ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Οπτική Ανίχνευση Εμποδίων για Αυτόνομα Οχήματα: Συγκριτική Αξιολόγηση σε CARLA, AirSim και Raspberry Pi

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Γρηγορίδου Αθανασίας

Επιβλέπων : κ. Καβαλλιεράτου Εργίνα

Μέλη εξεταστικής επιτροπής:

Σάμος, Σεπτέμβριος 2025

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο των σπουδών μου στο Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου. Αποτελεί το αποτέλεσμα μιας πολύχρονης προσπάθειας, κατά την οποία απέκτησα γνώσεις και δεξιότητες που συνέβαλαν ουσιαστικά στη διαμόρφωση της επιστημονικής και επαγγελματικής μου ταυτότητας. Η διαδικασία εκπόνησης της εργασίας μου επέτρεψε να εμβαθύνω σε ένα αντικείμενο που συνδυάζει σύχρονες τεχνολογίες με εφαρμογές μεγάλης σημασίας, όπως η αυτόνομη οδήγηση και η υπολογιστική όραση.

Θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη στην επιβλέπουσα διδάσκουσα κ. Καβαλλιεράτου Εργίνα, για την πολύτιμη και συνεχή καθοδήγηση που συνέβαλε καθοριστικά στην ολοκλήρωση της εργασίας αυτής. Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου για τη υποστήριξη που μου παρείχε καθ΄όλη τη διάρκεια των σπουδών μου.

© 2025
της
Γρηγορίδου Αθανασίας
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1 Εισαγωγή 1

