



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής
Εργαστήριο Επεξεργασίας Πληροφορίας και Υπολογιστών

Δημιουργία συμπεριφορών αυτόνομου
αυτοκινήτου με χρήση προσομοιωτή και απλών
καμερών

Διπλωματική Εργασία
Μαστραλέξη Χριστίνα Μαρία
ΑΕΜ 9284

Επιβλέποντες: **Ανδρέας Λ. Συμεωνίδης**
Καθηγητής Α.Π.Θ.
Τσαρδούλιας Εμμανουήλ
Μεταδιδακτορικός Ερευνητής

Σεπτέμβριος 2022

**Δημιουργία συμπεριφορών αυτόνομου
αυτοκινήτου με χρήση προσομοιωτή και απλών
καμερών**

Μαστραλέξη Χριστίνα Μαρία
cmastral@ece.auth.gr

Σεπτέμβριος 2022

Περίληψη

Η αυτόνομη οδήγηση είναι μια τεχνολογία η οποία αναπτύσσεται και εξελίσσεται ραγδαία τα τελευταία χρόνια. Τόσο η επιστημονική κοινότητα όσο και οι κατασκευάστριες εταιρίες καταβάλλουν αξιοσημείωτες προσπάθειες για την ανάπτυξη της τεχνολογίας προς την επίτευξη της αυτόνομης οδήγησης. Η αυτόνομη τεχνολογία οχημάτων θα μπορεί να είναι σε θέση να παρέχει ορισμένα πλεονεκτήματα σε σύγκριση με τα παραδοσιακά οχήματα, καθώς στα επόμενα χρόνια, θα αυξήσει δραματικά την ασφάλεια των πολιτών, θα μειώσει το χρόνο των μεταφορών αλλά και την κυκλοφοριακή συμφόρηση των δρόμων, συντελώντας στην εύρυθμη λειτουργία του κυκλοφοριακού συστήματος.

Για την επίτευξη μια τέτοιας πραγματικότητας είναι απαραίτητη η αντίληψη του χώρου γύρω από το αυτόνομο όχημα, η προσαρμογή του στο περιβάλλον και η άμεση απόκρισή του σε αλλαγές αυτού. Το όχημα θα πρέπει να πλοηγείται με ασφάλεια στο οδικό δίκτυο και να ανταποκρίνεται ανάλογα, αναγνωρίζοντας τόσο τα στατικά και όσο και τα δυναμικά εμπόδια. Επιπλέον, θα πρέπει να υπολογίζει και να αξιολογεί σενάρια αποφάσεων και να επιλέγει την κατάλληλη ανταπόκριση σύμφωνα με τις συνθήκες. Έτσι, τα αυτόνομα οχήματα θα πρέπει να είναι εξοπλισμένα με εξειδικευμένους αισθητήρες και με το κατάλληλο σύστημα ελέγχου και λήψης αποφάσεων.

Η παρούσα διπλωματική εργασία εστιάζει στον σχεδιασμό και την υλοποίηση ενός τέτοιου συστήματος ικανού να πλοηγείται αυτόνομα σε προσπελάσιμο δρόμο αφού τον αναγνωρίσει επιτυχώς, κάνοντας χρήση οπτικής πληροφορίας που λαμβάνεται από RGB κάμερα, η οποία είναι ενσωματωμένη στο όχημα. Το όχημα κινείται σε προσομοιωμένο περιβάλλον, και συγκεκριμένα στον προσομοιωτή CARLA. Γίνεται εκτενής χρήση συναρτήσεων της βιβλιοθήκης Υπολογιστικής Όρασης OpenCV για την διαμόρφωση και αξιοποίηση της ληφθείσας οπτικής πληροφορίας. Για την αναγνώριση των δυναμικών εμποδίων η οπτική πληροφορία αναλύεται από το μοντέλο βαθιάς μάθησης YOLO (You Only Look Once), ενώ χρησιμοποιείται Οπτική Αναγνώριση Χαρακτήρων (OCR) για την εξαγωγή πληροφορίας από τα σήματα οδικής κυκλοφορίας και γίνεται ανάλυση ιστογράμματος για Αναγνώριση Χρώματος (Color Recognition) των φωτεινών σηματοδοτών. Για την κατασκευή βασικού μονοπατιού δημιουργείται ένας κατευθυνόμενος γράφος του χάρτη και χρησιμοποιείται ο αλγόριθμος A* για την αναζήτηση της βέλτιστης διαδρομής. Τέλος, η πλοήγηση επιτυγχάνεται με ένα σύστημα διατήρησης των λωρίδων (Lane Keeping Assist System) και τη χρήση ενός PID ελεγκτή.

Μαστραλέξη Χριστίνα Μαρία
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών,
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Ελλάδα
Σεπτέμβριος 2022

Abstract

Camera-driven behavioral planning for autonomous vehicles on the CARLA Simulator

Autonomous driving is developing and evolving rapidly in the recent years. Both the scientific community and industry are making significant efforts to develop the necessary sophisticated technology and ultimately to achieve autonomous driving. Autonomous vehicle technology is expected to provide several advantages compared to traditional vehicles, as in the coming years, it will dramatically increase citizens' safety, reduce transportation time and road congestion.

In order to develop such a solution, it is necessary for the autonomous vehicles to have an excellent perception of the surrounding space, be able to adapt to it, and respond instantly to any changes of the surrounding environment. The vehicle must safely navigate on the road and properly respond to static and dynamic obstacles. In addition, it should evaluate decision scenarios and then, according to the circumstances, select the appropriate response. Thus, autonomous vehicles need to be equipped with appropriate state-of-the-art sensors and also with the appropriate control and decision-making system.

This dissertation focuses on the design and development of such an autonomous driving system. The purpose of the system is to safely navigate on the road recognized via optical means. The Computer Vision OpenCV library is extensively used throughout the project for preprocessing the received RGB image, so as to make it suitable for analysis. The perception system uses the programming interface of the CARLA simulator to integrate an RGB camera. For the identification of dynamic obstacles, visual information is analyzed by the deep learning model YOLO (You Only Look Once) while using Optical Character Recognition (OCR) to export information from road traffic signals and a histogram analysis is performed for traffic light color recognition. In the global path construction system, a guided graph of the map is created and the algorithm A* is used to search for the optimal route. Finally, a Lane Keeping Assistance system and a PID controller are used for the navigation control system.

Mastralexi Christina Maria
Electrical and Computer Engineering School,
Aristotle University of Thessaloniki, Greece
September 2022

Ευχαριστίες

Ευχαριστώ θερμά τον καθηγητή κ. Ανδρέα Συμεωνίδη για την εμπιστοσύνη που μου έδειξε με την ανάθεση της παρούσας διπλωματικής εργασίας, όπως και τον Ερευνητή του τμήματος, Δρ. Εμμανουήλ Τσαρδούλια για την καθοδήγηση, τις συμβουλές του και την συνεργασία που είχαμε καθ' όλη την διάρκεια των τελευταίων μηνών.

Ευχαριστώ την οικογένειά μου για τη στήριξή τους, όχι μόνο στο διάστημα εκπόνησης της διπλωματικής, αλλά και σε όλα τα χρόνια που προηγήθηκαν.

Τέλος, ευχαριστώ τους φίλους μου, για όλες τις σπιγμές που ζήσαμε κατά την διάρκεια των σπουδών μας αλλά και για τη συνεχή υποστήριξή τους.

Περιεχόμενα

1 Εισαγωγή	8
1.1 Περιγραφή προβλήματος	11
1.2 Στόχοι της διπλωματικής εργασίας	12
1.3 Διάρθρωση της διπλωματικής εργασίας	12
2 Θεωρητικό Υπόβαθρο	14
2.1 State-of-the-art συστήματα αυτόνομης οδήγησης	14
2.1.1 Εξελίξεις στη βιομηχανία	14
2.1.2 Προσομοιωτές αυτόνομης οδήγησης	15
2.2 Αισθητήρια όργανα ενός αυτόνομου οχήματος	16
2.2.1 Σύστημα κάμερας	17
2.2.2 Αισθητήρες Radar	18
2.2.3 Αισθητήρες Lidar	18
2.3 Ψηφιακή Επεξεργασία Εικόνας	18
2.3.1 Η εικόνα ως δισδιάστατο σήμα	18
2.3.2 Γεωμετρία Εικόνας	20
2.3.3 Επεξεργασία	20
2.4 Μηχανική Μάθηση (Machine Learning)	21
2.4.1 Εισαγωγή	21
2.4.2 Βαθιά Μάθηση (Deep Learning)	21
2.4.3 Συνελικτικά Δίκτυα (Convolutional Neural Networks)	22
2.5 Μηχανική όραση (Computer Vision)	27
2.5.1 Εισαγωγή	27
2.5.2 Τυπικές μέθοδοι συστήματος	27
2.5.3 Αναγνώριση αντικειμένων (Object Detection)	28
2.6 Αλγόριθμοι και εργαλεία	34
2.6.1 Προσομοιωτής CARLA	34
2.6.2 Η βιβλιοθήκη OpenCV	36
2.6.3 Μετασχηματισμός Hough	37
2.6.4 YOLO και darknet	38
2.6.5 Tesseract OCR	41
3 Μεθοδολογία	43
3.1 Παρουσίαση υποσυστημάτων	43
3.1.1 Υποσύστημα αντίληψης (Perception subsystem)	43
3.1.2 Υποσύστημα πλοήγησης και ελέγχου (Navigation subsystem)	44
3.1.3 Υποσύστημα κατασκευής βασικού μονοπατιού (Route planning subsystem)	44

3.2 Σύστημα αντίληψης (Perception System)	44
3.2.1 Εισαγωγή	44
3.2.2 Περιγραφή υλοποίησης	44
3.2.3 Αναγνώριση Λωρίδων (Lane Detection)	45
3.2.4 Αναγνώριση Αντικειμένων (Object Detection)	51
3.2.5 Αναγνώριση των σημάτων ταχύτητας με Tesseract OCR	54
3.2.6 Ανάλυση χρωματικού ιστογράμματος	55
3.3 Σύστημα πλοήγησης (Navigation System)	56
3.3.1 Πλοήγηση με χρήση PID ελεγκτή	56
3.3.2 Πλοήγηση με διατήρηση της λωρίδας με χρήση OpenCV	59
3.4 Κατασκευή της βασικής διαδρομής	61
3.4.1 Εισαγωγή	61
3.4.2 Περιγραφή υλοποίησης	61
4 Πειράματα - Αποτελέσματα	64
4.1 Εισαγωγή	64
4.2 Παραδοχές	64
4.3 Οι μετρικές αξιολόγησης του συστήματος	65
4.4 Πειράματα και αξιολόγηση του συστήματος	66
4.4.1 Παρουσιάση των χαρτών και του περιβάλλοντος του CARLA	66
4.4.2 Αναγνώριση Αντικειμένων (Object Detection)	71
4.4.3 Οπτική Αναγνώριση Χαρακτήρων (OCR)	73
4.4.4 Αναγνώριση χρώματος των φωτεινών σηματοδοτών (Traffic Light Color Recognition)	75
4.4.5 Αναγνώριση Λωρίδων (Lane Detection)	75
4.4.6 Πλοήγηση με χρήση PID ελεγκτή	77
4.4.7 Πλοήγηση με διατήρηση της λωρίδας	78
5 Συμπεράσματα & μελλοντική εργασία	79
5.1 Γενικά συμπεράσματα	79
5.2 Προβλήματα	80
5.3 Μελλοντικές Επεκτάσεις	81

Κατάλογος Σχημάτων

1.1 Επίπεδα αυτονομίας [8]	11
2.1 Vision Van της εταιρίας Mercedes-Benz [13]	15
2.2 Περιβάλλον του προσομοιωτή LGSVL [18]	16
2.3 Τοποθέτηση αισθητήτων πάνω στο όχημα [24]	19
2.4 Διάγραμμα Venn για την σχέση των μεθόδων της Τεχνητής Νοημοσύνης [27]	22
2.5 Συνελικτικό Νευρωνικό δίκτυο [29]	23
2.6 Η διαδικασία της συνέλιξης [29]	24
2.7 Πλήρως συνδεδεμένο επίπεδο [29]	26
2.8 Παράδειγμα της μεθόδου HOG [40]	30
2.9 Δομή του δικτύου AlexNet[43]	31
2.10 Δομή του R-CNN [44]	32
2.11 Δομή του δικτύου YOLO σε σύγκριση με το δίκτυο SSD [50]	33
2.12 Απόδοση των μοντέλων ανίχνευσης αντικειμένων [51]	34
2.13 CARLA simulator [52]	35
2.14 OpenCV [53]	37
2.15 Πολικές συνταταγμένες για τον Μετασχηματισμό Hough [57]	39
2.16 Διαίρεση εικόνας σε πλέγμα YOLO [29]	40
2.17 Μετρική Intersection over Union [58]	41
3.1 Διαδικασία εντοπισμού λωρίδων	46
3.2 Μετατροπή εικόνας σε κλίμακα του γκρι	47
3.3 Μείωση του θορύβου της εικόνας με Γκαουσιανό φίλτρο	48
3.4 Εφαρμογή του Canny Edge Detector	49
3.5 Συντεταγμένες της περιοχής ενδιαφέροντος	50
3.6 Τελική αναπαράσταση των λωρίδων στην εικόνα	50
3.7 Περιβάλλον labelImg	52
3.8 Διαδικασία αναγνώρισης σημάτων ταχύτητας	54
3.9 Μάσκες για εντοπισμό του πράσινου και του κόκκινου από ιστόγραμμα .	55
3.10 Ιστόγραμμα για πράσινο, κόκκινο και πορτοκαλί χρώμα αντίστοιχα . .	55
3.11 Block διάγραμμα PID ελεγκτή [66]	56
3.12 Steady state bias αναλογικού όρου [67]	57
3.13 Γραμμή κατεύθυνσης στο περιβάλλον του προσομοιωτή CARLA	59
3.14 Υπολογισμός της γωνίας διεύθυνσης	60
3.15 Εφαρμογή A* στην Πόλη 1	62
3.16 Εφαρμογή A* στην Πόλη 2	62
3.17 Εφαρμογή A* στην Πόλη 3	62

4.1 Χάρτης της Πόλης 1	67
4.2 Περιβάλλον της Πόλης 1	67
4.3 Χάρτης της Πόλης 2	68
4.4 Περιβάλλον της Πόλης 2	68
4.5 Χάρτης της Πόλης 3	69
4.6 Περιβάλλον της Πόλης 3	69
4.7 Χάρτης της Πόλης 4	70
4.8 Περιβάλλον της Πόλης 4	70
4.9 Χάρτης της Πόλης 5	71
4.10 Περιβάλλον της Πόλης 5	71
4.11 Αναγνώριση Αντικειμένων σε εικόνες από τον προσσομοιωτή CARLA	74
4.12 Εντοπισμός λωρίδας σε συνθήκες WetCloudyNoon	75
4.13 Εντοπισμός λωρίδας σε συνθήκες MidRainyNoon	76
4.14 Εντοπισμός λωρίδας σε συνθήκες ClearNoon	76
4.15 Εντοπισμός διασταύρωσης	77
4.16 Πλοήγηση με waypoints	78

Κατάλογος Πινάκων

3.1 Οι παράμετροι του μοντέλου	53
4.1 Ο ορισμός των TP, TN, FP, FN	65
4.2 Ορισμός των μετρικών	65
4.3 Εικόνες του συνόλου εκπαίδευσης και ελέγχου	72
4.4 Labeled και Unlabeled εικόνες στο σύνολο δεδομένων	72
4.5 Οι μετρικές του μοντέλου κατά την εκπαίδευση	72
4.6 Οι μετρικές απόδοσης mAP και IoU κατά τη διάρκεια εκπαίδευσης του μοντέλου	73
4.7 Απόδοση της μεθόδου Οπτικής Αναγνώρισης Χαρακτήρων	73
4.8 Απόδοση της μεθόδου αναγνώρισης χρώματος	75
4.9 Απόδοση της μεθόδου αναγνώρισης διασταυρώσεων	76
4.10 Ποσοστό εκτέλεσης διαδρομών με το σύστημα με PID controller	77
4.11 Ποσοστό εκτέλεσης διαδρομών με το σύστημα διατήρησης λωρίδας	78

Ακρωνύμια και συντομογραφίες

ADS Automated Driving Systems

NN Artificial Neural Network

CNN Convolutional Neural Network

DNN Deep Neural Network

ROI Region Of Interest

GPS Global Positioning System

ReLU Rectified Linear Unit

OCR Optical Character Recognition

Κεφάλαιο 1

Εισαγωγή

Η ραγδαία ανάπτυξη της τεχνολογίας έχει επηρεάσει όλους τους τομείς της καθημερινής ζωής και όπως είναι φυσικό και τον τομέα των μεταφορών, που αποτελεί βασικό και αναπόσπαστο κομμάτι κάθε λειτουργικά εύρυθμης κοινωνίας. Η αναζήτηση καλύτερων, ταχύτερων και ασφαλέστερων λύσεων μεταφοράς συμβάλλουν καθοριστικά στην ανάπτυξη των κοινωνιών και η αναγκαιότητα τους είναι προφανής. Με την αύξηση του παγκόσμιου πληθυσμού, η ανάγκη των μετακινήσεων αυξάνεται ραγδαία, ενώ παρατηρείται μεγάλη πρόοδος στην ενίσχυση των οδικών υποδομών και της παραγωγής οχημάτων σε μεγάλη κλίμακα, μειώνοντας το κόστος τους και δίνοντας την δυνατότητα προσιτών οικονομικά οχημάτων, βελτιώνοντας έτσι την ποιότητα ζωής των ανθρώπων. Αντίθετα, αυτές οι ραγδαίες αλλαγές έχουν δημιουργήσει διαφορετικής φύσης προβλήματα. Οι επιπτώσεις στο περιβάλλον και η εξάντληση των ορυκτών καυσίμων είναι κάποια από τα πιο σημαντικά, ωστόσο το κυριότερο από όλα είναι η αύξηση των αυτοκινητιστικών ατυχημάτων, τα οποία οδηγούν σε τραυματισμούς και στην απώλεια ανθρώπινων ζωών. Επομένως, είναι πλέον φανερή η ανάγκη για ένα αποδοτικότερο και ασφαλέστερο σύστημα μεταφοράς.

Οι αλλαγές στον τρόπο ζωής και η ταχύτατη ανάπτυξη της τεχνολογίας έφεραν μια από τις μεγαλύτερες καινοτομίες του 21ου αιώνα, η οποία είναι το αυτόνομο όχημα. Η έλευση των αυτόνομων οχημάτων είναι μια λύση ώστε να ξεπεραστεί ένα τελευταίο και σημαντικό πρόβλημα για την βελτιστοποίηση της διαδικασίας των μεταφορών, το οποίο είναι ο ανθρώπινος παράγοντας. Ο άνθρωπος καταφέρνει και οδηγεί τα οχήματα με τη δύναμη της συνήθειας και της διαίσθησης του ανθρώπινου εγκεφάλου που είναι ικανός να αναλύσει περίπλοκες καταστάσεις και μοτίβα και να πράξει αναλόγως. Η ανίχνευση των λωρίδων στο οδόστρωμα, τα σήματα του κώδικα οδικής κυκλοφορίας, η εναλλαγή της ημέρας και της νύχτας είναι πράγματα που κάθε οδηγός αντιμετωπίζει με σχετική ευκολία. Παρόλο που υπάρχει αυτή η ικανότητα, για την πλειοψηφία των τροχαίων ατυχημάτων που συμβαίνουν στους δρόμους ευθύνεται ο άνθρωπος, δηλαδή η μη επαρκής προσοχή του οδηγού ή ο λάθος χειρισμός του οχήματος. Ένα αυτόνομο όχημα μπορεί να κινηθεί χωρίς οδηγό, το οποίο κάνει την διαδικασία οδήγησης πολύ πιο αποτελεσματική και ασφαλή, καθώς αφαιρώντας τον ανθρώπινο παράγοντα και αυτοματοποιώντας την διαδικασία οδήγησης, η ποιότητα και η απόδοση της οδήγησης μπορεί να αυξηθεί σε μεγάλο βαθμό. Τα συστήματα αυτά είναι ικανά να παρακολουθούν διαρκώς την κατάσταση του δρόμου, τα γειτονικά οχήματα και τους πεζούς και να είναι ανά πάσα στιγμή έτοιμα να πάρουν την κατάλληλη απόφαση για να οδηγήσουν

τους επιβάτες με ασφάλεια στο προορισμό τους.

Οι κατασκευάστριες εταιρίες αυτοκινήτων συναγωνίζονται για την υπεροχή στην τεχνολογία αυτόνομων οχημάτων. Ωστόσο, είναι επίσης σαφές ότι υπάρχουν ακόμα αρκετά εμπόδια που θα πρέπει να ξεπεραστούν από τα υπάρχοντα συστήματα υποστήριξης οδηγού για διασφαλιστεί ότι είναι ικανά να οδηγήσουν τόσο καλά όσο και ο άνθρωπος. Για να μπορέσουν οι υπολογιστές να εξομοιώνουν την ανθρώπινη οδηγική συμπεριφορά, μπορούμε να χρησιμοποιήσουμε αλγορίθμους μηχανικής μάθησης. Αυτό με τη σειρά του έχει τις δικές του προκλήσεις όσον αφορά τον όγκο δεδομένων που θα χρειαστεί για να εκπαιδευτούν. Για να επιτευχθεί η εκπαίδευση και να μπορέσουν να γενικεύσουν την ανθρώπινη οδηγική συμπεριφορά, θα πρέπει να υπάρξει συλλογή τεράστιου όγκου δεδομένων από ανθρώπους – οδηγούς και να καταγράφονται τα διάφορα σενάρια που αντιμετωπίζουν και η ανταπόκριση τους σε αυτά. Εκτός από τις αμιγώς τεχνολογικές προκλήσεις, η αυτόνομη οδήγηση έχει προκαλέσει επίσης έναν κοινωνικό διάλογο που βρίσκεται ακόμη σε πλήρη εξέλιξη.

Η αυτοματοποίηση της λειτουργίας ενός οχήματος, περιλαμβάνει τον συντονισμό και προγραμματισμό πολλών ανεξάρτητων λειτουργιών του. Όσες περισσότερες λειτουργίες ενός οχήματος είναι αυτοματοποιημένες και πλήρως ανεξάρτητες από τον οδηγό, τόσο μεγαλύτερο βαθμό αυτονομίας φέρει το όχημα αυτό. Σύμφωνα με την Εθνική Διοίκηση Κυκλοφοριακής Ασφάλειας Αυτοκινητοδρόμων (NHTSA) [1], την Society of Automotive Engineers (SAE) [2] αλλά και από άλλους οργανισμούς, όπως οι German Association of the Automotive Industry (VDA) [3] και Germany Federal Highway Research Institute [4] της Γερμανίας, δημιουργήθηκε από τον Ιανουάριο του 2014, μια τεχνική ταξινόμησης που περιγράφει ποιες λειτουργίες εκτελεί το όχημα και ποιες πρέπει να εκπληρώσει ο οδηγός. Οι οδηγίες αυτές κατατάσσουν την αυτονομία του οχήματος από το επίπεδο 0 έως το επίπεδο 5. Σήμερα, οι περισσότερες αυτοκινητοβιομηχανίες κατασκευάζουν αυτοκίνητα με τεχνολογίες επιπέδου 2 και 3. Αυτή η ταξινόμηση περιλαμβάνει ενέργειες ορισμού κατεύθυνσης και ταχύτητας [5] [6] [7]. Τα επίπεδα αυτά είναι:

- **Επίπεδο 0: Μη αυτοματισμός (No automation)**

Το επίπεδο 0 καλύπτει ένα ευρύ φάσμα παλαιότερων οχημάτων, που δεν έχουν καθόλου βοηθήματα για την οδήγηση. Σε αυτό το επίπεδο το όχημα είναι πλήρως ελεγχόμενο από τον οδηγό, δεν ασκείται κανένας έλεγχος στην ταχύτητα και την κατεύθυνση του οχήματος από το ίδιο το όχημα.

- **Επίπεδο 1: Αυτοματισμός Συγκεκριμένων Λειτουργιών (Function-specific automation)**

Σε αυτό το επίπεδο ο έλεγχος του οχήματος βρίσκεται και πάλι στον οδηγό και αποτελεί το κατώτερο επίπεδο αυτονομίας. Περιλαμβάνεται η αυτοματοποίηση συγκεκριμένων λειτουργιών, όπως ο έλεγχος πλεύσης του οχήματος (cruise control), η καθοδήγηση των οριογραμμών (lane guidance) και η υποβοήθηση στο παρκάρισμα (automated parallel parking). Οι λειτουργίες αυτές αποσκοπούν στην διευκόλυνση του οδηγού.

- **Επίπεδο 2: Αυτοματισμός Συνδυαζόμενων Λειτουργιών (Combined autonomous functions)**

Και σε αυτό το επίπεδο ο έλεγχος παραμένει στον οδηγό, αλλά το όχημα μπο-

ρεί να ελέγχει αυτόνομα ταυτόχρονα την κατεύθυνση και την ταχύτητα του υπό συγκεκριμένες συνθήκες. Οι οδηγοί είναι υπεύθυνοι να ελέγχουν τις συνθήκες κυκλοφορίας στην οδό και να είναι σε θέση να αναλάβουν τον έλεγχο του οχήματος ανά πάσα στιγμή. Η ευθύνη του οχήματος παραμένει στον οδηγό και οι παραπάνω λειτουργίες τον υποβοηθούν.

- **Επίπεδο 3: Περιορισμένος Αυτοματισμός (Conditional automation)**

Τα οχήματα που βρίσκονται σε αυτό το επίπεδο αυτοματοποίησης διαθέτουν ένα προηγμένο σύστημα οδήγησης (ADS), το οποίο μπορεί να επιτελέσει το σύνολο των λειτουργιών και διαδικασιών που απαιτούνται για την αυτόνομη οδήγηση του οχήματος από ένα σημείο εκκίνησης προς ένα σημείο προορισμού. Το όχημα μπορεί να ανταποκριθεί στα δυναμικά γεγονότα του περιβάλλοντος και να εκτελέσει πλήρως διαδικασίες, όπως η αλλαγή λωρίδας ή να επιταχύνει με σκοπό την προσπέραση ενός αργά κινούμενο οχήματος. Ο οδηγός σε αυτή την περίπτωση θα πρέπει να πάρει τον έλεγχο αν το όχημα αποτύχει να φέρει εις πέρας την διαδικασία. Σε αυτό το επίπεδο αυτονομίας του οχήματος, οι υπολογιστές επιστρέφουν τον έλεγχο στον οδηγό με μία ειδοποίηση.

- **Επίπεδο 4: Υψηλό Επίπεδο Αυτοματοποίησης (High automation)**

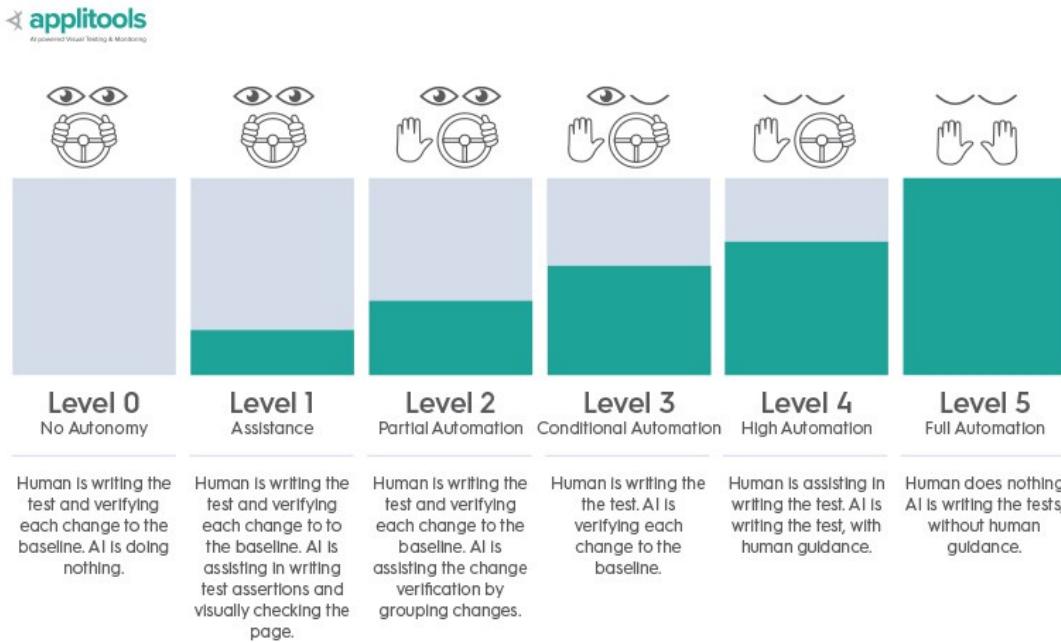
Το επίπεδο αυτό βρίσκεται πολύ κοντά με το προηγούμενο επίπεδο. Σε αυτό, πρέπει να υπάρχει ένα σύστημα με το οποίο το αυτοκίνητο είναι σε θέση να οδηγείται από μόνο του σχεδόν σε όλες τις περιστάσεις, η διαφορά όμως είναι ότι το επίπεδο αυτοματοποίησης είναι πολύ πιο ασφαλές. Αυτό περιλαμβάνει την οδήγηση σε αυτοκινητόδρομο με οποιαδήποτε ταχύτητα εντός ορίων, καθώς και την οδήγηση σε πόλη όπου οι οδικοί άξονες είναι πολύ πιο περίπλοκοι. Τα οχήματα μπορούν να πραγματοποιήσουν όλες τις οδηγικές λειτουργίες και να επιβλέψουν τις κυκλοφοριακές συνθήκες καθ' όλη τη διάρκεια ενός ταξιδιού, σε προκαθορισμένες όμως διαδρομές. Παρά την ικανότητα του οχήματος να μπορεί να πλοηγείται αυτόνομα, ο οδηγός πρέπει να μπορεί να πάρει τον έλεγχο του οχήματος οποιαδήποτε στιγμή το θελήσει.

- **Επίπεδο 5: Πλήρης Αυτοματοποίηση (Full Automation)**

Σε αυτό το επίπεδο το σύστημα (ADS) του οχήματος είναι σε θέση να πραγματοποιεί όλες τις λειτουργίες της οδήγησης και να παρακολουθεί τις κυκλοφοριακές συνθήκες του περιβάλλοντος στο οποίο κινείται, κάτω από οποιεσδήποτε συνθήκες, αντικαθιστώντας απόλυτα τον οδηγό. Τα οχήματα που λειτουργούν σε αυτό το επίπεδο δεν περιορίζονται σε συγκεκριμένο γεωγραφικό χώρο και μπορούν να λειτουργήσουν πλήρως αυτόνομα σε οποιοδήποτε σενάριο. Ο οδηγός του οχήματος δεν απαιτείται να έχει την οποιαδήποτε συνδρομή στη διαδικασία της οδήγησης πέραν από τον ορισμό του επιθυμητού προορισμού.

Στην εικόνα 1.1 παρουσιάζονται κατηγοριοποιημένα τα επίπεδα αυτοματοποίησης που περιγράφηκαν προηγουμένως.

Τα αυτόνομα οχήματα, όπως μπορεί να γίνει κατανοητό και από τα επίπεδα αυτονομίας που παρουσιάστηκαν, μπορούν να προσφέρουν καινοτόμες λύσεις και δυνατότητες στο χώρο της οδήγησης. Χωρίς αμφιβολία, τα οφέλη από την αυτόνομη οδήγηση μπορεί να είναι τεράστια. Παρ' όλα αυτά όμως, υπάρχει και η αντίθετη άποψη η οποία υποστηρίζει ότι το κόστος απόκτησης και λειτουργίας μπορεί να είναι υψηλότερα



Σχήμα 1.1: Επίπεδα αυτονομίας [8]

από τις υποθέσεις των αισιόδοξων προβλέψεων [9] [10] [11]. Τα πλήρως αυτόνομα οχήματα αναμένεται να αναπτυχθούν τεχνολογικά μέσα σε αυτή τη δεκαετία, αλλά θα χρειαστούν κάποια χρόνια για την προσαρμογή τους στις μεταφορικές υποδομές και την κοινωνία.

1.1 Περιγραφή προβλήματος

Τα προηγούμενα χρόνια έχει μελετηθεί εκτενώς το πεδίο της αυτόνομης οδήγησης, με αποτέλεσμα να έχουν αναπτυχθεί τεχνικές που επιλύουν με μεγάλη επιτυχία και ακρίβεια πολλά από προβλήματα που προκύπτουν από αυτές τις διαδικασίες. Ωστόσο, το πρόβλημα της οπτικής αντίληψης του περιβάλλοντος με κάμερα από αυτόνομα οχήματα σε ένα μη περιορισμένο, δυναμικό και δομημένο περιβάλλον με κανόνες κυκλοφορίας, όπως είναι το οδικό δίκτυο, είναι ένα πρόβλημα με περίπλοκη λύση.

Ένα όχημα μέσα σε ένα οδικό δίκτυο θα πρέπει να εντοπίζει την θέση των υπόλοιπων οχημάτων αλλά και των πεζών, ενώ παράλληλα κατά την κίνηση του θα πρέπει να αναγνωρίζει την σήμανση των δρόμων και να σέβεται τους κανόνες κυκλοφορίας. Η ταχύτητα του οχήματος ανάλογα με το περιβάλλον στο οποίο κινείται μπορεί να είναι αρκετά μεγάλη κάνοντας την διαδικασία ακόμα πιο δύσκολη. Ο όγκος των δεδομένων που πρέπει να επεξεργαστεί άμεσα το όχημα αυξάνεται ραγδαία με την αύξηση της ταχύτητας του οχήματος και κατ' επέκταση και ο χρόνος επεξεργασίας τους.

Στη βιβλιογραφία υπάρχει πληθώρα μεθόδων που περιγράφουν μερικώς και την διαδικασία για την πλοιόγηση του οχήματος, τον έλεγχο και τη λήψη αποφάσεων. Ωστόσο, δεν υπάρχει κάποια αποδειγμένη αρχιτεκτονική ή μοντέλο που να ορίζει με ποιον τρόπο θα πρέπει να κατασκευαστεί ένα τέτοιο σύστημα. Το πρόβλημα συνεχίζει

να βρίσκεται υπό μελέτη και να αναπτύσσονται συνεχώς νέες εξειδικευμένες τεχνικές τόσο για την ανάπτυξη των επί μέρους υποσυστημάτων αντίληψης και πλοήγησης όσο και για τον μεταξύ τους συνδυασμό.

1.2 Στόχοι της διπλωματικής εργασίας

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η παρουσίαση ενός συστήματος αυτόνομης οδήγησης, με χρήση της κάμερας ως βασικού αισθητήρα, το οποίο εκτελεί τρεις επιμέρους διεργασίες. Το όχημα θα πρέπει να μπορεί να αναγνωρίζει τις λωρίδες στο οδόστρωμα, να εντοπίζει τα εμπόδια στο περιβάλλον πλοήγησης και να ερμηνεύει τα σήματα οδικής κυκλοφορίας και τους φωτεινούς σηματοδότες. Επιπλέον, θα πρέπει να μπορεί να διατηρεί την θέση του ανάμεσα στις λωρίδες και να πλοηγείται από ένα αρχικό σημείο σε ένα τελικό μέσα σε έναν εκ των προτέρων γνωστό χάρτη. Το σύστημα χωρίζεται σε μικρότερα υποσυστήματα καθένα από τα οποία θα πρέπει:

- **Σύστημα Αντίληψης:** να αντιλαμβάνεται τα δυναμικά και στατικά εμπόδια του περιβάλλοντος του με χρήση των εικόνων που λαμβάνει από την κάμερα, να ερμηνεύει την πληροφορία, για παράδειγμα να μπορεί να προβλέψει τι είδους εμπόδιο συναντά, τι χρώμα έχουν οι φωτεινοί σηματοδότες, ποιο σήμα οδικής κυκλοφορίας συναντά, και ακόμα να μπορεί να εντοπίσει τις διαχωριστικές λωρίδες στον δρόμο, καθώς και τις διασταυρώσεις.
- **Σύστημα Πλοήγησης:** χρησιμοποιώντας της πληροφορίες που λήφθηκαν από το σύστημα αντίληψης σχετικά με την θέση του στο οδόστρωμα αλλά και την τοποθεσία του στον χάρτη διαδρομής, να επιτύχει πλοήγηση διατηρώντας τον έλεγχο ανάμεσα στα όρια των λωρίδων και να μπορεί να πλοηγηθεί σε έναν τελικό προορισμό, δίνοντας τιμές για τον πλευρικό και επιμήκη έλεγχο του οχήματος.
- **Σύστημα χάραξης βασικού μονοπατιού:** να μπορεί με βάση το αρχικό σημείο εκκίνησης να σχεδιάσει στον χάρτη το βέλτιστο βασικό μονοπάτι που θα το οδηγήσει στον τελικό προορισμό.

