date: 2024.09.27 14:22:34 +03'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος ΒΛΑΜΙΔΗΣ Κωνσταντίνος του Γεωργίου, με αριθμό μητρώου 1805, φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών «Γεωχωρικές Τεχνολογίες» του Τμήματος Μηχανικών Τοπογραφίας και Γεωπληροφορικής της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από εμένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας, τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο ΔΗΛΩΝ

Ευχαριστίες

Η διπλωματική αυτή εργασία, σηματοδοτεί την ολοκλήρωση των μεταπτυχιακών μου σπουδών στην Γεωπληροφορική. Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου, Λάζαρο Γραμματικόπουλο, για τη βοήθεια, την ενθάρρυνση και τη συνεργασία του, όχι μόνο κατά την εκπόνηση της διπλωματικής εργασίας, αλλά καθ' όλη τη διάρκεια των μεταπτυχιακών μου σπουδών. Ένα ακόμα μεγάλο ευχαριστώ, στην σύζυγό μου, για την αμέριστη συμπαράσταση, κυρίως τους τελευταίους αυτούς μήνες, που με κάθε τρόπο ήταν δίπλα μου. Χωρίς αυτήν δεν θα είχα φτάσει εδώ που βρίσκομαι σήμερα.

Στην σύζυγό μου…

ΠΕΡΙΛΗΨΗ

Η διπλωματική αυτή εργασία με τίτλο «Μέθοδος και Τεχνική Ανίχνευσης και Αναγνώρισης Αριθμού Κυκλοφορίας Οχήματος» εκπονήθηκε στο πλαίσιο του Μεταπτυχιακού Προγράμματος «Γεωχωρικές Τεχνολογίες» του Τμήματος Μηχανικών Τοπογραφίας και Γεωπληροφορικής του Πανεπιστημίου Δυτικής Αττικής. Παρουσιάζει την όραση των υπολογιστών, που σε συνδυασμό με τη χρήση μεθόδων μηχανικής μάθησης, μπορούν να δημιουργήσουν εφαρμογές που αναγνωρίζουν μοτίβα, όπως πινακίδες οχημάτων. Αρχικά, γίνεται μία ιστορική αναδρομή στις επιστήμες της υπολογιστικής όρασης και της μηχανικής μάθησης, όπως και στην αναγνώριση αντικειμένων. Εν συνεχεία, περιγράφεται ο αλγόριθμος που αναπτύχθηκε και τα στάδια υλοποίησης του. Τέλος, ακολουθεί η εφαρμογή του, μέσω ενός από τους state-of-art αλγορίθμους, του αλγορίθμου YOLOv4, ο οποίος εκπαιδεύτηκε σε ένα σύνολο δεδομένων, όπου ανακτήθηκε από ανοικτές πηγές διαδικτύου και εφαρμόστηκε σε πραγματικά δεδομένα που ελήφθησαν προς τον σκοπό αυτό, από Μη Επανδρωμένο Αεροσκάφος (Μ.Ε.Α).

Λέξεις Κλειδιά: Όραση Υπολογιστών, Μηχανική Μάθηση, Ανίχνευση Αντικειμένου, Εντοπισμός, Αναγνώριση, Ταξινόμηση, Βαθιά Μάθηση, Συνελικτικά Νευρωνικά Δίκτυα, Αλγόριθμος YOLO, Αριθμός Κυκλοφορίας Οχήματος, Πινακίδες Οχήματος, Μη Επανδρωμένο Αεροσκάφος, Αεροφωτογραφία

ABSTRACT

The Master Thesis “Visual Methods for the Detection and Recognition of Vehicle License Plates” was conducted in the framework of the Postgraduate Program “Geospatial Technologies” of the Department of Surveying and Geoinformatics Engineering of the University of West Attica. Initially, there is a throwback to the disciplines of computer vision and machine learning, in addition to the recognition of certain objects such as vehicle registration plate. Then, there is a deeper analysis in computer vision, by studying contour finding and convex hull algorithms, and the way they are improved through image processing. Finally, a certain state-of-art algorithm, YOLO v4, is being presented, which is the subject of the bachelor’s thesis. For the trained of it, images from open-source database, have been downloaded and the applicate being tested in real dataset, obtained for this purpose through an Unmanned Aerial Vehicle (UAV).

Keywords: Computer Vision, Machine Learning, Object Detection, Localization, Recognition, Classification, Deep Learning, Convolutional Neural Networks, YOLO algorithm, Vehicle Registration Plate, Unmanned Aerial Vehicle, aerial photography

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ 12
1.1. ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΑΠΚ 12
1.2. Η ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΑΠΚ 13
1.3. ΣΤΟΧΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ 13
1.4. ΔΟΜΗ 13
2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ 14
2.1. ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ 14
2.1.1. Αλγόριθμοι Machine Learning 15
2.1.1.1. Viola & Jones Detector 15
2.1.1.2. HOG Detector 16
2.1.1.3. DPM Detector 16
2.1.2. Αλγόριθμοι Deep Learning 17
2.1.2.1. Two-stage Network 17
2.1.2.1.1. R-CNN 18
2.1.2.1.2. Fast R-CNN 20
2.1.2.1.3. Faster R-CNN 21
2.1.2.2. One-stage Network 22
2.1.2.2.1. SSD 23
2.1.2.2.2. YOLO 24
2.2. ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ 27
2.2.1. Μηχανή Tesseract 28
2.2.2. Αρχιτεκτονική Tesseract 28
3. ΜΕΘΟΔΟΛΟΓΙΑ 29
3.1. ΜΕΘΟΔΟΛΟΓΙΑ MACHINE LEARNING 30
3.2. ΜΕΘΟΔΟΛΟΓΙΑ DEEP LEARNING 31
3.3. ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΓΑΣΙΑΣ 31
4. ΥΛΟΠΟΙΗΣΗ 34
4.1. ΣΥΛΛΟΓΗ ΥΛΙΚΟΥ 34
4.2. ΕΚΠΑΙΔΕΥΣΗ ΜΟΝΤΕΛΟΥ 35
4.3. ΑΡΧΙΚΟΠΟΙΗΣΗ YOLOv4 ΜΕ TENSORFLOW 38
4.4. ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ YOLOv4 41
4.5. ΔΟΚΙΜΕΣ ΣΕ ΕΙΚΟΝΕΣ ΚΑΙ ΒΙΝΤΕΟ ΔΙΑΔΙΚΤΥΟΥ 42

4.6. ΔΟΚΙΜΕΣ ΑΠΟ ΣμηΕΑ 50
5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ 53
5.1. ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ 53
5.2. ΠΙΘΑΝΕΣ ΒΕΛΤΙΩΣΕΙΣ 55
6. ΒΙΒΛΙΟΓΡΑΦΙΑ 56

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Φάσεις λειτουργίας του R-CNN 19
Εικόνα 2. Δομή του R-CNN 19
Εικόνα 3. Φάσεις λειτουργίας του Fast R-CNN 20
Εικόνα 4. Σύγκριση R-CNN με Fast R-CNN 21
Εικόνα 5. Δομή του Faster R-CNN 22
Εικόνα 6. Σύγκριση R-CNN, Fast R-CNN, Faster R-CNN 22
Εικόνα 7.Αρχιτεκτονική δικτύου VGG-16 23
Εικόνα 8. Bounding box του SSD 24
Εικόνα 9. Πειραματισμοί SSD με Two-stage Networks (mAP) 24
Εικόνα 10. Αρχιτεκτονική αλγορίθμου YOLO 25
Εικόνα 11. Κίτρινο πλαίσιο «υπεύθυνο» για τον εντοπισμό του αντικειμένου 25
Εικόνα 12. Συνοπτική αναπαράσταση λειτουργίας αλγορίθμου YOLO 27
Εικόνα 13. Δομή της μηχανής Tesseract 29
Εικόνα 14. Ροής εργασιών επίλυσης ενός προβλήματος ALPR 32
Εικόνα 15. Επισκόπηση της λογικής των ερευνητών Laroca, Rayson et al 33
Εικόνα 16. Αποτελέσματα ανίχνευσης αντικειμένων σε εικόνες μέσω YOLOv4 38
Εικόνα 17. Αποτελέσματα ανίχνευσης αντικειμένων σε βίντεο μέσω YOLOv4 39
Εικόνα 18. Αναγνώριση πινακίδων κυκλοφορίας 40
Εικόνα 19. Δοκιμές αλγορίθμου σε εικόνες 43
Εικόνα 20. Δοκιμές στις εικόνες αξιολόγησης, αναγνώρισης χαρακτήρων 47
Εικόνα 21. Σύνολο αποτελεσμάτων αναγνώρισης χαρακτήρων 48
Εικόνα 22. Δοκιμές αλγορίθμου σε βίντεο 50
Εικόνα 23. DJI MAVIC 2 Enterprise Zoom 50

1. ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια παρατηρείται ένα συνεχώς αυξανόμενο ενδιαφέρον για τις τεχνικές βαθιάς εκμάθησης σε διάφορες εφαρμογές, όπως η επεξεργασία ηχητικών και οπτικών ψηφιακών δεδομένων. Ο μεγάλος όγκος αυτών των δεδομένων (big data), σε συνδυασμό με τη ραγδαία ανάπτυξη των υπολογιστικών συστημάτων και την ανάγκη για αυτοματοποιημένα και ποιοτικά αποτελέσματα στον ελάχιστο δυνατό χρόνο, έχει αποτελέσει το έναυσμα για την ανάπτυξη και χρήση πολύπλοκων αλγορίθμων. Στην κατηγορία αυτή ανήκουν οι τεχνικές βαθιάς μηχανικής μάθησης, με κυριότερη αυτή των νευρωνικών δικτύων.

Στον κλάδο της όρασης των υπολογιστών και της αναγνώρισης και ταξινόμησης εικόνων, έχει αποδειχθεί από την επιστημονική κοινότητα ότι τα συνεχώς αναπτυσσόμενα δίκτυα εμφανίζουν ολοένα και καλύτερες αποδόσεις, ξεπερνώντας ακόμα και τον άνθρωπο. Η αυτόματη ανίχνευση και αναγνώριση αντικειμένων έχει μετατραπεί σε πραγματική πρόκληση, καθώς η ανάλυση των δεδομένων δεν περιορίζεται πλέον από τη διακριτική ικανότητα του ανθρώπινου ματιού ή την ταχύτητα επεξεργασίας του εγκεφάλου. Ο άνθρωπος λαμβάνει την τελική στοχευμένη πληροφορία, με αποτέλεσμα η δράση του να περιορίζεται μόνο στην ποιοτική αξιοποίησή της.

Η παρούσα εργασία καταπιάνεται με την δημιουργία μίας βάσης δεδομένων, με σκοπό την αυτοματοποίηση της διαδικασίας εντοπισμού και αναγνώρισης των πινακίδων κυκλοφορίας (ΑΠΚ) των οχημάτων, μέσω εναέριας κάμερας από ΣμηΕΑ.

1.1. ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΑΠΚ

Η χρήση εντός τέτοιου συστήματος ποικίλει, ιδιαίτερα σε πυκνο-κατοικημένες περιοχές όπου ο αριθμός των οχημάτων είναι συνεχώς αυξανόμενος. Μπορεί να υποβοηθήσει το έργο της Ελληνικής Αστυνομίας, τόσο στην διαχείριση της κυκλοφορίας και την προσαύξησης της ασφάλειας στους δρόμους, όσο και στην διαλεύκανση εγκληματικών ενεργειών. Τέλος, η αναγνώριση αριθμών κυκλοφορίας με χρήση ΣμηΕΑ, δίνει την δυνατότητα επιτήρησης και σάρωσης μεγάλων περιοχών από αέρα, ενώ παράλληλα

παρέχει την δυνατότητα συνεχής μετατόπισης της θέσης του οπτικού αισθητήρα καθώς αυτός είναι προσαρτημένος στο ιπτάμενο μέσο.

1.2. Η ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΑΠΚ

Ένα σύστημα αναγνώρισης πινακίδων κυκλοφορίας αποτελείται από τρία κύρια μέρη: την ανίχνευση της πινακίδας μέσα στην εικόνα, τον διαχωρισμό των χαρακτήρων της πινακίδας και την αναγνώρισή τους. Τα δύο τελευταία στάδια, δηλαδή ο διαχωρισμός και η αναγνώριση των χαρακτήρων, είναι πεδία με πολλές εφαρμογές, όπου η έρευνα έχει προχωρήσει με μεγάλο βαθμό επιτυχίας. Ωστόσο, η ανίχνευση της πινακίδας σε μη ελεγχόμενο περιβάλλον, και ιδιαίτερα από ΣμηΕΑ, αποτελεί το δυσκολότερο μέρος αυτής της διαδικασίας.

1.3. ΣΤΟΧΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Είναι η εξ' ολοκλήρου δημιουργία ενός συστήματος ΑΠΚ ώστε να δίνει ικανοποιητικά αποτελέσματα σε χρόνους εκτέλεσης κατάλληλους για εφαρμογές πραγματικού χρόνου. Ιδιαίτερη βάση δόθηκε στην ανίχνευση της θέσης της πινακίδας και στην εκπαίδευση κατάλληλου αλγορίθμου, μιας και αυτό είναι το βασικότερο πρόβλημα στην διαδικασία αναγνώρισης της. Η υλοποίηση έγινε σε γλώσσα προγραμματισμού Python με χρήση της βιβλιοθήκης OpenCV, με κάποιες τροποποιήσεις υπαρχουσών συναρτήσεων και μεθόδων, ώστε να είναι δυνατή η υλοποίηση του αλγορίθμου. Για το κομμάτι της οπτικής αναγνώρισης των χαρακτήρων επιχειρήθηκε η χρήση έτοιμης και ανοιχτής βιβλιοθήκης, της tesseract-OCR, η οποία θεωρείται από τις καλύτερες διαθέσιμες βιβλιοθήκες για αναγνώριση χαρακτήρων (κείμενο). Τέλος, για την αποτίμηση και την εξαγωγή συμπερασμάτων, χρησιμοποιήθηκαν δύο (2) σύνολα δεδομένων (βιντεοληπτικό υλικό). Το πρώτο σύνολο, αποτελεί υλικό σταθερής κάμερας και ανακτήθηκε από ανοιχτές πηγές, ενώ το δεύτερο από λήψεις ΣμηΕΑ που έγιναν για τον σκοπό της εργασίας.

1.4. ΔΟΜΗ

Η δομή της εργασίας έχει ως εξής. Αρχικά θα γίνει μια σύντομη ανάλυση των διάφορων μεθοδολογιών αναγνώρισης πινακίδων που έχουν αναπτυχθεί και τα πλεονεκτήματα

και μειονεκτήματα της καθεμιάς, ξεχωριστά για τη διαδικασία εντοπισμού της πινακίδας και της ανάγνωσης του περιεχομένου της. Στη συνέχεια θα αναλυθεί η μεθοδολογία που αναπτύχθηκε, στα πλαίσια της οποίας θα γίνει παρουσίαση των MSER και της βιβλιοθήκης tesseract που χρησιμοποιήθηκε για την οπτική αναγνώριση των χαρακτήρων. Ακολουθεί η περιγραφή της προγραμματιστικής διαδικασίας που χρησιμοποιήθηκε και οι παράμετροι που επιλέχθηκαν ώστε να επιτευχθούν καλύτερα αποτελέσματα. Τέλος θα γίνει παρουσίαση των αποτελεσμάτων και σύγκρισή τους με άλλες μεθόδους, καθώς και προτάσεις για τη βελτίωσή τους με μελλοντική εργασία.

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Όπως αναφέρθηκε και παραπάνω το πρόβλημα στην ΑΠΚ περιλαμβάνει τρία μέρη, την ανίχνευση της πινακίδας μέσα στην εικόνα, τον διαχωρισμό των χαρακτήρων της πινακίδας και την αναγνώρισή τους. Συχνά, τα δύο τελευταία στάδιο, ενοποιούνται σε ένα και αναφέρονται ως αναγνώριση χαρακτήρων, όπου θα λόγους ανάπτυξης της δομής της εργασίας, προκρίνεται η ως άνω σύντμηση. Στο κεφάλαιο αυτό γίνεται μία θεωρητική προσέγγιση των τεχνικών που χρησιμοποιούνται για την επίλυση αντίστοιχων προβλημάτων.

2.1. ΑΝΙΧΝΕΥΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Κάνοντας μία σύντομη αναζήτηση στην υπάρχουσα βιβλιογραφία, μπορούμε να χωρίσουμε ιστορικά τους αλγορίθμους ανίχνευσης αντικειμένων σε δύο περιόδους: έως το 2013, με αλγορίθμους εκμάθησης μηχανής (machine learning), και από το 2013 έως σήμερα, με αλγορίθμους βαθιάς μάθησης (deep learning). Οι αλγόριθμοι εκμάθησης μηχανής χαρακτηρίζονται από τη σχετικά απλή αρχιτεκτονική τους, βασίζονται σε μαθηματικά μοντέλα, και για την εφαρμογή τους απαιτείται περιορισμένη υπολογιστική ισχύς. Αντίθετα, οι αλγόριθμοι βαθιάς μάθησης έχουν πολύπλοκη δομή, απαιτούν μεγάλη επεξεργαστική ισχύ και έχουν μεγάλο βαθμό επεκτασιμότητας. Τέλος, τα μοντέλα που χρησιμοποιούνται στη βαθιά μάθηση ονομάζονται τεχνητά νευρωνικά δίκτυα.

2.1.1. Αλγόριθμοι Machine Learning

2.1.1.1. Viola & Jones Detector

Ο αλγόριθμος Viola & Jones δημοσιεύτηκε το 2001 από τους Paul Viola και Michael Jones [1], [2], και πήρε την ονομασία του από τους δημιουργούς του. Ο κύριος σκοπός του ήταν η αναγνώριση προσώπων από εικόνες, ενώ η ταχύτητά του σε σχέση με τους αντίστοιχους αλγόριθμους της εποχής ήταν χαρακτηριστική. Βασίζεται στην εξαγωγή χαρακτηριστικών τύπου Haar (Haar features), τα οποία είναι παρόμοια με τα συνελικτικά φίλτρα kernel. Τα χαρακτηριστικά Haar αποτελούν απλά μοτίβα που μπορούν να εντοπίσουν βασικά σχήματα και δομές σε μια εικόνα, όπως άκρα, γραμμές και γωνίες. Αυτά τα χαρακτηριστικά υπολογίζονται μέσω της διαφοράς φωτεινότητας μεταξύ γειτονικών περιοχών της εικόνας. Για την ταξινόμηση των υποψήφιων περιοχών της εικόνας, ο αλγόριθμος χρησιμοποιεί τον αλγόριθμο AdaBoost. Ο AdaBoost είναι μια μέθοδος ενίσχυσης (boosting) που συνδυάζει πολλούς αδύναμους ταξινομητές για να δημιουργήσει έναν ισχυρό ταξινομητή. Στην περίπτωση του Viola & Jones, οι αδύναμοι ταξινομητές είναι απλά αποφασιστικά δέντρα (decision stumps) που βασίζονται στα χαρακτηριστικά Haar. Βασίζεται σε τρία στάδια.

- Επιλογή Χαρακτηριστικών (Feature Selection): Από ένα μεγάλο σύνολο χαρακτηριστικών Haar, επιλέγονται τα πιο αντιπροσωπευτικά και αποδοτικά χαρακτηριστικά για την αναγνώριση του προσώπου.
- Εκπαίδευση με AdaBoost: Τα επιλεγμένα χαρακτηριστικά χρησιμοποιούνται για την εκπαίδευση του αλγορίθμου AdaBoost, ο οποίος δημιουργεί ένα ισχυρό μοντέλο ταξινόμησης από πολλούς αδύναμους ταξινομητές.
- Σύστημα Καταρράκτη (Cascade Classifier): Η ταξινόμηση των περιοχών της εικόνας γίνεται μέσω ενός συστήματος καταρράκτη, το οποίο εφαρμόζει σταδιακά τους ταξινομητές με αυξανόμενη πολυπλοκότητα. Αυτό το σύστημα επιτρέπει την ταχεία απόρριψη των περιοχών που δεν περιέχουν πρόσωπα και εστιάζει μόνο στις πιο πιθανές περιοχές.

Ο Viola & Jones detector αποτέλεσε ορόσημο στην ανίχνευση αντικειμένων λόγω της καινοτόμου προσέγγισής του, συνδυάζοντας απλότητα και αποτελεσματικότητα. Έθεσε

τις βάσεις για την ανάπτυξη πολλών σύγχρονων αλγορίθμων ανίχνευσης αντικειμένων και συνεχίζει να χρησιμοποιείται ευρέως σε διάφορες εφαρμογές υπολογιστικής όρασης

2.1.1.2. HOG Detector

Ο αλγόριθμος HOG (Histogram of Oriented Gradients) δημοσιεύτηκε το 2005 από τους N. Dalal και B. Triggs [3]. Αρχικά, χρησιμοποιήθηκε για την αναγνώριση ανθρώπων από στατικές φωτογραφίες, αλλά στη συνέχεια εξελίχθηκε ώστε να μπορεί να χρησιμοποιηθεί και ανά καρέ βιντεοληπτικού υλικού, καθιστώντας τον κατάλληλο για εφαρμογές όπως η ανίχνευση πεζών σε βίντεο. Ο αλγόριθμος εξάγει από την εικόνα πληροφορίες σχετικά με το σχήμα των αντικειμένων που περιέχει, και το αποτέλεσμά του είναι ένα διάνυσμα περιγραφής σχήματος (shape descriptor) για την εικόνα. Δεδομένου ότι το σχήμα ενός αντικειμένου είναι η συνισταμένη όλων των ακμών που το αποτελούν, ο αλγόριθμος υπολογίζει ένα ιστόγραμμα με τους προσανατολισμούς των ακμών. Συγκεκριμένα, η εικόνα διαιρείται σε μικρά κελιά, και για κάθε κελί υπολογίζεται το ιστόγραμμα των προσανατολισμών των διανυσμάτων (gradients). Η βάση στην οποία στηρίζεται ο αλγόριθμος είναι ότι οι εικόνες που απεικονίζουν παρόμοιο σχήμα θα έχουν παρόμοιο ιστόγραμμα ακμών. Κατά συνέπεια, η απόσταση μεταξύ των διανυσμάτων περιγραφής σχήματος αποτελεί το μέτρο ομοιότητας των εικόνων. Αυτή η τεχνική επιτρέπει την αποτελεσματική αναγνώριση και ταξινόμηση αντικειμένων βάσει του σχήματός τους, κάνοντας τον αλγόριθμο HOG ιδιαίτερα αποδοτικό για την αναγνώριση ανθρώπων και άλλων αντικειμένων σε ποικίλες συνθήκες φωτισμού και περιβάλλοντος. Ο αλγόριθμος HOG αποτελεί μια σημαντική εξέλιξη στην υπολογιστική όραση, προσφέροντας υψηλή ακρίβεια και αξιοπιστία στην αναγνώριση αντικειμένων, ενώ η χρήση του σε συνδυασμό με άλλες τεχνικές έχει επεκτείνει τις δυνατότητές του σε πολλούς τομείς της τεχνητής νοημοσύνης και της μηχανικής μάθησης.

2.1.1.3. DPM Detector

Ο αλγόριθμος DPM (Deformable Parts Model) δημοσιεύτηκε το 2008 από τον P. Felzenszwalb [4] και αποτέλεσε το αποκορύφωμα των μεθόδων Machine Learning, καθώς κατείχε τα καλύτερα αποτελέσματα σε χρόνο και ακρίβεια. Ο αλγόριθμος βασίζεται στη λογική της διαίρεσης του αντικειμένου προς αναζήτηση σε μικρότερα τμήματα. Για παράδειγμα, η ανίχνευση ενός ανθρώπου περιλάμβανε την ανίχνευση του κεφαλιού, των χεριών και των ποδιών. Το μοντέλο DPM χρησιμοποιεί μια σειρά από μεθόδους που επιτρέπουν την ανίχνευση παραμορφώσιμων τμημάτων (deformable parts) ενός

αντικειμένου, συνδυάζοντας τα αποτελέσματα για να προσδιορίσει την παρουσία και τη θέση του συνολικού αντικειμένου. Η χρήση αυτού του μοντέλου επιτρέπει την ανίχνευση αντικειμένων σε διαφορετικές θέσεις και καταστάσεις, προσφέροντας υψηλή ακρίβεια και ανθεκτικότητα σε παραμορφώσεις. Το επόμενο χρονικό διάστημα, μέχρι το 2013, εμφανίστηκαν πλήθος νέων αλγορίθμων που βασίζονταν σε τεχνικές όπως αυτές του DPM, με μικρές αλλαγές και βελτιώσεις. Ωστόσο, όλοι αυτοί οι αλγόριθμοι παρέμειναν εντός του πλαισίου των παραδοσιακών μεθόδων Machine Learning.

Το 2013 σηματοδότησε την έναρξη της ευρείας χρήσης των νευρωνικών δικτύων και των μεθόδων βαθιάς μάθησης (deep learning). Αυτές οι νέες μέθοδοι παρουσίασαν πρωτοφανείς ταχύτητες περάτωσης και ακρίβεια, φέρνοντας επανάσταση στον τομέα της ανίχνευσης αντικειμένων και θέτοντας τα θεμέλια για τις σύγχρονες εφαρμογές υπολογιστικής όρασης..

2.1.2. Αλγόριθμοι Deep Learning

Οι αλγόριθμοι βαθιάς μάθησης, χωρίζονται σε αλγορίθμους δύο σταδίων (two-stage networks) και ενός (one-stage networks). Οι μεν πρώτοι παρέχουν καλύτερα αποτελέσματα αλλά με σχετικά μεγάλο χρόνο ανταπόκρισης, ενώ αντίστοιχα οι δεύτεροι, μικρό χρόνο ανταπόκρισης, που πολλές φορές η επεξεργασία μας δίνει αποτελέσματα σε «πραγματικό» χρόνο, ενώ η ακρίβεια είναι μικρότερη.

2.1.2.1. Two-stage Network

Σε αυτή την κατηγορία αλγορίθμων, η ανίχνευση των αντικειμένων δίνεται σε δύο στάδια. Αρχικά, το μοντέλο προτείνει ένα σύνολο πρότασης περιοχών ενδιαφέροντος, από τα οποία εξάγονται χαρακτηριστικά σημεία και στην συνέχεια ένας ταξινομητής επεξεργάζεται τις περιοχές για τον τελικό εντοπισμό τους. Κάθε αλγόριθμος έχει την δική του λογική και στο κεφάλαιο αυτό θα δούμε τον βασικότερο, R-CNN, με τις επιμέρους αναβαθμίσεις του.

2.1.2.1.1. R-CNN

Ο αλγόριθμος R-CNN (Region-based Convolutional Neural Network) προτάθηκε από τον Ross Girshick το 2014 [5] και ανήκει στην κατηγορία των Region Based Frameworks. Αυτός ο αλγόριθμος αποτελεί ένα από τα πρώτα δίκτυα που επηρεάστηκαν από το AlexNet και χρησιμοποιεί κομμάτια αυτού του δικτύου, με μικρές αλλαγές. Η βασική διαφοροποίηση του R-CNN ήταν η δυνατότητα αναγνώρισης διαφόρων αντικειμένων σε μια εικόνα, αντί για την κατηγοριοποίηση της εικόνας ως ένα αντικείμενο. Για να επιτύχει αυτό, χρησιμοποιεί την τεχνική Selective Search [6], η οποία εντοπίζει κάθε περιοχή πιθανού αντικειμένου στην εικόνα και την εισάγει σε ένα παράθυρο. Ο τρόπος λειτουργίας του αλγορίθμου R-CNN μπορεί να περιγραφεί σε τρία απλά βήματα.

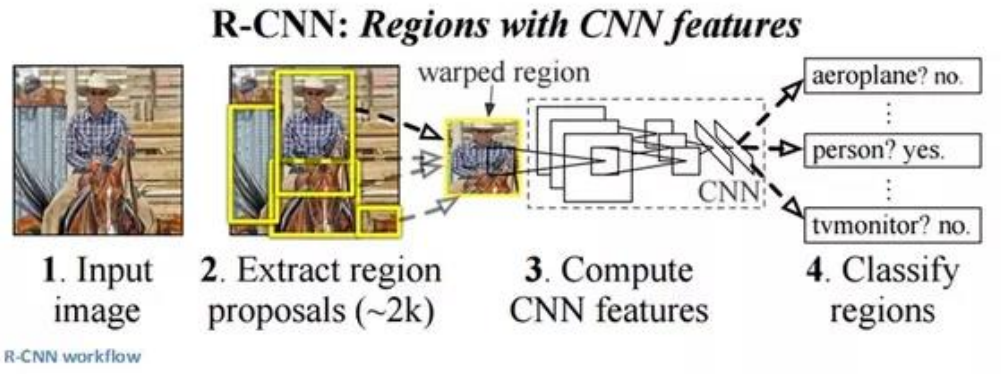
1ο βήμα Πραγματοποιείται σάρωση της αρχικής εικόνας για εντοπισμό αντικειμένων, χρησιμοποιώντας τον αλγόριθμο επιλεκτικής αναζήτησης (Selective Search), δημιουργώντας έτσι 2000 προτάσεις περιοχών (Region Proposals). Σε αυτές τις περιοχές υπάρχουν ενδεχόμενα αντικείμενα.

2ο βήμα Οι 2000 προτεινόμενες περιοχές εισάγονται σε ένα συνελικτικό νευρωτικό δίκτυο, με κύριο στόχο της εξαγωγή ενός διανύσματος 4096 διαστάσεων για την κάθε περιοχή. Με τον τρόπο αυτό το νευρωτικό δίκτυο λειτουργεί ως εξαγωγές χαρακτηριστικών της κάθε μίας περιοχής.

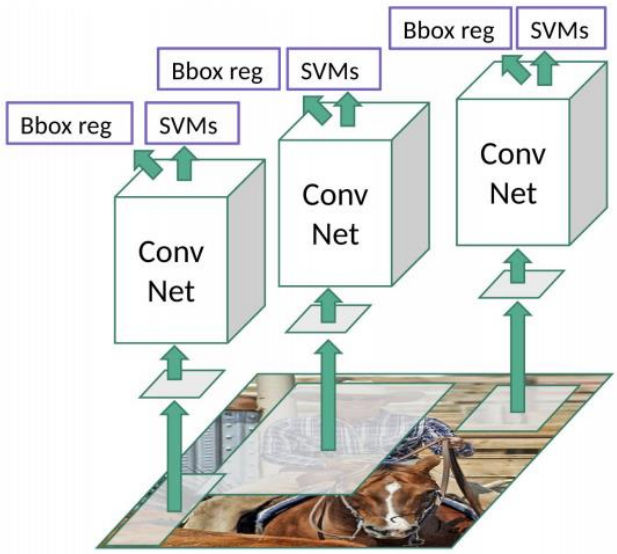
3ο βήμα Τα παραπάνω χαρακτηριστικά, εισάγονται σε ένα SVM για να ταξινομήσει την κάθε περιοχή αναλόγως των χαρακτηριστικών βάσει του διανύσματος. Ο σκοπός είναι η ταξινόμηση της παρουσίας των αντικειμένων μέσα στην υποψήφια περιοχή. Επίσης, σε κάθε υποψήφια περιοχή, ο αλγόριθμος προβλέπει τέσσερις τιμές οι οποίες οριοθετούν το πλαίσιο του αντικειμένου (bounding box).