1.1 Αυτόνομη Οδήγηση και Υπολογιστική Όραση 1

1.2 Αντικείμενο διπλωματικής 2

1.2.1 Ερευνητικοί στόχοι και προκλήσεις 2

1.3 Δομή της διπλωματικής 2

2 Θεωρητικό Υπόβαθρο και Σύγχρονες Τεχνολογίες στην Αυτόνομη Οδήγηση 4

2.1 Εισαγωγή 4

2.2 Ο αλγόριθμος Canny Edge Detection 5

2.3 Υψηλή Αντίθεση Φωτισμού στην Αυτόνομη οδήγηση 7

2.4 Συγκριτική Ανάλυση Μεθόδων Ανίχνευσης Ακμών 8

2.5 Εισαγωγή στους Προσομοιωτές 9

2.5.1 Προσομοιωτής CARLA 9

2.5.2 Προσομοιωτής AirSim 10

2.6 Το Sim2Real Gap στην Αυτόνομη Οδήγηση 12

2.7 Συμπεράσματα 14

3 Μεθοδολογία Ανίχνευσης, Απόφασης και Καταγραφής Δεδομένων 15

3.1 Εισαγωγή 15

3.2 Στάδια Επεξεργασίας Εικόνας 16

3.3 Απόσταση και Κριτήρια Απόφασης 17

3.4 Καταγραφή Δεδομένων για Επαλήθευση και Ανάλυση 18

3.5 Συμπεράσματα 19

4 Προσομοίωση στο CARLA 20

4.1 Εισαγωγή 20

4.2 Περιγραφή Περιβάλλοντος CARLA 21

4.3 Περιγραφή Σεναρίων 22

4.3.1 Σενάριο 1 – Υπερβολικός φωτισμός & Ηλιακή έκθεση 22

4.3.2 Σενάριο 2 – Μέτριος φωτισμός 22

4.3.3 Σενάριο 3 – Κανονικές συνθήκες ημέρας 23

4.3.4 Σενάριο 4 – Βρεγμένος δρόμος 24

4.3.5 Σενάριο 5 – Έντονη σκιά 24

4.4 Αποτελέσματα Ανίχνευσης 25

4.5 Συμπεράσματα 26

5 Προσομοίωση στο AirSim 27

5.1 Εισαγωγή 27

5.2 Περιγραφή περιβάλλοντος AirSim 28

5.3 Περιγραφή Σεναρίων 29

5.3.1 Σενάριο 1 – Κανονικές Συνθήκες Φωτισμού 29

5.3.2 Σενάριο 2 – Φωτισμός απο το εμπόδιο προς την κάμερα 29

5.3.3 Σενάριο 3 – Φωτισμός πάνω στο εμπόδιο 30

5.3.4 Σενάριο 4 – Φωτισμός πίσω από το εμπόσιο 30

5.4 Αποτελέσματα Ανίχνευσης 32

5.5 Συμπεράσματα 33

6 Πειραματική Εκτέλεση σε Raspberry Pi 4 34

6.1 Εισαγωγή 34

6.2 Περιγραφή Υλικού και Συνδεσμολογίας 35

6.3 Ανάγκη Ελέγχου Πορείας και Ενσωμάτωση Lane Detection 38

6.4 Γραφικό Περιβάλλον και Καταγραφή Συμβάντων 40

6.5 Περιγραφή Σεναρίων 41

6.5.1 Σενάριο 1 – Φυσιολογικός Φωτισμός 41

6.5.2 Σενάριο 2 – Φωτισμός προς το εμπόδιο από μπροστά 42

6.5.3 Σενάριο 3 – Φωτισμός από πάνω στο εμπόδιο 44

6.5.4 Σενάριο 4 – Φωτισμός από πλάγια προς την κάμερα 44

6.5.5 Σενάριο 5 – Φωτισμός από πίσω προς την κάμερα 45

6.6 Αποτελέσματα Ανίχνευσης 47

6.7 Συμπεράσματα 49

7 Αξιολόγηση Απόδοσης Ανίχνευσης Εμποδίων 50

7.1 Εισαγωγή 50

7.2 Μετρικές Αξιολόγησης 51

7.3 Ανάλυση Αποτελεσμάτων 52

7.4 Συμπεράσματα 54

8 Συμπεράσματα 55

8.1 Εισαγωγή 55

8.2 Συνοπτικά Αποτελέσματα 56

8.3 Περιορισμοί 57

8.4 Μελλοντικές Κατευθύνσεις 58

Βιβλιογραφία 60

Παράρτημα Ι: Επιλεγμένα Τμήματα Κώδικα 61

Λίστα Σχημάτων

Εικόνα 1- Διαδικασία ανίχνευσης ακμών με τον αλγόριθμο Canny σε πέντε διαδοχικά στάδια 6
Εικόνα 2- Παράδειγμα περιβάλλοντος στον προσομοιωτή CARLA 10
Εικόνα 3- Παράδειγμα περιβάλλοντος στον προσομοιωτή AirSim 11
Εικόνα 4- Διαδικασία εντοπισμού του χάσματος προσομοίωσης-πραγματικότητας 12
Εικόνα 5- Ροή Σταδίων Επεξεργασίας Εικόνας 16
Εικόνα 6 - Παράδειγμα καταγραφης δεδομένων σε CSV 18
Εικόνα 7- Σενάριο 1. Υπερβολικός φωτισμός και ηλιακή έκθεση 22
Εικόνα 8 - Σενάριο 2. Μέτριος φωτισμός. 23
Εικόνα 9 - Σενάριο 3. Κανονικές συνθήκες ημέρας. 23
Εικόνα 10 - Σενάριο 4. Βρεγμένος δρόμος. 24
Εικόνα 11 - Σενάριο 5. Έντονη σκιά. 24
Εικόνα 12 - Σενάριο 1. Κανονικές συνθήκες φωτισμού. 29
Εικόνα 13 - Σενάριο 2. Φωτισμός απο το εμπόδιο προς την κάμερα 29
Εικόνα 14 - Σενάριο 3. Φωτισμός πάνω στο εμπόδιο 30
Εικόνα 15 - Σενάριο 4. Φωτισμός πίσω από το εμπόδιο. 31
Εικόνα 16 - Raspberry Pi 4 Model B. Πηγή: DFRobot-“Raspberry Pi 4 Model B – 4 GB”, DFRobot.com. 35
Εικόνα 17 - L9110 H-Bridge Motor Driver. Πηγή: Robots Argentina-“Puente H: Placa controladora de motores L9110S”, Robots Didacticos, Robots-Argentina.com.ar. 36
Εικόνα 18 - Υπερηχητικός αισθητήρας HC-SR04. Πηγή: Elektronik-Kompendium- “HC-SR04-Ultrasonic Sensor (Ultraschallsensor)”, Elektronik-kompendium.de. 36
Εικόνα 19 - Ρομποτικό Οχημα που χρησιμοποιήθηκε 37
Εικόνα 20 - Συνδεσμολογία ρομποτικού οχήματος. 37
Εικόνα 21 - Παράδειγμα εικόνας του ρομποτικού οχήματος κατά την πορεία του στον διάδρομο κίνησης. 38
Εικόνα 22 - Παράδειγμα εικόνας γραφικού περιβάλλοντος για απεικόνιση και καταγραφή συμβάντων 40
Εικόνα 23 - Σενάριο 1. Έναρξη Ανίχνευσης (Φυσιολογικός Φωτισμός). 41
Εικόνα 24 - Σενάριο 1. Τέλος Ανίχνευσης (Φυσιολογικός Φωτισμός). 42
Εικόνα 25 - Σενάριο 2. κίτρινη επιφάνεια. Τέλος Ανίχνευσης (Φωτισμός Μπροστά), ... 42
Εικόνα 26 - Σενάριο 2. κίτρινη επιφάνεια. Έναρξη Ανίχνευσης (Φωτισμός Μπροστά) 42
Εικόνα 27 - Σενάριο 2, κόκκινη επιφάνεια. Τέλος Ανίχνευσης (Φωτισμός Μπροστά). 43
Εικόνα 28 - Σενάριο 2, κόκκινη επιφάνεια. Έναρξη Ανίχνευσης (Φωτισμός Μπροστά). 43
Εικόνα 29 - Σενάριο 3. Τέλος Ανίχνευσης (Φωτισμός απο πάνω) 44
Εικόνα 30 - Σενάριο 3. Έναρξη Ανίχνευσης (Φωτισμός από πανω) 44
Εικόνα 31 - Σενάριο 4. Τέλος Ανίχνευσης (Πλάγιος Φωτισμός) 45
Εικόνα 32 - Σενάριο 4. Έναρξη Ανίχνευσης (Πλάγιος Φωτισμός) 45
Εικόνα 33 - Σενάριο 5. Έναρξη Ανίχνευσης (Φωτισμός απο πίσω) 46
Εικόνα 34 - Σενάριο 5. Τέλος Ανίχνευσης (Φωτισμός απο πίσω) 46