Η συνεισφορά της διπλωματικής στοχεύει στην μελέτη των τεχνολογιών για τα επί μέρους συστήματα με σκοπό την ανάπτυξη ενός όσο το δυνατόν πιο λειτουργικού συστήματος αυτόνομης οδήγησης.

1.3 Διάρθρωση της διπλωματικής εργασίας

Στο παρόν κεφάλαιο έγινε μια σύντομη εισαγωγή παρουσιάζοντας το κίνητρο, το πρόβλημα και τους στόχους της παρούσας διπλωματικής εργασίας. Η εργασία διαιρέθηκε σε επιπλέον τέσσερα κεφάλαια, ως εξής:

- **Στο Κεφάλαιο 2** Γίνεται ανασκόπηση της ερευνητικής περιοχής που αφορά την τρέχουσα τεχνολογία των αυτόνομων οχημάτων, των κατηγοριών των συστημάτων και κάποιες βασικές αναφορές σε μεθόδους, έννοιες και διαδικασίες που είναι απαραίτητες για την κατανόηση της εργασίας. Επίσης, γίνεται μια εκτενής παρουσίαση των ήδη υπαρχουσών ερευνητικών προσπαθειών για την προσέγγιση προβλημάτων, παρεμφερών με αυτό της παρούσας διπλωματικής.

- **Στο Κεφάλαιο 3** Πλήρης περιγραφή της υλοποίησης του συστήματος αυτόνομης οδήγησης και των μοντέλων που αποτελούν τον πυρήνα της εργασίας, παραθέτοντας τις θεωρητικές βάσεις και σχολιάζοντας συγκεκριμένα κομμάτια της υλοποίησης των επιμέρους συστημάτων.
- **Στο Κεφάλαιο 4** Παρουσιάζεται αναλυτικά η μεθοδολογία των πειραμάτων και τα αποτελέσματα καθώς και τα τελικά συμπεράσματα. Τα πειράματα πραγματοποιήθηκαν με χρήση του προσομοιωτή CARLA.
- **Στο Κεφάλαιο 5** Αναφέρονται να συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας διπλωματικής και τα προβλήματα που παρουσιάστηκαν κατά την διάρκεια της ανάπτυξης του συστήματος αυτόνομης οδήγησης, ενώ παραθέτονται και μελλοντικά βήματα, αλλαγές και επεκτάσεις για τον περαιτέρω εμπλουτισμό του συστήματος που υλοποιήθηκε.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Το Κεφάλαιο 2 αποσκοπεί στο να παρουσιαστούν στον αναγνώστη έννοιες, τεχνολογίες και συστήματα που είναι θεμελιώδη για την κατανόηση των θεμάτων που αναλύονται στα πλαίσια της παρούσας διπλωματικής εργασίας. Παρουσιάζονται οι σύγχρονες τεχνολογίες State-of-the-art στο τομέα της αυτόνομης οδήγησης, πραγματοποιείται μια ανάλυση των χρησιμών εννοιών που πραγματεύεται η εργασία και τέλος, γίνεται αναφορά των εργαλείων και των μεθόδων που θα χρησιμοποιηθούν.

2.1 State-of-the-art συστήματα αυτόνομης οδήγησης

Τα συστήματα αυτόνομης οδήγησης (Automated Driving Systems) είναι ρομποτικά συστήματα που λειτουργούν σε περίπλοκα και δυναμικά περιβάλλοντα. Ως εκ τούτου, υπάρχουν πολλά σενάρια που περιέχουν άλυτα προβλήματα. Αυτή η ενότητα ασχολείται με τις υψηλού επιπέδου προκλήσεις που υπάρχουν γενικά στην αυτόνομη οδήγηση και έχει ως στόχο να παρουσιάσει ερευνητικές προσπάθειες πάνω σε βασικά θέματα που αφορούν τα αυτόνομα οχήματα.

2.1.1 Εξελίξεις στη βιομηχανία

Πολλές αυτοκινητοβιομηχανίες έχουν πραγματοποιήσει αρκετές επιτυχημένες δοκιμές στη λειτουργία του αυτόνομου οχήματος και πλέον το όραμα τους έχει επικεντρωθεί στο πως θα αναδειχτεί η εμπιστοσύνη μεταξύ των ανθρώπων και των οχημάτων αυτών. Τέτοια είναι, για παράδειγμα, το ερευνητικό όχημα F 015 [12] από την Mercedes-Benz, που περιέχει μεγάλες LED οθόνες στο εμπρόσθιο και πίσω μέρος του, ένα σύστημα λέιζερ απεικόνισης, καθώς και ένα σύνολο ήχων και ηχητικών εντολών, το οποίο χρησιμοποιείται ως μέσο επικοινωνίας βασισμένο στη Βαθιά Μάθηση (Deep Learning). Από την ίδια εταιρία, έχουμε το Vision Van [13], ένα ηλεκτρικό όχημα που χρησιμοποιείται για τη μεταφορά δεμάτων (Σχήμα 2.1).

Η εταιρεία Tesla έχει επίσης πραγματοποιήσει αρκετές επιτυχημένες προσπάθειες. Τα τελευταία χρόνια έχει δημιουργήσει το Model S και το Model X [14], με αρκετές νέες δυνατότητες, όπως αυτόματη αλλαγή λωρίδας, αυτόματο παρκάρισμα και σύστημα auto-steer, το οποίο διατηρεί το όχημα με σταθερή ταχύτητα στη λωρίδα που κινείται, κάνοντας χρήση των χαρτών υψηλής ευκρίνειας GPS, με τους αισθητήρες να αναγνωρίζουν τις κατάλληλες ενδείξεις προκειμένου να γνωρίζει το αυτοκίνητο πότε είναι μέσα



Σχήμα 2.1: Vision Van της εταιρίας Mercedes-Benz [13]

σε μια λωρίδα.

Τέλος, από το 2014 η εταιρία Google έχει παρουσιάσει έναν μεγάλο αριθμό λειτουργικών προγραμμάτων οδήγησης του αυτοκινήτου τους Google self-driving car, δοκιμάζοντας το σε πραγματικό οδικό περιβάλλον. Το σύστημα οδηγεί το όχημα στο όριο ταχύτητας που έχει αποθηκευτεί στους χάρτες και διατηρεί την απόστασή του από τα άλλα οχήματα που χρησιμοποιούν το σύστημα των αισθητήρων. Παρέχει μια λειτουργία που επιτρέπει σε έναν οδηγό να πάρει τον έλεγχο του αυτοκινήτου, ο οποίος το μόνο που πρέπει να κάνει είναι να πατήσει το φρένο ή την περιστροφή του τροχού, παρόμοια με τα συστήματα ελέγχου ταχύτητας ταξιδιού που ήδη βρίσκονται σε πολλά αυτοκίνητα σήμερα.

2.1.2 Προσομοιωτές αυτόνομης οδήγησης

Τα τελευταία χρόνια παρατηρείται μεγάλη ανάπτυξη στον τομέα των προσομοιωτών αυτόνομης οδήγησης, καθώς η ανάγκη δοκιμής και αξιολόγησης νέων μεθόδων και τεχνολογιών αυξάνεται.

Το 2015, δημιουργείται το Autoware [15], μια πλατφόρμα ανοιχτού κώδικα που χρησιμοποιείται για προσομοίωση αυτόνομων οχημάτων. Οι δυνατότητες του είναι εξειδικευμένες για πόλεις, ωστόσο υποστηρίζει επίσης αυτοκινητόδρομους και άλλες περιοχές. Ο προσομοιωτής είναι κατασκευασμένος για να παρέχει τρισδιάστατα δυναμικά και στατικά εμπόδια και υποστηρίζει πλήθος αισθητήρων, όπως κάμερες RGB, LiDAR, Radar και IMU. Είναι συμβατός με το λογισμικό ROS, είναι γραμμένος σε κώδικα Python και υποστηρίζει μόνο το λειτουργικό σύστημα Linux.

Το 2017 η εταιρία Microsoft κυκλοφόρησε τον προσομοιωτή αυτόνομης οδήγησης AirSim [16], κατασκευασμένο πάνω από τη μηχανή παιχνιδιών Unreal Engine. Το συγκεκριμένο λογισμικό επέτρεψε τον έλεγχο μεθόδων Βαθιάς Μάθησης και Μηχανικής Όρασης σε προσομοιωμένο περιβάλλον και αποτέλεσε ένα εξαιρετικό παράδειγμα για το πώς ένα εικονικό περιβάλλον μπορεί να μειώσει τον κίνδυνο, τόσο για το αυτοκίνητο, όσο και για αυτούς που βρίσκονται γύρω από αυτό.

Τον ίδιο χρόνο, δημιουργήθηκε ο CARLA simulator [17] από το Computer Vision Centre με την συνεισφορά του Toyota Research Institute. Πρόκειται για προσομοιωτή οδήγησης ανοιχτού κώδικα, ο οποίος στηρίζεται στο Unreal Engine για την εκτέλεση της προσομοίωσης και χρησιμοποιεί το πρότυπο OpenDRIVE (ανοιχτή μορφή αρχείου για τη λογική περιγραφή των οδικών δικτύων) για τον σχεδιασμό των δρόμων και κανόνων οδήγησης. Ο έλεγχος της προσομοίωσης παρέχεται μέσω ενός API (Application Programming Interface ή Διεπαφή προγραμματισμού εφαρμογών).



Σχήμα 2.2: Περιβάλλον του προσομοιωτή LGSVL [18]

Το 2019 η εταιρία LG [18] αξιοποιεί την κορυφαία πλατφόρμα ανάπτυξης 3D της Unity, και συγκεκριμένα το Unity Simulation. Αυτό το λογισμικό επιτρέπει την τέλεση πολλαπλών εκτελέσεών του σε κλίμακα και υποστηρίζεται από την υποδομή του Google Cloud, μειώνοντας με αυτόν τον τρόπο τα εμπόδια της εκπαίδευσης και το πλήθος των δοκιμών. Κατά συνέπεια, επιτυγχάνει την επικύρωση νέων προϊόντων και υπηρεσιών αυτόνομων οχημάτων, χρησιμοποιώντας ένα εικονικό περιβάλλον για τις δοκιμές τους. Ο καινοτόμος προσομοιωτής LGSVL, το περιβάλλον του οποίου φαίνεται στην εικόνα 2.2, μπορεί να εκτελέσει παράλληλα πολλά οδηγικά σενάρια, δημιουργώντας ψηφιακά αντίγραφα του πραγματικού κόσμου, με μοντέλα αισθητήρων ακριβείας για την εκπαίδευση των αυτόνομων οχημάτων.

2.2 Αισθητήρια όργανα ενός αυτόνομου οχήματος

Η δυνατότητα αντίληψης σε ένα αυτόνομο όχημα είναι πρωτεύουσας σημασίας για την αξιόπιστη και ασφαλή λειτουργία του, καθώς τα δεδομένα που λαμβάνονται από αυτές τις διαδικασίες τροφοδοτούν τη διαδικασία λήψης αποφάσεων. Τα αισθητήρια όργανα που χρησιμοποιούνται επιτρέπουν στο όχημα να κάνει μετρήσεις και παρατηρήσεις για συγκεκριμένες μεταβλητές του περιβάλλοντος. Κάποιοι αισθητήρες δίνουν την δυνατότητα μέτρησης της απόστασης ή της θέσης των αντικειμένων γύρω από μια περιοχή του οχήματος, ενώ κάποιοι άλλοι για την μέτρηση της ταχύτητας τους. Οι αισθητήρες που χρησιμοποιούνται παρουσιάζουν διάφορα πλεονεκτήματα ή

μειονεκτήματα ανάλογα με το περιβάλλον και τις συνθήκες οδήγησης για τις οποίες προορίζονται. Οι συνθήκες καιρού, όπως βροχή, χιόνι, ήλιος και ομίχλη μπορούν να επηρεάσουν σε μεγάλο βαθμό την ορατότητα, με αποτέλεσμα να επηρεάζουν και την απόδοση.

Ακολουθούν τα συστήματα αισθητήρων που συναντάμε πιο συχνά σε αυτόνομα οχήματα. Τα συγκεκριμένα συστήματα έχουν καθοριστικό ρόλο για την αντίληψη που έχει το όχημα για το περιβάλλον του και χρησιμοποιούνται ευρέως [19].

2.2.1 Σύστημα κάμερας

Τα αυτόνομα οχήματα έχουν συχνά βιντεοκάμερες και αισθητήρες για να βλέπουν και να ερμηνεύουν τα αντικείμενα γύρω από το όχημα, με το ίδιο τρόπο που οι άνθρωποι χρησιμοποιούν την όραση τους. Οι κάμερες έχουν την δυνατότητα να ανιχνεύουν αντικείμενα, να τα ταξινομούν και να καθορίζουν τις αποστάσεις μεταξύ αυτών και του οχήματος [20]. Για παράδειγμα, οι κάμερες μπορούν να αναγνωρίσουν αυτοκίνητα, πεζούς, ποδηλάτες, πινακίδες και σήματα κυκλοφορίας, φωτεινούς σηματοδότες και προστατευτικά κιγκλιδώματα.

Οι κάμερες τοποθετούνται στην οροφή ή στο πάνω μέρος του παρμπρίζ, για να αναπαραστήσουν το οπτικό πεδίο του οδηγού, ενώ μπορεί να τοποθετηθούν συμπληρωματικές κάμερες στο πίσω μέρος του οχήματος. Στο Σχήμα 2.3 παρουσιάζονται οι συνήθεις θέσεις των αισθητήρων πάνω σε ένα όχημα.

Η κάμερα ως βασικό αισθητήριο όργανο

Ένα από τα πιο επιθυμητά χαρακτηριστικά για τα αυτόνομα αυτοκίνητα είναι, όπως ήδη αναφέραμε, η δυνατότητα αυτόνομης πλοήγησης. Υπάρχουν πολλά σημαντικά και γνωστά παραδείγματα σε αυτόν τον τομέα, όπως το Darpa Challenge (2004 και 2005 Grand Challenges και 2007 Urban Challenge) [21] [22] που αποτελούν από τα πιο διάσημα έργα σε αυτό το πεδίο έρευνας.

Οι βασικές λειτουργίες ενός συστήματος πλοήγησης συνήθως συνοψίζονται σε: εντοπισμός θέσης (localization), χαρτογράφηση mapping, αντίληψη (perception), αξιολόγηση (assessment), σχεδιασμός αποφάσεων (decision planning) και λήψη αποφάσεων (decision making), και πλοήγηση (navigation) μέσω του ελέγχου των οχημάτων (vehicle control) [20]. Η πλοήγηση σχετίζεται με τη δυνατότητα του συστήματος να συλλέξει σωστά πληροφορίες για το περιβάλλον του από τους αισθητήρες και να τις επεξεργαστεί με σκοπό να κινηθεί στον χώρο με ασφαλή και αποτελεσματικό τρόπο. Πολλές από τις προσεγγίσεις που μελετήθηκαν χρησιμοποιούν για την πλοηγήση όλους τους αισθητήρες και GPS, ωστόσο έχει ιδιαίτερο ενδιαφέρον να εξερευνηθεί το πεδίο της πλοήγησης στον χώρο χρησιμοποιώντας την κάμερα σαν τον βασικό αισθητήρα του οχήματος.

Το όχημα στην περίπτωση αυτή θα πρέπει να μπορεί να κινείται μέσα σε ένα καθορισμένο μονοπάτι εντός των ορίων του δρόμου (navigable area), ακολουθώντας τις γραμμές, αναγνωρίζοντας τα εμπόδια που μπορεί να συναντήσει και να ακολουθεί τα φανάρια και τα σήματα οδικής κυκλοφορίας. Χρησιμοποιώντας επομένως μόνο την οπτική του αντίληψη για τον χώρο θα πρέπει να μπορεί να πλοηγείται και να λαμβάνει αποφάσεις για τις επόμενες κινήσεις του βάσει των οπτικών πληροφοριών που λαμβάνει από το αισθητήριο όργανο της κάμερας. Αυτές οι προσεγγίσεις χρησιμοποιούν συνήθως

νευρωνικά δίκτυα σε συνδυασμό με μια υλοποιήση μιας Μηχανής Καταστάσεων (FSM) [20].

Οι λόγοι επιλογής της κάμερας για βασικό ασιθητήρα σε πολλές εφαρμογές τα τελευταία χρόνια είναι πολλοί, καθώς έχουν αποδειχθεί ένα από τα καταλληλότερα συστήματα αισθητήρων για την πλοήγηση και την αποφυγή εμποδίων με ακρίβεια, ενώ το μικρό τους βάρος και η χαμηλή ενεργειακή τους κατανάλωση τις καθιστά ιδανικές για χρήση σε αυτόνομα οχήματα [23] [20]. Ακόμα, οι εικόνες, όπως είναι λογικό, μπορούν να δώσουν πολλές και διαφορετικές πληροφορίες για το περιβάλλον χωρίς να χρειαστεί απαραίτητα συνδυασμός αισθητήρων, καθιστώντας πολύ πιο έγκολη τη διαδικασία. Τέλος, το κόστος μειώνεται σημαντικά με τη χρήση καμερών.

2.2.2 Αισθητήρες Radar

Οι αισθητήρες Radar (Radio Detection And Range) συμβάλλουν καθοριστικά στη λειτουργία της αυτόνομης οδήγησης και μπορούν να διευκολύνουν πολύ την διαδικασία. Το Radar κατά την λειτουργία του στέλνει ραδιοκύματα για την ανίχνευση αντικειμένων και έπειτα υπολογίζει την απόσταση και την ταχύτητά τους σε σχέση με το αυτόνομο όχημα σε πραγματικό χρόνο. Τοποθετούνται συνήθως περιφερειακά του οχήματος και ο ανάλογα με την εμβέλειά τους παρουσιάζουν διαφορετικές λειτουργίες. Σε αντίθεση με τους αισθητήρες κάμερας που επηρεάζονται πολύ από τις καιρικές συνθήκες και τη φωτεινότητα, τα συστήματα ραντάρ συνήθως δεν έχουν κανένα πρόβλημα στην αναγνώριση αντικειμένων.

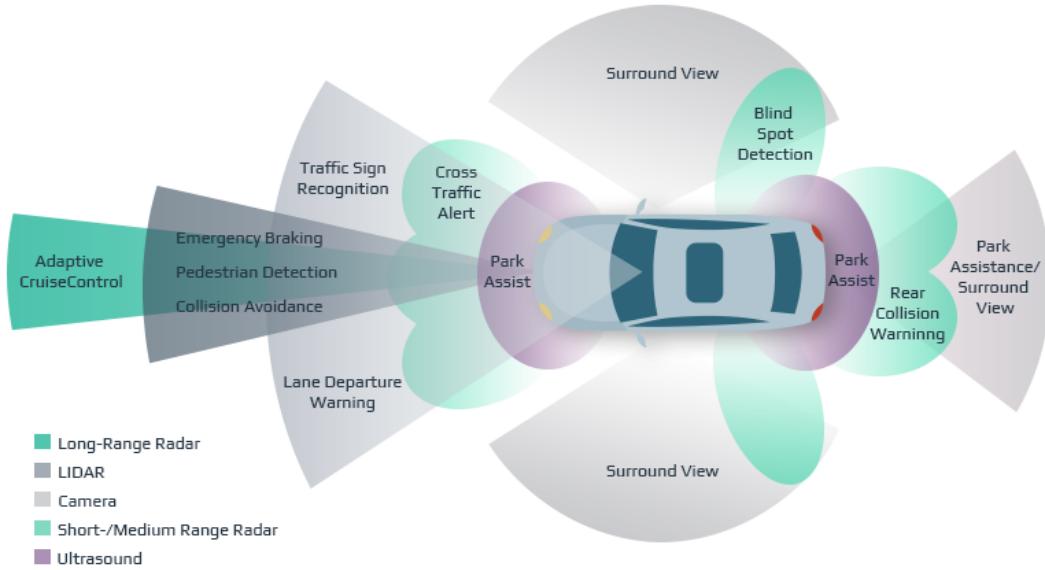
2.2.3 Αισθητήρες Lidar

Ο αισθητήρας Lidar λειτουργεί παρόμοια με το ραντάρ, με την έννοια ότι και οι δύο αισθητήρες εκπέμπουν και λαμβάνουν κύματα και υπολογίζουν τον χρόνο εκπομπής-λήψης για τον προσδιορισμό της απόστασης. Οι αισθητήρες Lidar (Light Detection And Ranging) στέλνουν κάθε δευτερόλεπτο πολλούς στοχευμένους παλμούς λέιζερ και δημιουργούν έναν χάρτη βάθους (depth map), ή έναν χάρτη σημείων (point cloud map), που απεικονίζει τα αντικείμενα στο χώρο σε μια περιοχή γύρω από το όχημα. Οι χάρτες αυτοί είναι αρκετά ακριβείς και μπορούν να χρησιμοποιηθούν για να δώσουν μια καλή πρόβλεψη για το είδος ενός αντικειμένου. Χρησιμοποιούνται για την αναγνώριση αντικειμένων και την παρακολούθηση της ταχύτητάς κίνησης τους. Ωστόσο, οι αισθητήρες Lidar βασίζονται στην πληροφορία που προκύπτει από το βάθος και, ως εκ τούτου, δεν μπορούν να εντοπίσουν κρίσιμα στοιχεία χωρίς βάθος, όπως για παράδειγμα τα φανάρια και τις πινακίδες.

2.3 Ψηφιακή Επεξεργασία Εικόνας

2.3.1 Η εικόνα ως δισδιάστατο σήμα

Η ψηφιακή εικόνα είναι ένα από τα δισδιάστατα ψηφιακά σήματα που παρουσιάζουν σημαντικό ενδιαφέρον στη σημερινή κοινωνία της πληροφορίας. Η ανάγκη για επεξεργασία της πληροφορίας της εικόνας οδηγεί κατ' αρχήν στη μαθηματική περιγραφή του περιεχομένου της. Την ασπρόμαυρη εικόνα μπορούμε να τη δούμε ως μία



Σχήμα 2.3: Τοποθέτηση αισθητήτων πάνω στο όχημα [24]

δισδιάστατη συνάρτηση των συντεταγμένων του χώρου x και y :

$$f \rightarrow f(x, y) \quad (2.1)$$

όπου η f περιγράφει μία επιφάνεια, η οποία έχει μεγάλη τιμή όταν η εικόνα είναι πιο λευκή, ενώ για μικρές τιμές της f η εικόνα πλησιάζει το μαύρο. Στην ουσία η f περιγράφει την αμαύρωση της εικόνας σε κάθε θέση (x,y) .

Κάθε εικόνα για να υποστεί ψηφιακή επεξεργασία θα πρέπει αρχικά να μετατραπεί σε ψηφιακή. Η διαδικασία αυτή είναι περίπου η ίδια όπως και στην περίπτωση του μονοδιάστατου σήματος. Έτσι, θα πρέπει να λάβουμε ισαπέχοντα δείγματα της συνάρτησης $f(x,y)$ στις διευθύνσεις x και y . Η πυκνότητα με την οποία θα ληφθούν τα δείγματα καθορίζεται και στην περίπτωση αυτή (δισδιάστατη συνάρτηση στο χώρο) από το θεώρημα δειγματοληψίας το οποίο ορίζεται ως εξής:

«Η απόσταση δύο διαδοχικών δειγμάτων στο επίπεδο (x,y) θα πρέπει να είναι μικρότερη από την ημιπερίοδο των ταχύτερων εναλλαγών της συνάρτησης $f(x,y)$.»

Συμπεραίνουμε ότι θα πρέπει να δειγματοληπτούμε αρκετά γρήγορα για να προλαβαίνουμε τις γρήγορες εναλλαγές της αμαύρωσης της εικόνας. Στη συνέχεια, τα δείγματά μας θα πρέπει να κβαντιστούν σε πεπερασμένο αριθμό σταθμών. Συνήθως οι στάθμες που χρησιμοποιούνται είναι 255 (2^8). Στη στάθμη 255 αντιστοιχούμε το λευκό, ενώ στη στάθμη 0 το μαύρο. Έτσι, ένας δυαδικός αριθμός των 8 bits (1 byte) επαρκεί για να περιγράψουμε την τιμή της αμαύρωσης ενός δείγματος της εικόνας που καλείται και εικονοστοιχείο (picture element - pixel).

Η ψηφιακή εικόνα παριστάνεται μαθηματικά ως η κβαντισμένη σε πλάτος συνάρτηση:

$$f \rightarrow f_q(n_1, n_2) \quad (2.2)$$

όπου οι διακριτές χωρικές μεταβλητές n_1 και n_2 αντιστοιχούν στις συνεχείς χωρικές μεταβλητές x και y . Στη γενική περίπτωση, η μέγιστη τιμή του n_1 είναι M και του n_2

είναι N. Έτσι, μπορούμε να θεωρήσουμε την ψηφιακή εικόνα ως μία μήτρα διαστάσεων $M \times N$ αριθμών που κάθε ένας από αυτούς εκπροσωπεί την τιμή του αντίστοιχου εικονοστοιχείου.

Σε περίπτωση που η εικόνα είναι έγχρωμη τότε σε κάθε θέση (n_1, n_2) η f έχει τρεις τιμές (άνυσμα), οι οποίες αντιστοιχούν στις τιμές των χρωμάτων κόκκινο, πράσινο και μπλε.

2.3.2 Γεωμετρία Εικόνας

Η εικόνα μπορεί να θεωρηθεί σαν κατανομή της πληροφορίας (βαθμός αμαύρωσης ή χρώμα) στο επίπεδο (x,y). Τα περισσότερα οπτικά συστήματα (φωτογραφικές μηχανές, βίντεο-κάμερες) αποδίδουν αναλογικές εικόνες, δηλαδή εικόνες στις οποίες οι συντεταγμένες (x,y) αλλά και ο βαθμός αμαύρωσης f είναι συνεχείς ποσότητες. Αντίθετα, οι ψηφιακές εικόνες, που πρόκειται να αποθηκευτούν στη μνήμη του υπολογιστή, ή να τύχουν επεξεργασίας από αυτόν, έχουν τιμή μόνο σε διακριτό αριθμό συντεταγμένων (n_1, n_2) που αποτελούν και τις θέσεις των εικονοστοιχείων. Επίσης, κάθε εικονοστοιχείο λαμβάνει πεπερασμένο αριθμό τιμών αμαύρωσης.

Ο προσανατολισμός των συντεταγμένων σε μία ψηφιακή εικόνα μπορεί να είναι αυθαίρετος. Στις περισσότερες όμως περιπτώσεις και κυρίως στις ποικίλες εφαρμογές λογισμικού που κυκλοφορούν στο εμπόριο, η αρχή των συντεταγμένων $(n_1, n_2)=(1, 1)$ λαμβάνεται στην επάνω αριστερή γωνία της εικόνας. Κάθε ένα εικονοστοιχείο έχει τις δικές του συντεταγμένες. Συνεπώς, μία εικόνα που έχει M εικονοστοιχεία στη διεύθυνση n_1 και N στη διεύθυνση n_2 περιέχει συνολικά $N \times M$ εικονοστοιχεία. Το μέγεθος της πληροφορίας της εικόνας είναι γενικά μεγάλο. Ακόμα και μικρές σε διαστάσεις εικόνες περιέχουν μεγάλο αριθμό εικονοστοιχείων, γεγονός που καθιστά χρονοθόρα την ψηφιακή τους επεξεργασία. Για τον λόγο αυτό, ο έλεγχος αλγορίθμων επεξεργασίας εικόνας γίνεται αρχικά σε ένα μικρό τμήμα της εικόνας.

2.3.3 Επεξεργασία

Η επεξεργασία της ψηφιακής εικόνας περιλαμβάνει στη γενικότερη περίπτωση πολλά και διαφορετικά στάδια ανάλογα με τη μορφή και την αρχική κατάσταση της ψηφιακής εικόνας και το επιθυμητό τελικό αποτέλεσμα.

Η πραγματική εικόνα θα καταγραφεί από το σύστημα απόκτησης της ψηφιακής εικόνας. Θα ακολουθήσει το στάδιο της εξαγωγής του θορύβου (noise filtering) και της διόρθωσης (restoration) σφαλμάτων που προκλήθηκαν είτε από το μέσο μεταξύ εικόνας και συστήματος απόκτησης είτε και από το ίδιο το σύστημα απόκτησης. Σφάλματα που παρουσιάζονται στην εικόνα κατά τη διάρκεια της απόκτησης της είναι ο θερμικός θόρυβος ή/και ο θόρυβος αιχμών (από το σύστημα απόκτησης) και η θόλωση της εικόνας.

Η βελτίωση της εικόνας αφορά κυρίως στον τονισμό συγκεκριμένων στοιχείων της εικόνας με σκοπό τη διευκόλυνση των διαδικασιών που θα ακολουθήσουν. Το τελευταίο στάδιο είναι η μηχανική όραση. Σε αυτό περιλαμβάνονται όλα τα αυτοματοποιημένα συστήματα επεξεργασίας εικόνας, αναγνώρισης αντικειμένων και λήψης αποφάσεων για συγκεκριμένες ενέργειες.

2.4 Μηχανική Μάθηση (Machine Learning)

2.4.1 Εισαγωγή

Η Μηχανική Μάθηση (Machine Learning) είναι ένα υπο-πεδίο της Τεχνητής Νοημοσύνης, με στόχο την ικανότητα της μηχανής να μιμείται ευφυή ανθρώπινη συμπεριφορά, ώστε να επιλύει περίπλοκα ζητήματα. Ο αρχικός ορισμός δόθηκε από τον Arthur Samuel το 1959, περιγράφοντάς την ως «το πεδίο έρευνας που δίνει στους υπολογιστές τη δυνατότητα να μαθαίνουν, χωρίς να έχουν προγραμματιστεί ρητά για να το κάνουν» [25].

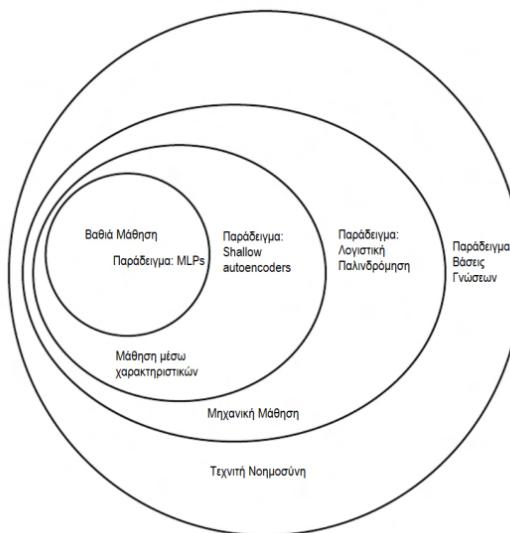
Οι αλγόριθμοι Μηχανικής Μάθησης εκμεταλλεύονται δεδομένα (αριθμητικά, κείμενο, εικόνα, ήχο, κτλ) και «μαθαίνουν» από αυτά για να κάνουν προβλέψεις. Τα δεδομένα εκπαίδευσης οφείλουν να είναι ικανοποιητικά τόσο σε ποσότητα, όσο και ποιότητα, για μια πετυχημένη εκπαίδευση ενός μοντέλου. Η δουλειά των προγραμματιστών είναι να παρέχουν στο μοντέλο κατάλληλα δεδομένα (εφαρμόζοντας και τεχνικές προ-επεξεργασίας), να διαλέξουν το κατάλληλο μοντέλο και να το παραμετροποιήσουν, ανάλογα με τις ιδιαιτερότητες του εκάστοτε προβλήματος προς επίλυση. Από κει και πέρα αναλαμβάνει η «μηχανή», υπεύθυνη να εκπαιδεύσει το μοντέλο και να βρει μοτίβα ή να κάνει προβλέψεις. Επιπλέον, εκτός από τα δεδομένα εκπαίδευσης, ένα κομμάτι των δεδομένων πρέπει να δεσμευτεί για να χρησιμοποιηθεί ως σύνολο ελέγχου· τα αποτελέσματα από τη διαδικασία ελέγχου μπορούν να μας παρέχουν με πολύ χρήσιμες μετρικές για το πόσο καλά συμπεριφέρεται το μοντέλο. Είναι αρκετά σύνηθες να διαχωρίζουμε τα μοντέλα Μηχανικής Μάθησης στις παρακάτω τρεις κατηγορίες [26]:

- Επιβλεπόμενη Μάθηση (Supervised learning)
- Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning)
- Ενισχυτική Μάθηση (Reinforcement learning)

Οι εφαρμογές της Μηχανικής μάθησης αφορούν υπολογιστικές εργασίες, που είναι εφικτός ο σχεδιασμός και η υλοποίηση αλγορίθμων σε προγραμματιστικό περιβάλλον. Παραδείγματα εφαρμογών αποτελούν τα φίλτρα σπαμ (spam filtering), η οπτική αναγνώριση χαρακτήρων (OCR), οι μηχανές αναζήτησης και η υπολογιστική όραση. Η Μηχανική μάθηση συγχέεται ακόμα και με την εξόρυξη δεδομένων, με σκοπό την εξερεύνηση και ανάλυση των δεδομένων, γνωστή και ως μη επιτηρούμενη μάθηση. Ακόμα, η μηχανική μάθηση εφαρμόζεται στο πεδίο της ανάλυσης των δεδομένων και αποτελεί μια μέθοδο που χρησιμοποιείται για την δημιουργία πολύπλοκων μοντέλων και αλγορίθμων που χρησιμοποιούνται για την πρόβλεψη αποτελεσμάτων. Τέλος, δύο βασικές προσεγγίσεις που αφορούν τη μηχανική μάθηση είναι τα τεχνητά νευρωνικά δίκτυα και η βαθιά μάθηση.

2.4.2 Βαθιά Μάθηση (Deep Learning)

Η βαθιά μάθηση είναι μέρος μιας ευρύτερης οικογένειας μεθόδων μάθησης που βασίζονται στα τεχνητά νευρωνικά δίκτυα. Ο χαρακτηρισμός «βαθιά» αναφέρεται στο πλήθος των επιπέδων που υπάρχουν στο δίκτυο, μέσα από τα οποία περνούν τα δεδομένα της εισόδου μέχρι να εξαχθεί το αποτέλεσμα. Η μάθηση μπορεί να είναι με εποπτεία ή χωρίς. Αυτά τα νευρωνικά δίκτυα προσπαθούν να προσομοιώσουν τη συμπεριφορά του ανθρώπινου εγκεφάλου επιτρέποντάς του να «μαθαίνει» μέσα από με-



Σχήμα 2.4: Διάγραμμα Venn για την σχέση των μεθόδων της Τεχνητής Νοημοσύνης [27]

γάλο όγκο δεδομένων. Ενώ ένα νευρωνικό δίκτυο με ένα μόνο στρώμα μπορεί να κάνει κατά προσέγγιση προβλέψεις, επιπλέον κρυμμένα επίπεδα σε αυτό μπορούν να βοηθήσουν στη βελτιστοποίηση και την ακρίβεια. Οι αρχιτεκτονικές βαθιάς μάθησης όπως τα βαθιά νευρωνικά δίκτυα, τα επαναλαμβανόμενα νευρωνικά δίκτυα και τα συνελικτικά νευρωνικά δίκτυα, έχουν εφαρμογή σε διάφορους τομείς όπως η μηχανική όραση, η αναγνώριση φωνής, η επεξεργασία φυσικής γλώσσας, η αναγνώριση ήχου, το φιλτράρισμα των κοινωνικών δικτύων, η ανάλυση ιατρικών εικόνων, η αυτόνομη οδήγηση κ.α.