Η εκπαίδευση ενός δικτύου R-CNN είναι μια χρονοβόρα διαδικασία, καθώς απαιτεί την επεξεργασία ενός πολύ μεγάλου όγκου εξαγόμενων χαρακτηριστικών. Για παράδειγμα, αν έχουμε N εικόνες για εκπαίδευση, τα εξαγόμενα χαρακτηριστικά θα είναι συνήθως N x 2000, ένας πολύ μεγάλος αριθμός. Επιπλέον, η διαδικασία εντοπισμού αντικειμένων

μέσω R-CNN περιλαμβάνει τη χρήση διαφόρων μοντέλων όπως νευρωνικά δίκτυα, γραμμικοί ταξινομητές SVM και μοντέλα παλινδρόμησης για την οριοθέτηση-πλαίσιο των αντικειμένων. Αυτό επιφέρει σημαντικό χρόνο ολοκλήρωσης για κάθε εικόνα, με τη διαδικασία να απαιτεί αρκετά δευτερόλεπτα. Ως αποτέλεσμα, ο εντοπισμός αντικειμένων σε πραγματικό χρόνο δεν είναι εφικτός με αυτόν τον αλγόριθμο. Αυτά τα περιοριστικά χαρακτηριστικά του R-CNN οδήγησαν στην ανάπτυξη πιο γρήγορων και αποδοτικών μεθόδων όπως το Fast R-CNN και το Faster R-CNN, τα οποία ξεπέρασαν αυτές τις προκλήσεις και επιτέλους επέτρεψαν τον εντοπισμό αντικειμένων σε πραγματικό χρόνο.



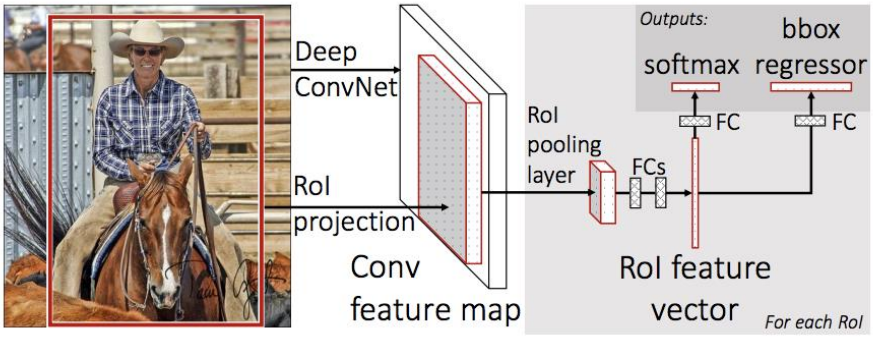
Εικόνα 1. Φάσεις λειτουργίας του R-CNN
(ΠΗΓΗ: εικόνα Girshick et al., 2014)



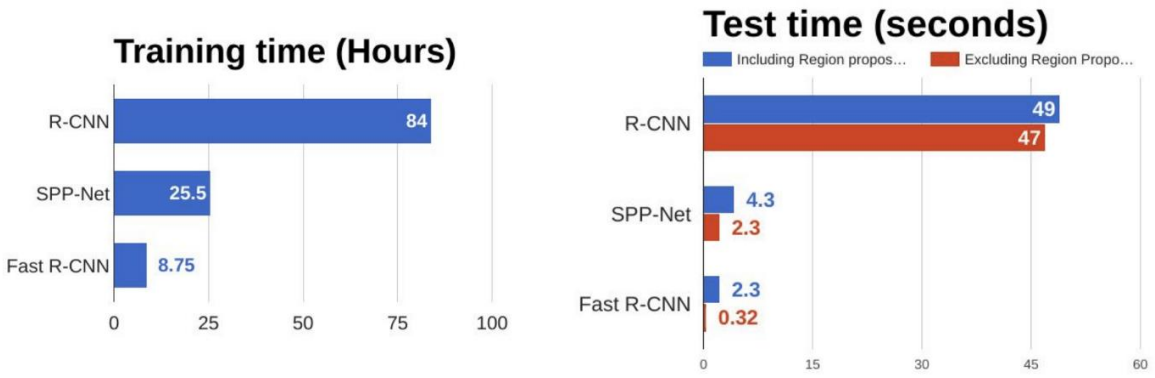
Εικόνα 2. Δομή του R-CNN
(ΠΗΓΗ: εικόνα https://towardsdatascience.com/) [7]

2.1.2.1.2. Fast R-CNN

Το Fast R-CNN, που προτάθηκε το 2015 από τον Ross Girshick [8], σχεδιάστηκε με στόχο να αντιμετωπίσει τα κύρια προβλήματα του αλγορίθμου R-CNN και να επιταχύνει την διαδικασία εντοπισμού αντικειμένων. Το πρώτο πρόβλημα, που αφορούσε τον υπολογιστικά ακριβό και χρονοβόρο χειρισμό των εξαγόμενων χαρακτηριστικών, λύθηκε με τη μέθοδο του Region of Interest Pooling (RoIPool). Αυτή η μέθοδος επιτρέπει την εφαρμογή της συνέλιξης μία φορά σε ολόκληρη την εικόνα και την υπολογιστική επεξεργασία pooling μόνο στα περιβλήματα των περιοχών ενδιαφέροντος (RoIs), αντί για κάθε πρόταση ξεχωριστά. Με αυτόν τον τρόπο, ο Fast R-CNN μείωσε σημαντικά τον αριθμό των απαιτούμενων υπολογιστικών επιχειρήσεων από περίπου 2000 σε μόνο μία. Το δεύτερο πρόβλημα, που αφορούσε τον μακροχρόνιο χρόνο εκτέλεσης λόγω των διαφορετικών μοντέλων που χρησιμοποιούνταν (SVMs για κατηγοριοποίηση, μοντέλα παλινδρόμησης για το bounding box), αντιμετωπίστηκε ενσωματώνοντας όλα αυτά τα μοντέλα σε ένα κοινό νευρωνικό δίκτυο. Η κατηγοριοποίηση των RoIs πλέον γίνεται με τη χρήση ενός Softmax classifier, το οποίο λαμβάνει ως είσοδο τα αποτελέσματα του επιπέδου RoIPool. Η παλινδρόμηση για την τοποθέτηση του bounding box επίσης ενσωματώθηκε σε αυτό το κοινό δίκτυο. Με αυτές τις βελτιώσεις, ο συνολικός χρόνος εκτέλεσης του Fast R-CNN μειώθηκε σημαντικά, ενώ ο χρόνος εκπαίδευσης του μοντέλου επίσης επιταχύνθηκε κατά 2.7 φορές. Επιπλέον, η ακρίβεια του μοντέλου (κριτήριο mAP) επίσης αυξήθηκε ελαφρώς, κάνοντας το Fast R-CNN μία αποδοτική και αποτελεσματική επιλογή για τον εντοπισμό αντικειμένων σε εικόνες.



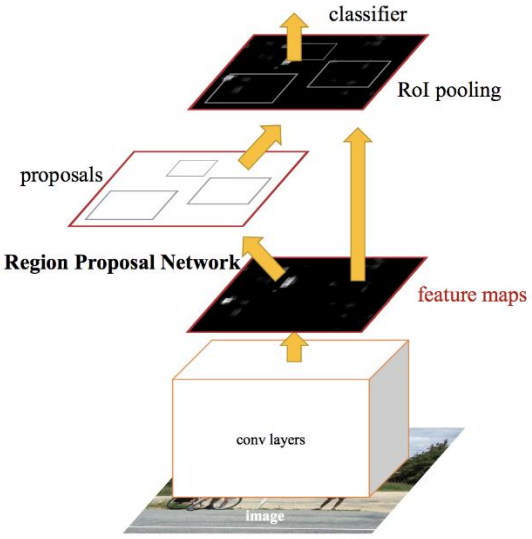
Εικόνα 3. Φάσεις λειτουργίας του Fast R-CNN
(ΠΗΓΗ: εικόνα Girshick et al., 2015)



Εικόνα 4. Σύγκριση R-CNN με Fast R-CNN
(ΠΗΓΗ: εικόνα https://towardsdatascience.com/) [7]

2.1.2.1.3. Faster R-CNN

Αποτελεί την επόμενη χρονικά πρόταση, η οποία προτάθηκε από το ερευνητικό τμήμα της Microsoft [9] και είναι η τροποποιημένη έκδοση του Fast R-CNN. Η διαφοροποίησή του είναι στον τρόπο της εύρεσης των περιοχών, η οποία πλέον δεν γίνεται με τον αλγόριθμο επιλεκτικής αναζήτησης (Selective Search), αλλά χρησιμοποιεί το “Region Proposal Network” (RPN). Το RPN δέχεται σαν είσοδο τους χάρτες χαρακτηριστικών και δημιουργεί ένα σύνολο προτάσεων για αντικείμενα, το καθένα με μια βαθμολογία αντικειμενικότητας ως έξοδο. Βέβαια, η λογική του δεν διαφέρει από τους προκατόχους του. Η αρχική εικόνα, τροφοδοτείται σε ένα νευρωτικό δίκτυο προκειμένου εξάγει τον χάρτη χαρακτηριστικά των πιθανών αντικειμένων και το RPN εφαρμόζεται σε αυτούς τους χάρτες και επιστρέφει προτάσεις περιοχών των αντικειμένων μαζί με μία τιμή, που είναι η πιθανότητα αναγνώρισης του αντικειμένου και τα όρια του πλαισίου που περιβάλει το αντικείμενο. Εν συνεχεία, σε όλες αυτές τις προτεινόμενες περιοχές αντικειμένων, εφαρμόζεται το RoIPool, όπως και στον Fast R-CNN. Τέλος, οι προτάσεις τροφοδοτούνται σε ένα πλήρως συνδεδεμένο δίκτυο το οποίο περιέχει ένα softmax στρώμα και ένα στρώμα γραμμικής παλινδρόμησης με σκοπό την ταξινόμηση των αντικειμένων και την εξαγωγή των πλαισίων οριοθέτησης για τα αντικείμενα.



Εικόνα 5. Δομή του Faster R-CNN
(ΠΗΓΗ: εικόνα https://towardsdatascience.com/) [7]

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Εικόνα 6. Σύγκριση R-CNN, Fast R-CNN, Faster R-CNN
(ΠΗΓΗ: εικόνα Evaluation and Evolution of Object Detection Techniques YOLO and R-CNN) [10]

Το κύριο πλεονέκτημα του Faster R-CNN είναι η σημαντική αύξηση της ταχύτητας. Συγκριτικά με τον Fast R-CNN, ο Faster R-CNN είναι περίπου 10 φορές ταχύτερος, ενώ η ακρίβεια του παραμένει στα ίδια επίπεδα. Αυτή η αποτελεσματική συνδυασμένη διαδικασία επιτρέπει την αξιοποίηση του Faster R-CNN σε πρακτικές εφαρμογές πραγματικού χρόνου, όπου η ταχύτητα εκτέλεσης είναι κρίσιμη (στο σύνολο δεδομένων VOC 2007) [10].

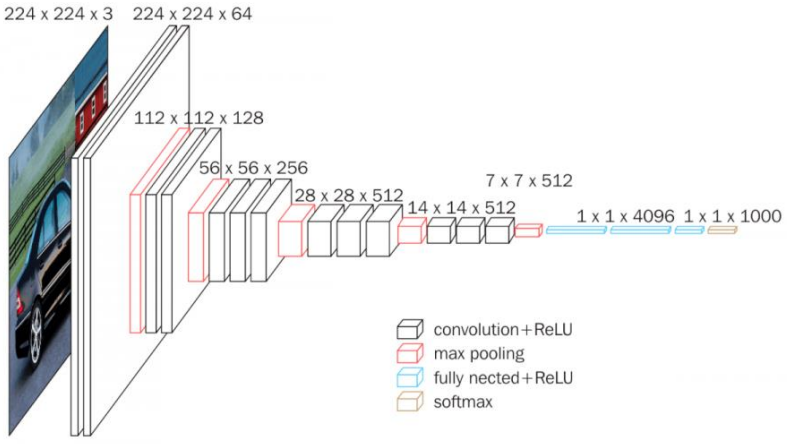
2.1.2.2. One-stage Network

Οι αλγόριθμοι, στην κατηγορία αυτή, έχουν μία διαφορετική προσέγγιση, σε σχέση με τους παραπάνω, καθώς παραλείπουν το στάδιο πρότασης περιοχών και εκτελούν τον εντοπισμό και την ταξινόμηση, απευθείας σε ένα συνελικτικό νευρωτικό δίκτυο.

Χαρακτηριστικό τους είναι η ταχύτητα περάτωσής τους, γεγονός το οποίο βοηθάει στην εξαγωγή αποτελεσμάτων σε πρακτικό χρόνο.

2.1.2.2.1. SSD

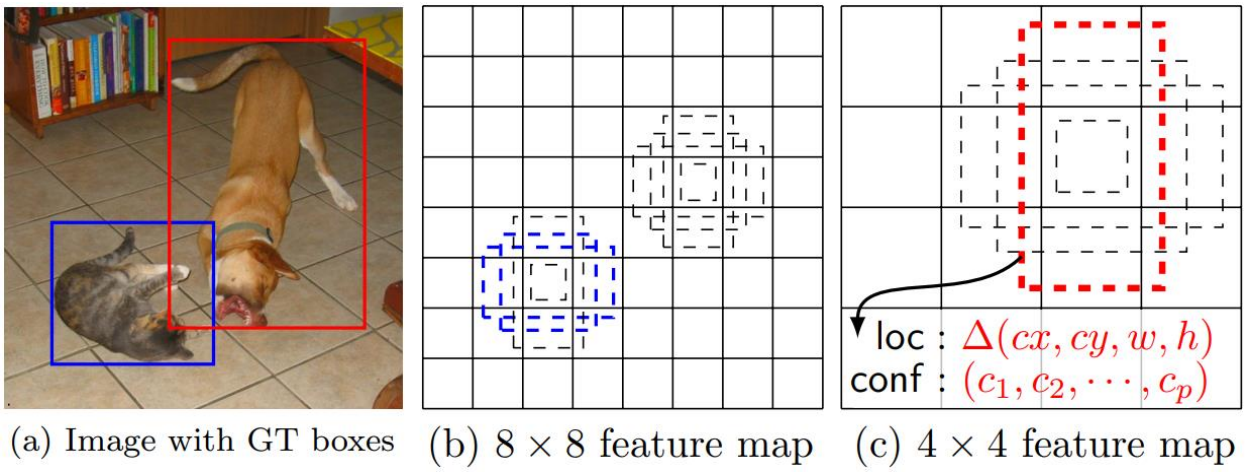
Ο αλγόριθμος SSD (Single Shot MultiBox Detector) αποτελεί μία προηγμένη έκδοση του Faster R-CNN, διατηρώντας κοινή αρχιτεκτονική με την προαναφερθείσα μέθοδο, αλλά προσφέροντας σημαντικές βελτιώσεις [11]. Λειτουργεί ως ένα βήμα, απευθείας εξάγοντας τα πλαίσια οριοθέτησης των αντικειμένων και τις πιθανότητες ταύτισης με τις κλάσεις, ουσιαστικά μειώνοντας δραστικά τον χρόνο εκτέλεσης (One-stage Network). Το SSD βασίζεται στο δίκτυο VGG-16 από τους Simonyan et al. [12], το οποίο χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από εικόνα (image recognition).



Εικόνα 7.Αρχιτεκτονική δικτύου VGG-16
(ΠΗΓΗ: εικόνα Very Deep Convolutional Networks for Large-Scale Image Recognition) [12]

Η λογική λειτουργίας του αλγορίθμου SSD βασίζεται σε ένα συνελικτικό νευρωνικό δίκτυο (CNN), το οποίο χρησιμοποιείται για την εξαγωγή χαρακτηριστικών από την εικόνα. Αυτό το δίκτυο παράγει μια συλλογή σταθερού μεγέθους πλαισίων οριοθέτησης και βαθμολογιών για την αναζήτηση αντικειμένων στην εικόνα. Τα αρχικά, μεγάλου μήκους συνελικτικά στρώματα αναγνωρίζουν καλύτερα μικρά αντικείμενα, καθώς είναι ειδικά σχεδιασμένα για να εντοπίζουν λεπτομέρειες στην εικόνα. Αντίθετα, τα βαθιά στρώματα αναγνωρίζουν καλύτερα τα μεγάλα αντικείμενα, επειδή έχουν μεγαλύτερο χωρικό πεδίο και μπορούν να ανιχνεύσουν συνολικές δομές στην εικόνα. Τα ενδιάμεσα στρώματα χρησιμοποιούν πλαίσια οριοθέτησης (bounding boxes) διαφορετικών διαστάσεων και κλίμακας για τον εντοπισμό αντικειμένων σε διάφορα μεγέθη και αναλογίες στην εικόνα. Αυτός ο ποικιλόμορφος σχεδιασμός πλαισίων οριοθέτησης επιτρέπει στον αλγόριθμο να αναγνωρίσει όσο το δυνατόν περισσότερα αντικείμενα στην εικόνα, καθιστώντας τον πιο

αποτελεσματικό σε πραγματικές συνθήκες. Ο παραπάνω ισχυρισμός αποδεικνύεται στην παρακάτω εικόνα.