Λίστα Πινάκων

Πίνακας 1- Κριτήρια ανά πλατφόρμα 17
Πίνακας 2 - Αποτελέσματα ανίχνευσης εμποδίων στον προσομοιωτή CARLA. 25
Πίνακας 3 - Αποτελέσματα ανίχνευσης εμποδίων στον προσομοιωτή AirSim 32
Πίνακας 4 - Συγκεντρωτικά αποτελέσματα ανίχνευσης εμποδίων στο Raspberry Pi. 47
Πίνακας 5 - Ανάλυση Απόδοσης Αλγορίθμου Ανίχνευσης Εμποδίων ανά Σενάριο και Προσομοιωτή. 52
Πίνακας 6 - Συνολική Επιτυχία Ανίχνευσης Εμποδίων στο Raspberry Pi 4 ανά Σενάριο. 53

Ακρωνύμια

AI	Artificial Intelligence
ML	Machine learning
CV	Computer Vision
ROS	Robot Operating System
CARLA	Car Learning to Act
AirSim	Aerial Informatics and Robotics Simulation
ROI	Region of Interest
API	Application Programming Interface
CNN	Convolutional Neural Network
LiDAR	Light Detection and Ranging
IMU	Inertial Measurement Unit
GPS	Global Positioning System
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη και αξιολόγηση μιας μεθοδολογίας ανίχνευσης εμποδίων. Επικεντρώνεται στη χρήση του αλγορίθμου Canny, διερευνώντας την ακρίβεια και την ανθεκτικότητά του σε συνθήκες μεταβαλλόμενου φωτισμού. Η μεθοδολογία αξιολογείται τόσο σε προσομοιωμένα περιβάλλοντα (CARLA και AirSim) όσο και σε πραγματικές υλοποιήσεις, με καταγραφή και ανάλυση ποσοτικών δεικτών απόδοσης.

Η εργασία εντιμετωπίζει προκλήσεις που σχετίζονται με τον θόρυβο, τις ψευδείς ανιχνεύσεις και τη μεταφορά των αποτελεσμάτων από προσομοίωση σε πραγματικές συνθήκες (Sim2Real Gap). Τα αποτελέσματα υπογραμμίζουν τη σημασία της πειραματικής επαλήθευσης και παρέχουν χρήσιμα συμπεράσματα για μελλοντικές βελτιώσεις αλγορίθμων ανίχνευσης και πλοήγησης σε αυτόνομα οχήματα.

Λέξεις Κλειδιά: *Αυτόνομη οδήγηση, ανίχνευση εμποδίων, αλγόριθμος Canny, Sim2Real, CARLA, AirSim, Raspberry Pi*

Abstract

This thesis focuses on the development and evaluation of an obstacle detection methodology. It emphasizes the use of the Canny algorithm, investigating its accuracy and robustness under varying lighting conditions. The methodology is evaluated both in simulated environments (CARLA and AirSim) and in real-word implementations, with the collection and analysis of quantitative performance metrics.

The work addresses challenges related to noise, false detections, and the transfer of results from simulation to real-world conditions (Sim2Real Gap). The results highlight the importance of experimental validation and provide valuable insights for future improvements in obstacle detection and navigation algorithms for autonomous vehicles.

Keywords: *Autonomous driving, obstacle detection, Canny algorithm, Sim2Real, CARLA, AirSim, Raspberry Pi*

1

Εισαγωγή

1.1 Αυτόνομη Οδήγηση και Υπολογιστική Όραση

Η αυτόνομη οδήγηση συγκαταλέγεται στις πλέον καινοτόμες εφαρμογές της σύγχρονης τεχνολογίας, συνδυάζοντας επιστήμες όπως η τεχνητή νοημοσύνη, η μηχανική μάθηση, η ρομποτική και η υπολογιστική όραση. Στόχος της είναι η ανάπτυξη οχημάτων ικανών να κινούνται με ασφάλεια και αποτελεσματικότητα, χωρίς την άμεση παρέμβαση του οδηγού. Η επιτυχής λειτουργία ενός αυτόνομου οχήματος προϋποθέτει την αντίληψη του περιβάλλοντος μέσω αισθητήρων, την κατανόηση της συνολικής κυκλοφοριακής κατάστασης, τη λήψη αποφάσεων και την εκτέλεση των κατάλληλων ενεργειών σε πραγματικό χρόνο.