Πιο συγκεκριμένα στην βαθιά μάθηση, κάθε επίπεδο μαθαίνει να μετασχηματίζει τα δεδομένα της εισόδου του σε μια ελαφρώς πιο αφηρημένη και σύνθετη αναπαράσταση. Τα χαμηλότερα στρώματα μπορεί να προσδιορίσουν τις άκρες ενός σχήματος, ενώ τα υψηλότερα στρώματα μπορούν να αναγνωρίσουν τις έννοιες που καταλαβαίνει και ο άνθρωπος όπως ψηφία, γράμματα ή πρόσωπα. Τα απαραίτητα χαρακτηριστικά διαμορφώνονται και γίνονται αντιληπτά από το νευρωνικό δίκτυο βάσει των δεδομένων που εκπαιδεύτηκε. Η πιο σημαντική διαφορά στην βαθιά διαδικασία εκμάθησης, σε σχέση με τα απλά νευρωνικά δίκτυα, είναι ότι μπορεί μόνο του το δίκτυο να μάθει να αναθέτει την εξαγωγή των χαρακτηριστικών στο κάθε στρώμα με βέλτιστο τρόπο.

Τα περισσότερα σύγχρονα μοντέλα βαθιάς μάθησης βασίζονται σε τεχνητά νευρωνικά δίκτυα, και μία πολύ σημαντική κατηγορία είναι τα συνελικτικά νευρωνικά δίκτυα ή πιο γνωστά ως Convolutional Neural Networks (CNN) [28].

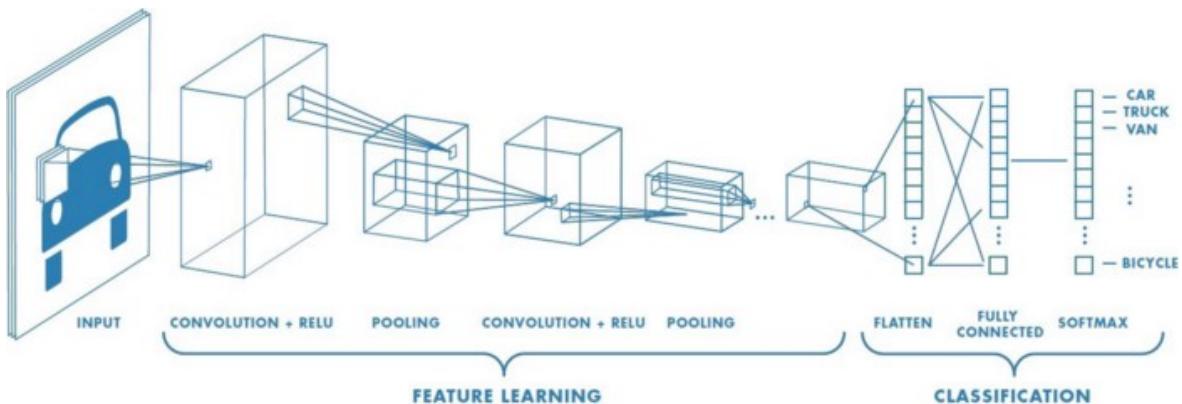
2.4.3 Συνελικτικά Δίκτυα (Convolutional Neural Networks)

Τα συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks ή CNN) είναι μια κατηγορία των βαθιών νευρωνικών δικτύων που συνήθως εφαρμόζονται στην ανάλυση οπτικών εικόνων. Το όνομα συνελικτικό νευρωνικό δίκτυο υποδηλώνει ότι χρησιμοποιεί ένα εξειδικευμένο είδος γραμμικής λειτουργίας που ονομάζεται συνέλιξη και χρησιμοποείται στον γενικό πολλαπλασιασμό σε τουλάχιστον ένα επίπεδο. Τα συνελι-

κά επίπεδα έχουν στην διάθεσή τους Πυρήνες (Kernels) ή αλλιώς Φίλτρα (Filters), τα οποία αποδεδειγμένα έχουν πολύ μεγάλη απόδοση στην ανίχνευση χαρακτηριστικών και μοτίβων των οπτικών δεδομένων.

Τα συνελικτικά νευρωνικά δίκτυα είναι μια τοπολογία νευρωνικού δικτύου που μοιάζει με πλέγμα και εμπνεύστηκαν από βιολογικές διεργασίες το μοτίβο της σύνδεσης μεταξύ των νευρώνων και ομοιάζουν με την οργάνωση του ζωτικού οπτικού φλοιού. Ο κάθε μεμονωμένος νευρώνας ανταποκρίνεται σε μια περιορισμένη περιοχή οπτικού πεδίου το λεγόμενο πεδίο υποδοχής. Όλα τα πεδία υποδοχής των νευρώνων συντελούν στην κάλυψη ολόκληρου οπτικού πεδίου.

Τα συνελικτικά νευρωνικά δίκτυα αποτελούν κανονικοποιημένη μορφή των MLP (Multi Layer Perceptrons) και είναι συνήθως πλήρως συνδεδεμένα μεταξύ τους. Αυτό το χαρακτηριστικό τα κάνει επιρρεπή στο φαινόμενο της υπερκπαίδευσης (overfitting). Τυπικοί τρόποι αντιμετώπισης του φαινομένου αυτού περιλαμβάνουν μέτρηση βαρών προς τη συνάρτηση απώλειας (loss function). Ωστόσο, τα συνελικτικά νευρωνικά δίκτυα υιοθετούν μια διαφορετική προσέγγιση, που εκμεταλλεύεται το ιεραρχικό πρότυπο στα δεδομένα, διασπώντας τα πιο περίπλοκα σχέδια σε μικρότερα και απλούστερα πρότυπα. Ως εκ τούτου, τα CNN βρίσκονται χαμηλά στην κλίμακα της συνδεσιμότητας και της πολυπλοκότητας.



Σχήμα 2.5: Συνελικτικό Νευρωνικό δίκτυο [29]

Τα CNN χρησιμοποιούν σχετικά μικρή προεπεξεργασία σε σύγκριση με άλλους αλγόριθμους ταξινόμησης εικόνων. Αυτό συμβαίνει γιατί το δίκτυο μαθαίνει τα φίλτρα μόνο του σε αντίθεση με άλλους αλγορίθμους που ο καθορισμός των φίλτρων γίνεται από τον άνθρωπο βάση των γνώσεων του. Αυτή η ανεξαρτησία από τις προηγούμενες γνώσεις και την ανθρώπινη προσπάθεια στον σχεδιασμό χαρακτηριστικών του νευρωνικού δικτύου αποτελεί σημαντικό πλεονέκτημα. Στο Σχήμα 2.5 φαίνεται μια αναπαράσταση της δομής ένος CNN.

Συστατικά μέρη

Ένα συνελικτικό νευρωνικό δίκτυο αποτελείται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και πολλά κρυφά επίπεδα. Τα κρυφά επίπεδα είναι αυτά που περιέχουν και τη διαδικασία της συνέλιξης. Η συνάρτηση ενεργοποίησης των επιπέδων αυτών είναι συνήθως η ReLU και ακολουθείται από επιπρόσθετες συνελίξεις (convolutions) όπως η συγκέντρωση επιπέδων (pooling layers), πλήρως συνδεδεμένα επίπεδα (fully connected layers) και επίπεδα κανονικοποίησης (normalization layers). Αυτά τα επίπεδα

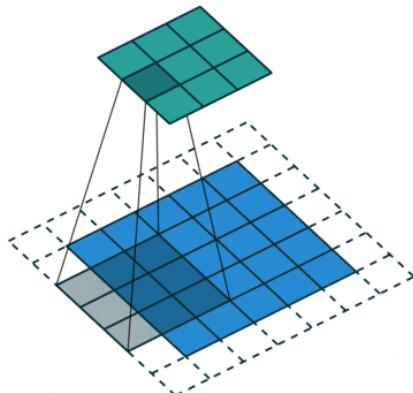
ονομάζονται κρυφά καθώς οι είσοδοι και έξοδοι τους καλύππονται από την συνάρτηση ενεργοποίησης και την τελική συνέλιξη. Η τελική συνέλιξη συχνά περιλαμβάνει την τεχνική του back-propagation όπου με μία καθορισμένη συνάρτηση απώλειας διορθώνει τα βάρη ώστε να βελτιώσει το αποτέλεσμα.

Συνέλιξη (Convolution)

Όταν κατασκευάζουμε ένα CNN, κάθε συνελικτικό επίπεδο εμπεριέχει τα εξής χαρακτηριστικά :

- Η είσοδος είναι ένας διάνυσμα με μορφή (αριθμός εικόνων) \times (πλάτος εικόνας) \times (ύψος εικόνας) \times (βάθος εικόνας).
- Οι συνελικτικοί νευρώνες έχουν υπερπαραμέτρους, το πλάτος, το ύψος και το βάθος των οποίων πρέπει να είναι ίσο με εκείνο της εικόνας.

Τα συνελικτικά επίπεδα επεξεργάζονται την είσοδο και μεταφέρουν το αποτέλεσμά της στο επόμενο επίπεδο. Παρόλο που τα πλήρως συνδεδεμένα εμπρόσθιας τροφοδότησης νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για να μάθουν χαρακτηριστικά καθώς και να ταξινομήσουν δεδομένα, δεν είναι πρακτικό να εφαρμοστεί αυτή η αρχιτεκτονική στις εικόνες. Θα ήταν απαραίτητος ένας πολύ μεγάλος αριθμός νευρώνων, λόγω των πολύ μεγάλων μεγεθών εισόδου που σχετίζονται με τις εικόνες, όπου κάθε εικονοστοιχείο είναι μια σχετική μεταβλητή. Για παράδειγμα, ένα πλήρως συνδεδεμένο στρώμα για μια μικρή εικόνα μεγέθους 100×100 έχει 10.000 βάρη για κάθε νευρώνα στο δεύτερο επίπεδο. Η διαδικασία της συνέλιξης (Σχήμα 2.6) φέρνει μια λύση σε αυτό το πρόβλημα καθώς μειώνει τον αριθμό των ελεύθερων παραμέτρων, επιτρέποντας περισσότερα επίπεδα με λιγότερες παραμέτρους [30].



Σχήμα 2.6: Η διαδικασία της συνέλιξης [29]

Το συνελικτικό επίπεδο είναι το κεντρικό δομικό στοιχείο ενός συνελικτικού νευρωνικού δικτύου. Οι παράμετροι του επιπέδου αποτελούνται από ένα σύνολο μαθηματικών φίλτρων. Κατά τη διάρκεια της προς τα εμπρός διάδοσης, κάθε φίλτρο διαπερνά κατά πλάτος και ύψος τι τιμές της εισόδου, δημιουργώντας ένα πίνακα 2-διαστάσεων ενεργοποίησης αυτού του φίλτρου. Ως αποτέλεσμα, το δίκτυο μαθαίνει φίλτρα, που ενεργοποιούνται όταν ανιχνεύει κάποιον συγκεκριμένο τύπο χαρακτηριστικού σε κάποια χωρική θέση στην είσοδο. Η τοποθέτηση των χαρτών ενεργοποίησης για όλα τα φίλτρα κατά μήκος του επιπέδου αποτελεί την πλήρη έξοδο του επιπέδου συνέλιξης.

Επίπεδο συνδυασμού (Pooling Layer)

Τα συνελικτικά δίκτυα ενδέχεται να περιλαμβάνουν συνδυαστικά επίπεδα (pooling layers), τα οποία εφαρμόζονται είτε καθολικά επηρεάζοντας όλους τους νευρώνες του δικτύου, είτε τοπικά σε ένα μόνο επίπεδο [31]. Τα συνδυαστικά επίπεδα σε τοπική εφαρμογή, μειώνουν τις διαστάσεις των δεδομένων συνδυάζοντας τις εξόδους των συστοιχιών νευρώνων σε μία στρώση σε έναν μόνο νευρώνα του επόμενου επιπέδου. Επιπλέον, ο συνδυασμός μπορεί να εφαρμοστεί για τον υπολογισμό μεγίστου ή μέσου όρου. Στην περίπτωση του μεγίστου ο συνδυασμός χρησιμοποιεί την μέγιστη τιμή από το προηγούμενο επίπεδο στο επόμενο, ενώ του μέσου όρου χρησιμοποιεί το μέσο όρο του για το επόμενο επίπεδο. Στα επίπεδα αυτά συγκεντρώνονται και συνδυάζονται τα χαρακτηριστικά που έχουν εξαχθεί από τα προηγούμενα επίπεδα συνέλιξης κατασκευάζοντας έτσι έναν χάρτη χαρακτηριστικών που θα χρησιμοποιηθεί από τα επόμενα επίπεδα για την περεταίρω ανάλυση της εικόνας.

Διαισθητικά, η ακριβής τοποθεσία ενός χαρακτηριστικού είναι λιγότερο σημαντική από την θέση του σε σχέση με άλλα χαρακτηριστικά. Αυτή είναι η ιδέα πίσω από τη χρήση του συνδυασμού σε συνελικτικά νευρωνικά δίκτυα. Το επίπεδο συνδυασμού χρησιμεύει για τη σταδιακή μείωση του χωροταξικού μεγέθους της αναπαράστασης της πληροφορίας εισόδου, τη μείωση του αριθμού των παραμέτρων, του αποτυπώματος μνήμης και του αριθμού των υπολογισμών στο δίκτυο και, ως εκ τούτου, του ελέγχου της υπερφόρτωσης. Είναι συνηθισμένο να εισάγετε περιοδικά ένα επίπεδο συνδυασμού μεταξύ διαδοχικών συνελικτικών στρώσεων σε μια αρχιτεκτονική CNN. Όπως θα δούμε παρακάτω ο τρόπος εξαγωγής των χαρτών αυτών είναι πολύ σημαντικός αφού καθορίζει την ταχύτητα και την ακρίβεια του δικτύου.

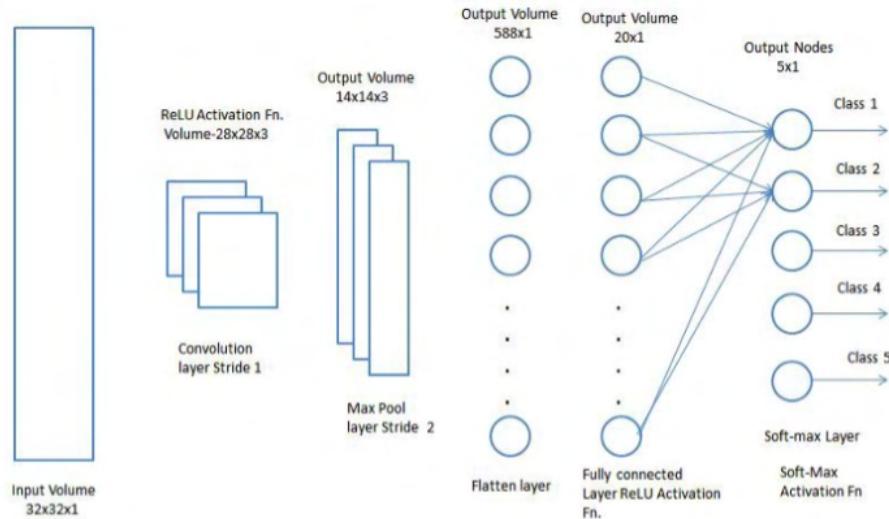
Πλήρως συνδεδεμένο επίπεδο (Fully Connected Layer)

Τα πλήρως συνδεδεμένα επίπεδα συνδέουν κάθε νευρώνα σε ένα επίπεδο με κάθε νευρώνα σε ένα άλλο επίπεδο. Η βάση του είναι ίδια με το παραδοσιακό πολυεπίπεδο νευρωνικό δίκτυο perceptron (MLP). Στα συνελικτικά νευρωνικά δίκτυα χρησιμοποιείται για να ταξινομήσει το αποτέλεσμα ανάλογα το σύνολο των χαρακτηριστικών που έχει εξαχθεί από τα προηγούμενα επίπεδα και συνεπώς το συναντούμε στα τελευταία επίπεδα του. Ουσιαστικά είναι αυτό που θα δώσει την τελική απόφαση του δικτύου για την ταξινόμηση του περιεχομένου της εικόνας. Ωστόσο, ένα CNN μπορεί και να μην περιέχει πλήρως συνδεδεμένο επίπεδο στο τέλος.

Επίπεδο κανονικοποίησης

Το επίπεδο κανονικοποίησης χρησιμοποιείται για την βελτίωση της ταχύτητας, της επίδοσης και της σταθερότητας ενός τεχνητού νευρωνικού δικτύου. Ο ορός αναφέρεται σαν batch normalization (κανονικοποίηση παρτίδας) καθώς τροποποιεί κατάλληλα τις τιμές στη είσοδο του ώστε να περάσουν στον επόμενο επίπεδο με τον βέλτιστο τρόπο. Η ανάγκη χρήσης του προέκυψε από το πρόβλημα της διαφοράς που δημιουργούνταν μεταξύ των τιμών των παραμέτρων στα επίπεδα κατά της διάρκεια της εκπαίδευσης. Έτσι το δίκτυο πρέπει να αναπροσαρμόζει σε κάθε διανομή τις τιμές ώστε να μην υπάρχουν μεγάλες αποκλίσεις μεταξύ των επιπέδων.

Ιδιαίτερα στα δίκτυα βαθιάς μάθησης μικρές μεταβολές παραμέτρων στα πρώτα επίπεδα μπορούν να δημιουργήσουν μεγάλες μεταβολές στα τελευταία επίπεδα. Με



Σχήμα 2.7: Πλήρως συνδεδεμένο επίπεδο [29]

την εισαγωγή της κανονικοποίησης ενδιάμεσα των επιπέδων, οι μεταβολές αυτές εξομαλύνονται οδηγώντας στη δημιουργία πιο αξιόπιστων μοντέλων. Επίσης το δίκτυο μπορεί να εκπαιδευτεί με μεγαλύτερο ρυθμό μάθησης χωρίς να χάνονται λεπτομέρειες χαρακτηριστικών. Τέλος είναι ευκολότερη η κατασκευή μοντέλου που μαθαίνει γενικά χαρακτηριστικά χωρίς την χρήση της μεθόδου dropout, η οποία απορρίπτει ένας μέρος των τιμών κατά τη διάρκεια της εκπαίδευσης, κάτι που οδηγεί στην πιθανή απώλεια λεπτών χαρακτηριστικών.

Συναρτήσεις απώλειας - ενεργοποίησης

Η συνάρτηση απώλειας (loss function) καθορίζει τον τρόπο με τον οποίο η εκπαίδευση επιβάλλει την ποινή στις τιμές που αποκλίνουν από την επιθυμητή έξοδο και συνήθως εκτελείται στο τελευταίο επίπεδο του νευρωνικού δικτύου. Η επιλογή της κατάλληλης συνάρτηση απώλειας εξαρτάται από τον σκοπό και την διεργασία που εκτελείται. Η πιο γνώστη συνάρτηση είναι η SoftMax, που χρησιμοποιείται για την πρόβλεψη μίας και μόνο κατηγορίας αμοιβαίως αποκλειόμενων κλάσεων. Ακόμα, η σιγμοειδής συνάρτηση χρησιμοποιείται για την εξαγωγή εκτιμήσεων ικανοποίησης για όλες τις ανεξάρτητες κλάσεις. Τέλος, η ευκλείδεια συνάρτηση χρησιμοποιείται για την παλινδρόμηση σε πραγματικές τιμές.

Η συνάρτηση ενεργοποίησης ενός κόμβου καθορίζει την έξοδο αυτού του κόμβου με δεδομένη είσοδο ή σύνολο εισόδων. Η πιο συνηθισμένη συνάρτηση ενεργοποίησης είναι η ReLU, η οποία αφαιρεί αποτελεσματικά τις αρνητικές τιμές από έναν χάρτη ενεργοποίησης θέτοντας τους στο μηδέν. Αυξάνει τις μη γραμμικές ιδιότητες της συνάρτησης λήψης αποφάσεων και του συνολικού δικτύου χωρίς να επηρεάζει τα πεδία υποδοχής του επιπέδου συνέλιξης. Η ReLU προτιμάται συχνά από άλλες συναρτήσεις, καθώς εκπαιδεύει το νευρικό δίκτυο αρκετές φορές πιο γρήγορα χωρίς σημαντική ποινή στην ακρίβεια γενίκευσης.

2.5 Μηχανική όραση (Computer Vision)

2.5.1 Εισαγωγή

Η Μηχανική όραση, υπολογιστική όραση ή τεχνητή όραση, η οποία αποτελεί τομέα της Τεχνητής Νοημοσύνης, είναι μέθοδος αυτοματοποίησης της ανάλυσης δεδομένων. Βασίζεται στη δυνατότητα των συστημάτων να εκπαιδευτούν από δεδομένα, να αναγνωρίσουν μοτίβα και να λάβουν αποφάσεις με την ελάχιστη ανθρώπινη παρέμβαση. Επιχειρεί, δηλαδή να αναπαράγει αλγορίθμικά την αίσθηση της όρασης, συνήθως σε ηλεκτρονικό υπολογιστή ή ρομπότ. Η μηχανική όραση σχετίζεται με τη θεωρία και την τεχνολογία, που εμπλέκονται στη σχεδίαση και κατασκευή συστημάτων, που λαμβάνουν και αναλύουν δεδομένα από ψηφιακές εικόνες. Τα εν λόγω δεδομένα μπορούν να είναι φωτογραφίες, βίντεο, όψεις από πολλαπλές κάμερες, πολυδιάστατες εικόνες κλπ.

Η μηχανική μάθηση έχει αναδειχθεί χρήσιμο εργαλείο για τη μοντελοποίηση των προβλημάτων που διαφορετικά είναι δύσκολο να διατυπώσουμε με ακρίβεια. Με τη μάθηση μηχανών, ένα μέρος της ανθρώπινης συμβολής αντικαθίσταται από έναν αλγόριθμο μάθησης. Η διαθεσιμότητα της υπολογιστικής ικανότητας και των δεδομένων έχει αυξηθεί, με αποτέλεσμα η μηχανική μάθηση έχει καταστεί όλο και πιο πρακτική και αναγκαία με την πάροδο των ετών.

2.5.2 Τυπικές μέθοδοι συστήματος

Η οργάνωση ενός συστήματος μηχανικής όρασης εξαρτάται σε μεγάλο βαθμό από την προς υλοποίηση εφαρμογή. Ορισμένα συστήματα αποτελούν αυτόνομες εφαρμογές που στοχεύουν στην επίλυση ενός συγκεκριμένου προβλήματος, μέτρησης ή ανίχνευσης, ενώ άλλα αποτελούν ένα υποσύστημα μεγαλύτερου σχεδιασμού το οποίο περιλαμβάνει, για παράδειγμα, υποσυστήματα για τον έλεγχο μηχανικών ενεργοποιητών, σχεδίασης, βάσεις δεδομένων, μηχανικές διεπαφές κλπ. Υπάρχουν τυπικές λειτουργίες που εφαρμόζονται σε πολλά συστήματα μηχανικής όρασης. Η συλλογή εικόνων είναι η πρώτη σε σειρά λειτουργία και αφορά την ψηφιακή εικόνα που παράγεται από έναν ή περισσότερους αισθητήρες εικόνας. Οι τιμές των εικονοστοιχείων αντιστοιχούν συνήθως στην ένταση του φωτός σε μία ή περισσότερες φασματικές ζώνες (γκρίζες εικόνες ή έγχρωμες εικόνες), αλλά μπορεί επίσης να σχετίζονται με διάφορα φυσικά μέτρα, όπως βάθος, απορρόφηση ή ανάκλαση ηχητικών ή ηλεκτρομαγνητικών κυμάτων.

Η προεπεξεργασία είναι μία λειτουργία που αφορά την επεξεργασία των δεδομένων προκειμένου να διασφαλιστεί ότι ικανοποιεί ορισμένες υποθέσεις. Οι συνθήκες αυτές υποδηλώνονται από τη μέθοδο που θα χρησιμοποιήσει η μηχανική όραση στα δεδομένα εικόνας προκειμένου να εξάγει κάποια συγκεκριμένη πληροφορία. Μερικές τεχνικές προεπεξεργασίας είναι οι εξής:

- Επανάληψη της δειγματοληψίας ώστε να επιβεβαιωθεί η ορθότητα του συστήματος συντεταγμένων εικόνας.
- Μείωση θορύβου για να διασφαλιστεί ότι ο θόρυβος του αισθητήρα δεν εισάγει ψευδείς πληροφορίες.
- Βελτίωση αντίθεσης εικόνας για να διασφαλιστεί η ανίχνευση σχετικών Αναπροσαρμογή διαστάσεων εικόνας σε κατάλληλες διαστάσεις.

Η εξαγωγή χαρακτηριστικών από τα δεδομένα εικόνας. Παραδείγματα τέτοιων χαρακτηριστικών είναι:

- Γραμμές, άκρες και κορυφογραμμές.
- Σημεία τοπικού ενδιαφέροντος, όπως γωνίες, κηλίδες ή σημεία.
- Πιο περίπλοκα χαρακτηριστικά μπορεί να σχετίζονται με την υφή, το σχήμα ή την κίνηση.

Η ανίχνευση / τμηματοποίηση αφορά την απόφαση που λαμβάνεται κατά τη διάρκεια της επεξεργασίας σχετικά με το ποια σημεία ή περιοχές της εικόνας είναι σημαντικά για περαιτέρω επεξεργασία. Παραδείγματα είναι:

- Επιλογή συγκεκριμένου συνόλου σημείου ενδιαφέροντος.
- Τμηματοποίηση μιας ή πολλαπλών περιοχών εικόνας που περιέχουν ένα συγκεκριμένο αντικείμενο ενδιαφέροντος.
- Τμηματοποίηση της εικόνας σε σκηνές που περιλαμβάνει το προσκήνιο, τις ομάδες αντικειμένων, τα μεμονωμένα αντικείμενα ή τα κυρίαρχα αντικείμενα.

Η επεξεργασία υψηλού επιπέδου αποτελεί το βήμα όπου η είσοδος είναι συνήθως ένα μικρό σύνολο δεδομένων, για παράδειγμα ένα σύνολο σημείων ή μια περιοχή εικόνας που θεωρείται ότι περιέχει ένα συγκεκριμένο αντικείμενο. Η επεξεργασίας αυτή αφορά, για παράδειγμα:

- Επαλήθευση ότι τα δεδομένα ικανοποιούν παραδοχές που βασίζονται σε μοντέλα και εφαρμογές.
- Εκτίμηση συγκεκριμένων παραμέτρων εφαρμογής, όπως η παρουσίαση αντικειμένου ή το μέγεθος αντικειμένου.
- Αναγνώριση εικόνων - ταξινόμηση ενός ανιχνευθέντος αντικειμένου σε διαφορετικές κατηγορίες.
- Καταχώρηση εικόνας - σύγκριση και συνδυασμός δύο διαφορετικών προβολών του ίδιου αντικειμένου.

Η λήψη αποφάσεων που αφορά την λήψη της τελικής απόφασης που απαιτείται για την εφαρμογή, για παράδειγμα:

- Pass/fail στις εφαρμογές αυτόματης επιθεώρησης.
- Ταίριασμα ή μη σε αντιστοιχία των εφαρμογών αναγνώρισης.
- Επισήμανση για περαιτέρω εξέταση από ανθρώπινο παράγοντα σε εφαρμογές ιατρικής, στρατιωτικής, ασφάλειας και αναγνώρισης.

2.5.3 Αναγνώριση αντικειμένων (Object Detection)

Η τεχνική της ανίχνευσης αντικειμένων (Object Detection) είναι ένα βασικό πεδίο μελέτης τα τελευταία χρόνια στο χώρο της μηχανικής όρασης (Computer Vision). Το κύριο πρόβλημα στην μηχανική όραση είναι η επεξεργασία εικόνας και ο συμπερασμός αν σε αυτή περιέχονται κάποια αντικείμενα, χαρακτηριστικά ή δραστηριότητα

ενδιαφέροντος. Τα πιο γνωστά προβλήματα αναγνώρισης που περιγράφονται στην βιβλιογραφία είναι:

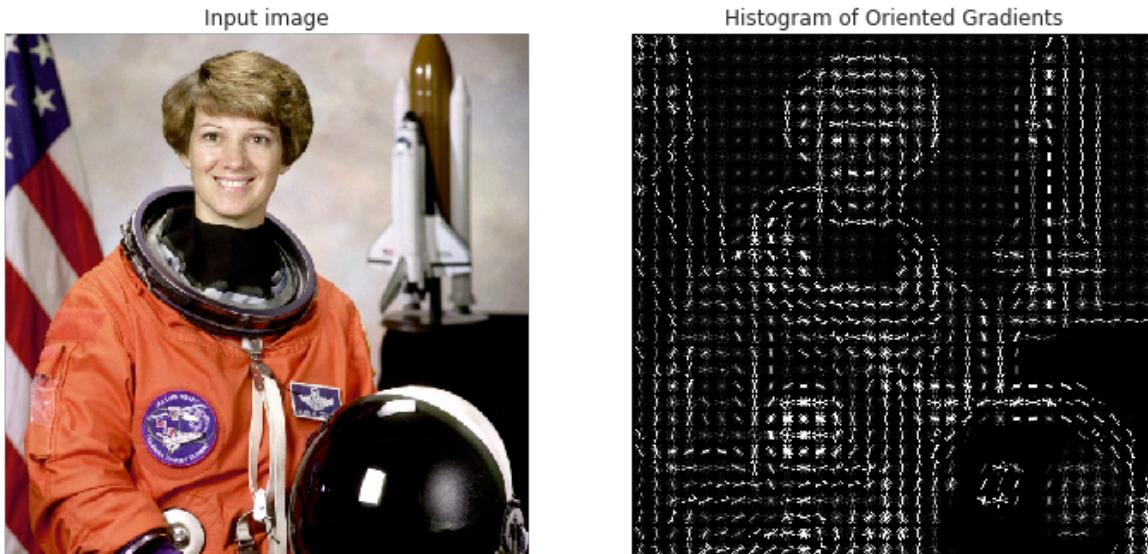
- Η αναγνώριση αντικειμένων (κατηγοριοποίηση αντικειμένων) αφορά ένα ή περισσότερα προσδιορισμένα αντικείμενα ή κλάσεις που μπορούν να αναγνωριστούν.
- Η ταυτοποίηση ή προσδιορισμός της κατηγορίας που ανήκει ένα αντικείμενο που αναγνωρίζεται.
- Η ανίχνευση που αφορά την σάρωση εικόνας σχετικά με μία συνθήκη.

Πολλές φορές γίνεται ανίχνευση μικρών περιοχών πιθανού ενδιαφέροντος στις οποίες εφαρμόζονται μετέπειτα πιο απαιτητικές υπολογιστικά αλγορίθμιμικές τεχνικές ώστε να καθοριστεί το τελικό συμπέρασμα. Τα τελευταία χρόνια λόγω της ραγδαίας ανάπτυξης της έρευνας στο πεδίο της αυτόνομης οδήγησης, έχουν παρουσιαστεί και πολλά διαφορετικά μοντέλα για την διαδικασία της αναγνώρισης αντικειμένων τα οποία προσφέρουν μεγάλη ακρίβεια στις προβλέψεις τους και χρησιμοποιούνται ευρέως [32] [33]. Οι πιο αποδοτικοί αλγόριθμοι για τέτοιου είδους εργασίες βασίζονται στα συνελικτικά νευρωνικά δίκτυα. Κυριότεροι εκπρόσωποι των δυνατοτήτων των δικτύων αυτών αποτελεί το ImageNet Large Scale Visual Recognition Challenge [34]. Το παραπάνω είναι σημείο αναφοράς στην κατηγοριοποίηση και ανίχνευση αντικειμένων, με εκατοντάδες κλάσεις και εκατομμύρια φωτογραφίες. Οι επιδόσεις των συνελικτικών δικτύων στα δοκιμαστικά του ImageNet πλησιάζει αυτή των ανθρώπων.

Παλαιότερα, οι παραδοσιακές μέθοδοι δεν περιλάμβαναν ANN. Οι πιο καθοριστικές για την εποχή τους ήταν οι Viola-Jones [35], HOG [36] και DPM [37]. Η πλειονότητα αυτών των μοντέλων ωστόσο, χρησιμοποιεί παρόμοια βήματα: δεν χρησιμοποιούσαν άμεσα τις RGB τιμές των εικονοστοιχείων της εικόνας ως χαρακτηριστικά εισόδου της κατηγοριοποίησης, αλλά κατείχαν ειδικούς μηχανισμούς εξαγωγής σημαντικών χαρακτηριστικών (feature extractor) από την εικόνα. Η μέθοδος Viola-Jones (V-J) [35] αναπτύχθηκε το 2001 από τους P. Viola και M. Jones και χρησιμοποιήθηκε αρχικά για τον εντοπισμό ανθρώπινων προσώπων. Το στοιχείο που την έκανε να ξεχωρίσει τότε ήταν η εξαγωγή Haar-τύπου χαρακτηριστικών (Haar-like features) από την εικόνα και η αξιοποίηση του αλγορίθμου AdaBoost ο οποίος περιόριζε το σύνολο των χαρακτηριστικών στα πιο αντιπροσωπευτικά. Οι τιμές των Haar-τύπου χαρακτηριστικών ορίζονται από τις εναλλαγές στην ένταση χρώματος σε διάφορες γειτονικές περιοχές της εικόνας, όπως για παράδειγμα αντιθέσεις μεταξύ φωτεινών και σκοτεινών περιοχών. Έτσι, γίνεται εφικτή η αντίληψη σχημάτων όπως γραμμές και γωνίες.

To 2005 προτάθηκε από τους N. Dalal και B. Triggs η μέθοδος Histogram of Oriented Gradients (HOG) [36]. Ομοίως με τη μέθοδο V-J, δεν χρησιμοποιεί απευθείας τις χρωματικές εντάσεις ως χαρακτηριστικά, αλλά χωρίζοντας την εικόνα σε υποπεριοχές εξάγει από κάθε μια τα ιστογράμματα των προσανατολισμένων κλίσεων (oriented gradients). Στη συνέχεια, τα εξαγόμενα χαρακτηριστικά κάθε υποπεριοχής εισάγονται σε έναν κατηγοριοποιητή Support Vector Machine (SVM) [38]. Στο Σχήμα 2.8 παρουσιάζεται ένα παράδειγμα χρήσης της μεθόδου. Η χρήση των προσανατολισμένων κλίσεων αποδείχθηκε μεν πιο ακριβής, αλλά ήταν πιο αργή σε σχέση με τη χρήση Haar-τύπου χαρακτηριστικών στο πρόβλημα εντοπισμού. Αυτό οδήγησε στην ανάπτυξη συστημάτων εντοπισμού που συνδυάζουν και τις δύο οικογένειες χαρακτηριστικών [39], κερδίζοντας ταυτόχρονα το πλεονέκτημα της ακρίβειας και της ταχύτητας.