Εικόνα 8. Bounding box του SSD
(ΠΗΓΗ: εικόνα SSD: Single Shot MultiBox Detector) [13]

Ο SSD, σύμφωνα με πειραματισμούς, σε διάφορα σύνολα δεδομένων (PASCAL , VOC, COCO), δείχνει ότι έχει ανταγωνιστική ακρίβεια (βλ. Εικόνα 9) σε σχέση με τους προηγούμενους μεθόδους δύο βημάτων (Two-stage Network). Παράλληλα είναι πιο γρήγορος, προσφέροντας μία ενοποιημένη μορφή, τόσο για την εκπαίδευσή του όσο και την εκτέλεση-περάτωσή του.

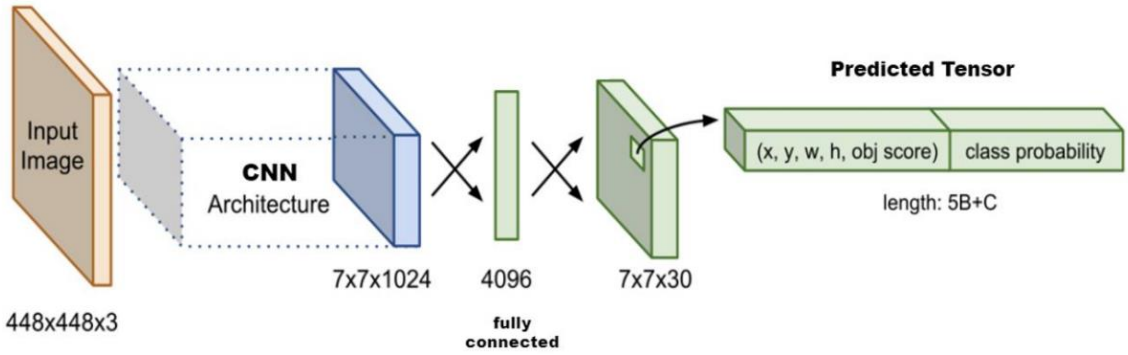
Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [2]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**

Εικόνα 9. Πειραματισμοί SSD με Two-stage Networks (mAP)
(ΠΗΓΗ: εικόνα SSD: Single Shot MultiBox Detector) [13]

2.1.2.2.2. YOLO

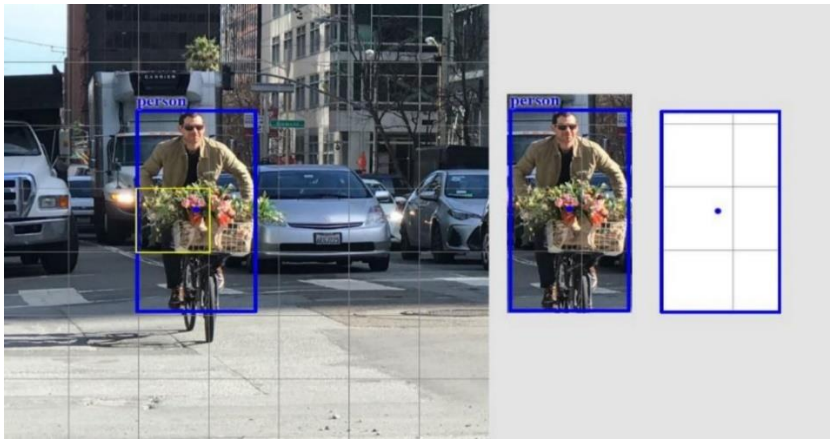
Ο αλγόριθμος YOLO (You Only Look Once) [14] προτάθηκε ως μία λύση για την αναγνώριση αντικειμένων σε πραγματικό χρόνο. Αρχικά δημοσιεύτηκε από τους Redmon et al. το 2016. Ακολούθησαν βελτιώσεις, όπως ο YOLOv2 (ή YOLO9000) την ίδια χρονιά [15], ο YOLOv3 του 2018 [16], YOLOv4 του 2020 [17] και τέλος, ο YOLOv5 από τον Glenn Jocher τον Μάιο του 2020 [20]. Η λογική του είναι η «ανάγνωση» μία φορά της εικόνας

και η πρόβλεψη των αντικειμένων και της θέσης τους, αντιμετωπίζοντας την ανίχνευση αντικειμένων σαν ένα πρόβλημα απλής παλινδρόμησης. Η αρχιτεκτονική του νευρωνικού αποτελείται από 24 επίπεδα συνέλιξης, υποδειγματοληψίας και δύο πλήρως συνδεδεμένα επίπεδα στο τέλος. Η έξοδος του μοντέλου είναι ένα διάνυσμα S x S x (5B + C) για κάθε εικόνα, όπου B είναι ο αριθμός των πλαισίων οριοθέτησης που προτείνει κάθε κελί στο πλέγμα και C είναι ο αριθμός των κατηγοριών που αναγνωρίζει το μοντέλο. Για να κατανοήσουμε τι αντιπροσωπεύει κάθε σύμβολο στην έξοδο του μοντέλου, χρειάζεται να κατανοήσουμε τη δομή του διανύσματος αυτού και τον τρόπο με τον οποίο κωδικοποιεί τις προβλέψεις για κάθε πλαίσιο οριοθέτησης και κατηγορία αντικειμένου στην εικόνα..



Εικόνα 10. Αρχιτεκτονική αλγορίθμου YOLO
(ΠΗΓΗ: εικόνα https://www.mdpi.com/2076-3417/10/2/612/htm)

Η διαδικασία εντοπισμού με την μέθοδο YOLO, ξεκινάει χωρίζοντας την εικόνα εισόδου σε ένα πλέγμα διαστάσεων S x S. Εάν το κέντρο ενός αντικειμένου, περιγράφεται από ένα κελί, τότε αυτό το τετράγωνο είναι υπεύθυνο για τον εντοπισμό του συγκεκριμένου αντικειμένου.

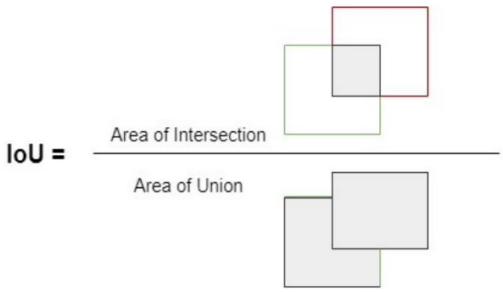


Εικόνα 11. Κίτρινο πλαίσιο «υπεύθυνο» για τον εντοπισμό του αντικειμένου
(ΠΗΓΗ: εικόνα Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3) [17]

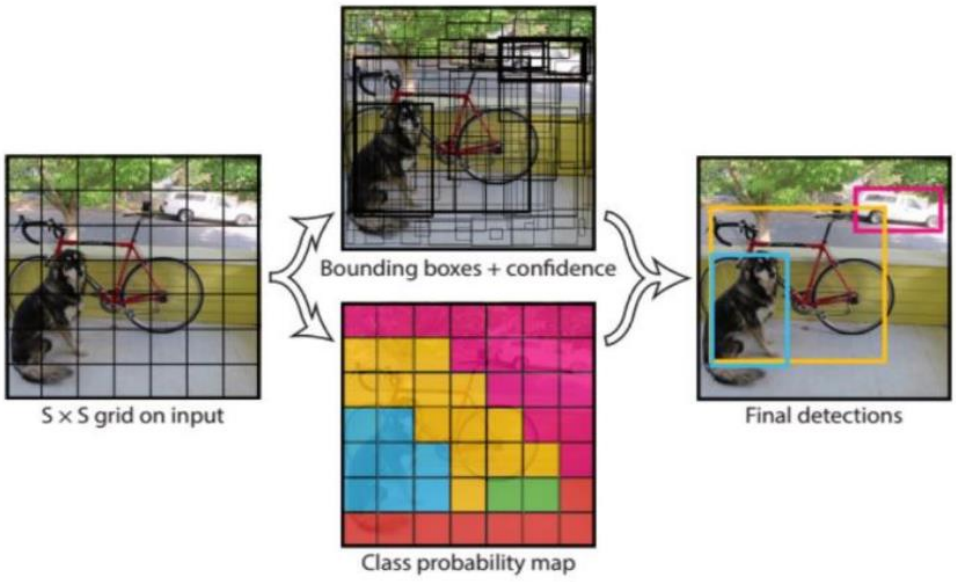
Επίσης κάθε πλαίσιο οριοθέτησης (bounding box) έχει και έναν βαθμός εμπιστοσύνης, όπου ο ρόλος του είναι η ποσοτική περιγραφή της βεβαιότητας του μοντέλου ότι το πλαίσιο οριοθέτησης περιέχει το αντικείμενο που προβλέπει [19]. Η εμπιστοσύνη βγαίνει από τον μαθηματικό τύπο,

$$Pr(Object) * IOU_{prediction}^{truth}$$

Όπου $IOU_{prediction}^{truth}$ (Intersection over Union) είναι η τομή του πλαισίου που έχει προβλεφθεί με το πραγματικό πλαίσιο που περιβάλλει το αντικείμενο προς την ένωση τους και Pr($Object$) η πιθανότητα να υπάρχει αντικείμενο. Εάν δεν υπάρχει αντικείμενο στο πλαίσιο, η τιμή της είναι 0 ενώ εφόσον υπάρχει αυτή μπορεί να κλιμακωθεί έως το 1. Η τιμή της προκύπτει από τον παρακάτω τύπο.



Κάθε πλαίσιο οριοθέτησης αποτελείται από 5 προβλέψεις (x, y, w, h, σκορ εμπιστοσύνης).
(x, y): το κέντρο του περιβλήματος
w = πλάτος ορθογωνίου / πλάτος εικόνας
h = ύψος ορθογωνίου / ύψος εικόνας
Επιπλέον, κάθε πλαίσιο έχει και C εξαρτώμενες πιθανότητες κλάσης Pr ($Classi|Object$), όπου C ο αριθμός των κλάσεων. Οπότε συνολικά η έξοδος του νευρωνικού είναι ένας τανυστής S x S x (5B + C) βάση του οποίου προκύπτει ο αριθμός των προβλέψεων. Στην εικόνα που δίνεται παρακάτω, φαίνεται η λογική του YOLO αλγορίθμου.



Εικόνα 12. Συνοπτική αναπαράσταση λειτουργίας αλγορίθμου YOLO (ΠΗΓΗ: εικόνα You only look once, 2016) [14]

2.2. ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΩΝ

Η οπτική αναγνώριση χαρακτήρων (OCR – Optical Character Recognition) αναφέρεται στη διαδικασία με την οποία μηχανικοί υπολογιστών και αλγόριθμοι μηχανικής μάθησης μπορούν να αναγνωρίσουν και να εξάγουν κείμενο από ψηφιακές εικόνες. Αυτό επιτυγχάνεται με τη μετατροπή των σαρωμένων εικόνων σε μορφή κειμένου που μπορεί να επεξεργαστεί ο υπολογιστής. Ο σκοπός της OCR είναι να αναγνωρίσει και να εξάγει όλους τους χαρακτήρες από μία εικόνα, προκειμένου να μετατραπεί το περιεχόμενο σε μηχανικά επεξεργάσιμη μορφή. Αυτή η τεχνολογία αναπτύσσεται συνεχώς και έχει εφαρμογές σε διάφορους τομείς. Για παράδειγμα, χρησιμοποιείται ως βοήθημα ανάγνωσης για τυφλούς, όπου μπορεί να μετατρέψει γραφικά σε κείμενο που να διαβάζεται από ηχητικά συστήματα. Επίσης, εφαρμόζεται στην αυτόματη αναγνώριση διευθύνσεων στα ταχυδρομεία, στην αναγνώριση πινακίδων αυτοκινήτων για ασφάλεια και διαχείριση κυκλοφορίας, καθώς και στην άμεση επεξεργασία χειρόγραφων κειμένων. Επιπλέον, χρησιμοποιείται για την αναγνώριση υπογραφών σε νομικά έγγραφα και άλλες εφαρμογές που απαιτούν τη μετατροπή αναγνωρίσιμων συμβόλων σε κείμενο που μπορεί να επεξεργαστεί ο υπολογιστής.

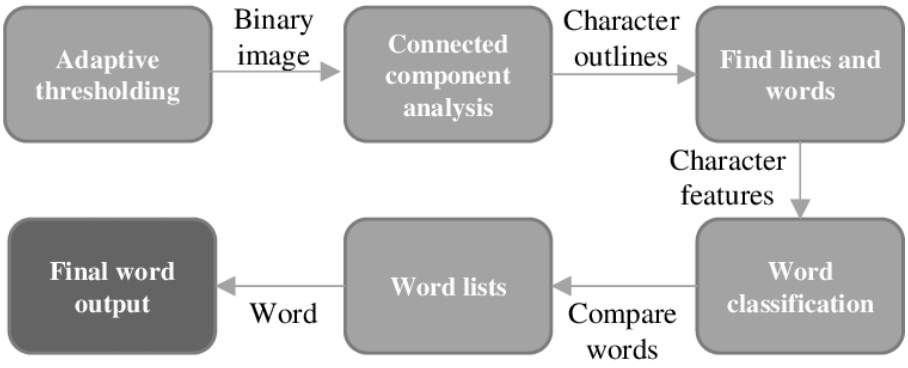
2.2.1. Μηχανή Tesseract

Η μηχανή Tesseract αναπτύχθηκε αρχικά ως εμπορικό πρόγραμμα από την Hewlett Packard στο Bristol της Αγγλίας και στο Greeley του Κολοράντο μεταξύ 1985 και 1994 σε γλώσσα προγραμματισμού C. Στη συνέχεια έγιναν κάποιες αλλαγές ώστε να είναι συμβατή με λειτουργικό σύστημα Windows και ο κώδικάς της να μπορεί να μεταγλωττιστεί με compiler της C++, ενώ οι προσθήκες στον κώδικα γίνονταν πλέον σε γλώσσα C++. Η εξέλιξή της σταμάτησε στα μέσα της δεκαετίας του 1990, και το 2005 δόθηκε ως πρόγραμμα ελεύθερου κώδικα από τη Hewlett Packard και το πανεπιστήμιο της Νεβάδας (UNLV). Από το 2006 η εξέλιξή της χρηματοδοτείται από τη Google. Σήμερα διατίθεται ως ελεύθερο λογισμικό με άδεια Apache 2.0. Είναι σχεδιασμένη να τρέχει από γραμμή εντολών, ενώ έχουν αναπτυχθεί από τρίτους προγράμματα για αλληλεπίδραση με το χρήστη (GUI) [21]. Η μηχανή Tesseract-OCR θεωρείται μία από τις καλύτερες ελεύθερες (Open Source) μηχανές στην οπτική αναγνώριση χαρακτήρων, καθώς μπορεί να αναγνωρίσει κείμενο σε περισσότερες από 35 γλώσσες. Ένας τρόπος για να επιτύχει αυτό είναι το ενσωματωμένο λεξιλόγιο, το οποίο βελτιώνει την ακρίβεια της αναγνώρισης και τη δυνατότητα αναγνώρισης σωστών λέξεων ακόμα και όταν οι χαρακτήρες αναγνωρίζονται λανθασμένα ως μεμονωμένοι [22].