Στον πυρήνα αυτής της διαδικασίας βρίσκεται η υπολογιστική όραση, η οποία επιτρέπει στα οχήματα να «βλέπουν» τον κόσμο και να αναγνωρίζουν κρίσιμα στοιχεία, όπως λωρίδες κυκλοφορίας, σήματα, πεζούς και εμπόδια. Οι κάμερες και οι αισθητήρες συλλέγουν δεδομένα που στη συνέχεια υφίστανται επεξεργασία μέσω αλγορίθμων, ώστε να εξαχθούν χρήσιμες πληροφορίες για την πλοήγηση. Τα συστήματα αυτά, πέρα από την τεχνική τους πολυπλοκότητα, οφείλουν να λειτουργούν αξιόπιστα σε ένα δυναμικό και συχνά απρόβλεπτο περιβάλλον, όπου οι συνθήκες φωτισμού, καιρικών φαινομένων και κυκλοφορίας μεταβάλλονται συνεχώς.

Παρά την εντυπωσιακή πρόοδο της τεχνολογίας, ο χώρος της αυτόνομης οδήγησης εξακολουθεί να αντιμετωπίζει κρίσιμες προκλήσεις. Μεταξύ αυτών συγκαταλέγεται η ακρίβεια στην αναγνώριση αντικειμένων υπό έντονες μεταβολές φωτισμού, η διαχείριση του θορύβου και των ψευδών ενδείξεων που μπορεί να προκύψουν από τους αισθητήρες, καθώς και το χάσμα μεταξύ ελεγχόμενων δοκιμών και πραγματικών συνθηκών οδήγησης. Τα προβλήματα αυτά αναδεικνύουν την ανάγκη για πιο αποδοτικές μεθόδους αντίληψης του περιβάλλοντος, οι οποίες θα μπορούν να διασφαλίσουν την σταθερότητα και την ασφάλεια των συστημάτων αυτόνομης πλοήγησης.

1.2 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία ασχολείται με την ανίχνευση εμποδίων σε συστήματα αυτόνομης οδήγησης, με έμφαση στη διαχείριση προβλημάτων που προκύπτουν σε συνθήκες υψηλής αντίθεσης φωτισμού. Το κύριο ζήτημα που επιχειρεί να αντιμετωπίσει είναι η αστάθεια των μεθόδων υπολογιστικής όρασης, οι οποίες συχνά αποτυγχάνουν να αναγνωρίσουν σωστά τα όρια αντικειμένων ή τα σήματα κυκλοφορίας όταν υπάρχουν έντονες εναλλαγές μεταξύ φωτεινών και σκοτεινών περιοχών. Το πρόβλημα αυτό είναι κρίσιμο για την ασφάλεια ενός αυτόνομου οχήματος, καθώς μπορεί να οδηγήσει σε καθυστερημένες ή λανθασμένες αντιδράσεις.

Για την επίλυση του προβλήματος, προτείνεται μια ενοποιημένη μεθοδολογία επεξεργασίας εικόνας, βασισμένη στον αλγόριθμο Canny για ανίχνευση ακμών. Η μεθοδολογία περιλαμβάνει στάδια προεπεξεργασίας εικόνας, εντοπισμού χαρακτηριστικών και λήψης απόφασης για την ύπαρξη ή μη εμποδίου. Η διαδικασία ενισχύεται με την ενσωμάτωση υπολογισμού απόστασης, ώστε να επιτυγχάνεται πιο αξιόπιστη διάκριση μεταξύ κρίσιμων και μη κρίσιμων καταστάσεων.

Η αξιολόγηση της μεθοδολογίας πραγματοποιείται σε τρία διαφορετικά περιβάλλοντα, στους προσομοιωτές CARLA και AirSim, καθώς και σε πραγματικό ρομποτικό όχημα με Raspberry Pi 4. Μέσα από τη συγκριτική ανάλυση των αποτελεσμάτων σε ελεγχόμενες αλλά και πραγματικές συνθήκες, επιδιώκεται η κατανόηση τόσο των πλεονεκτημάτων όσο και των περιορισμών της προτεινόμενης προσέγγισης.