To Deformable Part Model (DPM) [37] αποτελεί μια βελτιωμένη εκδοχή της με-



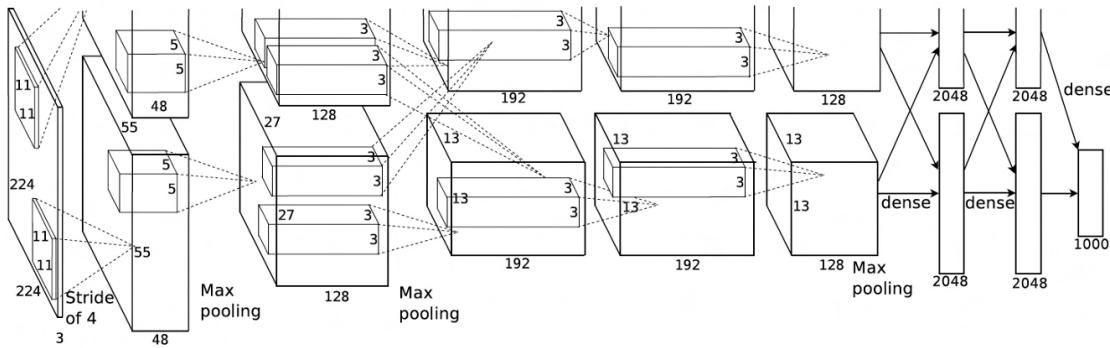
Σχήμα 2.8: Παράδειγμα της μεθόδου HOG [40]

Θόδου HOG. Προτάθηκε από τον P. Felzenszwalb κ.ά. το 2008 και στη συνέχεια βίωσε σημαντικές βελτιώσεις από τον R. Girshick που αύξησαν την ταχύτητα εντοπισμού κατά 10 φορές διατηρώντας σταθερή την ακρίβεια. Οι άριστες επιδόσεις του αλγορίθμου αναδείχθηκαν στο διαγωνισμό Pascal Visual Object Classes (VOC) [41] όπου εφαρμόστηκε σε σύνολο δεδομένων κοινών αντικειμένων και αποτέλεσε αναφορά για πολλαπλά νικητήρια συστήματα την περίοδο 2008-2012.

Την περίοδο που ακολούθησε, οι κλασσικοί εξαγωγείς χαρακτηριστικών της εποχής αδυνατούσαν να επιλύσουν πιο σύνθετα προβλήματα. Το 2012 ο διαγωνισμός ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [34] ωθησε την ανάπτυξη νέων καθοριστικών μεθόδων που είχαν ως βασικό συστατικό τα DNN. Στόχος του διαγωνισμού αυτού ήταν η αναγνώριση των κλάσεων των αντικειμένων από το σύνολο δεδομένων μεγάλης κλίμακας ImageNet [42], ένα από τα πιο σύνθετα και μεγάλα σύνολα του είδους του. Οι νικητές του διαγωνισμού, A. Krizhevsky, I. Sutskever και G. Hinton, ανέπτυξαν ένα CNN με το όνομα AlexNet [43], η δομή του οποίου φαίνεται στο Σχήμα 2.9. Το AlexNet ζεχώρισε χρησιμοποιώντας διάφορες καινοτόμες τεχνικές. Αρχικά καθοριστική ήταν η χρήση pooling μέγιστης τιμής σε επικαλυπτόμενες περιοχές της εικόνας. Το AlexNet κατέστησε τη μάθηση δυνατή με τη χρήση της υπολογιστικής δύναμης και μνήμης πολλαπλών GPU, κάτι πρωτοπόρο για την εποχή. Η πολλά υποσχόμενη επίδοση των CNN στην αναγνώριση κλάσης άνοιξε το δρόμο για την ανάπτυξη καινούργιων παραλλαγών τους με τη δυνατότητα όχι μόνο αναγνώρισης αλλά και εντοπισμού.

Η ανίχνευση αντικειμένων μπορεί να ομαδοποιηθεί σε δύο επιμέρους τεχνικές, τον εντοπισμό της θέσης των αντικειμένων (object localization) και την κατηγοριοποίηση αντικειμένων (object classification). Ο εντοπισμός της θέσης των αντικειμένων είναι η τεχνική κατά την οποία εντοπίζονται όλα τα αντικείμενα προς κατηγοριοποίηση στην εικόνα.

Υπάρχουν αρκετοί αλγόριθμοι για την ανίχνευση αντικειμένων και μπορούν να χωριστούν σε δύο κατηγορίες. Η ανίχνευση ενός μεμονωμένου στιγμιότυπου της κλάσης από την εικόνα ονομάζεται ανίχνευση αντικειμένου μεμονωμένης κλάσης (single class object detection), ενώ η ανίχνευση των κλάσεων όλων των αντικειμένων που υπάρχουν



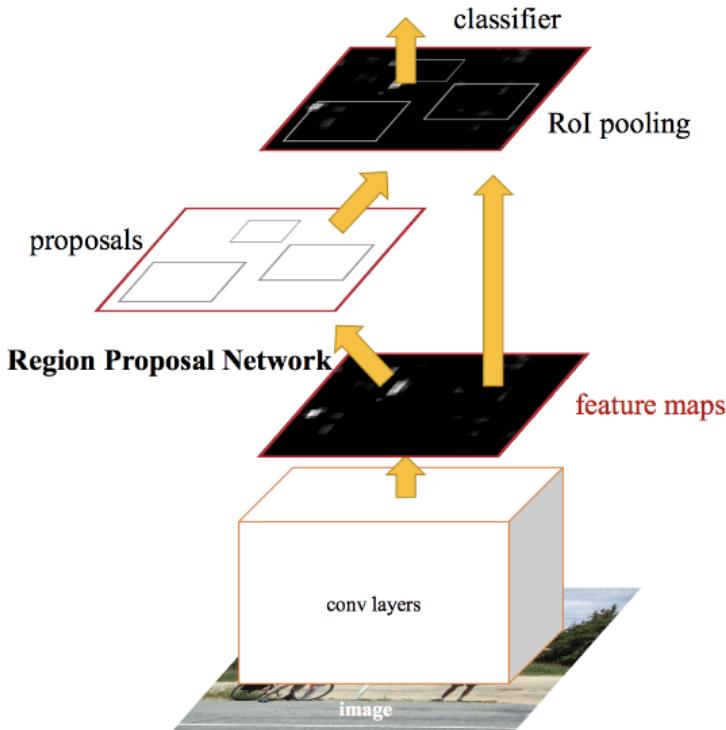
Σχήμα 2.9: Δομή του δικτύου AlexNet[43]

στην εικόνα είναι γνωστή ως ανίχνευση αντικειμένου πολλαπλών κλάσεων (multi class object detection).

Ακόμα, τα σύγχρονα μοντέλα εντοπισμού και αναγνώρισης αντικειμένου χωρίζονται σε δύο κατηγορίες αναλόγως με το πως διενεργούνται οι προβλέψεις στην κεφαλή του δικτύου: οι ανιχνευτές δύο σταδίων two-stage detectors και οι ανιχνευτές ενός σταδίου one-stage detectors. Στους two-stage detectors, το μοντέλο ξεκινάει με την φάση παραγωγής προτάσεων περιοχών ενδιαφέροντος (region proposals), δηλαδή περιοχές με υψηλή πιθανότητα να περιέχουν ένα αντικείμενο, και σε επόμενη φάση διενεργείται η ταξινόμηση (classification) του αντικειμένου χρησιμοποιώντας συνελικτικά νευρωνικά δίκτυα. Αντίθετα, στους one-stage detectors δεν παράγονται προτάσεις περιοχών και το μοντέλο διενεργεί την αναγνώριση κλάσης και τον εντοπισμό απευθείας από τα χαρακτηριστικά ολόκληρης της εικόνα, αντιμετωπίζοντας τη διαδικασία σαν ένα πρόβλημα παλινδρόμησης (regression). Τα one-stage μοντέλα, αν και φημίζονται για την υψηλή τους ταχύτητα, τείνουν να εμφανίζουν χαμηλότερα ποσοστά ακρίβειας σε σχέση με τα two-stage, είναι όμως καταλληλότερα για την αναγνώριση αντικειμένων σε πραγματικό χρόνο.

Στα πιο γνωστά two-stage μοντέλα ανήκει και το R-CNN (Region-based Convolutional Neural Network) που προτάθηκε το 2014[44][45]. Η μεθοδολογία του μοντέλου είναι απλή. Πρώτον, διενεργείται εξαγωγή προτάσεων περιοχών ενδιαφέροντος στην εικόνα με τη χρήση του αλγορίθμου Selective Search. Ο αλγόριθμος εκτελεί αρχική τμηματοποίηση της εικόνας σύμφωνα με τον αλγόριθμο του P. Felzenszwalb, με βάση την ομοιότητα σε χρώμα, υφή, μέγεθος και σχήμα μεταξύ των υποπεριοχών της εικόνας, και επαναληπτικά συγχωνεύει γειτονικά τμήματα με παρόμοιες ιδιότητες. Τα τελικά τμήματα οριοθετούνται σε πλαίσια που θα αποτελέσουν τις προτεινόμενες περιοχές ενδιαφέροντος. Οι προτάσεις αυτές είναι υποπεριοχές της εικόνας με υψηλή πιθανότητα να περιέχουν υποψήφιο αντικείμενο. Στη συνέχεια, κάθε περιοχή-πρόταση εισάγεται σε ένα CNN με στόχο την εξαγωγή σημαντικών χαρακτηριστικών σε μορφή διανύσματος. Τα διανύσματα χαρακτηριστικών τροφοδοτούνται εντός ενός κατηγοριοποιητή SVM ο οποίος είναι υπεύθυνος για την απόδοση κάποιας κλάσης στο οριοθετημένο αντικείμενο. Τέλος, μέσω μοντέλου παλινδρόμησης τελειοποιείται η ακρίβεια των περιβαλλόντων κουτιών γύρω από το αντικείμενο, ελαχιστοποιώντας το σφάλμα. Η δομή του R-CNN παρουσιάζεται στο Σχήμα 2.10.

Παρ' όλα αυτά, το R-CNN εμφάνιζε κάποια σημαντικά μειονεκτήματα. Αρχικά,



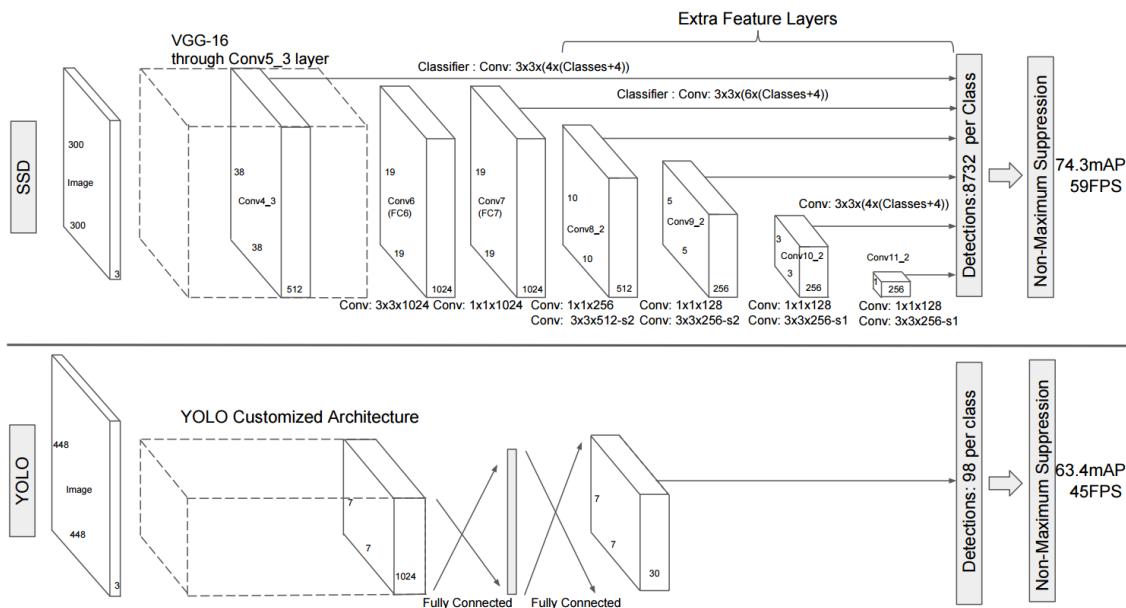
Σχήμα 2.10: Δομή του R-CNN [44]

λόγω του μεγάλου πλήθους των περιοχών-προτάσεων που παράγει το δίκτυο, η εισαγωγή κάθε μίας από αυτές στο CNN αυξάνει το κόστος τόσο στον χρόνο εντοπισμού όσο και στον χώρο αποθήκευσης των διανυσμάτων χαρακτηριστικών κάθε περιοχής. Επιπλέον, το CNN περιορίζεται σε εισόδους συγκεκριμένου μεγέθους λόγω των κατηγοριοποιητών SVM (ή fc επιπέδων) που έπονται των conv επιπέδων. Οπότε, κάθε περιοχή ενδιαφέροντος υπόκειται σε κανονικοποίηση στις κατάλληλες διαστάσεις, κάτι που επηρεάζει την ποιότητα της εικόνας σημαντικά. Για τους λόγους αυτούς, προτάθηκαν οι Fast-R-CNN [46], Faster-RCNN [47] και ο Mask-RCNN [48], οι οποίοι αποτελούν παραλλαγές του απλού R-CNN.

Το 2015 αναπτύχθηκε από τον J. Redmon κ.ά. το πρώτο one-stage detectors μοντέλο εντοπισμού αντικειμένων πραγματικού χρόνου, το You Only Look Once (YOLO)[49]. Οι δημιουργοί του ακολούθησαν μια ανατρεπτική τακτική ως προς τη διαδικασία εντοπισμού. Συγκεκριμένα ο εντοπισμός των περιβαλλόντων κουτιών και η αναγνώριση του αντικειμένου επιτυγχάνονταν ταυτόχρονα σε ένα CNN το οποίο παρήγαγε στην έξοδο τις συντεταγμένες όλων των κουτιών και τις πιθανότητες των κλάσεων τους με μια ανάγνωση ολόκληρης της εικόνας, εξού και το όνομα του. Η αρχιτεκτονική του YOLO απεικονίζεται στο Σχήμα 2.11. Η λογική που ακολουθούσε το YOLO ήταν απλή. Αρχικά, η εικόνα χωρίζεται σε ένα τετραγωνικό πλέγμα $k \times k$ κελιών. Μέσω των σον επιπέδων, κάθε κελί ανάγεται σε διάνυσμα M χαρακτηριστικών, το σύνολο των οποίων εισάγεται ως τένοσορας $k \times k \times M$ στα δύο διαδοχικά fc επίπεδα του δικτύου για τις προβλέψεις. Οι προβλέψεις του δικτύου ήταν δύο ειδών, του εντοπισμού και της αναγνώρισης κλάσης. Για τον εντοπισμό, το κάθε κελί ήταν υπεύθυνο για την οριοθέτηση του αντικείμενου του οποίου το κέντρο βρισκόταν εντός του κελιού. Έτσι, για κάθε κελί παράγεται πρόβλεψη ενός πλήθους περιβαλλόντων κουτιών του αντικειμένου. Το κάθε περιβάλλον κουτί αποτελεί διάνυσμα

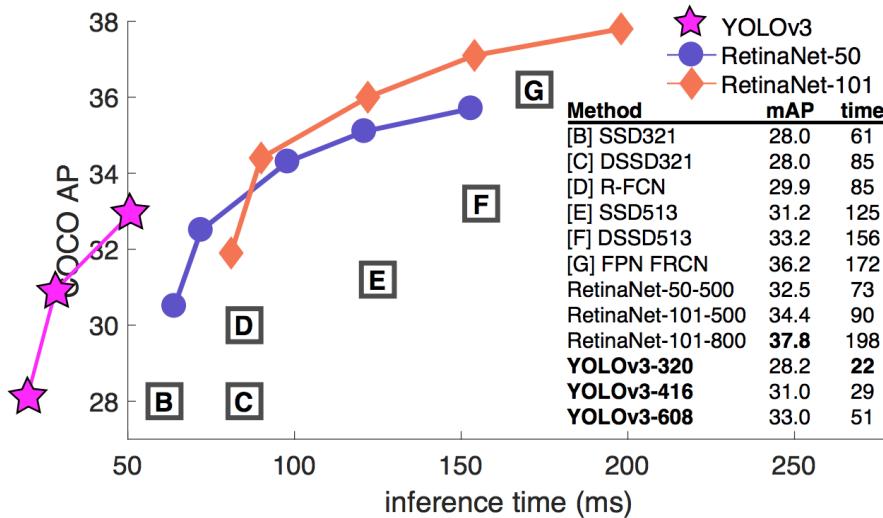
Το YOLO ήταν εξαιρετικά ταχύτερο ($6\times$ από το Faster R-CNN) και παράλληλα ανάλογα ακριβές (mAP 63,4% στο VOC2007) με άλλα state-of-the-art μοντέλα της εποχής. Αντί τεχνικών πρότασης περιοχών ενδιαφέροντος, το YOLO χρησιμοποιούσε την πληροφορία από ολόκληρη την εικόνα για να προβλέψει την τοποθεσία και την κλάση όλων των αντικειμένων. Έτσι, σχημάτιζε πιο σφαιρική εικόνα για την όψη των αντικειμένων, γενικεύοντας καλύτερα σε καινούργια δεδομένα συγκριτικά με το Fast R-CNN, το οποίο λανθασμένα εντόπιζε κάποια «μη αντικείμενα» από το φόντο της εικόνας πιο συχνά από το YOLO. Παρ' όλα αυτά, αντιμετώπιζε δυσκολία στον εντοπισμό πολύ μικρών και πυκνά τοποθετημένων αντικειμένων καθώς ήταν δυνατός ο εντοπισμός μονάχα ενός αντικειμένου ανά κελί. Στις επόμενες εκδόσεις του YOLO, v2, v3, v4 και v5, ο J. Redmon κ.ά. επιχείρησαν να λύσουν το πρόβλημα.

Ο W. Liu κ.ά. πρότειναν το 2015 το μοντέλο Single Shot MultiBox Detector (SSD) [50], το δεύτερο one-stage μοντέλο εντοπισμού μετά το YOLO. Το SSD διέφερε από τα υπόλοιπα μοντέλα εντοπισμού της εποχής στο ότι παρήγαγε για κάθε εικόνα χάρτες χαρακτηριστικών σε πολυάριθμες κλίμακες μεγέθους και στη συνέχεια χρησιμοποιούσε όλες τις αναπαραστάσεις αυτές για τον εντοπισμό, τεχνική που βελτιώθηκε στην πορεία στα δίκτυα FPN χρησιμοποιώντας παράλληλα υπερδειγματοληψία εντός της διαδικασίας. Έτσι, επέτρεπε υψηλή ακρίβεια στον εντοπισμό εικόνων διαφόρων αναλύσεων, ακόμα και πολύ μικρών. Επιπλέον, οι προβλέψεις των περιβαλλόντων κουτιών κάθε κλίμακας και των κλάσεων τους δεν πραγματοποιούνταν από fc επίπεδα αλλά από conv επίπεδα και εξάγονταν με τη μορφή μετατοπίσεων από κάποια προκαθορισμένα αρχικά κουτιά. Για κάθε αντικείμενο προβλέπονταν πολλαπλές υποψήφιες μετατοπίσεις, από τις οποίες επιλεγόταν ως τελικό περιβάλλον κουτί αυτή με τη μεγαλύτερη τιμή εμπιστοσύνης μέσω αλγορίθμου Non-Maximum Suppression (NMS). Το SSD ήταν ταχύτερο και πιο ακριβές από την πρώτη έκδοση του YOLO και το ίδιο ακριβές (mAP 74,3% στο VOC2007), αν όχι περισσότερο, από άλλα two-stage detectors μοντέλα, όπως το Faster R-CNN. Στο Σχήμα 2.11 παρουσιάζονται οι διαφορές στη δομή του YOLO και του SSD.



Σχήμα 2.11: Δομή του δικτύου YOLO σε σύγκριση με το δίκτυο SSD [50]

Στο Σχήμα 2.12 παρουσιάζεται σύνοψη των προαναφερθέντων μεθόδων ως προς την επίδοση τους σε ακρίβεια (mAP) και ταχύτητα.



Σχήμα 2.12: Απόδοση των μοντέλων ανίχνευσης αντικειμένων [51]

Μερικές προκλήσεις που χρίζουν χειρισμού κατά την εκτέλεση ανίχνευσης ενός αντικειμένου είναι η μερική/πλήρης συγκάλυψη του αντικειμένου, διαφορετικές συνθήκες φωτισμού, στάσεις στην κάμερα, κλιμάκωση, κλπ. Γενικότερα, η ανίχνευση αντικειμένων είναι το πρώτο βήμα σε οποιαδήποτε δραστηριότητα οπτικής αναγνώρισης.

2.6 Αλγόριθμοι και εργαλεία

2.6.1 Προσομοιωτής CARLA

Ο CARLA [17] είναι προσομοιωτής οδήγησης ανοιχτού κώδικα. Κατασκευάστηκε για να χρησιμοποιηθεί ως μία ευέλικτη διεπαφή προγραμματισμού (API) για αυτόνομο οχήματα και για την εκτέλεση μιας σειράς εργασιών που εμπλέκονται στο πρόβλημα της αυτόνομης οδήγησης, στις οποίες ο χρήστης μπορεί να αλλάξει και να ελέγχει παραμέτρους στη προσομοίωση. Έχει κατασκευαστεί πάνω από τη μηχανή παιχνιδιών Unreal Engine 4 για την εκτέλεση της προσομοίωσης και χρησιμοποιεί το πρότυπο OpenDRIVE (ανοιχτή μορφή αρχείου για τη λογική περιγραφή των οδικών δικτύων) για τον σχεδιασμό των δρόμων και κανόνων οδήγησης. Τα τελευταία χρόνια οι εκδόσεις ανανεώνονται συνεχώς. Η έκδοση 0.9.13 (CARLA 0.9.13) είναι αυτή που χρησιμοποιήθηκε στην παρούσα διπλωματική (νεότερη έκδοση) η οποία κυκλοφόρησε το 2021 και περιλαμβάνει όλα τα απαραίτητα χαρακτηριστικά (features) που χρειάστηκαν για την παρούσα διπλωματική [52].

Ένας από τους κύριους στόχους του προσομοιωτή CARLA είναι να δώσει την ευκαιρία σε περισσότερους χρήστες για ενασχόληση με την αυτόνομη οδήγηση, να χρησιμοποιηθεί ως ένα εργαλείο εύκολο στην πρόσθαση και στην προσαρμογή του από τους χρήστες. Για να γίνει αυτό, ο προσομοιωτής πρέπει να πληροί τις απαραίτησης διαφορετικών περιπτώσεων χρήσης στο γενικό πρόβλημα της οδήγησης (π.χ. εκμάθηση



Σχήμα 2.13: CARLA simulator [52]

πολιτικών οδήγησης, αλγόριθμοι εκπαίδευσης αντίληψης κλπ.). Ανάμεσα σε άλλα ο χρήστης μπορεί να καθορίσει μέσω της διεπαφής τον οδικό περιβάλλον (Town01 - Town10), να αλλάξει τις καιρικές συνθήκες τον αριθμό των οχημάτων και πεζών που θα υπάρχουν σε κάθε σενάριο κ.α. Ο έλεγχος της προσομοίωσης παρέχεται μέσω μιας διεπαφής (API), που μπορεί να χρησιμοποιηθεί με γλώσσες προγραμματισμού Python και C++. Το PythonAPI που είναι διαθέσιμο περιλαμβάνει GUI δημιουργημένο με το PyGame με το οποίο ο χρήστης μπορεί να ελέγχει το όχημα που έχει επιλέξει χειροκίνητα.

Για την παρούσα διπλωματική είναι αρκετά σημαντική η δυνατότητα να αλλάξουν οι συνθήκες καιρού, το φως και το οπτικό πεδίο μπροστά στο όχημα, κάτι που επηρεάζει ιδιαίτερα την αντίληψη των αντικειμένων του περιβάλλοντος και μπορεί να διευκολύνει ή να δυσχεραίνει τις προβλέψεις. Στη συγκεκριμένη εργασία, μια RGBκάμερα τοποθετήθηκε στο μπροστινό μέρος, ακριβώς στη μέση του οχήματος, ώστε το οπτικό πεδίο που καταγράφει να μοιάζει με αυτό του οδηγού.

Οι **καιρικές συνθήκες** και οι **συνθήκες φωτισμού** μπορούν να δοθούν από το χρήστη ή να επιλεχθούν από μια σειρά προκαθορισμένων ρυθμίσεων [52]:

- 0 - Default
- 1 - ClearNoon
- 2 - CloudyNoon
- 3 - WetNoon
- 4 - WetCloudyNoon
- 5 - MidRainyNoon
- 6 - HardRainNoon
- 7 - SoftRainNoon
- 8 - ClearSunset
- 9 - CloudySunset
- 10 - WetSunset

- 11 - WetCloudySunset
- 12 - MidRainSunset
- 13 - HardRainSunset
- 14 - SoftRainSunset

Περιλαμβάνει επίσης **10 χάρτες πόλεων** με διαφορετική διάταξη για την προσαρμογή σε κάθε σενάριο που μπορεί να θέλει να εκτελέσει ο χρήστης. Ο όρος **Actors** στον προσομοιωτή CARLA αναφέρεται σε οποιοδήποτε στοιχείο εκτελεί ενέργειες μέσα στο περιβάλλον του, δηλαδή οχήματα, πεζοί, αισθητήρες, φωτεινοί σηματοδότες και σήματα οδικής κυκλοφορίας. Ο προσομοιωτής έχει μεγάλη ποικιλία σε actors, καθώς ο χρήστης μπορεί να επιλέξει ανάμεσα σε 26 οχήματα, μια μεγάλη λίστα αισθητήρων, ποδήλατα, μηχανάκια, φορτηγά και πεζούς όλων των ηλικιών, ενώ επίσης ο χρήστης μπορεί να ελέγχει τους πεζούς. Η **φυσική του οχήματος** είναι αρκετά ρεαλιστική, όπως στην σύγκρουση των οχημάτων και κατά την ολίσθηση του οχήματος σε βρεγμένο δρόμο. Διαθέτει την δυνατότητα για έλεγχο των φυσικών χαρακτηριστικών του οχήματος, όπως το βάρος του οχήματος, η μέγιστη ταχύτητα, οι τριβές με το οδόστρωμα κλπ. Ακόμα, το Python API μπορεί να χρησιμοποιηθεί για τον σχεδιασμό συγκεκριμένων **σεναρίων οδήγησης (scenarios)**. Ο CARLA επιτρέπει να εκτελεστεί ξανά ακριβώς το ίδιο σενάριο πολλές φορές, δίνοντας ακριβώς τα ίδια δεδομένα από τους αισθητήρες και τις καταστάσεις των οχημάτων και των φαναριών [52].

Στην CARLA, υπάρχουν επίσης τα **σημεία διαδρομής (waypoints)** στον χάρτη σε κάθε πιθανή τοποθεσία οχήματος. Για κάθε λωρίδα, αυτά τα σημεία έχουν μια ορισμένη απόσταση μεταξύ τους. Για κάθε waypoint, η CARLA παρέχει πολλές πληροφορίες, όπως θέση και προσανατολισμός του σημείου πορείας σύμφωνα με τη λωρίδα του, αναγνώριση δρόμου και πληροφορίες λωρίδας μέσω του Python API. Οι πληροφορίες λωρίδας, αποτελούνται από το αναγνωριστικό λωρίδας, το πλάτος, την πληροφορία για τις διασταυρώσεις (αν επιτρέπεται η στροφή αριστερά ή/και δεξιά), τον τύπο (στάθμευση, πεζοδρόμιο, διάβαση, κλπ.), τύπος σήμανσης δεξιάς και αριστερής λωρίδας (συμπαγής, διακεκομμένη). Επιπροσθέτως, το Python API περιλαμβάνει μεθόδους που επιστρέφουν τα επόμενα ή τα προηγούμενα waypoints. Εάν δύο σημεία είναι πιο κοντά από ένα κατώφλι, θεωρούνται ως ένα από τον CARLA.

Όσο βρίσκεται σε λειτουργία αυτόματου πιλότου, για κάθε όχημα και πεζό η CARLA δημιουργεί μια λίστα με τυχαία waypoints και επιλέγει ένα από αυτά ως πρώτο προορισμό. Κάθε φορά που ένας actor φτάνει στον προορισμό του, ο CARLA επιλέγει τυχαία μια άλλη από τη λίστα ή δημιουργεί νέες τοποθεσίες εάν η λίστα είναι κενή. Κατά τη μετακίνηση, εάν υπάρχει μόνο ένα σημείο μπροστά από τον actor, τότε ο actor μετακινείται σε αυτό το σημείο. Εάν υπάρχουν περισσότερα σημεία μπροστά που σημαίνει ότι ο actor πρέπει να επιλέξει μεταξύ διαφορετικών λωρίδων ή δρόμων τότε χρησιμοποιούνται κάποιοι αλγόριθμοι σχεδιασμού διαδρομής για την επιλογή του κατάλληλου.

2.6.2 Η βιβλιοθήκη OpenCV

Η επεξεργασία εικόνας στη παρούσα διπλωματική επιτεύχθηκε με τη χρήση της βιβλιοθήκης OpenCV (Open Source Computer Vision). Η βιβλιοθήκη είναι ανεξάρτητη πλατφόρμας και υλοποιεί συναρτήσεις υπολογιστικής όρασης συμβατές σε Python, C,

C++, Java και Matlab όπως επίσης και στα λειτουργικά συστήματα Windows, Linux, Android και Mac. Περιέχει πάνω από 2500 αλγόριθμους και έχει σχεδιαστεί με στόχο την υψηλή υπολογιστική αποδοτικότητα και χρησιμοποιείται σε πολλές εφαρμογές, κυρίως πραγματικού χρόνου (real time), όπως συστήματα αναγνώρισης χαρακτηριστικών προσώπων, προηγμένα ρομποτικά συστήματα, συστήματα ανίχνευσης κίνησης κ.α. [53] [54]



Σχήμα 2.14: OpenCV [53]

Η OpenCV περιέχει περισσότερες από 500 συναρτήσεις κατάλληλες για πολλές διαφορετικές εφαρμογές στους τομείς της οπτικής επεξεργασίας. Διαθέτει επίσης τη Βιβλιοθήκη Μηχανικής Μάθησης (MLL), η οποία αποτελείται από διάφορες συναρτήσεις και εργαλεία που συμβάλλουν στη στατιστική ταξινόμηση, στην παλινδρόμηση και στην ομαδοποίηση των δεδομένων, καθώς και συναρτήσεις για τη προβολή και την αποθήκευση των βίντεο και ο,τιδήποτε άλλο έχει να κάνει με τη διεπαφή χρήστη-εφαρμογής. Η διάδοση της σε συνδυασμό με την απλότητα χρήσης της και το γεγονός ότι είναι ανοιχτού κώδικα (Open-Source) την καθιστούν ιδιαίτερα ελκυστική και είναι οι λόγοι επιλογής της στην παρούσα διπλωματική. Η έκδοση που θα χρησιμοποιηθεί είναι η 3.3.0, η οποία υλοποιήθηκε το 2017.

2.6.3 Μετασχηματισμός Hough

Ο Μετασχηματισμός Hough (Hough Transform) [55] αποτελεί μια τεχνική που χρησιμοποιείται για να απομονώσει κάποια χαρακτηριστικά ενός συγκεκριμένου σχήματος σε μια εικόνα, γραμμών, κύκλων και άλλων απλών μορφών. Οι Duda & Hart τον εισήγαγαν στην επεξεργασία εικόνας και στην υπολογιστική όραση. Στην παρούσα εργασία η έμφαση δίνεται στον Μετασχηματισμό Γραμμών Hough (Hough Lines Transform) δεδομένου ότι η ανίχνευση αφορά τις λωρίδες του δρόμου και τις διαστραυρώσεις, που είναι ευθείες γραμμές.

Ειδικότερα, μελετάμε την ανίχνευση ευθειών χρησιμοποιώντας μια διαδικασία «ψηφοφορίας». Η κεντρική ιδέα είναι η μετατροπή του προβλήματος ανίχνευσης ευθειών, σε ένα πιο απλό πρόβλημα ανίχνευσης μεγίστων στον χώρο των παραμέτρων της ευθείας, που ονομάζεται Χώρος Hough.

Αρχικά, εφαρμόζεται μια τεχνική ανίχνευσης ακμών όπως αυτή του Canny Edge Detector, όπως θα περιγραφεί αναλυτικά και σε επόμενη παράγραφο [56]. Η βασική θεωρητική σκέψη πίσω από τον Μετασχηματισμού Γραμμών Hough είναι πως κάθε σημείο - pixel σε μία δυαδική εικόνα μπορεί να είναι τμήμα κάποιας γραμμής. Μία γραμμή στο χώρο της εικόνας μπορεί να εκφραστεί με δύο μεταβλητές είτε στο Καρτεσιανό σύστημα αξόνων (Cartesian Coordinate System) είτε στο Πολικό σύστημα αξόνων

(Polar Coordinate System). Για να χρησιμοποιήσουμε τον μετασχηματισμό, χρειαζόμαστε έναν τρόπο για να εκφράσουμε μαθηματικά την ευθεία. Συγκεκριμένα,

$$y = mx + b \quad (2.3)$$

όπου m είναι η κλίση της ευθείας και b είναι το σημείου τομής της ευθείας με τον άξονα των y . Ωστόσο, όταν οι ευθείες τείνουν να γίνουν κάθετες, τα μεγέθη m και b τείνουν προς το άπειρο. Επομένως, μία πιο χρήσιμη αναπαράσταση μίας γραμμής είναι μέσω των πολικών συντεταγμένων:

$$x \cos \theta + y \sin \theta = \rho \quad (2.4)$$

Η εξίσωση εκφράζει μια ευθεία που περνάει από το σημείο (x, y) και είναι κάθετη στην ευθεία που ξεκινάει από την αρχή των αξόνων και καταλήγει στο σημείο (ρ, θ) των πολικών συντεταγμένων δηλαδή, $(\rho \cos \theta, \rho \sin \theta)$ στις καρτεσιανές συντεταγμένες, όπως φαίνεται και στο Σχήμα 2.15.

Σε μια ευθεία, τα θ και ρ σε κάθε σημείο είναι σταθερά. Για κάθε δεδομένο σημείο (x, y) , παίρνουμε τις ευθείες που περνούν από αυτό λύνοντας ως προς θ και ρ . Με την χρήση πιθανών τιμών της γωνίας θ , μπορούμε να λύσουμε με χρήση της εξίσωσης κατευθείαν ως προς ρ [55].

Δημιουργείται και καθορίζεται ένα πλέγμα από τις παραμέτρους (ρ, θ) και το επιλεγμένο βήμα. Στην συνέχεια, σε κάθε σημείο του πλέγματος συσσωρεύεται μια τιμή, η οποία φανερώνει πόσο καλά ένα αντικείμενο που έχει προκύψει από τις παραμέτρους που καθορίζονται για αυτό το σημείο αντιστοιχεί στην επεξεργαζόμενη εικόνα. Στο πλέγμα τα σημεία που συσσωρεύουν μεγάλες τιμές, είναι εκείνα που περιγράφουν χαρακτηριστικά που έχουν την δυνατότητα να προβληθούν πάνω στην αρχική εικόνα, ταιριάζοντας σε ένα βαθμό με χαρακτηριστικά που πραγματικά υπάρχουν στην εικόνα.

Υποθέτουμε ότι για την εύρεση του μετασχηματισμού Hough των ευθειών μιας εικόνας, επιλέγεται ένα συγκεκριμένο βήμα για τη γωνία, και γίνεται εναλλαγή ανάμεσα στις γωνίες που καθορίζονται σε αυτό το βήμα. Για κάθε γωνία θ λύνουμε ως προς ρ :

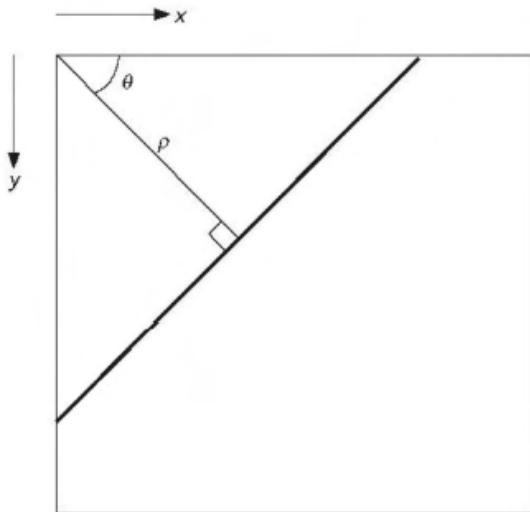
$$\rho = x \cos \theta + y \sin \theta \quad (2.5)$$

Στην συνέχεια, αυξάνεται η τιμή που βρίσκεται στο (ρ, θ) . Αυτή η διαδικασία μπορεί να χαρακτηρισθεί και ως ψήφος του σημείου (x, y) για την ευθεία που καθορίζεται από τις συντεταγμένες (ρ, θ) . Υπάρχουν αρκετές παραλλαγές για το πως δημιουργούνται οι ψήφοι ώστε να κάνουν το μετασχηματισμό περισσότερο σταθερό.