2.2.2. Αρχιτεκτονική Tesseract

Για την αναγνώριση των χαρακτήρων η μηχανή Tesseract, ακολουθεί συγκεκριμένα βήματα, όπως φαίνονται στην εικόνα. Αρχικά, η εικόνα εισόδου μετατρέπεται σε δυαδική εικόνα (binary image) μέσω της διαδικασίας "adaptive thresholding". Αυτό σημαίνει ότι οι γκρι αποχρώσεις της εικόνας μετατρέπονται σε μαύρο ή άσπρο, ανάλογα με την ένταση τους σε σχέση με τη γύρω περιοχή τους. Αυτή η διαδικασία βοηθά στην ευκολότερη ανίχνευση των χαρακτήρων. Στη συνέχεια, γίνεται η εξαγωγή των περιγραμμάτων των χαρακτήρων από τη δυαδική εικόνα. Αυτά τα περιγράμματα περιλαμβάνουν τα σύνορα των χαρακτήρων που θα αναγνωριστούν. Ακολούθως, τα περιγράμματα των χαρακτήρων συγκεντρώνονται και ενώνονται σε ομάδες που ονομάζονται "blobs". Τα blobs είναι περιοχές στην εικόνα όπου οι χαρακτήρες εντοπίζονται να είναι διαφορετικοί από τις γύρω περιοχές ως προς το χρώμα ή τη φωτεινότητα. Τέλος, τα blobs

τοποθετούνται σε γραμμές κειμένου, και οι γραμμές αναλύονται για να εξάγουν το κείμενο. Αυτή η διαδικασία περιλαμβάνει τον διαχωρισμό του κειμένου σε λέξεις, χρησιμοποιώντας καθορισμένους ή ασαφείς χώρους μεταξύ των λέξεων.



Εικόνα 13. Δομή της μηχανής Tesseract
(ΠΗΓΗ: εικόνα https://www.researchgate.net/figure/Tesseract-OCR-engine-architecture_fig4_265087843)

Τέλος, η μηχανή Tesseract, λειτουργεί με δύο περάσματα για την αναγνώριση κειμένου από εικόνες. Στο πρώτο πέρασμα, η Tesseract πραγματοποιεί την αναγνώριση κάθε λέξης μεμονωμένα. Κάθε λέξη που αναγνωρίζεται και πληροί τα κριτήρια αβεβαιότητας (uncertainty criteria), αποθηκεύεται σε έναν προσαρμοστικό ταξινομητή (adaptive classifier) ως δεδομένα εκπαίδευσης. Μετά το πρώτο πέρασμα, ο προσαρμοστικός ταξινομητής αναλύει τα δεδομένα που έχουν συλλεχθεί κατά το πρώτο πέρασμα. Αυτός ο ταξινομητής χρησιμοποιείται για να βελτιώσει την αναγνώριση του κειμένου στο δεύτερο πέρασμα. Στο δεύτερο πέρασμα, ο προσαρμοστικός ταξινομητής επιστρέφει στην εικόνα για να εξετάσει ξανά περιοχές όπου οι λέξεις είχαν μικρή ακρίβεια αναγνώρισης κατά το πρώτο πέρασμα. Στη διαδικασία αυτή, επιχειρεί να διορθώσει ασαφείς χώρους και να κάνει εναλλακτικές υποθέσεις για να εντοπίσει το κείμενο που δεν αναγνωρίστηκε σωστά στο πρώτο πέρασμα [23].

3. ΜΕΘΟΔΟΛΟΓΙΑ

Στο κεφάλαιο αυτό, θα προσεγγίσουμε θεωρητικά, τον τρόπο εκπαίδευσης για αλγορίθμους που ανήκουν τόσο στην κατηγορία της μηχανικής μάθησης όσο και αυτής της βαθιάς και στο τέλος θα αναλυθεί η μεθοδολογία που ακολουθήθηκε για τις ανάγκες της εργασίας αλλά και οι επιμέρους λόγοι.

3.1. ΜΕΘΟΔΟΛΟΓΙΑ MACHINE LEARNING

Η Μηχανική Μάθηση αποτελεί βασικό τμήμα της Τεχνητής Νοημοσύνης (AI) και επιτρέπει την αυτοματοποίηση λειτουργιών χωρίς την ανάγκη ανθρώπινης καθοδήγησης. Οι αλγόριθμοι της Μηχανικής Μάθησης διαφέρουν ανάλογα με τα δεδομένα εισόδου, τα αποτελέσματα που παράγουν, και το είδος του προβλήματος που επιχειρούν να επιλύσουν. Οι κύριες κατηγορίες αλγορίθμων είναι το Supervised Learning, το Unsupervised Learning, το Semi-supervised Learning, το Reinforcement Learning και το Self-learning. Για παράδειγμα, σε ένα πρόβλημα όπως η αναγνώριση αριθμού κυκλοφορίας οχήματος, η χρήση ενός Supervised αλγορίθμου θα μπορούσε να οδηγήσει σε εξαιρετικά ακριβή αποτελέσματα συγκριτικά με άλλες μεθόδους. Αυτοί οι αλγόριθμοι είναι ιδανικοί για αναγνώριση αντικειμένων που ανήκουν σε συγκεκριμένες κατηγορίες για τις οποίες έχουν εκπαιδευτεί. Η εκπαίδευσή τους, αποτελείται από τρία (3) βασικά στάδια τα οποία είναι, συλλογή δεδομένων, εξαγωγή χαρακτηριστικών, εκπαίδευση ταξινομητή (classifier).

- Το πρώτο στάδιο είναι κρίσιμο, καθώς η επιλογή σωστών δεδομένων είναι ουσιώδης για την επίτευξη ακριβών αποτελεσμάτων. Το μέγεθος των δεδομένων πρέπει να είναι ικανοποιητικό ώστε να επιτρέπει την εξαγωγή των κατάλληλων χαρακτηριστικών για το αντικείμενο που αναγνωρίζεται.
- Το δεύτερο στάδιο, ξεκινάει με την ολοκλήρωση του πρώτου και περιλαμβάνει όλα αυτά τα χαρακτηριστικά που θα χρειαστούν για την εν συνεχεία αυτόματη ανίχνευση από τον αλγόριθμο. Αποτελεί μία δύσκολη διαδικασία, η οποία χρήζει εξειδικευμένης εμπειρίας και ως αποτέλεσμα έχουμε αριθμητικά χαρακτηριστικά ή γραφήματα, τα οποία διαχωρίζουν το αντικείμενο που αναζητούμε από τα υπόλοιπα αντικείμενα.
- το τρίτο στάδιο, επιλέγεται και εκπαιδεύεται ο αλγόριθμος που θα χρησιμοποιηθεί. Η διαδικασία αυτή είναι αυτοματοποιημένη και απαιτεί συνήθως σχετικά λίγο χρόνο για την ολοκλήρωσή της, από λίγα λεπτά έως μερικές ώρες, ανάλογα με τον όγκο των δεδομένων και τον αλγόριθμο που χρησιμοποιείται.

Με την ολοκλήρωση αυτών των σταδίων, παράγεται ένα μοντέλο αλγορίθμου με συγκεκριμένη ακρίβεια αναγνώρισης, το οποίο μπορεί να εφαρμοστεί για την αναγνώριση

του αντικειμένου σε νέες εικόνες. Έτσι, για κάθε νέα εικόνα που παρέχεται, το μοντέλο μπορεί να αναγνωρίσει το ενδιαφέρον αντικείμενο

3.2. ΜΕΘΟΔΟΛΟΓΙΑ DEEP LEARNING

Η "βαθιά" μάθηση, γνωστή και ως deep learning, αποτελεί ένα κομμάτι της μηχανικής μάθησης που βασίζεται στην τεχνολογία των τεχνητών νευρωνικών δικτύων. Όπως και η μηχανική μάθηση, η εκπαίδευση ενός αλγορίθμου στη βαθιά μάθηση καθορίζεται από τον στόχο που θέλουμε να επιτύχουμε και χωρίζεται κυρίως σε τρεις κατηγορίες: Supervised learning, Semi-supervised learning και Unsupervised learning. Όταν η ανίχνευση αντικειμένου είναι σαφής, όπως στην περίπτωση της αναγνώρισης πινακίδων αυτοκινήτων, η χρήση ενός αλγορίθμου εκπαιδευόμενου με την τεχνική Supervised είναι προτιμότερη λόγω της μεγαλύτερης ακρίβειας αποτελεσμάτων. Η εκπαίδευση ενός τέτοιου αλγορίθμου (Supervised), ολοκληρώνεται σε μόλις δύο στάδια. Το πρώτο αφορά την συλλογή δεδομένων ενώ το δεύτερο στην εκπαίδευση του μοντέλου.

- Σε αντίθεση με τους αντίστοιχους αλγορίθμους της μηχανικής μάθησης που είδαμε στο προηγούμενο κεφάλαιο, οι αλγόριθμοι deep learning απαιτούν μεγάλο όγκο δεδομένων. Όσο περισσότερα δεδομένα συλλέγονται κατά το πρώτο στάδιο, τόσο αποτελεσματικότερη είναι η επίλυση του προβλήματος.
- Η διαδικασία εκπαίδευσης στη βαθιά μάθηση είναι πολύπλοκη και απαιτητική σε χρόνο. Χρειάζεται εξειδικευμένα εργαλεία και τη διαχείριση μεγάλων όγκων δεδομένων. Ο χρόνος ολοκλήρωσης μιας τέτοιας εκπαίδευσης μπορεί να φτάσει μέχρι και μερικές ημέρες, απαιτώντας συχνά τη χρήση υψηλής επεξεργαστικής ισχύος, όπως των μονάδων επεξεργασίας γραφικών (GPU), για την επιτάχυνση της διαδικασίας. Ωστόσο, μετά την ολοκλήρωση της εκπαίδευσης, ένα μοντέλο deep learning μπορεί, εισάγοντάς του δεδομένα, να ανιχνεύει αποτελεσματικά τα αντικείμενα ενδιαφέροντος.

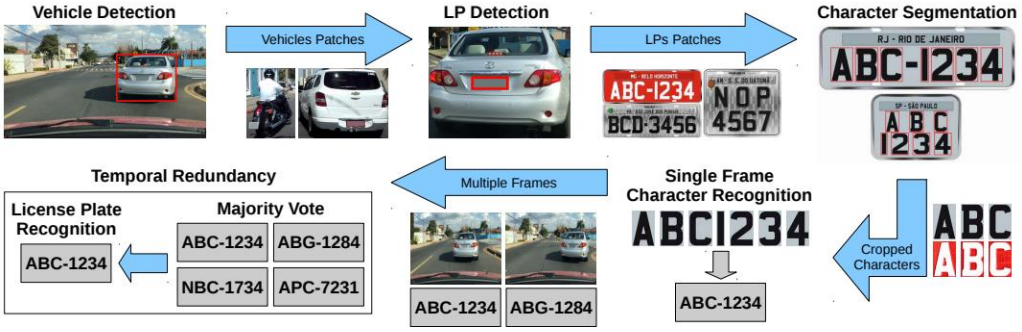
3.3. ΜΕΘΟΔΟΛΟΓΙΑ ΕΡΓΑΣΙΑΣ

Σε μία αναζήτηση σχετικών ερευνών που έχουν γίνει για αναγνώριση πινακίδων οχήματος σε πραγματικό χρόνο διαπιστώθηκε μεγάλος πλήθος. Όπως είναι φυσιολογικό, η συντριπτική πλειοψηφία, λόγω της ανάγκης αποτελεσμάτων σε πραγματικό χρόνο,

επέλεξε δίκτυα βαθιάς μάθησης (Deep learning), ενός βήματος (One-Stage Network). Χαρακτηριστικά αναφέρονται τέσσερεις μέθοδοι, με χρονολογική σειρά σύμφωνα με την δημοσίευσή τους.

- **Laroca, Rayson, et al. "A robust real-time automatic license plate recognition based on the YOLO detector." 2018** [24]

  Στην περίπτωση αυτή, το πρόβλημα της αυτόματης αναγνώρισης πινακίδων κυκλοφορίας (Automatic License Plate Recognition – ALPR) αντιμετωπίστηκε με την χρήση νευρωνικού δικτύου βαθιάς μάθησης, ενός βήματος. Συγκεκριμένα, για το πρόβλημα της αναγνώρισης πινακίδας (LP) έκανε μία σύγκριση μεταξύ του αλγορίθμου Fast-YOLO και YOLOv2 (YOLO9000), για να καταλήξει στον πρώτο λόγω του μικρού χρόνου περάτωσης έναντι του δευτέρου που είχε μεγάλο χρόνο περάτωσης αλλά μεγαλύτερη ακρίβεια. Στην συνέχεια ο ερευνητής, για την αναγνώριση των χαρακτήρων της πινακίδας, έκανε χρήση του δικτύου CR-NET [25]. Παρακάτω δίνεται εικόνα της ροής δεδομένων των εργασιών της προτεινόμενης λύσης στο πρόβλημα ALPR.



Εικόνα 14. Ροής εργασιών επίλυσης ενός προβλήματος ALPR
(ΠΗΓΗ: εικόνα A robust real-time automatic license plate recognition based on the YOLO detector) [24]

- **Hendry, Rung-Ching Chen "Automatic License Plate Recognition Via Sliding-Window Darknet-Yolo Deep Learning", 2019** [26]

  Αντίστοιχο πρόβλημα ALPR, απασχόλησε και τους ερευνητές Chen και Rung-Ching το 2019, οι οποίοι και πρότειναν την αναγνώριση της πινακίδας του οχήματος μέσω του αλγορίθμου βαθιάς μάθησης tiny YOLO, κάνοντας χρήση της πλατφόρμας SWSCD-YOLO (framework). Συγκεκριμένα, χρησιμοποιώντας το εργαλείο BBOX Label tool έφταιξε κλάσεις ανάλογες των χαρακτήρων που πρέπει να αναγνωρίζονται στην πινακίδα (25 letters), κλάση για την ίδια την πινακίδα (1

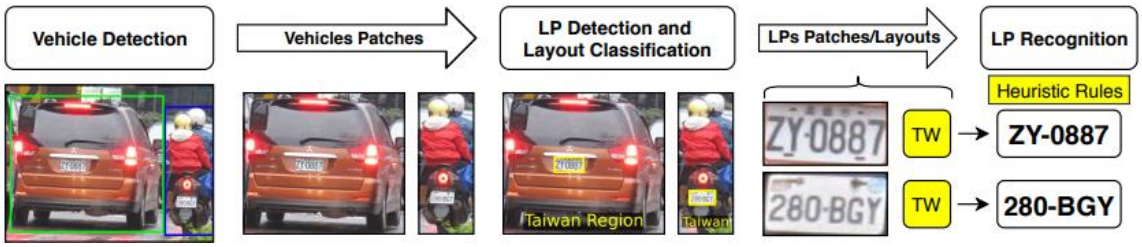
plate label) αλλά και κλάσεις για τον κάθε αριθμό (10 digits). Ως συμπέρασμα, κατέληξε ότι ο αλγόριθμος tiny YOLO είναι γρηγορότερος σε σχέση με την απλό YOLO, κάνοντας μία μικρή ακρίβεια στην ακρίβεια, ενώ η τεχνική δημιουργίας κλάσης καθενός χαρακτήρα της πινακίδας ήταν ιδιαίτερα αποτελεσματική, δίνοντας τελικά αποτελέσματα σε δεδομένα επίδειξης σε 825.81 ms και ακρίβειες, στον εντοπισμό της πινακίδας 98,22% και στην αναγνώριση 78%.

- **Riaz, Waqar, et al. "YOLO based recognition method for automatic license plate recognition." 2020** [27]

  Παρόμοια με την πρώτη περίπτωση ακολούθησε ο ερευνητής Riaz και οι λοιποί. Η διαφοροποίησή του, ήταν στο ότι χρησιμοποίησε για την αναγνώριση των πινακίδων (LP) τον αλγόριθμο YOLOv3, κάνοντας χρήση της πλατφόρμας (framework) darknet. Από το bounding box της πινακίδας, προχωρούσε στην κατάτμηση των γραμμάτων (segmentation) και στην αναγνώριση (recognition). Η παραπάνω προσέγγιση δίνει αποτελέσματα σε πολύ μικρό χρονικό διάστημα ενώ τόσο ο εντοπισμός όσο και η αναγνώριση των χαρακτήρων της πινακίδας είναι σε ικανοποιητικά επίπεδα.