1.2.1 Ερευνητικοί στόχοι και προκλήσεις

Κύριοι στόχοι της εργασίας είναι η ανάπτυξη μιας μεθοδολογίας ανίχνευσης εμποδίων που να είναι υπολογιστικά αποδοτική και εφαρμόσιμη σε συστήματα περιορισμένων πόρων, όπως το Raspberry Pi. Επιπλέον, επιδιώκεται η διερεύνηση της ακρίβειας και της ανθεκτικότητας του αλγορίθμου Canny σε συνθήκες μεταβαλλόμενου φωτισμού. Η εργασία στοχεύει επίσης στη σύγκριση της απόδοσης της μεθοδολογίας σε προσομοιωμένα περιβάλλοντα (CARLA, AirSim) και σε πραγματική υλοποίηση, καθώς και στην καταγραφή και ανάλυση δεδομένων, ώστε να εξαχθούν ποσοτικοί δείκτες απόδοσης, όπως precision, recall και επιτυχία ανίχνευσης.

Οι βασικές προκλήσεις που καλείται να αντιμετωπίσει η εργασία σχετίζονται με τον θόρυβο και τις ψευδείς ανιχνεύσεις σε περιβάλλοντα υψηλής αντίθεσης φωτισμού, τον περιορισμό των υπολογιστικών πόρων σε ενσωματωμένα συστήματα, καθώς και τη δυσκολία μεταφοράς των αποτελεσμάτων από προσομοίωση σε πραγματικές συνθήκες.

1.3 Δομή της διπλωματικής

Η δομή της διπλωματικής εργασίας περιλαμβάνει οκτώ κεφάλαια. Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο, με αναφορά σε βασικούς αλγορίθμους ανίχνευσης ακμών, στα προβλήματα που προκύπτουν από συνθήκες υψηλής αντίθεσης φωτισμού και στις δυνατότητες που προσφέρουν οι προσομοιωτές CARLA και AirSim. Το Κεφάλαιο 3 συζητά θέματα μοντελοποίησης, περιγράφοντας την προτεινόμενη μεθοδολογία ανίχνευσης, λήψης αποφάσεων και καταγραφής δεδομένων, η οποία εφαρμόζεται σε όλες τις πλατφόρμες αξιολόγησης. Στο Κεφάλαιο 4 αναπτύσσεται η εφαρμογή της μεθοδολογίας στον προσομοιωτή CARLA και παρουσιάζονται τα αντίστοιχα σενάρια και αποτελέσματα, ενώ στο Κεφάλαιο 5 γίνεται αντίστοιχη

ανάλυση με τον προσομοιωτή AirSim, περιγράφοντας τις δοκιμές και τις επιδόσεις του συστήματος. Το Κεφάλαιο 6 περιγράφει την υλοποίηση στο Raspberry Pi, τη συνδεσμολογία των αισθητήρων και τα αποτελέσματα από την πειραματική εκτέλεση σε πραγματικό ρομποτικό όχημα. Στο Κεφάλαιο 7 συγκεντρώνονται τα δεδομένα από όλες τις πλατφόρμες και πραγματοποιείται συγκριτική αξιολόγηση με τη χρήση μετρικών απόδοσης, ενώ το Κεφάλαιο 8 συνοψίζει τα συμπεράσματα της εργασίας, αναφέρει τους περιορισμούς που προέκυψαν και προτείνει μελλοντικές κατευθύνσεις έρευνας.

2

Θεωρητικό Υπόβαθρο και Σύγχρονες Τεχνολογίες στην Αυτόνομη Οδήγηση

2.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται η επιστημονική θεμελίωση της εργασίας, μέσω μιας εκτενούς επισκόπησης της σχετικής βιβλιογραφίας. Στόχος είναι να αποσαφηνστεί το θεωρητικό και τεχνολογικό υπόβαθρο που σχετίζεται με την ανίχνευση ακμών σε ψηφιακές εικόνες, τη χρήση τέτοιων μεθόδων σε εφαρμογές αυτόνομης οδήγησης, τις επιπτώσεις των συνθηκών φωτισμού στην απόδοση των συστημάτων υπολογιστικής όρασης, καθώς και τη χρήση προσομοιωτικών εργαλείων.

Ιδιαίτερη έμφαση δίνεται στον αλγόριθμο Canny, ο οποίος αποτελεί κλασική μέθοδο εντοπισμού ακμών και θεμελιώδη τεχνικήστην προεπεξεργασία εικόνων. Επιπλέον, αναλύονται τα προβλήματα που προκαλούν απο συνθήκες υψηλής αντίθεσης φωτισμού και παρουσιάζονται προησμένες μέθοδοι προσομοίωσης, όπως οι πλατφόρμες CARLA και AirSim, που χρησιμοποιούνται για την ανάπτυξη και αξιολόγηση αλγορίθμων σε ελεγχόμενο περιβάλλον. Τέλος, εξετάζεται το πρόβλημα του Sim2Real Gap, δηλαδη της διαφοράμεταξύ των προσομοιωμένων και των πραγματικών δεδομένων, και οι σύγχρονες τεχνικές γεφύρωσής του.