Ταυτόχρονα, για να ανιχνευθεί μία γραμμή απαιτείται και μία τιμή κατωφλίου, η οποία θα καθορίζει την ελάχιστη τιμή που απαιτείται για να θεωρηθεί ότι σχηματίζεται μία γραμμή.

2.6.4 YOLO και darknet

To YOLO (Joseph Redmon, 2016) [49] είναι ένας αλγόριθμος που χρησιμοποιεί νευρωνικά δίκτυα για τα παρέχει εντοπισμό και αναγνώριση αντικειμένων σε πραγματικό χρόνο. Ο συγκεκριμένος αλγόριθμος είναι γνωστός λόγω της ταχύτητας και της ακρίβειάς του. Έχει χρησιμοποιηθεί σε ποικίλες εφαρμογές για τον εντοπισμό πινακίδων, ανθρώπων, οχημάτων, ζώων κ.α.



Σχήμα 2.15: Πολικές συνταταγμένες για τον Μετασχηματισμό Hough [57]

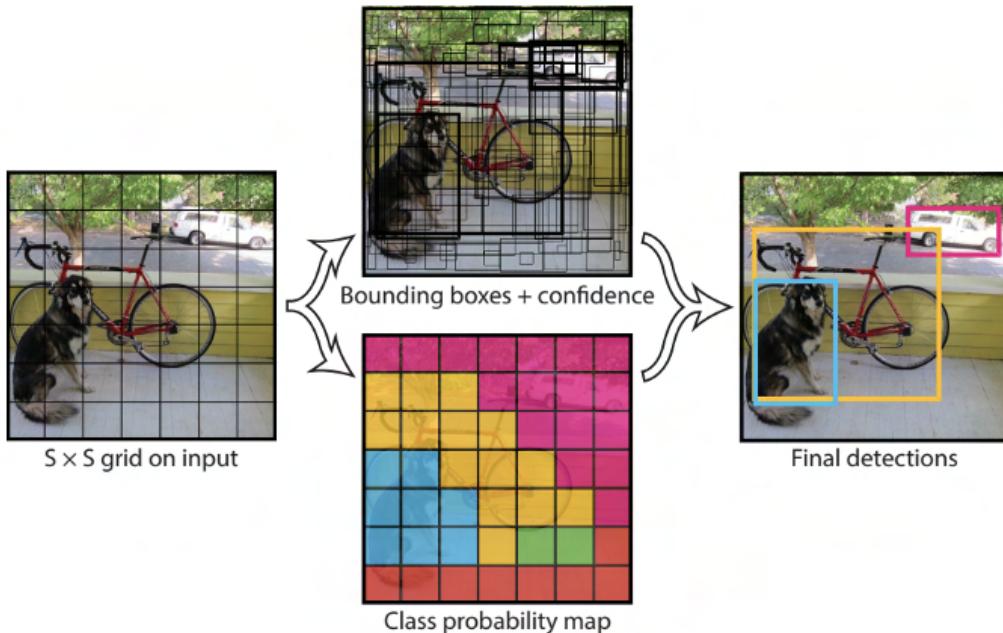
Ο συγκεκριμένος αλγόριθμος χρησιμοποιεί συνελικτικά νευρωνικά δίκτυα για την αναγνώριση αντικειμένων σε πραγματικό χρόνο. Χαρακτηρίζεται από το όνομά του You Only Look Once που υποδηλώνει πως τόσο κατά τον εντοπισμό/αναγνώρισή όσο και κατά την εκπαίδευση, το δίκτυο λαμβάνει την εικόνα εισόδου μία φορά και δεν την επεξεργάζεται ξανά μετά την είσοδο κατά την εκτέλεση του. Αυτό σημαίνει ότι η πρόβλεψη σε μια ολόκληρη εικόνα γίνεται μέσα σε ένα κύκλο του αλγορίθμου. Για το YOLO, η ανίχνευση είναι ένα απλό **πρόβλημα παλινδρόμησης** το οποίο παίρνει μια εικόνα εισόδου και επιστρέφει τις πιθανότητες των κλάσεων και τις συνταταγμένες του πλαισίου οριοθέτησης. Στην συγκεκριμένη διπλωματική χρησιμοποιείται το μοντέλο YOLOv4.

Ο συγκεκριμένος αλγόριθμος επιλέχθηκε για τους εξής λόγους:

- **Ταχύτητα:** Ο χρόνος εντοπισμού βελτιώνεται επειδή μπορεί να κάνει προβλέψεις σε πραγματικό χρόνο.
- **Υψηλή ακρίβεια:** Παρέχει ακριβή αποτελέσματα με ελάχιστα λάθη στο παρασκήνιο.
- **Μαθησιακές δυνατότητες:** Ο αλγόριθμος έχει εξαιρετικές μαθησιακές πιθανότητες που του επιτρέπουν να μάθει το παρουσιαστικό των αντικειμένων και να τα εφαρμόσει στην αναγνώριση αντικειμένων

Επειδή η επεξεργασία εισόδου γίνεται μία φορά, η αρχιτεκτονική του νευρωνικού αποτελείται από επίπεδα συνέλιξης, υποδειγματοληψίας και από δύο ολικά συνδεδεμένα επίπεδα στο τέλος. Όπως φαίνεται και στο Σχήμα 2.16, ο YOLO διαιρεί κάθε εικόνα σε πλέγμα $S \times S$ και κάθε πλέγμα προβλέπει N οριοθετούμενα πλαίσια και ένα σκορ εμπιστοσύνης. Το σκορ αυτό αυτό αντανακλά την ακρίβεια του πλαισίου και αν το πλαίσιο περιέχει πράγματι ένα αντικείμενο, ανεξάρτητα από την κλάση στην οποία θέλουμε να κατηγοριοποιήσουμε τα αντικείμενα. Ο YOLO προβλέπει επίσης τη βαθμολογία των κατηγοριών για κάθε πλαίσιο, για κάθε κατηγορία στη φάση της εκπαίδευσης. Επομένως, προβλέπονται συνολικά πλαίσια $S \times S \times N$. Ωστόσο, τα περισσότερα από αυτά τα πλαίσια έχουν χαμηλή βαθμολογία αξιοπιστίας. Θέτοντας μία επιθυμητή

τιμή κατωφλίου, κάθε πλαίσιο με βαθμό μεγαλύτερο του κατωφλίου, επιλέγεται και χρησιμοποιείται για την ανεύρεση ενός αντικειμένου.



Σχήμα 2.16: Διαίρεση εικόνας σε πλέγμα YOLO [29]

Η πληροφορία που περιγράφει ένα **πλαίσιο οριοθέτησης** είναι (x , y , w , h , σκορ εμπιστοσύνης), όπου:

(x, y) : το κέντρο του περιθλήματος

w : πλάτος ορθογωνίου / πλάτος εικόνας

h : ύψος ορθογωνίου / ύψος εικόνας

Επιπλέον κάθε κελί έχει και C πιθανότητες $Pr(Class_i | Object)$, $i = 1, \dots, C$.

Ως συναρτήσεις ενεργοποίησης, αντί για τις πλέον διαδεδομένες χρησιμοποιείται μια παραπλήσια μορφή:

$$\varphi(x) = \begin{cases} x, & x > 0 \\ 0.1x, & \text{else} \end{cases} \quad (2.6)$$

Η εκπαίδευση γίνεται με κύριο σκοπό την βελτιστοποίηση του τετραγωνικού σφάλματος της εξόδου.

Η μετρική **IoU (Intersection over Union)** περιγράφει πως αλληλεπικαλύπτονται το πραγματικό πλαίσιο σε σχέση με το πλαίσιο που προβλέφθηκε. Κάθε πλεγματικό κελί είναι υπεύθυνο για να προβλέπει τα περιμετρικά πλαίσια και το ποσοστό σιγουριάς τους. Το IoU είναι ίσο με 1 εάν το περιμετρικό πλαίσιο που βρέθηκε είναι το ίδιο με το πραγματικό. Αυτός ο μηχανισμός απορρίπτει περιμετρικά πλαίσια που δεν είναι ίδια με τα πραγματικά. Στο Σχήμα 2.17 παρουσιάζονται κάποια παραδείγματα σχετικά με την αξιολόγηση της μετρικής.

Όπως αναφέραμε, ο αλγόριθμος διατρέχει την εικόνα μόνο μια φορά. Ως εκ τούτου, ο YOLO είναι τάξεις μεγέθους ταχύτερος (45 καρέ ανά δευτερόλεπτο) από άλλους



Σχήμα 2.17: Μετρική Intersection over Union [58]

αλγόριθμους ανίχνευσης αντικειμένων. Μια άλλη βασική διαφορά είναι ότι το YOLO βλέπει την πλήρη εικόνα ταυτόχρονα σε αντίθεση με την εξέταση μόνο προτάσεων περιοχής που παρατηρήθηκαν σε άλλες μεθόδους. Έτσι, αυτές οι πληροφορίες που παρέχονται από τον αλγόριθμο, συμβάλλουν στην αποφυγή ψευδών θετικών. Μια άλλη περίπτωση είναι ότι ενώ προσπαθεί να γενικευθεί για μικρά και μεγάλα αντικείμενα, όταν αυτά βρίσκονται σε εικόνες με διαφορετικές αναλογίες διαστάσεων ή επαναλαμβάνονται σε ένα γκρουπ με μικρές διαστάσεις αποτυγχάνει τον εντοπισμό τους. Το σύστημα που θα αναπτυχθεί στην παρούσα εργασία, θα υλοποιηθεί με τον αλγόριθμο YOLOv4.

2.6.5 Tesseract OCR

Η Οπτική Αναγνώριση Χαρακτήρων (Optical Character Recognition) ή αλλιώς Αυτόματη Αναγνώριση Χαρακτήρων Κειμένου ονομάζεται η διαδικασία μετατροπής σαρωμένων εικόνων χειρογράφων ή έντυπων κειμένων σε κείμενο αναγνώσιμο από ηλεκτρονικό υπολογιστή. Είναι δηλαδή η μέθοδος με την οποία ψηφιοποιούνται έντυπα κειμένων έτσι ώστε να είναι προσπελάσιμα ηλεκτρονικά, να μπορούν αποθηκεύονται σε μικρότερο χώρο, να είναι προσβάσιμα από τον παγκόσμιο ιστό και αποτελεί πεδίο έρευνας για τους τομείς της αναγνώρισης προτύπων, της τεχνητής νοημοσύνης και της κατανόησης των εικόνων από τον ηλεκτρονικό υπολογιστή.

Η αντιστοίχηση με πρότυπα αφορά την αναγνώριση χαρακτήρων από έτοιμα πρότυπα ή περιγράμματα χαρακτήρων. Ο σαρωτής ψηφιοποιεί την εικόνα ενός εγγράφου στον υπολογιστή και το λογισμικό Οπτικής Αναγνώρισης προσπαθεί να ταιριάξει, με ένα βαθμό πιθανότητας, τους χαρακτήρες από το σαρωμένο αρχείο εικόνας με τα πρότυπα που έχει αποθηκευμένα. Αν η εικόνα ενός χαρακτήρα αντιστοιχεί με αναγνωρισμένο χαρακτήρα, τότε αντιστοιχίζεται με χαρακτήρα κειμένου για τον ηλεκτρονικό υπολογιστή.

Η μηχανή Tesseract-OCR θεωρείται μία από τις καλύτερες ελεύθερες (Open Source) στην οπτική αναγνώριση χαρακτήρων. Οι αρχικές εκδόσεις μπορούσαν να αναγνωρίσουν μόνο αγγλική γλώσσα, ενώ πλέον μπορεί να αναγνωρίσει κείμενο σε περισσότερες από 35 γλώσσες. Ένας τρόπος για να το επιτύχει αυτό είναι το ενσωματωμένο λεξιλόγιο, που της επιτρέπει να αναγνωρίζει σωστά λέξεις, ακόμα και αν κάποιος από τους χαρακτήρες θα αναγνωριζόταν λανθασμένα ως μεμονωμένος [59] [60] [61]. Η μηχανή Tesseract δίνει στο χρήστη τη δυνατότητα να προσαρμόσει κάποιες παραμέτρους,

ώστε να βελτιώσει τα αποτελέσματα αναγνώρισης. Κατά τη διαδικασία αρχικοποίησης της εφαρμογής, γίνεται φόρτωση κάποιας γλώσσας και του αντίστοιχου λεξικού, δηλαδή των λέξεων που περιέχει. Επίσης, δίνεται η δυνατότητα να προσδιοριστεί ένα υποσύνολο των χαρακτήρων, οι οποίοι θα είναι οι μόνοι που θα γίνονται δεκτοί ως αποτέλεσμα αναγνώρισης, η ονομαζόμενη «λευκή λίστα» (whitelist). Επιπλέον, μπορεί να γίνει διαχωρισμός, ώστε στην αρχή να αναγνωρίζει μόνο γράμματα και μετά μόνο αριθμούς, κάτι που επίσης μπορεί να βελτιώσει την αναγνώριση και αφορά ιδιαίτερα την παρούσα διπλωματική [59].

Η Tesseract είναι κατάλληλη για χρήση στο παρασκήνιο (backend), αλλά μπορεί να χρησιμοποιηθεί και για περισσότερο πολύπλοκα καθήκοντα OCR συμπεριλαμβανομένης ανάλυσης της διάταξης με τη χρήση ενός περιβάλλοντος διεπαφής (frontend) [62] [60]. Μερικές από τις εφαρμογές της επικεντρώνονται στην ανάγνωση κειμένων από σαρωμένες εικόνες από έντυπα, την εύρεση του κειμένου σε σύνθετες εικόνες, ή την ανάγνωση χειρόγραφου κειμένου.

Κεφάλαιο 3

Μεθοδολογία

Στην διπλωματική που παρουσιάζεται έχει αναπτυχθεί με χρήση της γλώσσας προγραμματισμού Python ένα σύστημα οδήγησης για αυτόνομο όχημα. Στο παρόν κεφάλαιο θα παρουσιαστεί το σύστημα και η μεθοδολογία που χρησιμοποιήθηκε για την ανάπτυξη του, καθώς και παραδείγματα για την καλύτερη κατανόηση.

3.1 Παρουσίαση υποσυστημάτων

Στα πλαίσια της παρούσας διπλωματικής εργασίας, υλοποιήθηκαν τρία υποσυστήματα αντίληψης, πλοϊγησης και κατασκευής βασικού μονοπατιού. Το υποσύστημα αντίληψης επιχειρεί να εξάγει και να ερμηνεύσει χρήσιμες πληροφορίες από τις εικόνες που λαμβάνει από την ενσωματωμένη κάμερα σε σχέση με τις λωρίδες του δρόμου, τα στατικά και δυναμικά εμπόδια του περιβάλλοντος. Στο υποσύστημα πλοϊγησης υλοποιείται ένα σύστημα διατήρησης λωρίδας χρησιμοποιώντας τον εντοπισμό των λωρίδων από το υποσύστημα αντίληψης, που επιχειρεί να διατηρήσει το όχημα στα όρια του δρόμου. Ακόμα, υλοποιείται ένα επιπλέον σύστημα που χρησιμοποιεί PID ελεγκτή και αποκλειστικά GPS για την πλοϊγηση, εντοπίζοντας την θέση του οχήματος και ακολουθώντας τα προκαθορισμένα σημεία του μονοπατιού. Τέλος, υλοποιήθηκε μια μέθοδος κατασκευής βασικού μονοπατιού, η οποία μπορεί να χρησιμοποιηθεί για την πλοϊγηση. Τα συστήματα περιγράφονται αναλυτικά στις παραγράφους που ακολουθούν.

3.1.1 Υποσύστημα αντίληψης (Perception subsystem)

Το σύστημα αντίληψης είναι υπεύθυνο για την καταγραφή και υπολογισμό των δεδομένων του περιβάλλοντος του οχήματος, δηλαδή όλης της απαραίτητης πληροφορίας που χρειάζεται το αυτόνομο όχημα για την λειτουργία του, όπως πληροφορίες για την δομή του δρόμου, τις θέσεις των οχημάτων και των πεζών γύρω του, των φωτεινών σηματοδοτών, της θέσης του οχήματος σε σχέση με τις λωρίδες του δρόμου και τα όρια του δρόμου. Επίσης, το υποσύστημα είναι υπεύθυνο για την διασύνδεση του CARLA simulator με το υπόλοιπο σύστημα αυτόνομου οχήματος, δηλαδή λειτουργεί σαν γέφυρα μεταξύ του λογισμικού της αυτόνομης πλοϊγησης που δημιουργήθηκε στην παρούσα διπλωματική με το CARLA simulator. Το σύστημα «αντίληψης» λειτουργεί θεωρώντας ότι λαμβάνεται όλη η απαραίτητη πληροφορία που σχετίζεται με τον δρόμο από την εγκατεστημένη κάμερα στο όχημα.

3.1.2 Υποσύστημα πλοϊγησης και ελέγχου (Navigation subsystem)

Το υποσύστημα «πλοϊγησης και ελέγχου» είναι υποσύστημα χαμηλού επιπέδου και χρησιμοποιείται για να ελέγξει τη κίνηση του οχήματος. Δεδομένης της επιθυμητής διαδρομής και της πληροφορίας που λαμβάνεται από το υποσύστημα αντίληψης, το υποσύστημα παράγει τις κατάλληλες τιμές για τα πεντάλ (γκάζι, φρένο) του οχήματος και για την γωνία περιστροφής του τιμονιού, ώστε το όχημα να έχει την επιθυμητή ταχύτητα και κατεύθυνση, και οι τιμές αυτές που προκύπτουν εφαρμόζονται στην συνέχεια στο όχημα μέσω του προσομοιωτή CARLA.

3.1.3 Υποσύστημα κατασκευής βασικού μονοπατιού (Route planning subsystem)

Το υποσύστημα «κατασκευής βασικού μονοπατιού» χρησιμοποιεί τον χάρτη του προσομοιωτή CARLA και ένα σημείο στόχο για να παράσχει την βασική διαδρομή που πρέπει να ακολουθήσει το όχημα. Επίσης, είναι υπεύθυνο για τον εντοπισμό του οχήματος μέσα στο μονοπάτι που παράχθηκε.

3.2 Σύστημα αντίληψης (Perception System)

3.2.1 Εισαγωγή

Η διαδικασία της αντίληψης σε ένα αυτόνομο όχημα πραγματοποιείται με τη χρήση καμερών, για την επεξεργασία και την κατανόηση του περιβάλλοντος γύρω από το όχημα σε πραγματικό χρόνο. Το σύστημα αντίληψης (perception system) ενός αυτόνομου οχήματος παρατηρεί το περιβάλλον και την αλληλεπίδρασή του με το όχημα. Είναι υπεύθυνο να επιτελεί λειτουργίες όπως η αίσθηση (sensing), η συλλογή δεδομένων από τους αισθητήρων (sensor fusion), ο εντοπισμός (localization) αντικειμένων στο περιβάλλον και γενικότερα λειτουργιών που είναι παρόμοιες με την αντίστοιχη διαδικασία της οπτικής αντίληψης του ανθρώπου. Είναι από τα πιο σημαντικά υποσυστήματα ενός αυτόνομου οχήματος καθώς είναι υπεύθυνο να συλλέγει συνεχώς πληροφορίες όπως λωρίδες κυκλοφορίας, σήματα οδικής κυκλοφορίας, την θέση των δυναμικών και στατικών αντικειμένων στο περιβάλλον γύρω του κ.α. Με την ενσωμάτωση όλων αυτών των εργασιών, γίνεται κατανοητή η κατάσταση του εξωτερικού κόσμου με βάση τα δεδομένα των αισθητήρων. Επιπλέον, θα πρέπει να είναι ικανό να λειτουργεί το ίδιο αποτελεσματικά υπό διαφορετικές καιρικές συνθήκες και καθ' όλη τη διάρκεια της ημέρας και της νύχτας. Δεδομένου ότι το σύστημα αντίληψης επεξεργάζεται και παράγει πληροφορίες και για τα υπόλοιπα υποσυστήματα του οχήματος, είναι εμφανές ότι ένα αξιόπιστο και ακριβές σύστημα αντίληψης είναι εξαιρετικά κρίσιμο.

3.2.2 Περιγραφή υλοποίησης

Για την απόκτηση πληροφορίας από το περιβάλλον σχετιζόμενη με τα αντικείμενα και τον χώρο χρησιμοποιήθηκε η RGB κάμερα που προσφέρει ο προσομοιωτής CARLA. Οι πληροφορίες που λαμβάνονται επεξεργάζονται κατάλληλα και μετατρέπονται σε πληροφορία για το αυτόνομο όχημα. Θεωρούμε ότι η κάμερα έχει τοποθετηθεί στην οροφή του οχήματος και η οπτική που καταγράφει είναι αντίστοιχη με αυτή του οδηγού.

Τα βασικά δεδομένα λαμβάνονται από την κάμερα και στη συνέχεια επεξεργάζονται μέσω του PythonAPI του CARLA για να εξάγουμε τις εξής πληροφορίες:

- Οι λωρίδες κυκλοφορίας στο οδόστρωμα.
- Τα σημεία διασταύρωσεων.
- Η θέση και η τιμή επερχόμενου σήματος ορίου ταχύτητας.
- Η κατάσταση και η θέση των φωτεινών σηματοδοτών.
- Η θέση των δυναμικών εμποδίων, δηλαδή των οχημάτων, των φορτηγών και των πεζών.

Το σύστημα αντίληψης που κατασκευάστηκε στην παρούσα διπλωματική περιέχει όλες τις κατάλληλες αλγορίθμικές διαδικασίες ώστε να μετατρέψει τα παραπάνω δεδομένα που λαμβάνονται από την κάμερα σε χρήσιμη πληροφορία για το όχημα.

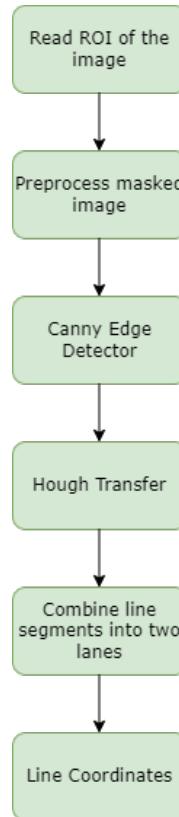
3.2.3 Αναγνώριση Λωρίδων (Lane Detection)

Η αναγνώριση των λωρίδων στο οδόστρωμα είναι μέρος του συστήματος αντίληψης του αυτόνομου οχήματος, ενώ με τις πληροφορίες που θα εξαχθούν από τη συγκεκριμένη διαδικασία σε σχέση με τη θέση του οχήματος στο οδόστρωμα θα πρέπει να γίνουν οι αντίστοιχες ενέργειες σε άλλα υποσυστήματα. Ο αλγόριθμος που αναπτύχθηκε έχει συγκεκριμένα βήματα και η μεθοδολογία είναι αρκετά κοινή για τα περισσότερα προβλήματα ίδιου τύπου.

Η τυπική διαδικασία που ακολουθείται για την προεπεξεργασία και τον εντοπισμό των λωρίδων είναι η εξής [63]:

1. συλλογή της εικόνας
2. μετατροπή της εικόνας σε κλίμακα του γκρι (grayscale)
3. εφαρμογή Γκαουσιανού φίλτρου για μείωση του θορύβου
4. χρήση του Canny Edge Detector για τον εντοπισμό των περιοχών της εικόνας που έχουν απότομη αλλαγή της φωτεινότητας
5. απομόνωση μιας περιοχής ενδιαφέροντος της εικόνας
6. Μετασχηματισμός Hough
7. εντοπισμός των ακμών στην περιοχή ενδιαφέροντος
8. ομιδοποίηση και τελική επιλογή των ακμών που σηματοδοτούν την αριστερή και δεξιά λωρίδα
9. υπολογισμός των συντεταγμένων των αρχικών και τελικών σημείων των ακμών που θα αναπαραστήσουν τις λωρίδες στην εικόνα

Εφαρμόσαμε τον αλγόριθμο σε εικόνες που έχουν συλλεχθεί από τον προσομοιωτή και έπειτα το σύστημα ενσωματώθηκε στο CARLA. Η διαδικασία εντοπισμού των λωρίδων είναι ένα σχετικά απλούστερο πρόβλημα για ιδανικές καιρικές συνθήκες και φωτισμού, ενώ μπορεί να αυξηθεί ο βαθμός δυσκολίας του σε άλλες συνθήκες, όπως σκιά, ομίχλη, έντονη βροχή, νυχτερινό τοπίο. Για τον λόγο αυτό, ο αλγόριθμος είναι σημαντικό να ελεγχθεί σε πολλές διαφορετικές συνθήκες.



Σχήμα 3.1: Διαδικασία εντοπισμού λωρίδων

Στη συνέχεια, περιγράφονται αναλυτικά όλα τα βήματα που εφαρμόστηκαν.

Μετατροπή σε κλίμακα του γκρι

Το χρώμα ενός εικονοστοιχείου (pixel) σε μια εικόνα δημιουργείται από έναν συνδυασμό τριών χρωμάτων, του κόκκινου, του πράσινου και του μπλε (Red Green Blue (RGB)). Οι τιμές χρώματος RGB αναπαρίστανται από ένα διάνυσμα τριών διαστάσεων (x, y, z). Μια έγχρωμη εικόνα εξαρτάται ποιοτικά από το χρώμα που αντιπροσωπεύει ο αριθμός των bit. Αν κάθε θέση για το κόκκινο, πράσινο και μπλε χρώμα καταλαμβάνει 8 bit τότε ο συνδυασμός RGB καταλαμβάνει 24 bit και υποστηρίζει 2^{24} διαφορετικά χρώματα.

Τα 24 bit αντιπροσωπεύουν το χρώμα ενός εικονοστοιχείου στην έγχρωμη εικόνα, ενώ η αντίστοιχη εικόνα της κλίμακας του γκρι (grayscale image) παριστάνεται με φωτεινότητα (luminance) χρησιμοποιώντας 8 bit. Η φωτεινότητα μιας τιμής εικονοστοιχείου σε κλίμακα του γκρι κυμαίνεται από 0 έως 255. Με την τιμή 0 να υποδηλώνει μηδενική φωτεινότητα, δηλαδή μαύρο χρώμα, ενώ με την τιμή 255 να υποδηλώνει μέγιστη φωτεινότητα, δηλαδή άσπρο χρώμα. Η μετατροπή (conversion) μιας έγχρωμης εικόνας σε μια εικόνα κλίμακας του γκρι μετατρέπει τις τιμές RGB, δηλαδή γίνεται μετατροπή από 24 bit σε 8 bit.

Είναι σημαντικό να κατανοήσουμε ότι μια εικόνα μπορεί να την αναπαρασταθεί ως πίνακας, όπου κάθε εικονοστοιχείο της εικόνας αποτελεί μια αντίστοιχη εγγραφή στον πίνακα. Υπολογίζοντας την κλίση (gradient) της εικόνας γίνεται η εύρεση αλλαγών στην φωτεινότητα σε μια σειρά από γειτονικά εικονοστοιχεία. Μια μεγάλη κλίση υποδηλώνει

μια απότομη αλλαγή, ενώ μια μικρή κλίση αντιπροσωπεύει μια μικρή αλλαγή. Αυτό μας βοηθάει να προσδιορίσουμε τις ακμές στην εικόνα, αφού μια ακμή ορίζεται από τη διαφορά στις τιμές φωτεινότητας. Μετατρέποντας μια έγχρωμη εικόνα σε εικόνα κλίμακα του γκρι, όπως φαίνεται στο Σχήμα 3.2, καταφέρνουμε να ελαχιστοποιήσουμε τον υπολογιστικό χρόνο επεξεργασίας της εικόνας κάτι που είναι επιθυμητό στη διαδικασία εντοπισμού των λωρίδων σε πραγματικό χρόνο.



Σχήμα 3.2: Μετατροπή εικόνας σε κλίμακα του γκρι

Μείωση Του Θορύβου (Noise Reduction)

Επειδή η ανίχνευση των ακρών είναι επιρρεπής στο θόρυβο, είναι απαραίτητο, σαν πρώτο βήμα, να αφαιρεθεί ο θόρυβος από την εικόνα. Αυτό επιτυγχάνεται με την εφαρμογή κάποιων φίλτρων στην αρχική εικόνα.

Στην επεξεργασία εικόνας, το Gaussian Filter ή αλλιώς Gaussian Blur χρησιμοποιείται για τη μείωση του θορύβου στο στάδιο προεπεξεργασίας της εικόνας. Το Γκαουσιανό Φίλτρο είναι ένα χαμηλοπερατό φίλτρο (low-pass filter) που χρησιμοποιεί μια Γκαουσιανή συνάρτηση για να υπολογίσει το μετασχηματισμό που πρέπει να εφαρμοστεί σε κάθε πιζελ της εικόνας.

Κατά την εφαρμογή του φίλτρου, πραγματοποιείται αλλαγή στην τιμή ενός εικονοστοιχείου με τη μέση τιμή της φωτεινότητας των εικονοστοιχείων γύρω από αυτό. Ο μέσος όρος των εικονοστοιχείων της εικόνας για τη μείωση του θορύβου θα γίνει με τον πυρήνα (kernel), ο οποίος παίρνει τιμές από την κανονική κατανομή (normal distribution).

Συγκεκριμένα, ο πυρήνας κανονικά κατανεμημένων αριθμών τρέχει σε ολόκληρη την εικόνα και ορίζει το καθένα εικονοστοιχείο ίσο με τον σταθμισμένο μέσο όρο των γειτονικών εικονοστοιχείων, εξομαλύνοντας έτσι την εικόνα από τον θόρυβο. Στην εικόνα

3.3 χρησιμοποιούμε θόλωση σε μια γκρι εικόνα με έναν πυρήνα μεγέθους 5×5 και τυπικής απόκλισης $\sigma = 0.0$.



Σχήμα 3.3: Μείωση του θορύβου της εικόνας με Γκαουσιανό φίλτρο

Εφαρμογή της συνάρτησης Canny Edge Detector

Η Ανίχνευση Ακρών Canny (Canny Edge Detection) είναι ένας δημοφιλής αλγόριθμος ανίχνευσης ακρών. Αναπτύχθηκε από τον John F. Canny το 1986. Η τεχνική εύρεσης των ακρών της εικόνας είναι μια μέθοδος που επιτρέπει αφενός τον καθορισμό των δομικών πληροφοριών των αντικειμένων που περιέχονται σε μια εικόνα, αφετέρου μειώνει δραματικά την ποσότητα των πληροφοριών που πρέπει να επεξεργαστεί ένα σύστημα προκειμένου να ολοκληρώσει κάποιες διεργασίες, όπως την ανίχνευση των λωρίδων. Στην εικόνα 3.4 παρουσιάζεται το αποτέλεσμα μετά την εφαρμογή του Canny Edge Detector σε μία εικόνα που έχει ήδη μετατραπεί σε κλίμακα του γκρι και έχει υποστεί μείωση θορύβου.

Ο αλγόριθμος Canny Edge Detector έχει επικρατήσει χάρη στα τρία κριτήρια απόδοσης:

- **Υψηλά ποσοστά ανίχνευσης:** Στόχος είναι να υπάρχει χαμηλή πιθανότητα να ανιχνευθεί ένα σημείο ψευδώς ως ακμή και ταυτόχρονα να είναι χαμηλή και η πιθανότητα να μην αναγνωριστεί ένα πραγματικό σημείο ακμής ως τέτοιο.
- **Καλή τοπικοποίηση (good localization):** Τα σημεία που χαρακτηρίζονται άκρες από τον τελεστή πρέπει να είναι όσο το δυνατόν πιο κοντά στις πραγματικές άκρες της εικόνας.



Σχήμα 3.4: Εφαρμογή του Canny Edge Detector

Επιλογή Περιοχής Ενδιαφέροντος (ROI)

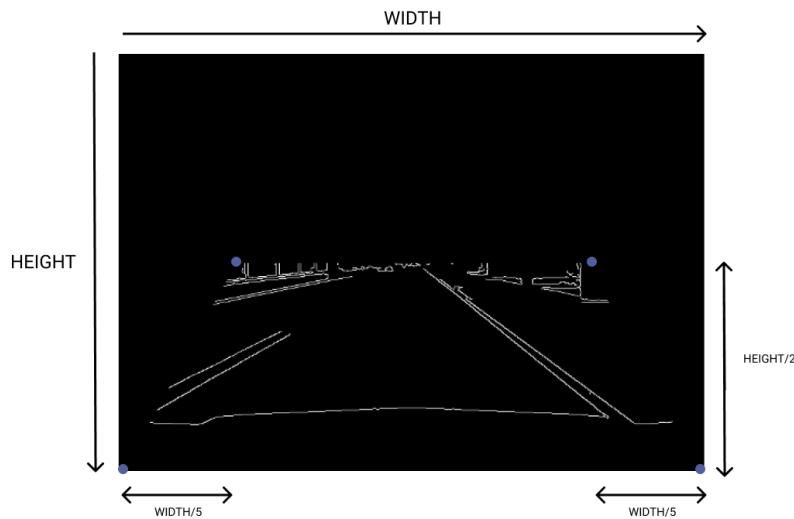
Πολλές φορές για να αυξηθεί η ταχύτητα επεξεργασίας χρειάζεται να διασφαλιστεί ότι οι υπολογιστικοί πόροι δεν δαπανώνται για περιπτούς υπολογισμούς. Για να επιτευχθεί αυτό, χρειάζεται να επιλέξουμε την περιοχή της εικόνας που μας ενδιαφέρει να επεξεργαστούμε, αφήνοντας την υπόλοιπη αμετάβλητη. Η Περιοχή Ενδιαφέροντος (Region of Interest – ROI) είναι ένα τμήμα της εικόνας στο οποίο πρόκειται να γίνει κάποια επεξεργασία. Στις εφαρμογές υπολογιστικής όρασης το (ROI) ορίζεται ως τα όρια της περιοχής που περιέχει το προς εξέταση αντικείμενο.

Η βιβλιοθήκη της OpenCV παρέχει έτοιμη συνάρτηση επιλογής περιοχών ενδιαφέροντος. Ωστόσο, υπάρχουν και εναλλακτικοί τρόποι επιλογής ενός τέτοιου παραθύρου, όπως ο καθορισμός ενός συγκεκριμένου σχήματος παραθύρου επιλογής, όπως έγινε στη παρούσα διπλωματική και μπορούμε να παρατηρήσουμε στο Σχήμα 3.5. Δημιουργώντας μια μάσκα και περιορίζοντας την περιοχή ενδιαφέροντος μόνο στον χώρο του οδοστρώματος μπροστά από τον οδηγό, εντοπίζονται μόνο τις γραμμές του δρόμου που βρίσκονται μπροστά στο όχημα.

Hough Line Transform και εντοπισμός των λωρίδων

Μετά το στάδιο της προεπεξεργασίας, στην εικόνα εφαρμόζεται ο Μετασχηματισμός Hough για τον τελικό εντοπισμό των ακμών. Έχουμε ήδη αναφέρει αναλυτικά το θεωρητικό υπόβαθρο του μετασχηματισμού Hough και του χώρου Hough στο Κεφάλαιο 2. Η βιβλιοθήκη OpenCV μας προσφέρει έτοιμη τη συνάρτηση για την εφαρμογή του μετασχηματισμού, η οποία επιστρέφει τις παραμέτρους των ευθειών που βρίσκει στην περιοχή ενδιαφέροντος.