- **Laroca, Rayson, et al. "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector", 2021** [28]

  Και στην περίπτωση αυτή, οι ερευνητές επέλεξαν την ίδια μεθοδολογία με μία μικρή διαφοροποίηση. Συγκεκριμένα, μέσω του αλγορίθμου YOLOv4, επιτυγχάνονταν η αναγνώριση ολόκληρου το οχήματος και εν συνεχεία, μέσω στο πλαίσιο αυτό, γινόταν η αναγνώριση της πινακίδας. Στο τελευταίο στάδιο, από την πινακίδα πραγματοποιούταν η αναγνώριση των χαρακτήρων της, με την χρήση του CR-NET [25].



Εικόνα 15. Επισκόπηση της λογικής των ερευνητών Laroca, Rayson et al
(ΠΗΓΗ: εικόνα An efficient and layout-independent automatic license plate recognition system based on the YOLO detector) [28]

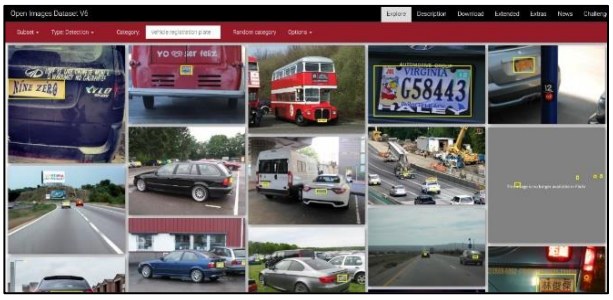
Μελετώντας όλα τα παραπάνω και για τις ανάγκες της διπλωματικής εργασίας, προκλήθηκε η επιλογή εκπαίδευσης ενός μοντέλου εξ' αρχής, το οποίο θα είναι σε θέση να εντοπίζει απευθείας τις πινακίδες των οχημάτων. Επιπρόσθετα, επιλέχθηκε ο αλγόριθμος YOLOv4, ο οποίος αποτελεί έναν state of art αλγόριθμο, και η εξ' αρχής εκπαίδευση του να γίνει με επεξεργαστική ισχύς της Google, μέσω του Colab. Τέλος, ως πλατφόρμα εκπαίδευσης χρησιμοποιήθηκε η darknet η οποία χρησιμοποιεί ως βάση δεδομένων την coco dataset η οποία διαθέτει 80 κατηγορίες αναγνώρισης αντικειμένων (classes).

4. ΥΛΟΠΟΙΗΣΗ

Σκοπός της εργασίας, ήταν η δημιουργία και εκπαίδευση μοντέλου για την αυτοματοποιημένη αναγνώριση πινακίδων οχημάτων (ALPR) με την χρήση υπολογιστικής όρασης. Για τις ανάγκες της εργασίας, ακολουθήθηκε συγκεκριμένη μεθοδολογία, η οποία αναλύεται ως ακολούθως.

4.1. ΣΥΛΛΟΓΗ ΥΛΙΚΟΥ

Για την ανάγκες της εκπαίδευσης (training set) και ελέγχου του βαθμού εκπαίδευσης (validation set) ανακτήθηκαν 1500 φωτογραφίες και 300 αντίστοιχα, στις οποίες εμφαίνονται διαφόρων τύπων πινακίδες οχημάτων. Η λήψη έγινε μέσω του εργαλείου OIDv4_ToolKit από ανοιχτή βάση δεδομένων της εταιρείας Google (open image dataset). Παράλληλα ανακτήθηκαν και οι ετικέτες (labels) της εκάστοτε φωτογραφίας καθώς ήταν διαθέσιμες.



Open Image Dataset

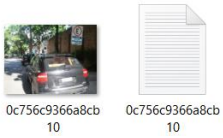
Δεδομένα

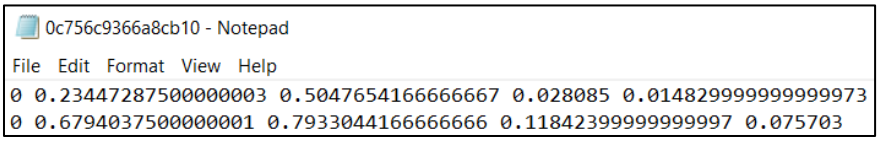
Σε κάποιες εκ των φωτογραφιών όπου οι ετικέτες δεν ήταν διαθέσιμες ή παρατηρήθηκαν σφάλματα, η διαδικασία έγινε χειροκίνητα για την κάθε μία εξ' αυτών. Συγκεκριμένα,

χρησιμοποιήθηκε το πρόγραμμα Labellmg, όπου στην ουσία, ανοίγοντας κάθε φωτογραφία, περιγράφονταν το αντικείμενο ενδιαφέροντος (για την περίπτωσή μας η πινακίδα κυκλοφορίας) εντός τετραγώνου.

Labellmg

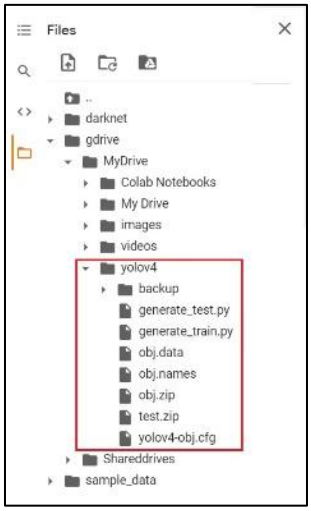
Τέλος, προκειμένου οι ετικέτες χρησιμοποιηθούν για την εκπαίδευση του αλγορίθμου YOLOv4, τροποποιήθηκαν ώστε έχουν την κατάλληλη μορφή.





4.2. ΕΚΠΑΙΔΕΥΣΗ ΜΟΝΤΕΛΟΥ

Προκειμένου ξεκινήσει η διαδικασία της εκπαίδευσης του αλγορίθμου, συνέδεσα τον λογαριασμό μου Google Drive, ώστε αποθηκεύονται απευθείας όλα τα δεδομένα που προκύπτουν από την εκπαίδευση του μοντέλου μου. Για τον σκοπό αυτό έφτιαξα έναν φάκελο με την ονομασία “yolov4” και εντός αυτού έναν έτερο φάκελο με την ονομασία “back up”. Ο λόγος της συγκεκριμένης δομής αναλύεται παρακάτω. Η χρήση του αλγορίθμου πλέον μπορεί να γίνει μέσα από το εν λόγω cloud server, με τα δεδομένα, είτε εικόνα είτε βίντεο, να μπορεί να τα φορτώσει από το ίδιο το δίκτυο (στο αντίστοιχο φάκελο image εικόνες και στο video το βιντεοληπτικό υλικό).



Εν συνεχεία και προκειμένου ξεκινήσει η διαδικασία της εκπαίδευσης, φόρτωσα τις φωτογραφίες που ανέκτησα στο προηγούμενο στάδιο, μαζί με τα txt αρχεία τους (labels) μέσα στον cloud λογαριασμό. Τόσο το αρχείο εκπαίδευσης (training set) των 1500 φωτογραφιών, όσο και αυτό της αξιολόγησης (validation set), συμπιέστηκαν και ονομάστηκαν obj.zip kai test.zip αντίστοιχα. Και τα δύο αρχεία, τοποθετήθηκαν στον φάκελο “yolov4” του Google Drive. Επίσης, στον ίδιο φάκελο φορτώθηκαν και άλλα αρχεία, η λειτουργία των οποίων δίνεται στον παρακάτω πίνακα.

Όνομα αρχείου	Πρακτική λειτουργία
obj.data	Εντός αυτού περιγράφεται η αριθμός των αντικειμένου που θέλουμε να εκπαιδεύσουμε, το όνομα της κλάσης (names), η θέση του train.txt και valid.txt καθώς και η θέση που θα αποθηκεύεται το αποτέλεσμα της εκπαίδευσης (backup).
obj.names	Περιγράφεται το όνομα της κλάσης (στην περίπτωση μας “license plate”)
generate_test.py	Κώδικας σε γλώσσα python για την εγγραφή του test.txt
generate_train.py	Κώδικας σε γλώσσα python για την εγγραφή του train.txt
Yolov4-obj.cfg	Καθορίζει τους παραμέτρους της εκπαίδευσης

Με το πέρας όλων των παραπάνω, μπορεί να ξεκινήσει η διαδικασία της εκπαίδευσης του μοντέλου. Η εκπαίδευση, ανά 100 iteration, δημιουργεί νέο αρχείο με την ονομασία “yolov4-obj_last.weights” στον φάκελο “back up” ενώ ανά 1000, δημιουργεί καινούργιο ξεχωριστώ αρχείο. Όσο πιο «εις βάθος» (iteration) επαναλήψεις πραγματοποιήσουμε, τόσο βελτιώνεται το δίκτυό μας. Στην περίπτωση μου, έκανα μέχρι 3000 iteration με την μέση ακρίβεια ανά 1000 iteration να δίνεται παρακάτω.

```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
300
 detections_count = 2409, unique_truth_count = 402
class_id = 0, name = license_plate, ap = 82.92%          (TP = 329, FP = 85)

 for conf_thresh = 0.25, precision = 0.79, recall = 0.82, F1-score = 0.81
 for conf_thresh = 0.25, TP = 329, FP = 85, FN = 73, average IoU = 59.84 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.829155, or 82.92 %
Total Detection Time: 6 Seconds
```

1000 iteration – 82.92% ακρίβεια

```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
300
 detections_count = 936, unique_truth_count = 402
class_id = 0, name = license_plate, ap = 86.55%          (TP = 339, FP = 40)

 for conf_thresh = 0.25, precision = 0.89, recall = 0.84, F1-score = 0.87
 for conf_thresh = 0.25, TP = 339, FP = 40, FN = 63, average IoU = 73.11 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.865525, or 86.55 %
Total Detection Time: 6 Seconds
```

2000 iteration – 86.55% ακρίβεια

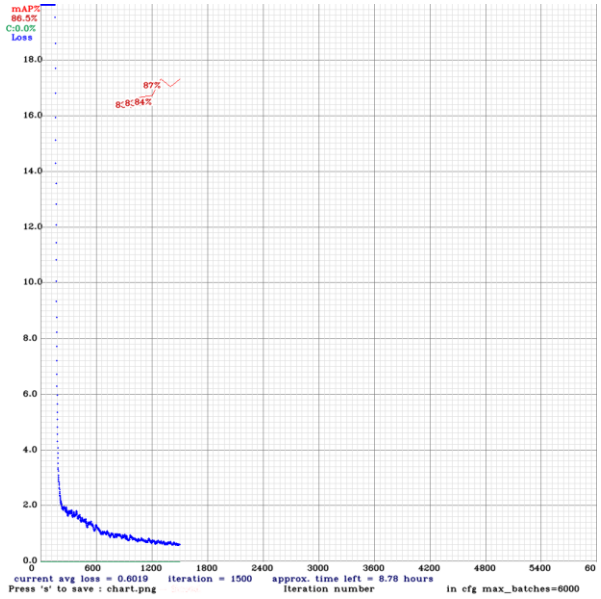
```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
300
 detections_count = 746, unique_truth_count = 402
class_id = 0, name = license_plate, ap = 86.96%          (TP = 343, FP = 32)

 for conf_thresh = 0.25, precision = 0.91, recall = 0.85, F1-score = 0.88
 for conf_thresh = 0.25, TP = 343, FP = 32, FN = 59, average IoU = 73.39 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.869648, or 86.96 %
Total Detection Time: 7 Seconds
```

3000 iteration – 86.96 % ακρίβεια

Μάλιστα παράλληλα με την εκπαίδευση του μοντέλου δημιουργείται σχεδιάγραμμα (chart) το οποίο δείχνει την πορεία εκπαίδευσης του μοντέλου.

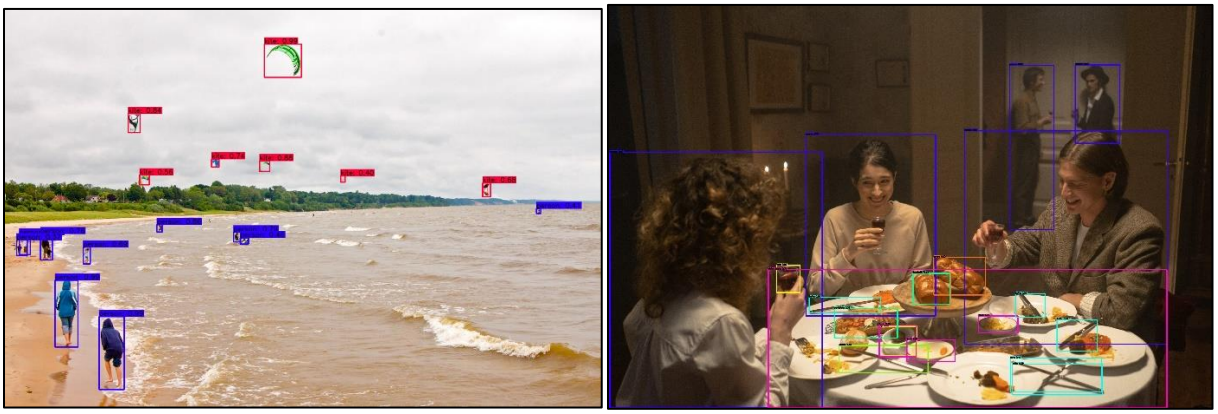


Στο τέλος της διαδικασίας, μέσα στον φάκελο “back up” δημιουργήθηκαν 4 αρχεία (yolov4-obj_1000.weights, yolov4-obj_2000.weights, yolov4-obj_3000.weights, yolov4-obj_last.weights) από τα οποία χρησιμοποίησα για την εκτέλεση και έλεγχο του μοντέλου αυτό με την μεγαλύτερη ακρίβεια.

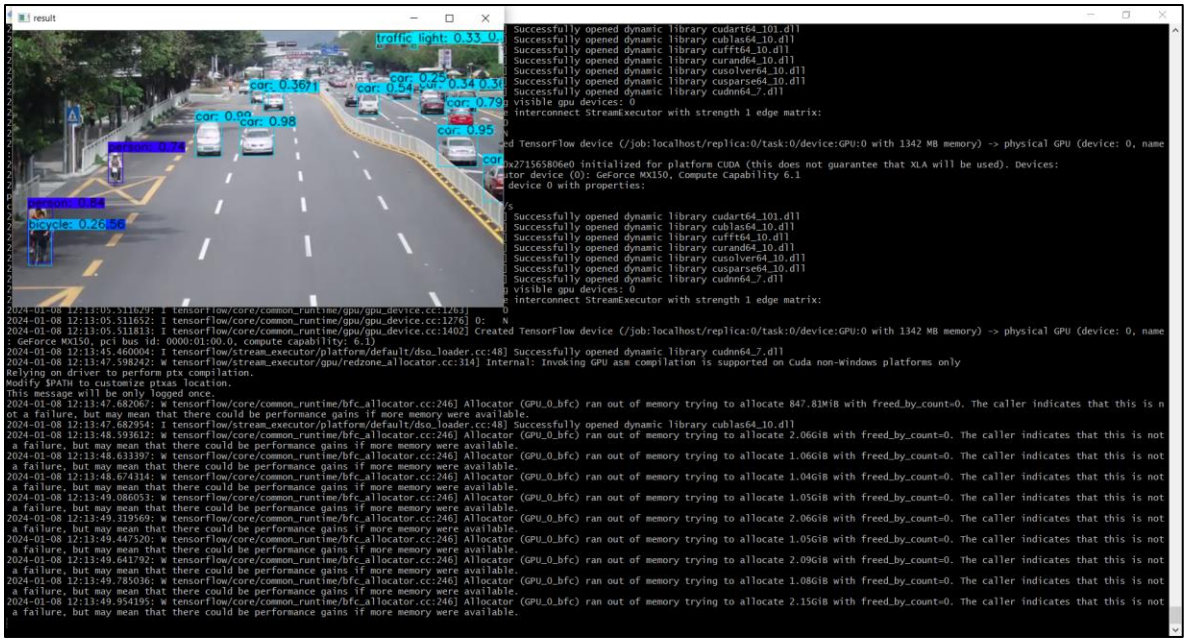
4.3. ΑΡΧΙΚΟΠΟΙΗΣΗ YOLOv4 ΜΕ TENSORFLOW

Αρχικά έτρεξα τον αλγόριθμο yolov4 για να δω ότι λειτουργεί κανονικά στον υπολογιστή μου κάνοντας κάποιες βασικές δοκιμές ανίχνευσης (detection). Μάλιστα, για λόγους απλοποίησης των διαδικασιών και ταχύτερης επεξεργασίας, χρησιμοποίησα την βιβλιοθήκη tensorflow και το εργαλείο CUDA της NVIDIA, προκειμένου μπορέσω εκμεταλλευτώ την NVIDIA κάρτα γραφικών (NVIDIA GeForce MX150 με compute capability 6.1), που διέθετε ο υπολογιστής που χρησιμοποίησα για τις δοκιμές. Προκειμένου γίνει αυτό, έπρεπε να τροποποιήσω το μοντέλο του darknet (darknet weights) σε αντίστοιχα αναγνωρίσιμο από την βιβλιοθήκη του tensorflow. Για τις ανάγκες υλοποίησης, χρησιμοποίησα τον αλγόριθμο που βρήκα στην ηλεκτρονική βιβλιοθήκη github hunglc007/tensorflow-yolov4-tflite (https://github.com/hunglc007/tensorflow-yolov4-tflite).