2.2 Ο αλγόριθμος Canny Edge Detection

Ο αλγόριθμος ανίχνευσης ακμών Canny, προτάθηκε από τον John F. Canny το 1986 (Canny, 1986), αποτελεί μία απο τις πιο αποτελεσματικές και διαδεδομένες τεχνικές στην επεξεργασία εικόνας. Η ανάπτυξη του βασίστηκε σε τρία θεμελιώδη κριτήρια βέλτιστης απόδοσης: (α) χαμηλό ποσοστό ψευδών εντοπισμών, δηλαδή η δυνατότητα ανίχνευσης όλων των σημαντικών ακμών της εικόνας με ταυτόχρονη ελαχιστοποίηση ψευδώς θετικών αποτελεσμάτων, (β) ακριβής χωρικός εντοπισμός των ακμών, ώστε η ανιχνευθείσα θέση τους να είναι όσο το δυνατόν πιο κοντά στη φυσική τους θέση, και (γ) μοναδική απόκριση, με στόχο η ίδια ακμή να ανιχνεύεται μία και μόνο φορά, αποφεύγοντας την εμφάνιση πολλαπλών παράλληλων γραμμών. Λόγω της ικανότητας του να ισορροπεί μεταξύ ευαισθησίας, ακρίβειας και ανθεκτικότητας στον θόρυβο, ο αλγόριθμός Canny έχει καθιερωθεί ως σημείο αναφοράς στην ανίχνευση ακμών(Sekehravani, Babulak and Masoodi, 2020).

Η διαδικασία του Canny εκτελείται σε πέντε διαδοχικά στάδια. Αρχικά, πραγματοποιείται εξομάλυνση της εικόνας με φιλτράρισμα Gaussian , καθώς η ανίχνευση ακμών είναι εξαιρετικά ευαίσθητη στον θόρυβο λόγω της χρήσης παραγώγων, οι οποίες ενισχύουν μικρές τοπικές διακυμάνσεις φωτεινότητας. Η Gaussian συνάρτηση δύο διαδτάσεων δίνεται από τη σχέση:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Όπου x και y είναι οι αποστάσεις από το κέντρο του πυρήνα και σ η τυπική απόκλιση, η οποία καθορίζει το επίπεδο εξομάλυνσης. Μικρές τιμές σ οδηγούν σε ήπια θόλωση, ενώ μεγαλύτερες τιμές προκαλούν ισχυρή θόλωση με συνέπεια τη μείωση λεπτομερειών(Sekehravani, Babulak and Masoodi, 2020).

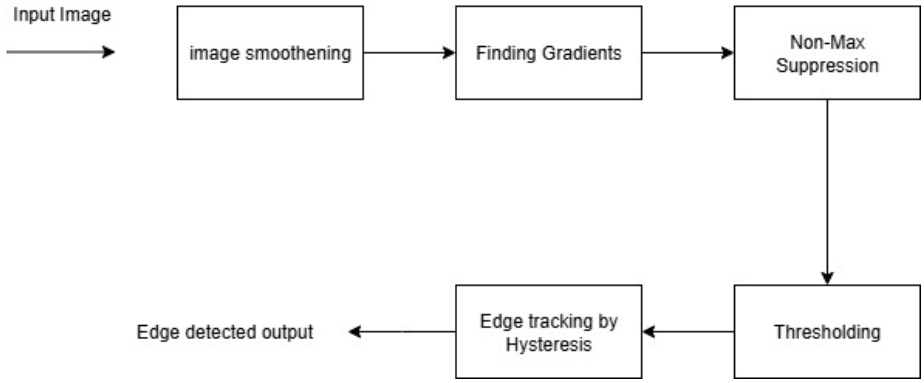
Στη συνέχεια, υπολογίζεται το διάνυσμα κλίσης (gradient) της εικόνας, το οποίο εκφράζει τον ρυθμό μεταβολής της φωτεινότητας και χρησιμοποιείται για την ανίχνευση περιοχών με έντονες αλλαγές φωτεινότητας, δηλαδή πιθανές ακμές. Οι οριζόντιες και κατακόρυφες συνιστώσες του gradient, G_x και G_y, μπορούν να υπολογιστούν με τη χρήση διακριτών τελεστών όπως οι Sobel(Sekehravani, Babulak and Masoodi, 2020):

$$G_x = \frac{\partial I}{\partial x}, \quad G_y = \frac{\partial I}{\partial y}$$

Το μέτρο της κλίσης προκύπτει από τη σχέση: $G = \sqrt{G_x^2 + G_y^2}$, ενώ η διεύθυνση της κλίσης δίνεται από την σχέση: $\theta = \arctan\left(\frac{G_y}{G_x}\right)$. Η γωνία θ στρογγυλοποιείται σε μία απο τις τέσσετις γωνίες $(0°, 45°, 90°, 135°)$ για να διευκολυνθεί η επόμενη φάση της διαδικασίας.