Όπως είναι λογικό, οι ευθείες οι οποίες προκύπτουν συνήθως είναι πολλές και δεν είναι όλες αντιπροσωπευτικές των λωρίδων. Για τον λόγο αυτό, θα πρέπει να γίνει ένα



Σχήμα 3.5: Συντεταγμένες της περιοχής ενδιαφέροντος

φιλτράρισμα αυτών. Αρχικά, ομιδοποιούμε τις ακμές σε αριστερή και δεξιά λωρίδα σύμφωνα με την κλίση τους και τις παραμέτρους που επιστράφηκαν από τον Μετα-σχηματισμό Hough, ενώ κρατάμε σε κάθε λίστα μόνο αυτές με κλίσεις που ξεπερνάνε ένα κατώφλι. Αυτό γίνεται για να αποφύγουμε πιθανά λάθη στον εντοπισμό, όπως για παράδειγμα ευθείες που τείνουν να γίνουν οριζόντιες και αλλοιώνουν το αποτέλεσμα. Η αρνητική κλίση σηματοδοτεί την αριστερή λωρίδα, ενώ θετική κλίση τη δεξιά. Έπειτα, βρίσκουμε τον μέσο όρο (average) για κάθε λίστα ακμών, ορίζουμε το αρχικό και τελικό σημείο κάθε ακμής που προέκυψε και προβάλλουμε τις γραμμές που προέκυψαν στην αρχική εικόνα, όπως παρουσιάζεται στην εικόνα 3.6.



Σχήμα 3.6: Τελική αναπαράσταση των λωρίδων στην εικόνα

3.2.4 Αναγνώριση Αντικειμένων (Object Detection)

Η τεχνική της ανίχνευσης αντικειμένων (Object Detection) αποτελεί μια από της βασικές λειτουργίες του συστήματος αντίληψης του αυτόνομου οχήματος. Όπως ήδη έχει αναφερθεί στο Κεφάλαιο 2, για την ανίχνευση αντικειμένων χρειάζεται να εντοπιστεί η θέση των αντικειμένων (object localization) και να γίνει κατηγοριοποίηση των αντικειμένων (object classification) στην εικόνα που λαμβάνεται από τον προσομοιωτή.

Από τους αλγορίθμους που είναι έχουν αναπτυχθεί για αυτή τη διαδικασία επιλέχθηκε ο YOLO (You Only Look Once). Το πρόβλημα έχει μεγάλες απαιτήσεις στη ταχύτητα ανίχνευσης, επομένως θεωρήθηκε η βέλτιστη επιλογή για την διαδικασία ανίχνευσης αντικειμένων σε πραγματικό χρόνο, θυσιάζοντας μικρά ποσοστά ακρίβειας σε σχέση με τις μεθόδους δύο σταδίων που έχουν ήδη αναφερθεί.

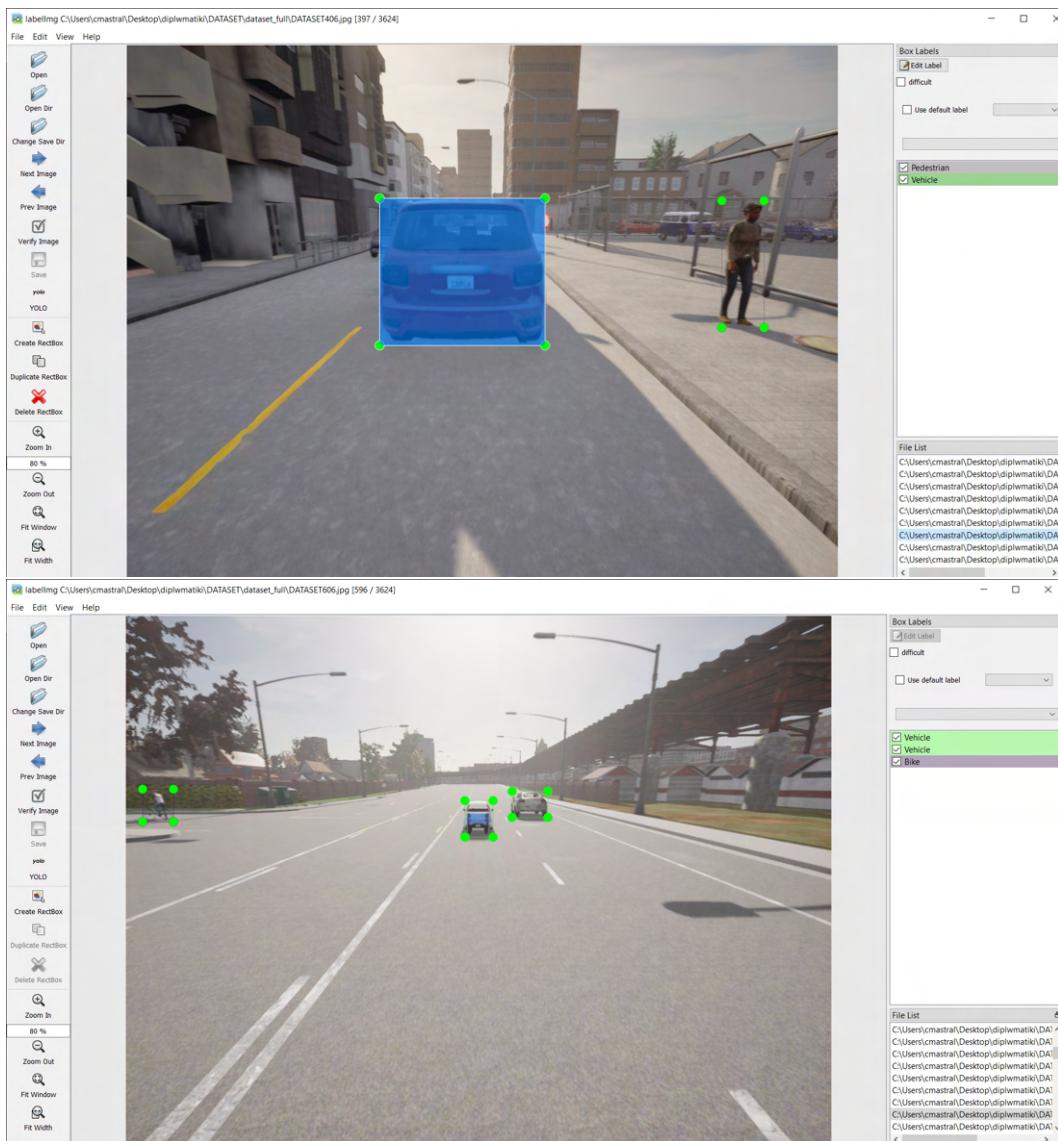
Σε αυτή την ενότητα, θα αναλυθεί η σχεδίαση της διαδικασίας και η εκπαίδευση του μοντέλου για την αναγνώριση των αντικειμένων.

Συλλογή δεδομένων (Dataset)

Για την εφαρμογή του αλγορίθμου χρησιμοποιήθηκε μία συλλογή εικόνων που συγκεντρώθηκαν από την RGB κάμερα που τοποθετήθηκε στο όχημα στο περιβάλλον του CARLA simulator αλλά και από μία ήδη υπάρχουσα συλλογή δεδομένων [64]. Το σύνολο δεδομένων είναι δημόσια προσβάσιμο για τον αναγνώστη μεταβαίνοντας εδώ ή ακολουθώντας τον σύνδεσμο: <https://drive.google.com/drive/folders/1YeOAByuXxyu1U1rZykbhRvpYQ23bFff>.

Το dataset αποτελείται από 3624 εικόνες, οι οποίες συλλέχθηκαν καθώς το όχημα οδηγούσε σε λειτουργία αυτόματου πιλότου (autopilot mode), η οποία προσφέρεται από τον προσομοιωτή. Είναι σημαντικό να αναφερθεί ότι σε αυτή τη διαδικασία ήταν επιθυμητό να συλλεχθούν εικόνες από διαφορετικές πόλεις, χάρτες και σενάρια του CARLA, με διαφορετικές καιρικές συνθήκες, όπως επίσης, είναι απαραίτητο στο οδικό δίκτυο να υπάρχουν όλα τα δυναμικά εμπόδια που θα θέλαμε αναγνωριστούν στην τελική διαδικασία.

Οι κλάσεις που ορίστηκαν στην παρούσα διπλωματική είναι 7: «vehicle», «bus», «bike», «motorbike», «pedestrian», «traffic light» και «traffic sign». Λαμβάνοντας υπόψιν τις προϋποθέσεις για την οργάνωση που ακολουθεί ένα σύνολο δεδομένων για εκπαίδευση με το YOLO χρειάζεται ένα ακόμα στάδιο επεξεργασίας ώστε να κατασκευαστεί η απαραίτητη δομή. Μετά την εξαγωγή των εικόνων απαιτείται και η δημιουργία ενός αρχείου που περιέχει όλα τα αντικείμενα που βρίσκονται στην εικόνα, δηλαδή ένα αντίστοιχο αρχείο αναφοράς αντικειμένων (labels). Η μορφή αυτού του αρχείου ακολουθεί το πρότυπο YOLO και είναι τύπου txt, ενώ η ονομασία του θα πρέπει να αναφέρεται στο αντίστοιχο αρχείο εικόνας. Το περιεχόμενο του αρχείου είναι οργανωμένο σε γραμμές, όπου σε κάθε μία υπάρχει αρχικά ο αριθμός της κλάσης και έπειτα η συντεταγμένες του αντικειμένου. Ο αριθμός της κλάσης αναφέρεται στην δείκτη της θέσης της μέσα στον πίνακα κλάσεων που έχουμε ορίσει. Η διαδικασία αυτή έγινε με τη χρήση του ανοιχτού λογισμικού labellImg [65], το οποίο μας προσφέρει την δυνατότητα να θέσουμε χειροκίνητα ετικέτες σε κάθε φωτογραφία σε διάφορες μορφές. Στο Σχήμα 3.7 παρουσιάζονται κάποια στιγμιότυπα από το περιβάλλον του labellImg κατά τη διαδικασία της δημιουργίας ετικετών.



Σχήμα 3.7: Περιβάλλον labelImg

Το σύνολο των δεδομένων χωρίζεται σε σετ εκπαίδευσης, το οποίο περιέχει εικόνες και σετ ελέγχου, το οποίο περιέχει εικόνες. Η ανάλυση των εκπαιδευόμενων εικόνων είναι 1600×1200 .

Google Colab

Στην παρούσα εργασία, χρησιμοποιείται το διαδραστικό περιβάλλον προγραμματισμού (Interactive Python Programming Environment), Google Colaboratory (Google Colab). Το Google Colab, αποτελεί ένα προϊόν της Google Research που επιτρέπει στους προγραμματιστές να γράφουν και να εκτελούν κάθικες γλώσσας προγραμματισμού Python, μέσω του προγράμματος περιήγησής τους. Το περιβάλλον αυτό αποτελεί εξαιρετικό εργαλείο για εργασίες που απαιτούν βαθιά μάθηση. Ουσιαστικά, είναι μία εξειδικευμένη μορφή του Jupyter notebook, που δεν χρειάζεται κάποια εγκατάσταση. Το Google Colab, αποτελεί ένα κατά βάση ανοιχτό περιβάλλον προγραμματισμού, προσφέροντας υπολογιστικούς πόρους, όπως GPU και TPU, οι οποίες αποτελούν μονάδες επεξεργασίας γραφικών και μειώνουν σημαντικά τον χρόνο που απαιτείται για την ε-

κτέλεση ενός προγράμματος. Επιπροσθέτως, το Google Colab έχει ήδη εγκατεστημένες σχεδόν όλες τις απαραίτητες βιβλιοθήκες, οι οποίες είναι διαθέσιμες ανά πάσα στιγμή. Ένα ιδιαίτερο χαρακτηριστικό του προγράμματος είναι πως πραγματοποιεί αυτόματη αποθήκευση στο Cloud και έτσι δεν υπάρχει κίνδυνος να καταστραφεί ο αλγόριθμος που έχει ήδη υλοποιηθεί. Η παρούσα εργασία χρησιμοποιεί την έκδοση 3.7.12 της γλώσσας προγραμματισμού Python, καθώς επίσης υλοποιεί την μονάδα επεξεργασίας γραφικών GPU Tesla K80.

Προετοιμασία των αρχείων για εκπαίδευση

Όπως αναφέραμε, το μοντέλο που χρησιμοποιήθηκε στην παρούσα διπλωματική είναι το YOLOv4, η δομή του οποίου έχει αναλυθεί στο Κεφάλαιο 2. Αρχικά, θα πρέπει να επεξεργαστούμε το αντίστοιχο **αρχείο διαμόρφωσης (configuration file)**, το οποίο χρησιμοποιείται για τη διαμόρφωση των παραμέτρων και των αρχικών ρυθμίσεων του δικτύου ώστε να επεξεργαστούμε αυτές τις παραμέτρους. Πρώτο βήμα της διαδικασίας είναι να αλλάξουμε τη μεταβλητή **batch size** από 1 σε 64. Αυτή η μεταβλητή δηλώνει των αριθμό των δειγμάτων (εικόνων) που θα περνάνε από το δίκτυο σε κάθε επανάληψη, και αυτή η τιμή έχει βρεθεί να είναι η βέλτιστη τόσο για την μνήμη που απαιτείται κατά τη διαδικασία της εκπαίδευσης και την ταχύτητα της όσο και για την ακρίβεια. Επίσης, η μεταβλητή **max batches** διαμορφώνεται σύμφωνα με τον τύπο (classes x 2000), δηλαδή 14.000. Αυτό συμβαίνει επειδή έχει αποδειχτεί ότι 2000 επαναλήψεις για κάθε κλάση είναι κατα προσέγγιση οι επαναλήψεις που χρειάζονται να κάνει το δίκτυο κατά την εκπαίδευση. Ο **ρυθμός μάθησης (learning rate)** θα παραμείνει 0.001. Η μεταβλητή **steps** ορίζεται σύμφωνα με τον τύπο (80% του max batches, 90% του max batches), δηλαδή (11.200, 12.600). Επίσης, το **μέγεθος του δικτύου** ορίζεται να είναι 416x16 και η μεταβλητή **classes** αλλάζει σε 7, που είναι οι κλάσεις σε κάθε [yolo]-layer. Τέλος, αλλάζει και ο **αριθμός των φίλτρων filters=255** σύμφωνα με τον τύπο filters=(classes + 5)x3 σε κάθε τελευταίο convolutional-layer πριν από κάθε yolo-layer.

Πίνακας 3.1: Οι παράμετροι του μοντέλου

Training Network Configuration			
batch	64	learning rate	0.001
subdivisions	16	max batches	14000
width	416	steps	11200, 12600
height	416	classes	7
channels	3	flip	0
decay	0.0005	scales	0.1, 0.1
angle	0	burn in	1000

Έπειτα, δημιουργούμε ένα αρχείο **obj.names** στην τοποθεσία (directory) build/darknet/x64/data/, με τα ονόματα των κλάσεων το καθένα σε νέα σειρά.

Δημιουργούμε το αρχείο **obj.data** στην τοποθεσία (directory) build/darknet/x64/data/, το οποίο περιέχει τα παρακάτω στοιχεία:

Τον αριθμό των κλάσσεων: classes = 7

Την τοποθεσία για το train.txt αρχείο: train = data/train.txt

Την τοποθεσία για το test.txt αρχείο: valid = data/test.txt

Την τοποθεσία για το obj.names αρχείο: names = data/obj.names

Την τοποθεσία ενός backup folder ώστε να αποθηκεύονται εκεί όλα τα απαραίτητα αρχεία σε περίπτωση που τελειώσει ή διακοπεί απότομα η διαδικασία της εκπαίδευσης: backup = backup/

Ο φάκελος που περιέχει το dataset τοποθετείται στο directory build/darknet/x64/data/obj/ και περιέχει, όπως σχολιάστηκε ήδη, τις εικόνες (.jpg) και τα αντίστοιχα labels (.txt) για κάθε μια από τις εικόνες. Στη συνέχεια, δημιουργούμε το αρχείο **train.txt** στη τοποθεσία build/darknet/x64/data/, το οποίο περιέχει τα ονόματα των αρχείων εικόνων που έχουν μπει στο σύνολο ελέγχου, για παράδειγμα:

data/obj/img1.jpg

data/obj/img2.jpg

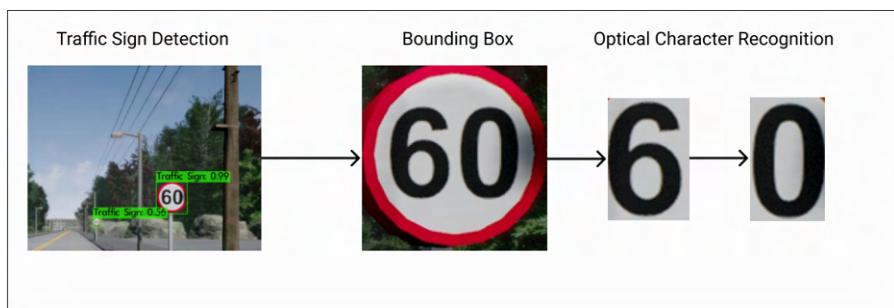
data/obj/img3.jpg

Την ίδια διαδικασία ακολουθούμε και για τα αρχεία που έχουμε επιλέξει να μπουν στο σύνολο ελέγχου, και για αυτά δημιουργούμε το αντίστοιχο αρχείο **test.txt**.

Τέλος, μπορούμε να αλλάξουμε κάποιες παραμέτρους στο darknet, όπως για παράδειγμα κάναμε στη περίπτωση αυτής της διπλωματικής όπου αποφασίσαμε να αποθηκεύουμε τα βάρη του δικτύου (.weights file) ανά **100 επαναλήψεις** αντί για 1000 που ήταν το προκαθορισμένο από το δίκτυο.

3.2.5 Αναγνώριση των σημάτων ταχύτητας με Tesseract OCR

Όπως ήδη αναφέρθηκε στην θεωρητική ανάλυση, η χρήση της μηχανής Tesseract OCR προσφέρει τη δυνατότητα αναγνώρισης κειμένου από ψηφιακή εικόνα. Ο σκοπός της στη παρούσα εργασία είναι να αναγνωρίσει το περιέχομενο που αναγράφεται στα σήματα που συναντά το όχημα στο περιβάλλον του προσομοιωτή CARLA. Συγκεκριμένα, υπάρχουν μόνο σήματα για το όριο της ταχύτητας κυκλοφορίας των οχημάτων (30, 60 και 90 χιλιόμετρα/ώρα).



Σχήμα 3.8: Διαδικασία αναγνώρισης σημάτων ταχύτητας

Δεδομένου ότι η προσφέρει διαφορετικές λειτουργίες σχετικές με τη διάρθωση του κειμένου στη εικόνα εισόδου (Page Segmentation Modes - PSMs) ανάλογα με τον τύπο της αναγνώρισης στην οποία αποσκοπεί ο χρήστης, στην διπλωματική χρησιμοποίησαμε τον τύπο «9 - Treat the image as a single word in a circle». Αυτό που αναμένει από την

εικόνα εισόδου, είναι να υπάρχει μία λέξη στο πλαίσιο ενός κυκλικού σχήματος, καθώς τα σήματα στο προσομοιωτή CARLA μοιάζουν όπως φαίνεται στην Εικόνα 3.8.

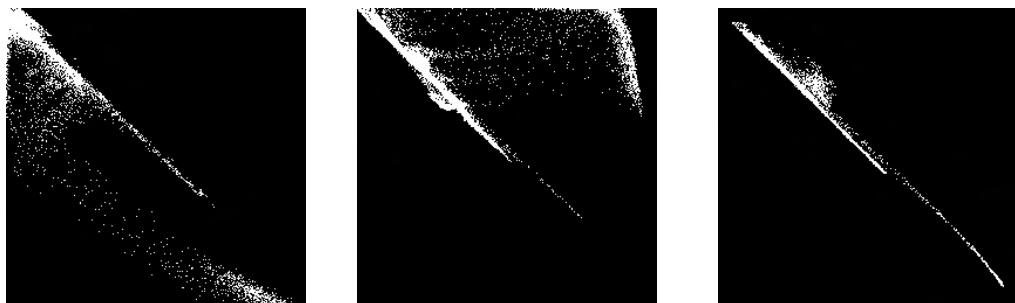
3.2.6 Ανάλυση χρωματικού ιστογράμματος

Για την αναγνώριση του χρώματος των φωτεινών σηματοδοτών, χρησιμοποιείται η πληροφορία ιστογράμματος της εικόνας. Το ιστόγραμμα της εικόνας είναι μονοδιάστατο για μονόχρωμες εικόνες και πολυδιάστατο για έγχρωμες. Η ανίχνευση κορυφών ή κοιλάδων στο ιστόγραμμα, μπορούν να οδηγήσουν στην ανίχνευση αντικειμένων και στον διαχωρισμό του όντου αντίστοιχα.

Στις πολύχρωμες εικόνες βέβαια, όπως στη περίπτωσή μας, η διαδικασία είναι πιο περίπλοκη. Τα τρία ιστογράμματα (RGB histograms) μπορούν να επεξεργαστούν χωριστά και να συνδυαστούν τα αποτελέσματα. Υπάρχει ένα ιστόγραμμα ανά κανάλι R, G και B αλλά και ένα διάγραμμα ιστογράμματος το οποίο παράγεται από τον μέσο όρο των τριών καναλιών $(R+G+B)/3$.



Σχήμα 3.9: Μάσκες για εντοπισμό του πράσινου και του κόκκινου από ιστόγραμμα



Σχήμα 3.10: Ιστόγραμμα για πράσινο, κόκκινο και πορτοκαλί χρώμα αντίστοιχα

Αντί να βλέπουμε την κατανομή της φωτεινότητας συνδυαστικά για την εικόνα, την βλέπουμε για κάθε χρωματικό κανάλι ξεχωριστά. Η διαδικασία προέκυψε από τη παρατήρηση του συνδυασμού G και R σε ένα 2D ιστόγραμμα. Όπως ήδη αναφέρθηκε με την βοήθεια των 2D ιστογραμμάτων μπορούμε να αναλύσουμε σε ζεύγη τα κανάλια (RG, GB, BR) ή όλα μαζί σε ένα 3D ιστόγραμμα. Στη συγκεκριμένη περίπτωση, αυτό που έχει ενδιαφέρουσα πληροφορία είναι το GR ιστόγραμμα, όπως φαίνεται στο Σχήμα 3.10. Ακολουθεί η προσαρμογή/κανονικοποίηση των τιμών στη κλίμακα $(0,1)$ και η δημιουργία δύο μασκών (Σχήμα 3.9), ώστε να μετρηθούν τα αντίστοιχα σημεία της εικόνας που έχουν υψηλές τιμές στο κόκκινο και στο πράσινο αντίστοιχα, ενώ ορίζονται

ένα κατώφλι μπορούμε τελικά να εξάγουμε συμπέρασμα για το χρώμα που εμφανίζεται στην εικόνα.

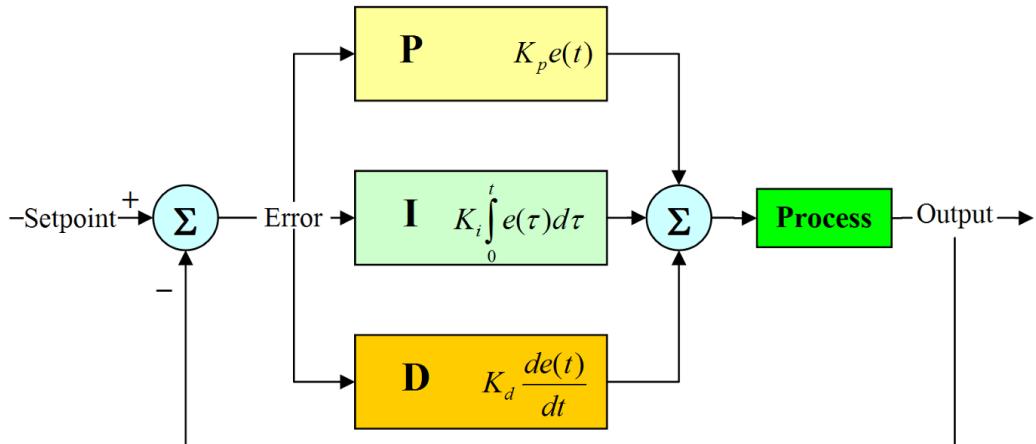
Όπως φαίνεται στο Σχήμα 3.10, όταν στην εικόνα υπάρχει το κόκκινο χρώμα παρουσιάζονται μη μηδενικά στοιχεία στο πάνω και δεξί μέρος της ιστογράμματος, για πράσινο στο κάτω αριστερό μέρος του ιστογράμματος και πορτοκαλί στη διαγώνιο.

3.3 Σύστημα πλοϊγησης (Navigation System)

3.3.1 Πλοϊγηση με χρήση PID ελεγκτή

Στην πάροδο των χρόνων και με τον τομέα της αυτόνομης οδήγησης να αναπτύσσεται συνεχώς, έχουν προταθεί πολλοί τρόποι ελέγχου του οχήματος τόσο με παραδοσιακές όσο και με πιο σύγχρονες μεθόδους, που περιλαμβάνουν τεχνικές μηχανικής μάθησης και νευρωνικά δίκτυα. Στη παρούσα διπλωματική θα εξετάσουμε κάποιες από τις πιο κλασικές μεθόδους.

Μία αποτελεσματική, ευρέως διαδεδομένη αλλά και με μικρές απαιτήσεις πόρων από το σύστημα είναι αυτή της χρήσης PID ελεγκτών (PID Controllers), η οποία θα ενσωματωθεί στον προσωμοιωτή CARLA για να τον έλεγχο του οχήματος στο οδικό περιβάλλον σε μια προκαθορισμένη διαδρομή.



Σχήμα 3.11: Block διάγραμμα PID ελεγκτή [66]

Οι ελεγκτές δεν είναι τίποτα άλλο από υποσυστήματα, κύρια δραστηριότητα των οποίων είναι να τροποποιούν το σφάλμα μεταξύ της επιθυμητής (setpoint) και της πραγματικής εξόδου έτσι ώστε να γίνει περισσότερο αντιληπτό από το κεντρικό σύστημα (plant) και να μηδενίσει (ή να ελαχιστοποιήσει) το τελικό σφάλμα (error signal) γρηγορότερα για να επιτύχει την επιθυμητή τιμή για κάποια ελεγχόμενη μεταβλητή (control variable). Τα αρχικά PID αντιπροσωπεύουν τρεις λέξεις: Proportional (αναλογικός), Integral (ολοκληρωτικός) και Derivative (διαφορικός). Με άλλα λόγια το σφάλμα εισόδου πολλαπλασιάζεται με μία σταθερά (Proportional), ολοκληρώνεται (Integral) και παραγωγίζεται (Derivative). Στη συνέχεια, οι τρεις αυτές ποσότητες αθροίζονται και αποτελούν την πραγματική είσοδο του κυρίως συστήματος [67], όπως μπορούμε να παρατηρήσουμε στο διάγραμμα του ελεγκτή στο Σχήμα 3.11.

Οι όροι K_P , K_I και K_D είναι αντίστοιχα:

- K_P = Αναλογικό κέρδος
- K_I = Ολοκληρωτικό κέρδος
- K_D = Διαφορικό κέρδος

Για παράδειγμα, για την παραμονή ενός οχήματος ανάμεσα στις λωρίδες του δρόμου, το κέντρο των λωρίδων του δρόμου θα ήταν το setpoint, δηλαδή η επιθυμητή τιμή, το σφάλμα παρακολούθησης θα ήταν η απόκλιση από την επιθυμητή θέση, η είσοδος θα ήταν η τιμή που πρέπει να δώσουμε στο σύστημα ώστε να επιστρέψει το όχημα στη μέση των λωρίδων, δηλαδή να μειώσει το σφάλμα, δεδομένου του setpoint.

Για να γίνει αντιληπτή η σημασία των όρων που αναφέρθηκαν στη συνάρτηση μεταφοράς του ελεγκτή και των ιδιοτήτων που προσφέρουν θα φέρουμε ένα παράδειγμα που αφορά άμεσα τα αυτόνομα οχήματα, αυτό του ελέγχου της ταχύτητας. Σε αυτή τη περίπτωση η ελεγχόμενη μεταβλητή είναι η ταχύτητα και το setpoint είναι η επιθυμητή ταχύτητα, ενώ το σύστημα είναι η φυσική του οχήματος και η μηχανή.

Αναλογικός όρος

Η διαφορά μεταξύ της επιθυμητής ταχύτητας και της ταχύτητας του οχήματος καλείται σφάλμα. Όταν η ταχύτητα είναι μικρότερη από την επιθυμητή το σφάλμα είναι θετικό, ενώ αν είναι μεγαλύτερη είναι αρνητικό.

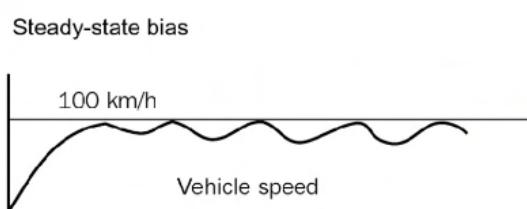
$$\text{error}_{\text{speed}} = \text{setpoint}_{\text{speed}} - \text{current}_{\text{speed}} \quad (3.1)$$

Ο αναλογικός όρος είναι ο πολλαπλασιασμός αυτής της διαφοράς με ένα αναλογικό κέρδος (K_P).

$$K_p * \text{error}_{\text{speed}} \quad (3.2)$$

Οπότε η τιμή που θα έπρεπε να δωθεί στο γκάζι του οχήματος θα ήταν:

$$\text{Throttle} = K_p * \text{error}_{\text{speed}} \quad (3.3)$$



Σχήμα 3.12: Steady state bias αναλογικού όρου [67]

Σε αυτή τη περίπτωση όσο το σφάλμα ταχύτητας ελαττώνεται, τόσο και η τιμή για το γκάζι ελαττώνεται μέχρι να επιτευχθεί μηδενικό σφάλμα και να μηδενιστεί και η τιμή για το γκάζι. Αυτό ωστόσο σημαίνει ότι ποτέ δεν θα μπορούσε το σύστημα να μείνει σταθερά στην επιθυμητή ταχύτητα, καθώς κάθε φορά που θα την φτάνει, η τιμή του γκαζιού θα μηδενίζεται και απευθείας η ταχύτητα θα πέφτει και πάλι κάτω από

την επιθυμητή, όπως φαίνεται στο Σχήμα 3.12. Επομένως, ο αναλογικό έλεγχος έχει ως αποτέλεσμα την ελλάττωση του χρόνου ανύψωσης, κάνει δηλαδή το σύστημα πιο γρήγορο, ενώ ρυθμίζει το μόνιμο σφάλμα και τείνει να το μηδενίσει, αλλά δεν το κάνει ποτέ.

Ολοκληρωτικός όρος

Ο ολοκληρωτικός όρος δίνει την λύση σε αυτό το πρόβλημα ενσωματώνοντας όλα τα προηγούμενα σφάλματα στο σύστημα:

$$\text{total_error}_{\text{speed}} = \text{total_error}_{\text{speed}} + \text{current_error}_{\text{speed}} \quad (3.4)$$

Το συνολικό σφάλμα υπολογίζεται και πολλαπλασιάζεται και πάλι με έναν συντελεστή, K_I , που ονομάζεται ολοκληρωτικό κέρδος.

$$K_I * \text{total_error}_{\text{speed}} \quad (3.5)$$

$$\text{Throttle} = K_P * \text{error}_{\text{speed}} + K_I * \text{total_error}_{\text{speed}} \quad (3.6)$$

Όπως εξηγήθηκε, αν δεν υπήρχε αυτός ο όρος η ταχύτητα θα ήταν πάντα μικρότερη από την επιθυμητή οπότε το συνολικό σφάλμα θα ήταν θετικό. Όσο περνάει ο χρόνος τόσο μεγαλύτερος γίνεται ο όρος με το συνολικό σφάλμα. Επιλέγοντας το κατάλληλο K_I η τιμή του γκαζιού θα παραμένει μεγαλύτερη από το 0 ακόμα κι αν μηδενιστεί ο όρος του σφάλματος ταχύτητας. Συνεπώς, ο ολοκληρωτικός έλεγχος αυξάνει τον τύπο του συστήματος και μηδενίζει το μόνιμο σφάλμα, αλλά μεταβάλλει προς το χειρότερο τη μεταβατική συμπεριφορά καθώς προκαλεί εμφάνιση ταλαντώσεων γύρω από την επιθυμητή τιμή.

Διαφορικός όρος

Το τελευταίο πρόβλημα διορθώνεται με τον τελευταίο όρο. Ο διαφορικός όρος είναι η λύση σε αυτό καθώς προσαρμόζει την τιμή του γκαζιού ανάλογα με το πόσο γρήγορα ο όρος πλησιάζει στο setpoint. Ο ρυθμός της αλλαγής του σφάλματος ορίζεται:

$$K_D * (\text{current_error}_{\text{speed}} - \text{previous_error}_{\text{speed}}) / (\text{time}_{\text{current}} - \text{time}_{\text{previous}}) \quad (3.7)$$

Οπότε καταληγούμε:

$$K_D * \text{error}_{\text{speed}} * d/dt \quad (3.8)$$

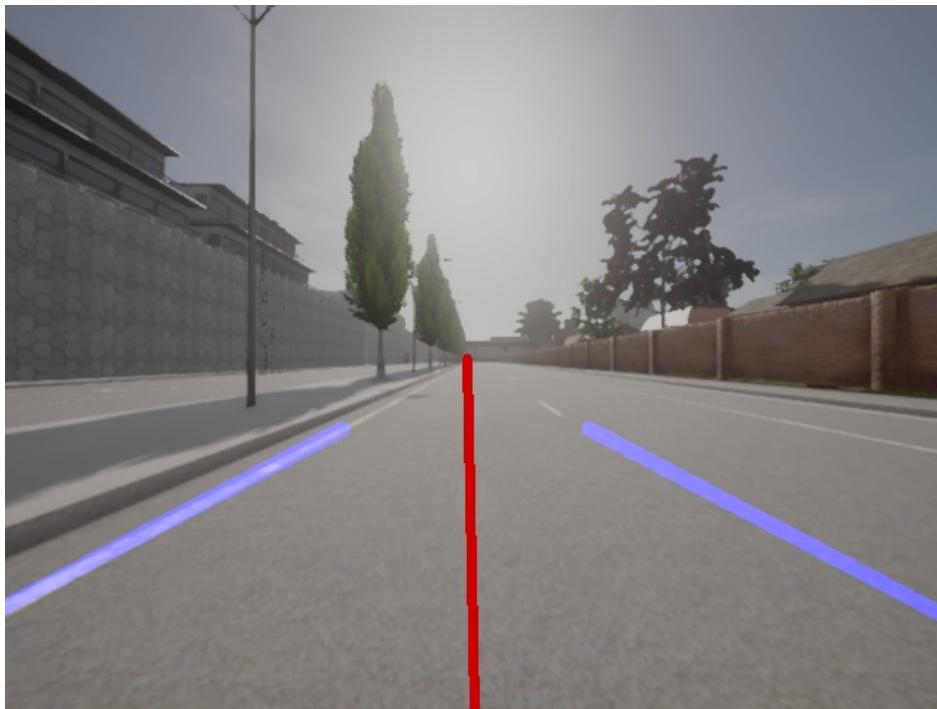
Όσο το σφάλμα μειώνεται, δηλαδή η τιμή προσεγγίζει την επιθυμητή, ο όρος θα είναι αρνητικός και θα αναγκάζει την τιμή να μειωθεί εφόσον τώρα υπολογίζουμε το άθροισμα όλων των όρων που εξηγήθηκαν.

$$\text{Throttle} = K_P * \text{error}_{\text{speed}} + K_I * \text{total_error}_{\text{speed}} + K_D * \text{error}_{\text{speed}} * d/dt \quad (3.9)$$

Για να πραγματοποιηθούν τα παραπάνω, η σωστή επιλογή των κερδών είναι ο βασικός παράγοντας. Η διαδικασία που παρουσιάστηκε ακολουθήθηκε για τον έλεγχο ταχύτητας, πέδησης και κατεύθυνσης του οχήματος, ενσωματώνοντας στο σύστημα τον VehiclePIDController του προσομοιωτή για τους δύο ελέγχους, επιλέγοντας τις τιμές των κερδών και δίνοντας την επιθυμητή ταχύτητα. Η επιθυμητή διαδρομή μπορεί να οριστεί είτε ακολουθώντας το μονοπάτι που μπορεί να δημιουργήσει ο προσομοιωτής είτε ακολουθώντας τα ωμοποιητικά που δημιουργήσαμε με τον αλγόριθμο A*.