Στις δοκιμές για την αρχική εγκατάσταση που έκανα, όλα λειτούργησαν κανονικά, τόσο σε ανίχνευση αντικειμένων από εικόνα, όσο και ανίχνευση αντικειμένων σε βίντεο. Παραδείγματα από την ανίχνευση, δίνονται ως ακολούθως.



Εικόνα 16. Αποτελέσματα ανίχνευσης αντικειμένων σε εικόνες μέσω YOLOv4



Εικόνα 17. Αποτελέσματα ανίχνευσης αντικειμένων σε βίντεο μέσω YOLOv4

Στην συνέχεια, πραγματοποίησα δοκιμές, βάζοντας το δικό μου μοντέλο ανίχνευσης πινακίδας κυκλοφορίας οχήματος (custom weight) και περιορίζοντας τις κατηγορίες αναγνώρισης από 80 που είχα μέσω του coco dataset, σε μόλις 1 (license plate). Επιπρόσθετα, τροποποίησα το μοντέλο του darknet (darknet custom weights) σε αντίστοιχα αναγνωρίσιμο από την βιβλιοθήκη του tensorflow. Για σκοπούς αξιολόγησης ορθής λειτουργίας του μοντέλου, ανακτήθηκαν, επιπλέον 25 φωτογραφίες, από ανοιχτές πηγές διαδικτύου, με τα αποτελέσματα αναγνώρισης να δίνονται στον πίνακα που ακολουθεί.

	Φωτογραφίες
Ορθή αναγνώριση	22
Μερική αναγνώριση	3
Λανθασμένη αναγνώριση	-

Όπως πολύ εύκολα μπορεί κάποιος να διαπιστώσει, το μοντέλο για την αναγνώριση του πλαισίου των πινακίδων, εμφάνισε ικανοποιητικά αποτελέσματα. Μάλιστα, υπήρχαν φωτογραφίες, όπου τόσο η ποιότητα τους, όσο και η γωνία λήψης τους, εμφάνιζαν δυσκολίες για την αναγνώριση της πινακίδας, πλην όμως ο αλγόριθμος δεν αντιμετώπισε δυσκολίες. Παραδείγματα από την ανίχνευση πινακίδων κυκλοφορίας οχημάτων, δίνονται ως ακολούθως.

Εικόνα 18. Αναγνώριση πινακίδων κυκλοφορίας

4.4. ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ YOLOv4

Σκοπός της παρούσας διπλωματικής, πέραν της αρχικής αναγνώρισης πινακίδων κυκλοφορίας οχημάτων, είναι και sη ανάγνωση του αλφαριθμητικού μέρους της πινακίδας. Όπως αναφέρθηκε και παραπάνω, η λειτουργία αυτή, γίνεται μέσω της βιβλιοθήκης tesseract-OCR, η οποία θεωρείται από τις καλύτερες διαθέσιμες βιβλιοθήκες για αναγνώριση κειμένου. Παράλληλα, προκειμένου «βοηθηθεί» ο αλγόριθμος, στην αναγνώριση των χαρακτήρων της πινακίδας, πραγματοποιήθηκε μία σειρά ενεργειών (workflow – core/function.py).

- Το πρώτο βήμα της διαδικασίας ήταν να λάβω τις συντεταγμένες του πλαισίου (bounding box coordinates) από το YOLOv4 και απλά να εξάγουμε την υποεικόνα περιοχής εντός των ορίων του πλαισίου.

```
# get box coords
xmin, ymin, xmax, ymax = boxes[i]
# crop detection from image (take an additional 5 pixels around all edges)
cropped_img = img[int(ymin)-5:int(ymax)+5, int(xmin)-5:int(xmax)+5]
```

- Στην συνέχεια, μετατρέπω την εικόνα σε κλίμακα του γκρι.

```
# grayscale region within bounding box
gray = cv2.cvtColor(box, cv2.COLOR_RGB2GRAY)
```

- Μετά, η εικόνα υφίσταται ένα κατώφλι (thresholding) ώστε τα λευκά κείμενα να είναι σε μαύρο φόντο, ενώ εφαρμόζεται και η μέθοδος του Otsu. Αυτό το κείμενο σε λευκό φόντο βοηθά στον εντοπισμό των περιγραμμάτων της εικόνας.

```
# threshold the image using Otsus method to preprocess for tesseract
thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
```

```
# separate coordinates from box
xmin, ymin, xmax, ymax = boxes[i]
# get the subimage that makes up the bounded region and take an additional 5 pixels on each side (crop ROI)
box = img[int(ymin)-5:int(ymax)+5, int(xmin)-5:int(xmax)+5]
```

- Εφαρμόζω ένα φίλτρο Gaussian blur για την μερική αποθορυβοποίησής της

```
# perform a median blur to smooth image slightly
blur = cv2.GaussianBlur(gray, (5,5), 0)
```

- Τις περισσότερες φορές η εικόνα αυτή ήταν ιδιαίτερα μικρή με αποτέλεσμα να την μεγεθύνω με την εντολή cv2.resize() κατά τρεις φορές από το αρχικό της μέγεθος.

```
# resize image to triple the original size as tesseract does better with certain text size
blur = cv2.resize(blur, None, fx = 3, fy = 3, interpolation = cv2.INTER_CUBIC)
```

– Τέλος, τρέχω την μηχανή tesseract προκειμένου εξαχθούν από την εικόνα, οι χαρακτήρες τις πινακίδας

```
# run tesseract and convert image text to string
try:
    text = pytesseract.image_to_string(blur, config='--psm 11 --oem 3')
    print("Class: {}, Text Extracted: {}".format(class_name, text))
except:
    text = None
```

4.5. ΔΟΚΙΜΕΣ ΣΕ ΕΙΚΟΝΕΣ ΚΑΙ ΒΙΝΤΕΟ ΔΙΑΔΙΚΤΥΟΥ

Προκειμένου δούμε την αποτελεσματικότητα της λειτουργίας functions.py έγιναν αρκετές δοκιμές, τόσο σε βίντεο όσο και σε εικόνες. Στην περίπτωση των εικόνων, έγιναν δοκιμές σε τέσσερεις (4) επιλεγμένες φωτογραφίες (ιδανική λήψη), οι οποίες είχαν εμφανή τους χαρακτήρες των πινακίδων. Μάλιστα στις εικόνες που επιλέχθηκαν, οι δύο περιείχαν μόνο μία πινακίδα, ενώ οι άλλες δύο, από δύο πινακίδες. Τα αποτελέσματά τους, δίνονται παρακάτω.

Περίπτωση 1η

Περίπτωση 2η

Περίπτωση 3η

Περίπτωση 4η

Εικόνα 19. Δοκιμές αλγορίθμου σε εικόνες

Εκτός από τις παραπάνω, ιδανικής λήψεως φωτογραφίες, έγιναν δοκιμές και στο σύνολο των 25 φωτογραφιών, που χρησιμοποιήθηκαν για την αξιολόγηση της αναγνώρισης των πινακίδων. Τα αποτελέσματα δίνονται στον παρακάτω πίνακα.

	Αριθμός φωτογραφιών
Ορθή αναγνώριση	4
Μερική αναγν. – Ορθή	4
Μερική αναγν. – Λανθασμένη	3
Λανθασμένη αναγνώριση	3
Καμία αναγνώριση	11

Ορθή αναγνώριση

Μερική αναγνώριση - Ορθή

Μερική αναγνώριση - Λανθασμένη

Λανθασμένη αναγνώριση

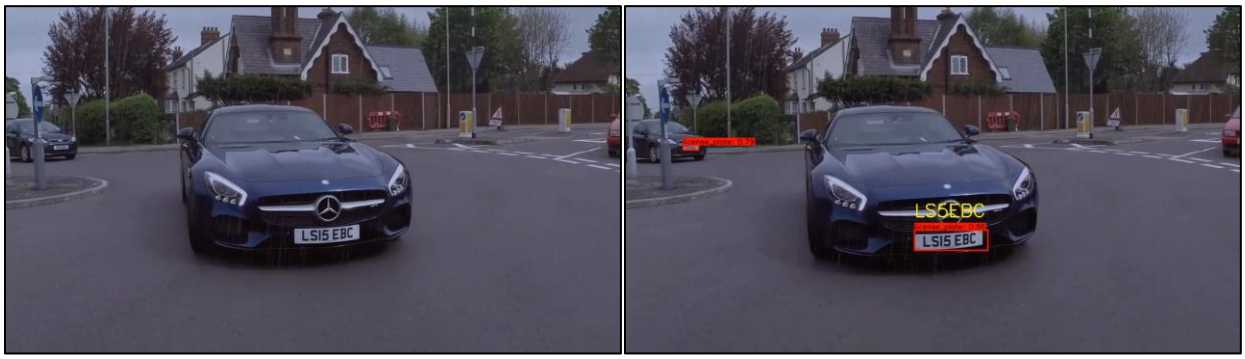
Εικόνα 20. Δοκιμές στις εικόνες αξιολόγησης, αναγνώρισης χαρακτήρων

Εικόνα 21. Σύνολο αποτελεσμάτων αναγνώρισης χαρακτήρων

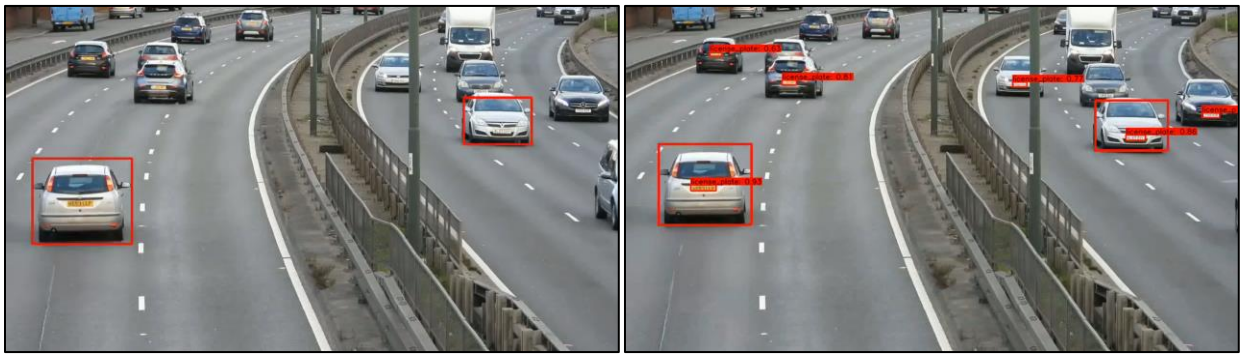
Αυτό το οποίο διαπιστώνεται στις εικόνες είναι ότι το μοντέλο αναγνώρισης πινακίδων λειτουργεί απροβλημάτιστα. Τόσο στις φωτογραφίες ένα όχημα, όσο και σε αυτές με περισσότερα, οι αναγνωρίσεις γίνονται κανονικά, έχοντας μάλιστα και μεγάλο βαθμό βεβαιότητας. Το πρόβλημα εντοπίζεται στην αναγνώριση των χαρακτήρων των

πινακίδων, στις περιπτώσεις των πολλαπλών οχημάτων, όπου εμφανίζεται μεγάλος βαθμός σφάλματος και στις λήψεις, όπου είτε είναι υπό γωνία, είτε έχουν χαμηλή ποιότητα. Αντίστοιχες δοκιμές, έγιναν σε βίντεο που ανακτήθηκαν από ανοιχτές πηγές. Συγκεκριμένα, επιλέχθηκαν ένα βίντεο από σταθερή κάμερα επιτήρησης κυκλοφορίας οχημάτων, υψηλής ανάλυσης (HD) και δύο (2) βίντεο από κάμερες εδάφους, με καλή ορατότητα στο εμπρόσθιο μέρος των οχημάτων. Όπως προέκυψε από τον αλγόριθμο, οι πινακίδες κυκλοφορίας αναγνωρίζονταν κανονικά, πλην όμως σε καμία πινακίδα δεν αναγνωρίστηκαν πλήρως οι χαρακτήρες τους, εμφανίζοντας πολλά προβλήματα. Τα αποτελέσματά τους, δίνονται παρακάτω.

Περίπτωση 1η



Περίπτωση 2η



Περίπτωση 3η

Εικόνα 22. Δοκιμές αλγορίθμου σε βίντεο

4.6. ΔΟΚΙΜΕΣ ΑΠΟ ΣμηΕΑ

Για τις δοκιμές λήψης του βιντεοληπτικού υλικού και των φωτογραφιών, χρησιμοποιήθηκε το DJI Mavic 2 Enterprise Zoom.

Εικόνα 23. DJI MAVIC 2 Enterprise Zoom
(ΠΗΓΗ https://www.dji.com/gr/mavic-2-enterprise)

Η κάμερα του ΣμηΕΑ, αποτελείται από έναν αισθητήρα 1/2.3” CMOS, 12 Megapixels. Επιπρόσθετα, έχει την δυνατότητα λήψης βίντεο μέχρι 4K με 30 καρέ το δευτερόλεπτο, ενώ παράλληλα, δίνεται η δυνατότητα οπτικής εστίασης (x2) και ψηφιακής (x3).

Ο τρόπος λήψης του βιντεοληπτικού υλικού βασίστηκε σε τρεις κατηγορίες. Η πρώτη κατηγορία δοκιμών επικεντρώθηκε στην αναγνώριση πινακίδων κυκλοφορίας σε σταθμευμένα οχήματα, χρησιμοποιώντας τόσο οπτική όσο και ψηφιακή μεγέθυνση (zoom in). Στόχος ήταν να αξιολογηθεί η απόδοση του συστήματος σε συνθήκες όπου η θέση των οχημάτων και των πινακίδων είναι σταθερή και γνωστή. Η δεύτερη κατηγορία δοκιμών αφορούσε την αναγνώριση πινακίδων κυκλοφορίας σε οχήματα, τα οποία κινούνταν με μικρή ταχύτητα σε αστικό περιβάλλον. Η λήψη των δεδομένων

πραγματοποιήθηκε με τη χρήση του ΣμηΕΑ, το οποίο πετούσε σε ύψος που επέτρεπε την σαφή καταγραφή των πινακίδων. Επιπρόσθετα, ακολουθήθηκε η ανάποδη διαδικασία με την πρώτη περίπτωση, δηλαδή από εστιασμένη περιοχή πηγαίνουμε σε μεγαλύτερη περιοχή (zoom out). Τέλος, η τρίτη περίπτωση αφορούσε οχήματα που βρισκόντουσαν σε πλήρη κίνηση. Όλο το βιντεοληπτικό που χρησιμοποιήθηκε ήταν σε ανάλυση FHD με 25 καρέ/δευτερόλεπτο. Επιπρόσθετα, τραβήχτηκαν και φωτογραφίες υψηλής ανάλυσης, για την αξιολόγηση του αλγορίθμου σε φωτογραφίες.