Ακολουθεί το τρίτο στάδιο, η καταστολή μη τοπικών μεγίστων (Non-Maximum Suppression), με σκοπό τη μείωση του πάχους των ακμών σε ένα μόνο pixel και την απομάκρυνση αδύναμων αποκρίσεων. Κάθε pixel συγκρίνεται με τα γειτονικά του κατά μήκος της κατεύθυνσης θ, και

διατηρείται μόνο αν αποτελεί τοπικό μέγιστο. Σε διαφορετική περίπτωση, η τιμή του μηδενίζεται(Sekehravani, Babulak and Masoodi, 2020).



Εικόνα 1- Διαδικασία ανίχνευσης ακμών με τον αλγόριθμο Canny σε πέντε διαδοχικά στάδια

Στο τέταρτο στάδιο εφαρμόζεται η μέθοδος διπλού κατωφλίου για τη διάκριση των ακμών σε ισχυρές, αδύναμες και μη ακμές. Εισάγονται δύο τιμές κατωφλίου, T_1 (υψηλό) και T_2 (χαμηλό) με $T_1 > T_2$. Αν $G > T_1$, το pixel χαρακτηρίζεται ως ισχυρή ακμή, αν $T_2 < G \leq T_1$ χαρακτηρίζεται ως αδύναμη ακμή, ενώ αν $G \leq T_2$ απορρίπτεται ως μη ακμή. Οι τιμές αυτές μπορεί να οριστούν εμπειρικά ή με αυτόματες μεθόδους όπως το Otsu thresholding.

Τέλος, εφαρμόζεται η διαδικασία ανίχνευσης ακμών μέσω υστέρησης (edge tracking by hysteresis). Οι ισχυρές ακμές διατηρούνται πάντα στο τελικό αποτέλεσμα, ενώ οι αδύναμες ακμές διατηρούνται μόνο εφόσος συνδέονται χωρικά με ισχυρές ακμές βάσει 8-γειτονικής συνδεσιμότητας. Με αυτόν τον τρόπο απομακρύνονται οι ασύνδετες ψευδο-ακμές και διατηρούνται μόνο οι ουσιαστικές δομές.

Ο αλγόριθμος Canny, παρά την υψηλή του ακρίβεια παρουσιάζει ορισμένους περιορισμούς. Η ευαισθησία του στον θόρυβο μπορεί να οδηγήσει σε απώλεια λεπτών ακμών σε περιοχές χαμηλού contrast, ενώ η χρήση σταθερών κατωφλίων μειώνει την προσαρμοστηκότητά του σε συνθήκες μεταβαλλόμενου φωτισμού. Σε περιβάλλοντα αυτόνομης οδήγησης, όπου η υψηλή αντίθεση φωτισμού (π.χ. μετάβαση από ηλιόλουστες σε σκιερές περιοχές, έντονα φώτα οχημάτων τη νύχτα) είναι συχνή, οι περιορισμοί αυτοί μπορούν να οδηγήσουν σε λανθασμένη ή ελλιπή ανίχνευση ορίων αντικειμένων και οδοσήμανσης. Αυτό ενδέχεται να επηρεάσει αρνητικά την ακρίβεια της αντίληψης του οχήματος και, κατ'επέκταση, την ασφάλεια πλοήγησης. Για την αντιμετώπιση των παραπάνω προβλημάτων, έχουν προταθεί διάφορες βελτιώσεις, όπως είναι η χρήση προσαρμοστικών κατωφλίων που μεταβάλλονται ανάλογα με τις τοπικές συνθήκες φωτισμού, η αντικατάσταση των τελεστών gradient με πιο ανθεκτικους σε θόρυβο και φωτιστικές μεταβολές υπολογισμούς, καθώς και η ενσωμάτωση τεχνικών μηχανικής μάθησης για την αυτόματη ρύθμιση των παραμέτρων(Canny, 1986; Sekehravani, Babulak and Masoodi, 2020).

Οι παραπάνω προκλήσεις αναδεικνύονται ακόμη περισσότερο όταν το σύστημα όρασης καλείται να λειτουργήσει σε συνθήκες υψηλής αντίθεσης φωτισμού, φαινόμενο που εμφανίζεται συχνά στην αυτόνομη οδήγηση. Η μετάβαση από έντονα φωτισμένες σε σκιερές περιοχές ή η παρουσία έντονων φωτεινών πηγώ κατά τη διαρκεια της νύχτας, δημιουργούν συνθήκες στις οποίες οι σταθερές παράμετροι του Canny ενδέχεται να αποτύχουν να προσαρμοστούν. Ως αποτέλεσμα, η

αξιοπιστία της ανίχνευσης ακμών μπορεί να μειωθεί σημαντικά. Για τον λόγο αυτό, η κατανόηση και η διαχείριση της υψηλής αντίθεσης φωτισμού αποτελούν κρίσιμους παράγοντες για την ανάπτυξη αποδοτικών συστημάτων όρασης υπολογιστή.