3.3.2 Πλοιήγηση με διατήρηση της λωρίδας με χρήση OpenCV

Ένα σύστημα πλοιήγησης που αποσκοπεί το όχημα να διατηρεί τη λωρίδα του (Lane Keeping Assist System) περιλαμβάνει δύο επιμέρους διαδικασίες: την αντίληψη και αναγνώριση των λωρίδων στο οδόστρωμα (Lane Detection) και του καθορισμού της κίνησης (Motion Planning). Όπως έχουμε ήδη αναφέρει στο υποσύστημα αντίληψης, η αναγνώριση των λωρίδων στο οδόστρωμα έχει υλοποιηθεί με χρήση της βιβλιοθήκης OpenCV και έχει την δυνατότητα από τις εικόνες που λαμβάνει από την εγκατεστημένη κάμερα να εξάγει όλα τα απαραίτητα χαρακτηριστικά, ώστε να καταλήξουμε στις συντεταγμένες των ευθειών που είναι οι λωρίδες του οδοστρώματος. Η δεύτερη διαδικασία αφορά τον καθορισμό της κίνησης. Εφόσον το σύστημα πλέον διαθέτει όλες τις πληροφορίες για την θέση των λωρίδων, το επόμενο βήμα είναι να καθορίσουμε την διεύθυνση (steer) με σκοπό το όχημα να παραμείνει μέσα στα όρια των λωρίδων και ιδανικά να πλοηγείται στη μέση αυτών. Επομένως, δεδομένων των πληροφοριών για τις λωρίδες καλούμαστε να υπολογίσουμε την γωνία διεύθυνσης (steering angle) του οχήματος.



Σχήμα 3.13: Γραμμή κατεύθυνσης στο περιβάλλον του προσομοιωτή CARLA

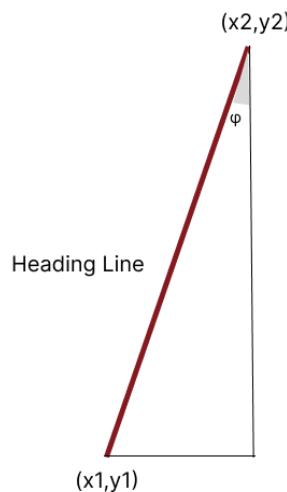
Γραμμή κατεύθυνσης (Heading Line)

Πρώτο βήμα της διαδικασίας είναι να ορίσουμε τη γραμμή κατεύθυνσης (Heading Line). Εφόσον το όχημα πρέπει να κινείται στη μέση των λωρίδων, ο πιο εύκολος τρόπος να γίνει αυτό είναι αρχικά να υπολογίσουμε το πιο απομακρυσμένο σημείο (endpoint) αυτής της γραμμής που θα βρίσκεται στη μέση των λωρίδων που έχουν εντοπιστεί. Στο Σχήμα 3.13 φαίνεται η γραμμή με το σημείο αυτό να είναι το μέσο των πιο απομακρυσμένων σημείων των δύο λωρίδων. Αντίστοιχα, εφόσον γνωρίζουμε την οπτική της εγκατεστημένης κάμερας στη μέση του οπτικού πεδίου του οχήματος, το αρχικό σημείο είναι στο χαμηλότερο σημείο και πάλι στη μέση της εικόνας.

Γωνία διεύθυνσης (Steering Angle)

Η τελευταία διαδικασία που απομένει είναι να μετατρέψουμε αυτή τη πληροφορία σε γωνία διεύθυνσης. Όπως θα δούμε και στο Σχήμα 3.14 που ακολουθεί, θα πρέπει πρώτα να υπολογιστούν οι αποκλίσεις σε κάθε άξονα. Εφόσον θεωρούμε ότι η κάμερα βρίσκεται εγκατεστημένη στη μέση του οχήματος, θα υπολογίσουμε την απόκλιση του μέσου των λωρίδων από το μέσο του πλάτους της εικόνας κι αυτό θα είναι το x_{offset} , ενώ το y_{offset} ορίζεται $height/2$. Όπως μπορούμε να παρατηρήσουμε και στην εικόνα, η γωνία που θέλουμε να υπολογίσουμε είναι η ϕ .

$$\phi = \tan^{-1}\left(\frac{\text{offset}_x}{\text{offset}_y}\right) \quad (3.10)$$



Σχήμα 3.14: Υπολογισμός της γωνίας διεύθυνσης

Επομένως, μπορούμε εύκολα να υπολογίσουμε την εφαπτομένη της γωνίας, όπως φαίνεται στο Σχήμα 3.14, και να καταλήξουμε στην τιμή της η οποία βρίσκεται σε rad. Μπορούμε να την μετατρέψουμε σε μοίρες (degrees) και έπειτα σε τιμή κατάλληλη για το όχημα στον προσομοιωτή CARLA. Η τιμή του steer στον προσομοιωτή μπορεί να πάρει τιμές από -1 έως 1, με τις αρνητικές τιμές να κάνουν το όχημα να κινηθεί προς τα αριστερά και με τις θετικές προς τα δεξιά. Οι τιμές αυτές αντιστοιχούν σε -70 και 70 μοίρες αντίστοιχα, αλλά σε διαφορετικά οχήματα αυτές οι τιμές ίσως αλλάζουν. Με βάση αυτά, η τιμή steer υπολογίζεται ως εξής:

$$\text{steer} = \phi \cdot \frac{180}{70\pi} \quad (3.11)$$

Ανάλογα με τη κλίση της ευθείας θα δωθεί και το κατάλληλο πρόσημο και η τιμή θα ελεγχθεί να βρίσκεται ανάμεσα στο διάστημα των τιμών που πρέπει να πάρει. Τέλος, για να αποφευχθούν απότομες κινήσεις θα ορίσουμε μια μέγιστη τιμή που μπορεί να πάρει κάθε φορά το steer, ώστε να βρεθούμε προοδευτικά στο επιθυμητό σημείο χωρίς να αποσταθεροποιείται το όχημα. Μετά από αυτή τη διαδικασία, η τιμή του steer μπορεί να τροφοδοτηθεί στον έλεγχο.

3.4 Κατασκευή της βασικής διαδρομής

3.4.1 Εισαγωγή

Η κύρια χρήση ενός συστήματος αυτόνομης οδήγησης είναι η μεταφορά από ένα αρχικό σημείο σε έναν τελικό προορισμό. Αυτή η διαδικασία προϋποθέτει την σχεδίαση ενός βασικού μονοπατιού ή βασικής διαδρομής (route path), από το αρχικό στο τελικό σημείο. Η διαδρομή αυτή θα πρέπει να είναι η βέλτιστη ως προς μερικά κριτήρια, τα οποία αφορούν κυρίως το μήκος μονοπατιού και το χρόνο ολοκλήρωσης της διαδρομής. Επιπλέον, το μονοπάτι που θα κατασκευαστεί θα πρέπει να τηρεί τον κώδικα οδικής κυκλοφορίας και να μη βγαίνει εκτός των ορίων του αυτοκινητόδρομου, ενώ ταυτόχρονα να οδηγεί ως προς τη σωστή φορά σύμφωνα με τις λωρίδες κυκλοφορίας, το οποίο πραγματοποιείται λαμβάνοντας αυτές τις πληροφορίες για τον δρόμο από το API του CARLA.

Στην ενότητα αυτή θα αναλυθεί ο τρόπος και τα κριτήρια σχεδιασμού του βασικού μονοπατιού, με το κριτήριο για την σχεδίαση να έχει επιλεχθεί να είναι το μήκος του μονοπατιού. Για την εύρεση και την σχεδίαση του μονοπατιού έχει χρησιμοποιηθεί ο αλγόριθμος A*. Ο A* είναι ένας από τους πιο γνωστούς αλγορίθμους στο τομέα της ρομποτικής για την εύρεση μονοπατιών, στα πλεονεκτήματα του περιλαμβάνεται ο χρόνος εκτέλεσης και επίσης το ότι η λύση είναι κοντά στην βέλτιστη, ενώ το πιο βασικό μειονέκτημα είναι ότι δεν βρίσκει την βέλτιστη διαδρομή. Στην περίπτωση μας η επιλογή του A* είναι πολύ καλή, η χρήση κάποιου άλλου αλγορίθμου που μπορεί να δώσει την βέλτιστη λύση όπως για παράδειγμα ο Dijkstra αλγόριθμος θα έκανε το χρόνο εκτέλεσης απαγορευτικά μεγάλο λόγω της πολύπλοκης δομής του δυναμικού χώρου και των κριτηρίων που πρέπει να ληφθούν υπόψη για την σχεδίαση.

3.4.2 Περιγραφή υλοποίησης

Ο A* είναι από τους πιο γνωστούς και ευρέως χρησιμοποιημένους αλγόριθμους της Τεχνητής Νοημοσύνης που χρησιμοποιείται για την αναζήτηση μονοπατιού και για την διάσχιση γράφου. Ανήκει στην κατηγορία των αλγορίθμων πληροφορημένης ή ευριστικής αναζήτησης. Οι αλγόριθμοι αυτοί χρησιμοποιούν ευριστική συνάρτηση με τη βοήθεια της οποίας αξιολογούν τις επόμενες πιθανές καταστάσεις και επιλέγουν την καλύτερη από αυτές. Σκοπός του αλγορίθμου είναι η εύρεση της βέλτιστης λύσης δηλαδή το συντομότερο μονοπάτι με το λιγότερο κόστος μεταξύ 2 σημείων που ονομάζονται κόμβοι. Ο αλγόριθμος βρίσκει ευρεία χρήση λόγω της ακρίβειας και των επιδόσεων του. Αρχικά, παρουσιάστηκε το 1968 από τον Peter Hart, τον Nils Nilsson και τον Bertram Rapheal του Ερευνητικού Ινστιτούτου Stanford. Είναι αποτέλεσμα συνδυασμού της άπληστης αναζήτησης με την αναζήτηση με βάση το κόστος. Πραγματοποιεί μια ταξινόμηση των καταστάσεων (κόμβων) με βάση την συνάρτηση $f(k) = g(k) + h(k)$. Η συνάρτηση g αντιπροσωπεύει το κόστος για να φθάσουμε από το αρχική κατάσταση στην τρέχουσα κατάσταση, ενώ η συνάρτηση h είναι μια ευριστική συνάρτηση που εκτιμά την απόσταση από την παρούσα κατάσταση στην κατάσταση στόχου. Στις εικόνες που ακολουθούν παρουσιάζονται τρία παραδείγματα σχεδίασης διαδρομής με A* στις πόλεις 1,2 και 3 του προσομοιωτή CARLA. Τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα παρουσιάζονται παρακάτω στον Αλγόριθμο 1.



Σχήμα 3.15: Εφαρμογή A^* στην Πόλη 1



Σχήμα 3.16: Εφαρμογή A^* στην Πόλη 2



Σχήμα 3.17: Εφαρμογή A^* στην Πόλη 3

Algorithm 1 Algorithm A*

```

1: Result: route
2: Node  $\leftarrow$  Class: Object with parameters: parent, position, g , h, f
   return route
3: # parent: Parent node of the current node
4: # position: Current node
5: # g: Distance value from starting position
6: # h: Distance value from ending position
7: # f: g + h
8: start: starting position
9: goal: ending position
10: open_list  $\leftarrow$  empty list
11: closed_list  $\leftarrow$  empty list
12: open_list  $\leftarrow$  append start
13: while open_list is not empty do
14:   current  $\leftarrow$  node with the lowest f score in the open list
15:   if this node is our destination node then
16:     Path Found
17:   else
18:     Put the current node in the closed list and look at all of its neighbors
19:     for each neighbor of the current node do
20:       if neighbor has < g value than current AND is in the closed list then
21:         replace the neighbor with the new lower g value
22:         current node is the neighbor's parent
23:       else if current g value is lower this neighbor is in open list then
24:         replace the neighbor with the new lower g
25:         change the neighbor's parent to our current node
26:       else if this neighbor is not in both lists then
27:         add it to the open list and set its g
28:       end if
29:     end for
30:   end if
31: end while
   return route

```

Κεφάλαιο 4

Πειράματα - Αποτελέσματα

4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα πραγματοποιηθεί η αξιολόγηση του συστήματος αυτόνομης οδήγησης οχήματος που αναπτύχθηκε κατά την εκπόνηση της διπλωματικής εργασίας. Αρχικά θα παρουσιαστούν οι παραδοχές που έγιναν για το περιβάλλον στο οποίο εξετάστηκε το σύστημα, καθώς και οι μετρικές σύμφωνα με τις οποίες θα γίνει η εκτίμηση για την ποιότητα και την απόδοση του συστήματος. Τα κριτήρια αξιολόγησης στηρίζονται κυρίως στην ακρίβεια εντοπισμού των λωρίδων και των εμποδίων, της πληροφορίας των σημάτων οδικής κυκλοφορίας και του χρώματος των φαναριών, όπως επίσης και τη μέτρηση της απόδοσης του συστήματος ως προς την ικανότητα του οχήματος να πλοηγείται και να ολοκληρώνει τη διαδρομή του. Τα πειράματα εκτελούνται σε μία από τις πόλεις του προσωμοιωτή CARLA. Τέλος, θα παρουσιαστούν τα αποτελέσματα των πειραμάτων και θα εξαχθούν τα τελικά συμπεράσματα.

4.2 Παραδοχές

Το όχημα κινείται σε περιβάλλον οδικού δικτύου, οι δρόμοι περιέχουν λωρίδες κυκλοφορίας και μπορεί να είναι μονής ή διπλής κατεύθυνσης. Οι δρόμοι περιέχουν σήμανση για το όριο ταχύτητας και φωτεινούς σηματοδότες. Τα δυναμικά εμπόδια του περιβάλλοντος είναι οχήματα ή πεζοί. Στην λίστα των οχημάτων περιλαμβάνονται επιβατηγά οχήματα, φορτηγά, μοτοσυκλέτες και ποδήλατα. Στη κατηγορία των αντικειμένων που θα πρέπει να αναγνωρίζει το όχημα συγκαταλέγονται όλα τα παραπάνω, δηλαδή όλα τα οχημάτα, οι πεζοί, οι σημάνσεις των ορίων ταχύτητας καθώς και οι φωτεινοί σηματοδότες. Τα οχήματα κινούνται σύμφωνα με τους κανόνες οδικής κυκλοφορίας αλλά μπορεί να τους παραβιάσουν. Το αυτόνομο όχημα θεωρείται ότι λαμβάνει δεδομένα που σχετίζονται με πληροφορίες για το δρόμο από ψηφιακό προκατασκευασμένο χάρτη. Τα στατικά εμπόδια που βρίσκονται εκτός του δρόμου δεν λαμβάνονται υπόψιν.

Για τα πειράματα χρησιμοποιήθηκε υπολογιστής με επεξεργαστή Intel Core i7-6500U και συχνότητα επεξεργαστή 2.50GHz, μνήμη RAM 8 GB και κάρτα γραφικών AMD Radeon R5 M335 4GB. Η εκπαίδευση του δικτύου YOLO έγινε στο Google Colab που χρησιμοποιεί την μονάδα επεξεργασίας γραφικών GPU Tesla K80.

4.3 Οι μετρικές αξιολόγησης του συστήματος

Πριν προχωρήσουμε στην παρουσίαση των πειραμάτων, οφείλουμε να μιλήσουμε εν συντομίᾳ για τον τρόπο αξιολόγησης των μοντέλων. Σύμφωνα και με το [68], ορίζουμε τα tp, fn, fp, tn για ένα δυαδικό πρόβλημα ταξινόμησης (η λογική επεκτείνεται και σε προβλήματα πολλαπλών κλάσεων, όπως το δικό μας), όπως φαίνεται στον πίνακα 4.1. Μπορούμε τώρα να ορίσουμε (πίνακας 4.2) σε ένα συνοπτικό πίνακα τις συνήθεις μετρικές αξιολόγησης.

Πίνακας 4.1: Ο ορισμός των TP, TN, FP, FN

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True positive (tp)	False negative (fn)
Predicted Negative Class	False positive (fp)	True negative (tn)

Πίνακας 4.2: Ορισμός των μετρικών

Μετρική	Τύπος	Χρησιμότητα
Accuracy (acc)	$\frac{tp+tn}{tp+fp+tn+fn}$	Μετράει την αναλογία σωστών προβλέψεων ως προς τις συνολικές προβλέψεις
Precision (p)	$\frac{tp}{tp+fp}$	Μετράει τις σωστές θετικές προβλέψεις ως προς το σύνολο όλων των στοιχείων που έπρεπε να ταξινομηθούν στη θετική κλάση
Recall (r)	$\frac{tp}{tp+fn}$	Μετράει το κλάσμα των σωστών θετικών προβλέψεων που ταξινομήθηκαν σωστά
F-Measure (FM)	$\frac{2*p*r}{p+r}$	Αντιπροσωπεύει τον αρμονικό μέσο ανάμεσα στις τιμές του recall και precision

Η μετρική **IoU (Intersection over Union)** περιγράφει πως αλληλεπικαλύπτονται το πραγματικό πλαίσιο σε σχέση με το πλαίσιο που προβλέφθηκε και αποτελεί μια μετρική απόδοσης που χρησιμοποιείται στην ανίχνευση αντικειμένων. Το IoU θα είναι το ο λόγος του «εμβαδόν» της περιοχής που είναι κοινή και για τα 2 περιμετρικά πλαίσια προς το «εμβαδόν» όλης της περιοχής που τα δύο πλαίσια καλύπτουν. Το IoU είναι ίσο με 1 εάν το περιμετρικό πλαίσιο που βρέθηκε είναι το ίδιο με το πραγματικό.

Ο δείκτης ο οποίος χρησιμοποιείται προκειμένου να αξιολογήσουμε την απόδοση του μοντέλου YOLO ονομάζεται **μέσος μέσης ακρίβειας (mAP)**. Η μέση ακρίβεια (AP) είναι μία δημοφιλής μετρική για την μέτρηση της ακρίβειας ανιχνευτών αντικειμένων, όπως το YOLO. Η μέση ακρίβεια υπολογίζει την μέση τιμή ακρίβειας για την τιμή της ανάκλησης (Recall) από 0 έως 1.

Η μετρική που θα χρησιμοποιηθεί για την **αξιολόγηση ολοκλήρωσης των δια-**

δρομών είναι το ποσοστό ολοκλήρωσης κατά μέσο όρο για N διαδρομές:

$$\frac{1}{N} \sum_{n=i}^N p_i$$

Για την εύρεση του **σφάλματος της μεθόδου αναγνώρισης λωρίδων** εντοπίστηκαν οι πραγματικές ακμές που αναπαριστούν τις λωρίδες και οι ακμές από την εφαρμογή της μεθόδου και βρέθηκε η μεταξύ τους απόκλιση, υπολογίζοντας την γωνία μεταξύ των ευθειών. Για να βρούμε την οξεία γωνία ϕ που σχηματίζουν δύο ευθείες e_1 και e_2 , αρχικά θεωρούμε διανύσματα $\vec{\delta}_1 \parallel e_1$ και $\vec{\delta}_2 \parallel e_2$ και βρίσκουμε τη γωνία $\omega = (\widehat{\vec{\delta}_1}, \widehat{\vec{\delta}_2})$ χρησιμοποιώντας τη σχέση:

$$\text{συν}(\widehat{\vec{\delta}_1}, \widehat{\vec{\delta}_2}) = \frac{\vec{\delta}_1 \cdot \vec{\delta}_2}{|\vec{\delta}_1| |\vec{\delta}_2|}$$

Αν $\text{συν}(\widehat{\vec{\delta}_1}, \widehat{\vec{\delta}_2}) > 0$, τότε $\omega < 90^\circ$ και η ζητούμενη γωνία είναι η $\phi = \omega$.

Επομένως, μπορούμε να βρούμε το σφάλμα μεταξύ των ευθειών κατά μέσο όρο:

$$\frac{1}{N} \sum_{n=i}^N \omega_i$$

Επιπλέον, η τυπική απόκλιση που εκφράζει το ποσό της μεταβολής ή της διασποράς των τιμών μπορεί να βρεθεί από τον τύπο:

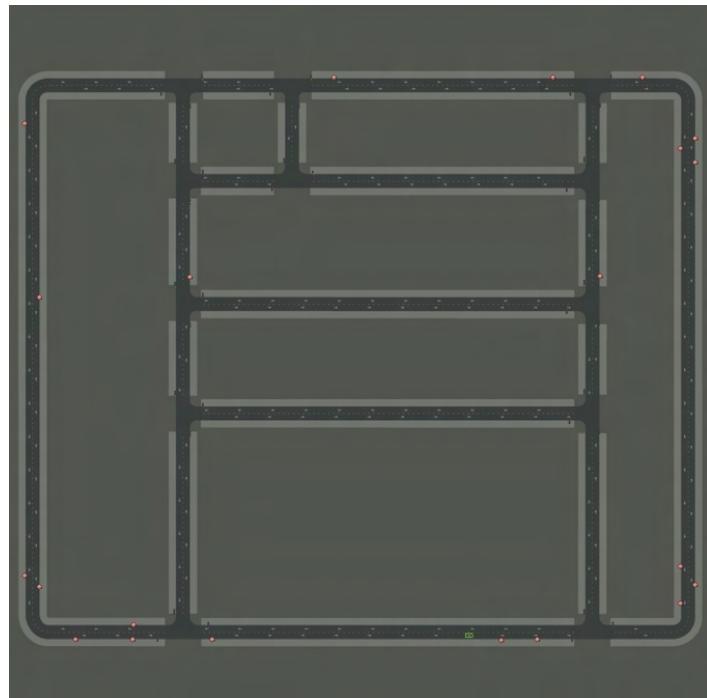
$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

4.4 Πειράματα και αξιολόγηση του συστήματος

4.4.1 Παρουσιάση των χαρτών και του περιβάλλοντος του CARLA

Town01

Η πόλη 1 αποτελεί μια σχετικά μικρή πόλη όπου όλες οι διασταυρώσεις είναι σε μορφή T, έχει λίγα φανάρια και δεν περιέχει δύσκολες κυκλοφοριακές δομές. Σε μία τέτοια πόλη μπορεί να προστεθεί μικρός αριθμός πεζών και οχημάτων.



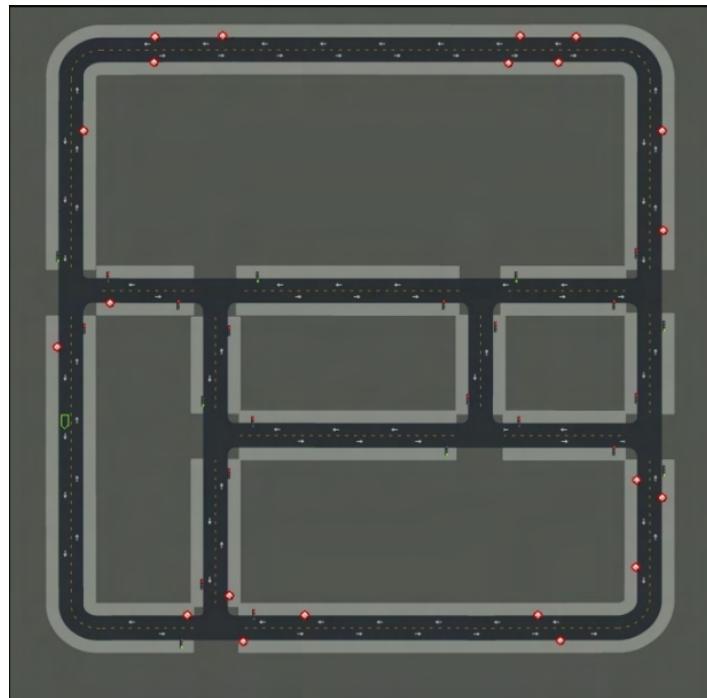
Σχήμα 4.1: Χάρτης της Πόλης 1



Σχήμα 4.2: Περιβάλλον της Πόλης 1

Town02

Η πόλη 2 μοιάζει αρκετά με την πόλη 1, αλλά είναι μικρότερη σε μέγεθος. Έχει το ίδιο κυκλοφοριακό περιβάλλον με την πόλη 1 και μπορεί να προστεθεί μικρός αριθμός πεζών και οχημάτων.



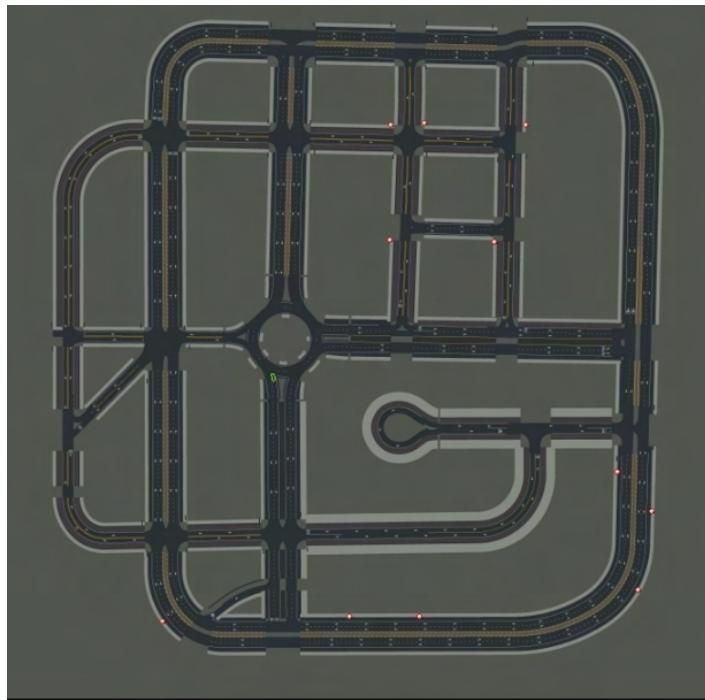
Σχήμα 4.3: Χάρτης της Πόλης 2



Σχήμα 4.4: Περιβάλλον της Πόλης 2

Town03

Αυτή η πόλη προσφέρει αρκετά πιο σύνθετα κυκλοφοριακά σενάρια καθώς έχει διασταυρώσεις με 4 λωρίδες κυκλοφορίας, ενώ περιέχει επίσης κυκλικό κόμβο, σήραγγα και πλάγιες οδούς.



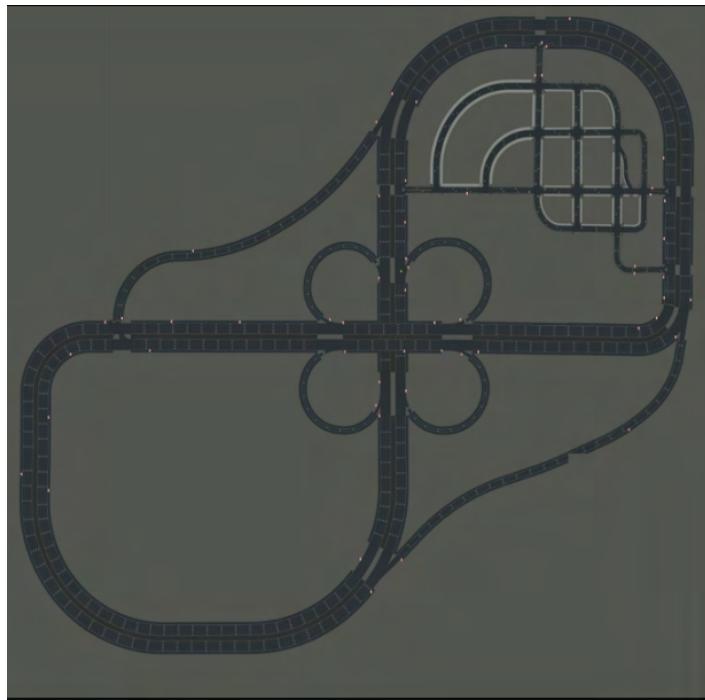
Σχήμα 4.5: Χάρτης της Πόλης 3



Σχήμα 4.6: Περιβάλλον της Πόλης 3

Town04

Η πόλη 4 υποστηρίζει μεγάλη ροή κυκλοφορίας καθώς έχει 4 λωρίδες για κάθε κατεύθυνση. Επιπλέον περιέχει έξοδο από τον αυτοκινητόδρομο που κατευθύνει σε μία μικρή πόλη αλλά και είσοδο στον αυτοκινητόδρομο.



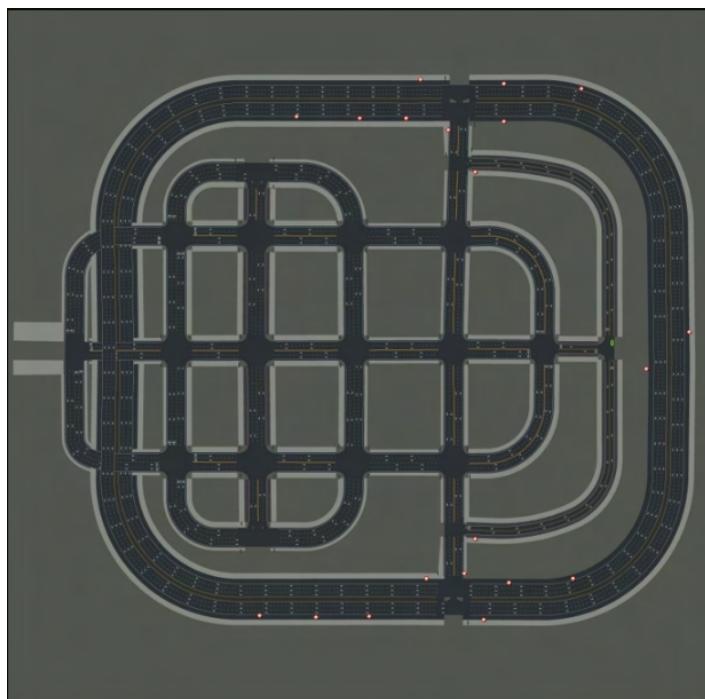
Σχήμα 4.7: Χάρτης της Πόλης 4



Σχήμα 4.8: Περιβάλλον της Πόλης 4

Town05

Αποτελεί μία πόλη μεγάλου μεγέθους που αποτελείται από επιμέρους τετράγωνα πλέγματα. Περιέχει μία μεγάλη γέφυρα, μία έως τρεις λωρίδες κυκλοφορίας και είναι ιδανική για μικρό και μεγάλο αριθμό οχημάτων και πεζών.



Σχήμα 4.9: Χάρτης της Πόλης 5



Σχήμα 4.10: Περιβάλλον της Πόλης 5

4.4.2 Αναγνώριση Αντικειμένων (Object Detection)

Όπως ήδη αναφέρθηκε στο Κεφάλαιο 3, συγκεντρώθηκε ένα σύνολο δεδομένων με 3624 εικόνες από το περιβάλλον του CARLA και οι πίνακες 4.3 και 4.4 που ακολουθούν περιέχουν τα στοιχεία για τις συνθήκες οδήγησης και την πόλη, καθώς και το ποσοστό των εικόνων που περιείχε αντικείμενα προς αναγνώριση (labeled) και το ποσοστό που δεν είχε (unlabeled). Εικόνες χωρίς αντικείμενα προς αναγνώριση είναι απαραίτητο να προστεθούν στο σύνολο δεδομένων προκειμένου το μοντέλο να αποκτήσει την ικανότητα να γενικεύει επαρκώς. Τελικά, στο σύνολο εκπαίδευσης έχουμε 2345 εικόνες, ενώ στο σύνολο ελέγχου 450.

Πίνακας 4.3: Εικόνες του συνόλου εκπαίδευσης και ελέγχου

Dataset		
Town	Weather	Images
Town01	ClearNoon	422
Town02	ClearNoon	387
Town03	ClearNoon	301
Town04	ClearNoon	389
Town05	ClearNoon	386
Town01	CloudyNoon	300
Town02	CloudyNoon	59
Town03	WetCloudyNoon	115
Town04	SoftRainyNoon	132
Town05	CloudyNoon	183
Town01	SoftRainyNoon	121

Πίνακας 4.4: Labeled και Unlabeled εικόνες στο σύνολο δεδομένων

Unlabeled	Labeled
755	1390

Πίνακας 4.5: Οι μετρικές του μοντέλου κατά την εκπαίδευση

Μετρικές του μοντέλου						
Weights Files	Precision	Recall	F1-Score	TP	FP	FN
1000 iterations	0.70	0.72	0.71	627	265	240
2000 iterations	0.81	0.78	0.78	680	120	187
3000 iterations	0.84	0.86	0.85	748	140	119
4000 iterations	0.89	0.84	0.87	731	90	136
5000 iterations	0.88	0.87	0.88	754	98	113
6000 iterations	0.89	0.88	0.88	761	97	106
7000 iterations	0.89	0.87	0.89	751	76	116
8000 iterations	0.89	0.89	0.89	770	98	97
9000 iterations	0.89	0.89	0.89	774	95	93
10000 iterations	0.88	0.88	0.88	766	100	101
11000 iterations	0.90	0.88	0.89	751	65	116
12000 iterations	0.89	0.89	0.89	768	87	99
13000 iterations	0.89	0.88	0.89	766	83	101
14000 iterations	0.88	0.88	0.89	768	84	99

Οι μετρικές mAP, IoU καθώς και οι TP, FP, FN παρατηρήθηκαν από τα αρχεία των βαρών της εκπαίδευσης του μοντέλου προκειμένου να εξαχθεί ένα συμπέρασμα σχετικά με τα καλύτερα βάρη, που είναι αυτά που θα χρησιμοποιηθούν τελικά με σκοπό τις καλύτερες προβλέψεις. Οι πίνακες 4.5 και 4.6 που ακολουθούν περιέχουν τα αποτελέσματα.

Αναλύοντας τα αποτελέσματα μπορούμε να εξάγουμε το συμπέρασμα ότι στην επανάληψη 11000 δημιουργούνται τα βάρη με την υψηλότερη απόδοση στο σύνολο ελέγχου, και είναι αυτά που θα χρησιμοποιηθούν για τις καλύτερες δυνατές προβλέψεις.

Πίνακας 4.6: Οι μετρικές απόδοσης mAP και IoU κατά τη διάρκεια εκπαίδευσης του μοντέλου

Weights Files	mAP	IoU
1000 iterations	63.83%	51.88%
2000 iterations	80.76%	68.61%
3000 iterations	84.97%	67.96%
4000 iterations	86.43%	73.17%
5000 iterations	87.40%	71.90%
6000 iterations	87.54%	72.68%
7000 iterations	87.02%	75.38%
8000 iterations	87.51%	73.03%
9000 iterations	88.21%	73.74%
10000 iterations	88.28%	73.29%
11000 iterations	88.99%	76.65%
12000 iterations	90.65%	76.15%
13000 iterations	90.28%	76.56%
14000 iterations	90.26%	76.51%

Αξίζει επίσης να παρατηρήσουμε ότι μετά από αυτή την επανάληψη η απόδοση αρχίζει προοδευτικά να μειώνεται, κάτι που οφείλεται στο φαινόμενο της υπερεκπαίδευσης. Στο Σχήμα 4.11 φαίνονται κάποια από τα αποτελέσματα σε εικόνες από το περιβάλλον του προσομοιωτή.