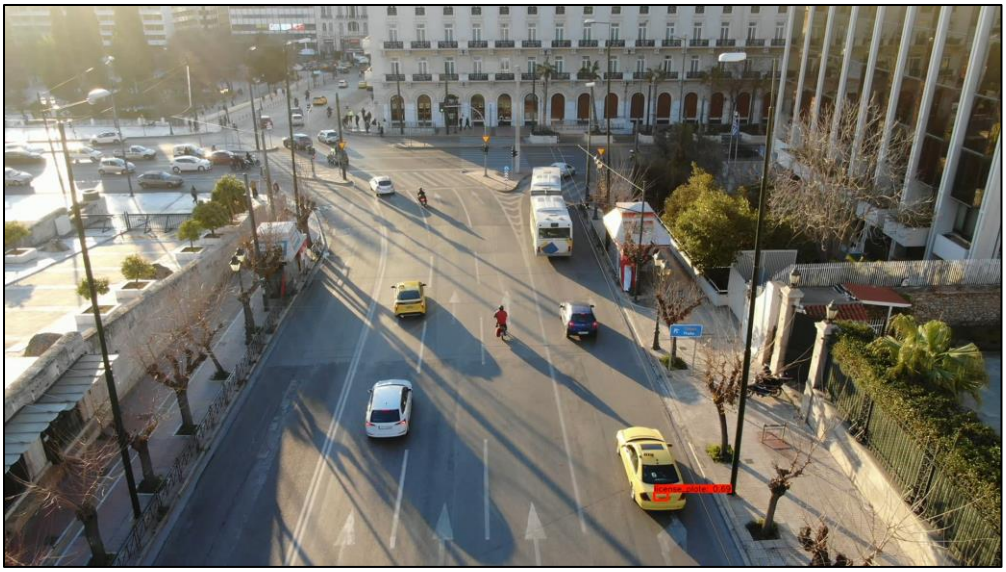
Περίπτωση 1η

ΣΧΟΛΙΑΣΜΟΣ: Στο γενικό πλάνο της λήψης, δεν αναγνωρίστηκε καμία πινακίδα. Αντιθέτως, κατά την διάρκεια της πλήρης εστίασης (x6, optical x2 – digital x3), αναγνωρίστηκαν οι πινακίδες κυκλοφορίας, χωρίς ωστόσο να εξαχθούν τα αλφαριθμητικά μέρη τους. Σημαντικό είναι να τονιστεί, πως κατά την πλήρη εστίαση (zoom), το βίντεο ήταν μη ευκρινές (blur) με αποτέλεσμα το αλφαριθμητικό μέρος των πινακίδων, να μην ήταν ευδιάκριτο.

Περίπτωση 2η

ΣΧΟΛΙΑΣΜΟΣ: Κατά την διάρκεια της εστίασης, αναγνωρίστηκαν οι πινακίδες κυκλοφορίας, πλην όμως το αλφαριθμητικό μέρος και στις δύο περιπτώσεις ήταν τελείως ανακριβής. Μάλιστα, όσο ανοίγαμε την περιοχή ενδιαφέροντος, τόσο δυσκολότερη ήταν η αναγνώριση των πινακίδων, ενώ καμία αναγνώριση αλφαριθμητικού μέρους δεν έγινε.

Περίπτωση 3η



ΣΧΟΛΙΑΣΜΟΣ: Η περίπτωση αυτή, ήταν η λήψη γενικού πλάνου, χωρίς κάποια εστίαση. Όπως προέκυψαν και στις προηγούμενες περιπτώσεις, η αναγνώριση πινακίδων κυκλοφορίας δεν ήταν δυνατή, πλην ελαχίστων περιπτώσεων. Μάλιστα, στην περίπτωση αυτή, έγινε λήψη και φωτογραφίας, προκειμένου ελεγχθεί η αποτελεσματικότητα του αλγορίθμου, με αντίστοιχα όμως αποτελέσματα. Στα παραπάνω αποτελέσματα, η πρώτη

εικόνα αποτελεί καρέ του βιντεοληπτικού υλικού, ενώ η δεύτερη η φωτογραφία που τραβήχτηκε.

Επιπρόσθετες δοκιμές έγιναν και σε 18 φωτογραφίες, υψηλής ανάλυσης, που λήφθηκαν από το ΣμηΕΑ. Τα αποτελέσματά τους δίνονται στον παρακάτω πίνακα.

	Αριθμός Φωτογραφιών
Ορθή αναγνώριση	2
Μερική αναγνώριση	16
Λανθασμένη αναγνώριση	-

	Αριθμός Φωτογραφιών
Ορθή αναγνώριση	-
Μερική αναγν. – Ορθή	2
Μερική αναγν. – Λανθασμένη	3
Λανθασμένη αναγνώριση	2
Καμία αναγνώριση	11

5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η επιτυχής ανάπτυξη ενός συστήματος ανίχνευσης και αναγνώρισης αριθμών κυκλοφορίας οχημάτων (ΑΠΚ) βασισμένου σε αλγόριθμους υπολογιστικής όρασης και μηχανικής μάθησης αποτελεί τον πυρήνα της παρούσας διπλωματικής εργασίας. Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά τα αποτελέσματα που προέκυψαν από την υλοποίηση και τις δοκιμές του συστήματος, επισημαίνοντας τις επιδόσεις και την ακρίβεια του σε διάφορα περιβάλλοντα και συνθήκες. Επιπλέον, αναφέρονται προτάσεις για μελλοντικές επεκτάσεις και βελτιώσεις που μπορούν να αυξήσουν την αποτελεσματικότητα και την ευελιξία του συστήματος, επιτρέποντας την εφαρμογή του σε ευρύτερο φάσμα πραγματικών σεναρίων

5.1. ΑΞΙΟΛΟΓΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Η αξιολόγηση των αποτελεσμάτων του συστήματος ανίχνευσης και αναγνώρισης πινακίδων πραγματοποιήθηκε με τη χρήση δύο συνόλων δεδομένων. Το πρώτο σύνολο

αποτελείται από υλικό που ανακτήθηκε από σταθερές κάμερες και ανοιχτές πηγές, ενώ το δεύτερο περιλαμβάνει βιντεοληπτικό υλικό από ΣμηΕΑ (Συστήματα μη Επανδρωμένων Αεροσκαφών), το οποίο συλλέχθηκε ειδικά για τους σκοπούς της παρούσας έρευνας. Τα αποτελέσματα δείχνουν ότι το σύστημα που αναπτύχθηκε αποδίδει ικανοποιητικά σε πραγματικό χρόνο, ιδιαίτερα κατά την ανίχνευση της θέσης της πινακίδας. Η ακρίβεια και η ταχύτητα της αναγνώρισης των χαρακτήρων εξαρτώνται από την ποιότητα των εικόνων και τις συνθήκες λήψης, με τα καλύτερα αποτελέσματα να επιτυγχάνονται υπό σταθερές και ευνοϊκές συνθήκες φωτισμού. Οι δοκιμές έδειξαν ότι το μοντέλο YOLOv4, που χρησιμοποιήθηκε για την ανίχνευση αντικειμένων, ανταποκρίνεται αποτελεσματικά και μπορεί να αναγνωρίσει με ακρίβεια τις πινακίδες σε διάφορα περιβάλλοντα. Συγκεκριμένα:

Ως προς την ακρίβεια ανίχνευσης:

- Το σύστημα ανίχνευσε επιτυχώς τις πινακίδες σε στατικές εικόνες και βίντεο με ακρίβεια που κυμαίνεται από 88%, ανάλογα με τις συνθήκες φωτισμού και την ποιότητα της εικόνας.
- Η ανίχνευση από ΣμηΕΑ παρουσίασε προκλήσεις λόγω της κίνησης και των μεταβαλλόμενων γωνιών λήψης, με την ακρίβεια να μειώνεται δραματικά σε ορισμένες περιπτώσεις.

Ως προς την αναγνώριση χαρακτήρων:

- Η βιβλιοθήκη Tesseract-OCR κατάφερε να αναγνωρίσει τους χαρακτήρες σχετική ακρίβεια που κυμαίνεται από στατικές εικόνες (32%).
- Οι αναγνωρίσεις σε βίντεο ήταν λιγότερο ακριβείς, κυρίως λόγω της θολότητας και των κινήσεων, με χαμηλή ακρίβεια. Μάλιστα στις περισσότερες περιπτώσεις, η αναγνώριση ήταν είτε αδύνατη είτε τελείως ανακριβής – μερική.

Ορισμένα προβλήματα που εντοπίστηκαν κατά τη διάρκεια της υλοποίησης και των δοκιμών μπορούν να αντιμετωπιστούν με τις ακόλουθες βελτιώσεις:

- Βελτιστοποίηση Αλγορίθμου. Χρήση διαφορετικού αλγορίθμου (YOLOv5 και άνω) για καλύτερη απόδοση σε μη ελεγχόμενα περιβάλλοντα και από ΣμηΕΑ.
- Βελτίωση Υλικού. Η κάμερα του Μη Επανδρωμένου Αεροσκάφους που χρησιμοποιήθηκε είναι περιορισμένων δυνατοτήτων, κάτι που επηρέασε την ποιότητα των εικόνων και την ακρίβεια της αναγνώρισης. Η αντικατάσταση με

κάμερα υψηλότερης ανάλυσης και καλύτερης απόδοσης σε συνθήκες χαμηλού φωτισμού θα μπορούσε να βελτιώσει σημαντικά τα αποτελέσματα.

- Επέκταση Συνόλων Δεδομένων. Το μοντέλο εκπαιδεύτηκε με φωτογραφίες από ανοιχτές πηγές, οι οποίες δεν αντικατοπτρίζουν πλήρως τις πραγματικές συνθήκες λήψης από ΜΕΑ. Η συλλογή και χρήση φωτογραφιών από ΜΕΑ, που λαμβάνονται υπό γωνία και σε διάφορες συνθήκες, θα βελτιώσει την ακρίβεια και την ανθεκτικότητα του συστήματος.

5.2. ΠΙΘΑΝΕΣ ΒΕΛΤΙΩΣΕΙΣ

Η μελλοντική εργασία μπορεί να επικεντρωθεί σε αρκετούς τομείς για τη βελτίωση του συστήματος. Αρχικά, θα μπορούσαν να εξεταστούν νέες αρχιτεκτονικές μοντέλων νευρωνικών δικτύων που πιθανώς να προσφέρουν καλύτερη ακρίβεια ή ταχύτητα ανίχνευσης και αναγνώρισης. Επιπλέον, η ενσωμάτωση προηγμένων τεχνικών προ-επεξεργασίας εικόνας, όπως η βελτίωση της ανάλυσης και η εξάλειψη θορύβου, θα μπορούσε να συμβάλει στη βελτίωση της απόδοσης του συστήματος. Ενδεικτικά δίνονται οι παρακάτω προτάσεις:

- Επέκταση Συνόλων Δεδομένων:
  - Συλλογή και χρήση μεγαλύτερων και πιο ποικιλόμορφων συνόλων δεδομένων για την εκπαίδευση του συστήματος.
  - Ενσωμάτωση δεδομένων από διάφορες πηγές και περιβάλλοντα για πιο γενικευμένες και ανθεκτικές αναγνωρίσεις.
- Εφαρμογές σε Πραγματικό Χρόνο:
  - Ανάπτυξη και δοκιμή του συστήματος σε εφαρμογές πραγματικού χρόνου, όπως συστήματα ελέγχου κυκλοφορίας και παρακολούθησης.
  - Διερεύνηση της δυνατότητας ενσωμάτωσης του συστήματος σε φορητές συσκευές και πλατφόρμες cloud για αυξημένη ευελιξία και προσβασιμότητα.
- Εξερεύνηση Νέων Τεχνολογιών:
  - Ενσωμάτωση τεχνολογιών όπως η τεχνητή νοημοσύνη και οι αλγόριθμοι βαθιάς μάθησης για περαιτέρω βελτιώσεις στην ανίχνευση και αναγνώριση.

- Διερεύνηση της χρήσης άλλων αισθητήρων και δεδομένων, όπως Lidar και ραντάρ, για ενίσχυση της απόδοσης του συστήματος σε ποικίλα περιβάλλοντα.

Με αυτές τις επεκτάσεις, το σύστημα μπορεί να βελτιωθεί σημαντικά και να χρησιμοποιηθεί σε ένα ευρύ φάσμα εφαρμογών, συμβάλλοντας στην πρόοδο της τεχνολογίας ανίχνευσης και αναγνώρισης πινακίδων κυκλοφορίας

6. ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] P. Viola και M. Jones, «"Rapid object detection using a boosted cascade of simple features",» Computer Vision and Pattern Recognition 2001, IEEE Computer Society Conference 2001.

[2] P. Viola και M. Jones, «“Robust real-time face detection",» International Journal of Computer Vision, 2004.

[3] N. Dalal και B. Triggs, «"Histograms of Oriented Gradients for Human Detection",» Computer Vision and Pattern Recognition 2005, IEEE, 2005.

[4] P. Felzenszwalb, D. McAllester και D. Ramanan, «"A discriminatively trained, multiscale, deformable part",» Computer Vision and Pattern Recognition 2008, IEEE Conference 2008.

[5] Ross Girshick κ.ά. ``Rich feature hierarchies for accurate object detection and semantic segmentation''. Στο: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014, σσ. 580–587.

[6] Jasper RR Uijlings κ.ά. ``Selective search for object recognition''. Στο: International journal of computer vision 104.2 (2013), σσ. 154–171

[7] Gandhi, R. 2018. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Towards Data Science (https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e)

[8] Girshick R. (2015), Fast R-CNN, IEEE International Conference on Computer Vision (ICCV)

[9] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99)

[10] Evaluation and Evolution of Object Detection Techniques YOLO and R-CNN. 2019. International Journal of Recent Technology and Engineering, 8(2S3), pp.824–829

[11] Dumitru Erhan κ.ά. ``Scalable object detection using deep neural networks". Στο: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014, σσ. 2147–2154.

[12] Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG-16) https://arxiv.org/pdf/1409.1556.pdf

[13] SSD: Single Shot MultiBox Detector https://arxiv.org/pdf/1512.02325.pdf

[14] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[15] Redmon J. and Farhadi A., (2016), YOLO9000: better, faster, stronger, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[16] Redmon J., Farhadi A., (2018), Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767v1.

[17] Bochkovskiy A., Wang C.-Y., Liao H.-Y. M., (2020), YOLOv4: optimal speed and accuracy of object detection, arXiv:2004.10934v1 [cs.CV]

[18] Bochkovskiy A., Wang C.-Y., Liao H.-Y. M., (2020), YOLOv4: optimal speed and accuracy of object detection, arXiv:2004.10934v1 [cs.CV]

[19] Hui, J. 2019. Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. Medium. https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2- 28b1b93e2088.

[20] YOLOv5 Open source code on Github (https://github.com/ultralytics/yolov5)

[21] Apache, "http://www.apache.org/licenses/LICENSE-2.0," 2004.

[22] Google, "code.google.com/p/tesseract-ocr/"

[23] Ray Smith Google Inc. theraysmith@gmail.com “An Overview of the Tesseract OCR Engine’

[24] Laroca, Rayson, et al. "A robust real-time automatic license plate recognition based on the YOLO detector." 2018 international joint conference on neural networks (ijcnn). IEEE, 2018.

[25] S. Montazzolli and C. R. Jung, “Real-time brazilian license plate detection and recognition using deep convolutional neural networks,” in 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images, Oct 2017, pp. 55–62

[26] Chen, Rung-Ching. "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning." Image and Vision Computing 87 (2019): 47-56.

[27] Riaz, Waqar, et al. "YOLO based recognition method for automatic license plate recognition." 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). IEEE, 2020.

[28] Laroca, Rayson, et al. "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector." IET Intelligent Transport Systems 15.4 (2021): 483-503.