4.4.3 Οπτική Αναγνώριση Χαρακτήρων (OCR)

Η χρήση της μηχανής Tesseract OCR χρησιμοποιήθηκε για να αναγνωρίσει το περιέχομενο που αναγράφεται στα σήματα που συναντά το όχημα στο περιβάλλον του προσομοιωτή CARLA. Υπενθυμίζεται ότι υπάρχουν μόνο σήματα για το όριο της ταχύτητας κυκλοφορίας των οχημάτων (30, 60 και 90 χιλιόμετρα/ώρα).

Παρόλο που αυτή η διαδικασία αναγνώρισης των σημάτων θα μπορούσε να πραγματοποιηθεί και απευθείας στο πλαίσιο την Αναγνώρισης Αντικειμένων, προσθέτοντας εξαρχής στο σύστημα 3 ακόμα κλάσεις για τα 30,60 και 90 χιλιόμετρα/ώρα αντίστοιχα, επιλέχθηκε ιεραρχική προσέγγιση του προβλήματος. Για αυτή την διαδικασία αρχικά βρίσκονται τα πλαίσια εντοπισμού (bounding boxes) και στη συνέχεια απομονώνεται η πληροφορία για να γίνει η Οπτική Αναγνώριση Χαρακτήρων. Μια διαφορετική προσέγγιση του προβλήματος θα αύξανε πολύ το υπολογιστικό κόστος, καθώς θα υπήρχαν πλέον 10 κλάσεις προς αναγνώριση.

Πίνακας 4.7: Απόδοση της μεθόδου Οπτικής Αναγνώρισης Χαρακτήρων

30 km/h	37
60 km/h	39
90 km/h	24
Testing Set Accuracy	0.76

Για την αξιολόγηση της συγκεκριμένης διαδικασίας, έχουν εξαχθεί από τις εικόνες



Σχήμα 4.11: Αναγνώριση Αντικειμένων σε εικόνες από τον προσομοιωτή CARLA

του συνόλου δεδομένων 100 εικόνες που περιέχουν σήματα ταχυτήτων και σε σχέση με αυτές υπολογίστηκε η ακρίβεια των προβλέψεων, όπως παρατηρούμε στον πίνακα 4.7. Το σύνολο αυτό περιλαμβάνει εικόνες σε διαφορετικές καιρικές συνθήκες και συνθήκες φωτισμού.

Αυτό που θα μπορούσαμε να παρατηρήσουμε σε αυτή την περίπτωση, από τον πίνακα 4.7 είναι πως ενώ πρόκειται για μια υψηλή τιμή ακρίβειας, ο συγκεκριμένος αλγόριθμος υστερεί σε κάποιες περιπτώσεις. Συγκεκριμένα παρατηρήθηκε πως σε περιπτώσεις ακραίων καιρικών συνθηκών, όπως έντονη βροχή και ομίχλη, σε περιπτώσεις σκιερού τοπίου και σε περιπτώσεις που το σήμα εντοπίζεται, αλλά βρίσκεται αρκετά μακριά από το όχημα, τότε ο αλγόριθμος αδυνατεί να κάνει σωστές πρόβλεψεις.

4.4.4 Αναγνώριση χρώματος των φωτεινών σηματοδοτών (Traffic Light Color Recognition)

Για την Αναγνώριση Χρώματος για τους φωτεινούς σηματοδότες ακολουθήθηκε ο αλγόριθμος που περιγράφεται αναλυτικά στο Κεφάλαιο 3. Για την αξιολόγηση της μεθόδου απομονώθηκε από το σύνολο δεδομένων ένα πλήθος 100 εικόνων με την βοήθεια των οποίων υπολογίστηκε η ακρίβεια των προβλέψεων. Αξίζει να σημειωθεί ότι και σε αυτή τη περίπτωση το σύνολο που απομονώθηκε περιέχει εικόνες που έχουν ληφθεί υπό διαφορετικές καιρικές συνθήκες και συνθήκες φωτισμού. Τα αποτελέσματα παρουσιάζονται στον πίνακα 4.8.

Πίνακας 4.8: Απόδοση της μεθόδου αναγνώρισης χρώματος

Red Traffic Lights	27
Green Traffic Lights	55
Yellow Traffic Lights	18
Testing Set Accuracy	0.96

4.4.5 Αναγνώριση Λωρίδων (Lane Detection)

Η αναγνώριση των λωρίδων στο οδόστρωμα έγινε με την βοήθεια του Μετασχηματισμού ευθειών Hough, με μεθοδολογία που αναφέρθηκε αναλυτικά στο Κεφάλαιο 3. Στις παρακάτω εικόνες παρουσιάζονται κάποιες από τις δοκιμές που έγιναν με διάφορεταικές συνθήκες φωτισμού και καιρού στο περιβάλλον του προσομοιωτή.



Σχήμα 4.12: Εντοπισμός λωρίδας σε συνθήκες WetCloudyNoon

Μπορούμε να παρατηρήσουμε ότι ακόμα και σε συνθήκες βροχής (Σχήμα 4.13) ή έντονης σκιάς από εμπόδιο πάνω στην λωρίδα κατεύθυνσης (Σχήμα 4.14), η μέθοδος αποδίδει ικανοποιητικά.

Για την αξιολόγηση της ακρίβειας της μεθόδου επιλέχθηκαν και χρησιμοποιήθηκαν 50 εικόνες από το σύνολο δεδομένων, στις οποίες επικρατούν τυπικές συνθήκες καιρού και φωτισμού σε κατάσταση ημέρας, συγκεκριμένα ClearNoon, CloudyNoon και WetNoon. Σε αυτές εντοπίστηκαν οι πραγματικές ακμές που αναπαριστούν τις λωρίδες κι αυτές που εντοπίστηκαν από τη μέθοδο, και βρέθηκε η μεταξύ τους απόκλιση. Υπολογίζοντας την γωνία μεταξύ των ευθειών, βρίσκοντας την γωνία μεταξύ δύο διανυσμάτων παράλληλων στις ευθείες, βρέθηκε ότι αυτή είναι κατά μέσο όρο $1,06^\circ$ με τυπική



Σχήμα 4.13: Εντοπισμός λωρίδας σε συνθήκες MidRainyNoon



Σχήμα 4.14: Εντοπισμός λωρίδας σε συνθήκες ClearNoon

απόκλιση $\sigma = 0.8$. Επομένως, η μέθοδος εντόπισε αρκετά αποτελεσματικά τις λωρίδες σε αυτές τις περιπτώσεις.

Αναγνώριση Διασταυρώσεων (Junctions)

Στο περιβάλλον του προσομοιωτή CARLA οι διασταυρώσεις σε μεγάλες οδούς σηματοδοτούνται με οριζόντια γραμμή στο οδόστρωμα, όπως φαίνεται στην εικόνα 4.15. Η ίδια διαδικασία με την αναγνώριση των λωρίδων ακολουθήθηκε και για αυτή την αναγνώριση, με τη διαφορά ότι ορίζεται ένα κατώφλι σχετικά με την κλίση των ευθειών που αναγνωρίζονται ώστε να προσεγγίζει αυτή των οριζόντιων γραμμών εφόσον αυτές βρίσκονται στη περιοχή ενδιαφέροντος μπροστά από το όχημα.

Για την αξιολόγηση της συγκεκριμένης διαδικασίας, έχουν εξαχθεί από τις εικόνες του συνόλου δεδομένων 50 εικόνες που περιέχουν τέτοιους είδους διασταυρώσεις δρόμων και σε σχέση με αυτές υπολογίστηκε η ακρίβεια των προβλέψεων, όπως φαίνεται στον πίνακα 4.9.

Πίνακας 4.9: Απόδοση της μεθόδου αναγνώρισης διασταυρώσεων

Testing Accuracy	0.70
-------------------------	-------------



Σχήμα 4.15: Εντοπισμός διασταύρωσης

Η συγκεκριμένη μέθοδος λειτουργεί με ακρίβεια υπάρχουν όμως περιπτώσεις, κατά τις οποίες το όχημα έχει παρεκλίνει από την θέση του ανάμεσα στις γραμμές του οδοιστρώματος και επαναπροσεγγίζει την θέση του, στις οποίες αυτή η μέθοδος υστερεί. Δεδομένου ότι χρησιμοποιείται μόνο η κάμερα και η μέθοδος χρησιμοποιεί την κλίση των ευθειών ως παράγοντα απόφασης αυτή η αδυναμία της μεθόδου ήταν αναμενόμενη.

4.4.6 Πλοήγηση με χρήση PID ελεγκτή

Η διαδικασία πλοήγησης με χρήση του ελεγκτή, ενσωματώνοντας στο σύστημα τον VehiclePIDController του προσομοιωτή, για τους ελέγχους ταχύτητας, πέδησης και κατεύθυνσης του οχήματος, εκτελέστηκε τοπιθετώντας το όχημα στο σημείο εκκίνησης της διαδρομής, με εντοπισμό της θέσης του με GPS μέσα στο μονοπάτι που ορίζεται κάθε φορά αποτελούμενο από waypoints τα οποία το όχημα ακολουθεί. Όλα τα πειράματα εκτελέστηκαν στις πόλεις 1,2 και 3 του CARLA θέτοντας διαφορετικές διαδρομές εκτέλεσης και μεταβλητό μήκος διαδρομών. Εκτελέστηκαν δύο πειράματα σε κάθε πόλη ακολουθώντας μια διαφορετική διαδρομή σε κάθε πείραμα, και η εκτέλεση της διαδρομής βαθμολογείται με 0 εφόσον δεν ολοκληρωθεί με επιτυχία μια διαδρομή και με 1 εφόσον το όχημα εκτελέσει με επιτυχία την πλοήγηση στη διαδρομή που ορίστηκε. Το ποσοστό εκτέλεσης διαδρομών φαίνεται στον πίνακα 4.10.

Πίνακας 4.10: Ποσοστό εκτέλεσης διαδρομών με το σύστημα με PID controller

Testing Accuracy	0.83
-------------------------	-------------

Στην συγκεκριμένη μεθοδολογία, χρειάζεται ιδιαίτερη προσοχή και πολλά πειράματα ώστε να βρεθούν οι κατάλληλες τιμές για τα κέρδη του ελεγκτή, ανάλογα με τις αποστάσεις των waypoints και την ταχύτητα του οχήματος. Αυτή η διαδικασία

έγινε επαναληπτικά εως ότου να βρεθούν οι κατάλληλες τιμές, ωστόσο μία από αυτές τις διαδρομές δεν κατάφερε να ολοκληρωθεί και θα έπρεπε να αλλάξουν κάποιες από τις παραμέτρους. Στην εικόνα 4.16 βλέπουμε ένα στιγμιότυπο μίας από τις διαδρομές.



Σχήμα 4.16: Πλοήγηση με waypoints

4.4.7 Πλοήγηση με διατήρηση της λωρίδας

Για την διαδικασία της πλοήγησης με διατήρηση λωρίδας το όχημα επιχειρεί, ξεκινώντας από μια αρχική θέση που ορίζεται από τον χρήστη και χωρίς χρήση GPS, να ολοκληρώσει μια ευθεία διαδρομή παραμένοντας ανάμεσα στα όρια των λωρίδων μόνο από τον εντοπισμό τους και δίνοντας εντολές τιμονιού με βάση αυτές.

Για την συγκεκριμένη μέθοδο, επιλέχθηκαν κάποιες διαδρομές χωρίς διασταυρώσεις για την αξιολόγησή της. Όπως αναλύθηκε σε προηγούμενη παράγραφο, οι πόλεις 1 και 2 διαθέτουν οδικό δίκτυο χωρίς σύνθετες δομές, επιμένως επιλέχθηκαν αυτές οι πόλεις για την διεξαγωγή των πειραμάτων. Σε κάθε πόλη εκτελέστηκαν 3 διαδρομές. Το αποτέλεσμα φαίνεται στον πίνακα 4.11 που ακολουθεί.

Πίνακας 4.11: Ποσοστό εκτέλεσης διαδρομών με το σύστημα διατήρησης λωρίδας

Testing Accuracy	0.67
-------------------------	-------------

Σε αυτήν την περίπτωση παρατηρήθηκε ότι το όχημα μπορούσε να πλοηγηγεί με επιτυχία μέχρι το τέλος διαδρομής σε περιπτώσεις που οι λωρίδες του δρόμου φαίνονταν ξεκάθαρα. Ωστόσο, σε περιπτώσεις που μια μικρή ανωμαλία του δρόμου μπορεί να προκαλούσε κάποια παραμόρφωση για κάποια δευτερόλεπτα στις λωρίδες που αναγνώριζε το σύστημα, τότε το όχημα έκανε σχετικά απότομες κινήσεις ώστε να βρεθεί στη μέση των λωρίδων, αποσταθεροποιούνταν χωρίς να φτάσει πάλι στην επιθυμητή θέση. Κάτι τέτοιο θα μπορούσε να αποφευχθεί χρησιμοποιώντας ακόμη πιο τιμή για τη μέγιστη τιμή που μπορεί να πάρει ο έλεγχος από το τιμόνι ή με μια καλύτερη μέθοδο σταθεροποίησης της γωνίας διεύθυνσης.

Κεφάλαιο 5

Συμπεράσματα & μελλοντική εργασία

Στο παρόν και τελευταίο κεφάλαιο παρουσιάζονται συνοπτικά τα συμπεράσματα που προέκυψαν από την έκβαση των πειραμάτων, καθώς και προτάσεις για περαιτέρω βελτίωση και εξέλιξη του συστήματος που αναπτύχθηκε.

5.1 Γενικά συμπεράσματα

Στα πλαίσια της παρούσας διπλωματικής εργασίας, υλοποιήθηκαν τρία επιμέρους υποσυστήματα αντίληψης, πλοήγησης και κατασκευής βασικού μονοπατιού. Για το πρώτο υποσύστημα, αναλύσαμε τον τρόπο με τον οποίο ένας ηλεκτρονικός υπολογιστής μπορεί να εξάγει χρήσιμες πληροφορίες από εικόνες ή από βίντεο που έχουν προέλθει από την ενσωματωμένη κάμερα και μέσω της Βαθιάς Μάθησης καταφέραμε να επεξεργαστούμε εικόνες και να εξάγουμε πληροφορίες σε σχέση με τα στατικά και δυναμικά εμπόδια του περιβάλλοντος, με αρκετά ικανοποιητικά αποτελέσματα. Ακόμα, μπορέσαμε να ερμηνεύσουμε την πληροφορία των σημάτων οδικής κυκλοφορίας με OCR, ενώ με τεχνικές ψηφιακής επεξεργασίας εικόνας επιτεύχθηκε η αναγνώριση των λωρίδων στον οδόστρωμα, των διασταυρώσεων και η αναγνώριση χρώματος των φαναριών. Στην συνέχεια, στο υποσύστημα πλοήγησης υλοποιήθηκε ένα σύστημα διατήρησης λωρίδας χρησιμοποιώντας πληροφορία από το υποσύστημα αντίληψης και ένα σύστημα που χρησιμοποιεί PID ελεγκτή και GPS για την πλοήγηση, εντοπίζοντας την θέση του οχήματος και ακολουθώντας τα προκαθορισμένα σημεία του μονοπατιού. Τέλος, υλοποιήθηκε μια μέθοδος κατασκευής βασικού μονοπατιού για την πλοήγηση.

Με βάση τα αποτελέσματα των πειραμάτων του Κεφαλαίου 4 που αφορούν την ικανότητα του οχήματος να εκτελεί τις επιμέρους λειτουργίες που αναφέρθηκαν, παρατηρήθηκαν τα εξής:

- Το σύστημα μπόρεσε να αναγνωρίσει επιτυχώς τα στατικά και δυναμικά εμπόδια του οδικού δικτύου, δίνοντας ενθαρρυντικά αποτελέσματα, και συνεπώς εκπληρώθηκε ένας από τους βασικούς στόχους της παρούσας εργασίας. Παρατηρήθηκε ότι στα συγκεκριμένα πειράματα το σύστημα λειτουργούσε καλύτερα όταν τα εμπόδια προς αναγνώριση βρίσκονταν αρκετά κοντά στο όχημα, ενώ υπήρχε μεγαλύτερη αδυναμία σε απομακρυσμένα εμπόδια, γεγονός που αναμενόταν και

δεν αποτελεί ιδιαίτερο πρόβλημα για την διαδικασία που ακολουθήσαμε.

2. Στα πειράματα εντοπισμού των λωρίδων, η μέθοδος που χρησιμοποιήθηκε κατάφερε να έχει αρκετά καλά αποτελέσματα και να αναταποκριθεί σε διαφορετικές συνθήκες καιρού και φωτισμού. Παρόμοια ήταν και η συμπεριφορά του συστήματος για τον εντοπισμό των διασταυρώσεων, ωστόσο παρατηρήθηκε ότι η μέθοδος επηρεαζόταν σε μεγάλο βαθμό από τη θέση του οχήματος στο οδόστρωμα.
3. Παρατηρήθηκε πως η διαδικασία της οπτικής αναγνώρισης χαρακτήρων επηρεάστηκε πολύ περισσότερο από τις υπόλοιπες μεθόδους από τις καιρικές συνθήκες και τις συνθήκες φωτισμού.
4. Για την διαδικασία πλόγησης με διατήρηση λωρίδας, το σύστημα αντίληψης λειτούργησε ικονοποιητικά, ώστε να έχουμε αρκετά καλά αποτελέσματα στα πειράματα που το αφορούσαν. Ωστόσο, δεδομένου ότι στις στροφές σε μεγαλύτερους δρόμους δεν υπάρχουν λωρίδες η μέθοδος αυτή δεν θα λειτουργούσε.

5.2 Προβλήματα

Ένα από τα αρχικά προβλήματα που παρουσιάστηκαν κατά την διάρκεια της εκπόνησης της διπλωματικής εργασίας, ήταν η δυσκολία στην εύρεση κατάλληλων τεχνικών, που να υποδεικνύουν τον τρόπο με τον οποίο θα έπρεπε να συνδυαστούν οι μέθοδοι με βασικό αισθητήρα την κάμερα, που παρουσιάζονται στη βιβλιογραφία, για την επίτευξη μιας ολοκληρωμένης συμπεριφοράς. Παρόλο που βρέθηκαν κάποιοι μέθοδοι για μεμονωμένα συστήματα, όπως αυτά που παρουσιάστηκαν και στην εργασία, δεν υπήρχαν αρκετές αναφορές για πλήρεις αρχιτεκτονικές κατασκευής που να αναδεικνύουν τον συνδυασμό όλων των υποσυστημάτων.

Ακόμα ένα πρόβλημα που παρουσιάστηκε ήταν η επιλογή των τιμών του PID ελεγκτή για τον έλεγχο ταχύτητας και θέσης του συστήματος. Η ρύθμιση πραγματοποιήθηκε εμπειρικά δοκιμάζοντας το πρόγραμμα σε διαφορετικά σενάρια και όταν πρόκυπτε κάποιο πρόβλημα θα έπρεπε να αφιερωθεί χρόνος εντοπισμού της αιτίας του προβλήματος και της επίλυσης του.

Το μεγαλύτερο εμπόδιο που αντιμετωπίστηκε κατά τη διάρκεια της εργασίας ήταν οι υψηλές απαιτήσεις για υπολογιστικούς πόρους του CARLA. Όπως παρατηρήθηκε και στα πειράματα σε μεγάλο αριθμό οχημάτων και πεζών, όπως επίσης και σε υψηλές ταχύτητες, το πρόγραμμα προσομοίωσης έφτανε κάτω και από 5 fps ξεπερνώντας το όριο κατά πολύ για την ομαλή λειτουργία του. Πιο συγκεκριμένα, χρησιμοποιήθηκε μνήμη RAM 8GB και κάρτα γραφικών Radeon R5 M335 με μνήμη γραφικών 4 GB. Ο προσομοιωτής συνιστούσε την χρήση γραφικών τουλάχιστον 470 GTX με 8 GB εσωτερικής μνήμης, οπότε ήταν αναμενόμενο να υπάρχει δυσκολία κατά την εκτέλεση. Έτσι, τα αποτελέσματα που προέκυψαν είναι σχετικά ακριβή για διαδικασίες χαμηλού υπολογιστικού κόστους, δηλαδή για τις μικρότερες πόλεις με μικρό αριθμό οχημάτων και πεζών και για τις διαδικασίες εξαγωγής εικόνων και επεξεργασίας εικόνων, ωστόσο περισσότερα πειράματα για τη διαδικασία της πλοϊγησης, και ειδικότερα σε μεγαλύτερες πόλεις με πιο σύνθετες δομές, θα μπορούσαν να γίνουν μόνο με άλλους υπολογιστικούς πόρους.

5.3 Μελλοντικές Επεκτάσεις

Μια βελτίωση που θα μπορούσε να πραγματοποιηθεί στην παρούσα διπλωματική είναι στο σύστημα αντίληψης του αυτόνομου οχήματος. Το σύστημα που κατασκευάστηκε χρησιμοποιεί το προγραμματιστικό περιβάλλον του προσομοιωτή CARLA για να λάβει τις απαραίτητες πληροφορίες για το περιβάλλον γύρω από το οχήμα. Με την εξαγωγή των εικόνων δημιουργήθηκε ένα σύνολο δεδομένων για την εκπαίδευση του μοντέλου YOLO. Για το σύνολο των δεδομένων θα θέλαμε έναν μεγάλο όγκο εικόνων για την εκπαίδευση, οπότε ένα βήμα που θα μπορούσε να γίνει για τη βελτίωση της διαδικασίας θα ήταν να αυξηθεί το πλήθος του συνόλου δεδομένων και ελέγχου για να αποκτήσουμε μια καλύτερη και πιο αντιπροσωπευτική ένδειξη για το πως συμπεριφέρεται το μοντέλο. Ωστόσο, η ανάκτηση ενός τόσο μεγάλου όγκου δεδομένων είναι μια αρκετά χρονοθόρα διαδικασία.

Ακόμα, θα μπορούσε να υλοποιηθεί ένα σύστημα πλοιόγησης που να λειτουργεί εξ ολοκλήρου με τη χρήση κάμερας, μελετώντας πιο σύνθετες μεθόδους. Χρησιμοποιώντας, δηλαδή, μόνο την οπτική του αντίληψη για τον χώρο να μπορεί να πλοιηγείται και να λαμβάνει αποφάσεις για τις επόμενες κινήσεις του βάσει των οπτικών πληροφοριών που λαμβάνει από το αισθητήριο όργανο της κάμερας. Αυτές οι προσεγγίσεις χρησιμοποιούν συνήθως νευρωνικά δίκτυα σε συνδυασμό με μια υλοποιήση μιας Μηχανής Καταστάσεων (FSM). Κάτι τέτοιο θα ενσωμάτωνε και ένα επιπλέον σύστημα επιλογής συμπεριφοράς με σκοπό τον συνδυασμό των επιμέρους υποσυστημάτων.

Μια επιπρόσθετη βελτίωση για το σύστημα θα ήταν μελέτη και η χρήση μιας αναλυτικής μεθόδου για τον καθορισμό των κανόνων αξιολόγησης κατά την διαδικασία αξιολόγησης των μεθόδων πλοιόγησης. Θα μπορούσε να χρησιμοποιηθεί ένας προκατασκευασμένος πίνακας που αντικατοπτρίζει την αξία ενός κανόνα για κάθε συμπεριφορά και πως επηρεάζει την μετρική αξιολόγησης, με επιλογή βαρών για κάθε παράβαση που μπορεί να εκτελεί το όχημα.

Επιπλέον, για την αξιολόγηση της μεθόδου αναγνώρισης των λωρίδων θα μπορούσε να γίνει αξιολόγηση με μέρος του συνόλου δεδομένων που να περιλαμβάνει περισσότερες εικόνες με ακραίες συνθήκες φωτισμού και καιρού, για να υπάρξει μια πιο ακριβής προσέγγιση σε σχέση με την αποτελεσματικότητα της μεθόδου.

Τέλος, θα μπορούσε να δημιουργηθεί ένα σύστημα εναλλακτικών διαδρομών σε περίπτωση αποφυγής εμποδίων ή προσπέρασης οχήματος, το οποίο θα λάμβανε πληροφορίες από το σύστημα αντίληψης.

Bibliography

- [1] NHTSA. *Preliminary Statement of Policy Concerning Automated Vehicle*. 2013. URL: http://www.nhtsa.gov/staticfiles/20rulemaking/pdf/Automated_Vehicles_Policy.pdf.
- [2] SAE International standard J3016 SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2014.
- [3] VDA (German Association of the Automotive Industry). *Automated Driving*. 2016. URL: <https://www.vda.de/de/themen/innovation-undtechnik/automatisiertes-fahren/automatisiertes-fahren.html>.
- [4] Tom M Gasser and Daniel Westhoff. “BASt-study: Definitions of automation and legal issues in Germany”. In: *Proceedings of the 2012 road vehicle automation workshop*. Bergisch Gladbach: German Federal Highway Research Institute. 2012.
- [5] Lloyd’s. *Autonomous Vehicles: Handling Over Control: Opportunities and Risks for Insurance*. 2014. URL: <https://assets.lloyds.com/assets/pdf-autonomous-vehicles/1/pdf-autonomous-vehicles.pdf>.
- [6] Tomtom. *What are the six levels of autonomous driving?* 2019. URL: <https://www.tomtom.com/blog/automated-driving/different-levels-of-autonomous-driving>.
- [7] T. Litman. “Autonomous Vehicle Implementation Predictions”. In: (June 2020). URL: <https://www.vtpi.org/avip.pdf>.
- [8] *The Six Levels of Autonomous Testing Automated Visual Testing Applitools*. <https://applitools.com/blog/not-only-cars-the-six-levels-of-autonomous/>. Feb. 2019.
- [9] Araz Taeihagh and Hazel Si Min Lim. “Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks”. In: *Transport Reviews* 39.1 (2019), pp. 103–128. doi: 10.1080/01441647.2018.1494640. eprint: <https://doi.org/10.1080/01441647.2018.1494640>. URL: <https://doi.org/10.1080/01441647.2018.1494640>.
- [10] Keshav Bimbaw. “Autonomous Cars: Past, Present and Future - A Review of the Developments in the Last Century, the Present Scenario and the Expected Future of Autonomous Vehicle Technology”. In: *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings* 1 (Jan. 2015), pp. 191–198. doi: 10.5220/0005540501910198.
- [11] Ricardo A. Daziano, Mauricio Sarrias, and Benjamin Leard. “Are consumers willing to pay to let cars drive for them? Analyzing response to autonomous vehicles”. In: *Transportation Research Part C: Emerging Technologies* 78 (2017), pp. 150–164. ISSN: 0968-090X. doi: <https://doi.org/10.1016/j.trc>.

- 2017.03.003. URL: <https://www.sciencedirect.com/science/article/pii/S0968090X17300682>.
- [12] *The Mercedes-Benz F 015 Luxury in Motion Forerunner of a mobility revolution.* <https://group-media.mercedes-benz.com/marsMediaSite/en/instance/ko/The-Mercedes-Benz-F-015-Luxury-in-Motion-Forerunner-of-a-mobility-revolution.xhtml?oid=9906573>.
- [13] *Mercedes-Benz Innovation Autonomous.* <https://www.mercedes-benz.com/en/innovation/autonomous/>. Sept. 2020.
- [14] Motortrend. *Tesla Model S.* <https://www motortrend.com/cars/tesla/model-s/2015/>. Dec. 2015.
- [15] Autoware. *Vision and Mission.* <https://www.autoware.org/visionandmission>. Aug. 2021.
- [16] Shital Shah et al. *AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles.* 2017. doi: 10.48550/ARXIV.1705.05065. URL: <https://arxiv.org/abs/1705.05065>.
- [17] Alexey Dosovitskiy et al. “CARLA: An open urban driving simulator”. In: *Conference on robot learning*. PMLR. 2017, pp. 1–16.
- [18] *LG ELECTRONICS UNITY TECHNOLOGIES COLLABORATE ON AUTONOMOUS VEHICLE SIMULATION LG Singapore.* <https://www.lg.com/sg/about-lg/press-and-media/lg-electronics-unity-technologies-collaborate-on-autonomous-vehicle-simulation>. Apr. 2021.
- [19] Jelena Kocić, Nenad Jovičić, and Vujo Drndarević. “Sensors and Sensor Fusion in Autonomous Vehicles”. In: *2018 26th Telecommunications Forum (TELFOR)*. 2018, pp. 420–425. doi: 10.1109/TELFOR.2018.8612054.
- [20] Daniel Sales et al. “Vision-Based Autonomous Navigation System Using ANN and FSM Control”. In: *Latin American Robotics Symposium and Intelligent Robotics Meeting 0* (Oct. 2010), pp. 85–90. doi: 10.1109/LARS.2010.26.
- [21] Sebastian Thrun et al. “Stanley: The robot that won the DARPA Grand Challenge”. In: *Journal of field Robotics* 23.9 (2006), pp. 661–692.
- [22] Chris et al. Urmson. “Autonomous Driving in Urban Environments: Boss and the Urban Challenge”. In: *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–59. ISBN: 978-3-642-03991-1. doi: 10.1007/978-3-642-03991-1_1. URL: https://doi.org/10.1007/978-3-642-03991-1_1.
- [23] Simon Herman and Khalid N. Ismail. “Single Camera Object Detection for Self-Driving Vehicle: A Review”. In: *Journal of the Society of Automotive Engineers Malaysia* (2021).
- [24] *How Autonomous Vehicles Sensors Fusion Helps Avoid Deaths.* <https://intellias.com/sensor-fusion-autonomous-cars-helps-avoid-deaths-road/>. Nov. 2018.
- [25] A. L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. doi: 10.1147/rd.33.0210.
- [26] Osvaldo Simeone. *A Brief Introduction to Machine Learning for Engineers.* 2017. doi: 10.48550/ARXIV.1709.02840. URL: <https://arxiv.org/abs/1709.02840>.
- [27] Χρήστος Α. Διδαχός. *Χρήση Τεχνικών Βαθιάς Μάθησης Για Λογισμικό Αυτόνομης Οδήγησης.* 2021. URL: <http://hdl.handle.net/10889/14454>.

- [28] Keiron O’Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. doi: 10.48550/ARXIV.1511.08458. URL: <https://arxiv.org/abs/1511.08458>.
- [29] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks*. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Dec. 2018.
- [30] Hamed Habibi Aghdam and Elnaz Jahani Heravi. “Guide to convolutional neural networks”. In: *New York, NY: Springer* 10.978-973 (2017), p. 51.
- [31] Dan C. Ciresan et al. “Flexible, High Performance Convolutional Neural Networks for Image Classification”. In: *IJCAI*. 2011.
- [32] G. E. Hinton and R. R. Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science* 313.5786 (2006), pp. 504–507. doi: 10.1126/science.1127647. eprint: <https://www.science.org/doi/pdf/10.1126/science.1127647>. URL: <https://www.science.org/doi/abs/10.1126/science.1127647>.
- [33] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proc. IEEE* 86 (1998), pp. 2278–2324.
- [34] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *Int. J. Comput. Vision* 115.3 (Dec. 2015), pp. 211–252. ISSN: 0920-5691. doi: 10.1007/s11263-015-0816-y. URL: <https://doi.org/10.1007/s11263-015-0816-y>.
- [35] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I-I. doi: 10.1109/CVPR.2001.990517.
- [36] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. doi: 10.1109/CVPR.2005.177.
- [37] Pedro Felzenszwalb, David Mcallester, and Deva Ramanan. “A Discriminatively Trained, Multiscale, Deformable Part Model”. In: vol. 8: June 2008. doi: 10.1109/CVPR.2008.4587597.
- [38] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT ’92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, pp. 144–152. ISBN: 089791497X. doi: 10.1145/130385.130401. URL: <https://doi.org/10.1145/130385.130401>.
- [39] Xin Yuan, Xiaosen Shan, and Li Su. “A Combined Pedestrian Detection Method Based on Haar-Like Features and HOG Features”. In: *2011 3rd International Workshop on Intelligent Systems and Applications* (2011), pp. 1–4.
- [40] *Using Histogram of Oriented Gradients HOG for Object Detection*. <https://iq.opengenus.org/object-detection-with-histogram-of-oriented-gradients-hog/>. Mar. 2020.
- [41] M. Everingham et al. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

- [42] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.
- [43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- [44] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2013. doi: 10.48550/ARXIV.1311.2524. URL: <https://arxiv.org/abs/1311.2524>.
- [45] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. *Mesh R-CNN*. 2019. doi: 10.48550/ARXIV.1906.02739. URL: <https://arxiv.org/abs/1906.02739>.
- [46] Ross Girshick. *Fast R-CNN*. 2015. doi: 10.48550/ARXIV.1504.08083. URL: <https://arxiv.org/abs/1504.08083>.
- [47] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. doi: 10.48550/ARXIV.1506.01497. URL: <https://arxiv.org/abs/1506.01497>.
- [48] Kaiming He et al. *Mask R-CNN*. 2017. doi: 10.48550/ARXIV.1703.06870. URL: <https://arxiv.org/abs/1703.06870>.
- [49] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [50] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *Computer Vision - ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 21–37. ISBN: 978-3-319-46448-0.
- [51] Jonathan Hui. *Object detection: speed and accuracy comparison*. <https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359>. Oct. 2020.
- [52] CARLA Simulator. <https://carla.readthedocs.io/en/latest/>. Nov. 2021.
- [53] OpenCV. <https://opencv.org/>. June 2002.
- [54] Kari Pulli et al. "Realtime Computer Vision with OpenCV". In: *Queue* 10 (2012), pp. 40–56.
- [55] Linda G Shapiro, George C Stockman, et al. *Computer vision*. Vol. 3. Prentice Hall New Jersey, 2001.
- [56] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.
- [57] Byambaa Dorj and Deokjin Lee. "A Precise Lane Detection Algorithm Based on Top View Image Transformation and Least-Square Approaches". In: *Journal of Sensors* 2016 (Nov. 2015), pp. 1–13. doi: 10.1155/2016/4058093.
- [58] *Intersection over Union IoU for object detection*. <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. Nov. 2016.
- [59] *Tesseract documentation Tesseract OCR*. <https://tesseract-ocr.github.io/>. May 2015.

- [60] R. Smith. “An Overview of the Tesseract OCR Engine”. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. Vol. 2. 2007, pp. 629–633. doi: 10.1109/ICDAR.2007.4376991.
- [61] Ray W. Smith. “History of the Tesseract OCR engine: what worked and what didn’t”. In: *Document Recognition and Retrieval XX*. Ed. by Richard Zanibbi and Bertrand Coërasnon. Vol. 8658. International Society for Optics and Photonics. SPIE, 2013, p. 865802. doi: 10.1117/12.2010051. url: <https://doi.org/10.1117/12.2010051>.
- [62] Chirag I. Patel, Atul Patel, and Dr. Dharmendra Patel. “Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study”. In: *International Journal of Computer Applications* 55 (2012), pp. 50–56.
- [63] A Galip Ulsoy, Huei Peng, and Melih Çakmakci. *Automotive control systems*. Cambridge University Press, 2012.
- [64] Martí Sánchez Juanola. “Speed traffic sign detection on the CARLA simulator using YOLO”. In: 2019. url: <http://hdl.handle.net/10230/42548>.
- [65] Heartex. “LabelImg”. In: *GitHub Repository* 6 (2015).
- [66] Mattia Maldini. *PID control explained Medium*. <https://maldus512.medium.com/pid-control-explained-45b671f10bc7>.
- [67] Krishtof Korda Luca Venturi. “Hands On Vision and Behavior for Self Driving Cars”. In: Oct. 2020. ISBN: 978-1-80020-358-7.
- [68] Mohammad Hossin and Sulaiman M.N. “A Review on Evaluation Metrics for Data Classification Evaluations”. In: *International Journal of Data Mining Knowledge Management Process* 5 (Mar. 2015), pp. 01–11. doi: 10.5121/ijdkp.2015.5201.