2.3 Υψηλή Αντίθεση Φωτισμού στην Αυτόνομη οδήγηση

Η υψηλή αντίθεση φωτισμού αποτελεί μία από τις σημαντικότερες προκλήσεις για τα συστήματα αυτόνομης οδήγησης, καθώς επηρεάζει άμεσα την ποιότητα των εικόνων που λαμβάνονται από τους αισθητήρες και την ακρίβεια των αλγορίθμων αναγνώρισης αντικειμένων. Σε περιβάλλοντα με ακραίες συνθληκες φωτισμού, όπως η έντονη ηλιακή ακτινοβολία, οι έντονες σκιές και οι νυχτερινές συνθήκες, τα αυτόνομα οχήματα ενδέχεται να δυσκολευτούν νς σντιληφθούν το περιβάλλον τους με ακρίβεια.

Ένα από τα βασικά προβλήματα που προκαλεί η υψηλή αντίθεση φωτισμού είναι η ύπαρξη υπερφωτισμένων και σκοτεινών περιοχών μέσα στην ίδια εικόνα. ΄Οταν ένα αντκείμενο βρίσκεται εν μέρει στη σκιά και εν μέρει εκτίθεται σε έντονο φωτισμό, οι αισθητήρες του οχήματος ενδέχεται να μην καταγράψουν σωστά τις λεπτομέρειές του, γεγονός που μπορεί να επηρεάσει την ανίχνευση πεζών, οχημάτων και σημάτων κυκλοφορίας. Επιπλέον, η έντονη ηλιακή ακτινοβολία μπορεί να προκαλέσει θάμπωση στις κάμερες, μειώνοντας την ικανότητα του οχήματος να αναγνωρίσει κρίσιμα στοιχεία, όπως διαβάσεις πεζών ή φωτεινούς σηματοδότες. Παράλληλα, οι ανακλάσεις από υγρές επιφάνειες ή μεταλλικά αντικείμενα μπορούν να παραπλανήσουν τα συστήματα όρασης, οδηγώντας σε λανθασμένες εκτιμήσεις απόστασης(Bouchner, Novotný and Piekník, 2007).

Η μειωμένη απόδοση των αισθητήρων σε συνθήκες χαμηλού φωτισμού είναι ένα ακόμα σημαντκό ζήτημα. Τη νύχτα, η διαφορά φωτεινότητας μεταξύ των φώτων του οχήματος και του περιβάλλοντος μπορει να περιορίσει την ικανότητά του να αντιληφθεί εμπόδια στον δρόμο. Επιπλέον, οι σκιές που δημιουργούνται από τα φώτα των οχημάτων ή από στοιχεία του περιβάλλοντος μπορούν να αποκρύψουν αντικείμενα, αυξάνοντας τον κίνδυνο ατυχημάτων.

Για την αντιμετώπιση των προκλήσεων που προκαλεί η υψηλή αντίθεση φωτισμού, έχουν αναπτυχθεί διάφορες τεχνικές. Μία από αυτές είναι η χρήση τεχνικών βελτίωσης της αντίθεσης στις εικόνες που λαμβάνουν οι αισθητήρες. Τεχνικές όπως Contrast limited Adaptive Histogram Equalization (CLAHE) μπορούν να βελτιώσουν την αντίθεση των εικόνων, διατηρώντας παράλληλα τη φυσικότητα των χρωμάτων και μειώνοντας τον θόρυβο(Yuan *et al.*, 2023), Παράλληλα, αλγόριθμοι Deep Learning μπορούν να εκπαιδευτούν ώστε να προσαρμόζουν δυναμικά τη φωτεινότηταα και την αντίθεση των εικόνων ανάλογα με τις συνθήκες φωτισμού.

Ένας άλλος αποτελεσματικός τρόπος βελτίωσης της αντίληψης του περιβάλλοντος από τα αυτόνομα οχήματα είναι η χρήση συνδυασμού αισθητήρων. Οι κάμερες ορατού φάσματος μπορούν να συνεργαστούν με LiDAR και υπέρυθρες κάμερες, επιτρέποντας πιο αξιόπιστη ανίχνευση αντικειμένων ανεξαρτήτως φωτισμού. Οι αισθητήρες ραντάρ, οι οποίοι δεν επηρεάζονται από τις αλλαγές του φωτισμού, μπορούν επίσης να παρέχουν συμπληρωματικές πληροφορίες για τον εντοπισμό εμποδίων. Τα προηγμένα συστήματα αντίληψης που βασίζονται σε τεχνητή νοημοσύνη επιτρέπουν στα συστήματα αυτόνομης οδήγησης να προβλέπουν και να αντισταθμίζουν τις επιπτώσεις της υψηλής αντίθεσης φωτισμού.

2.4 Συγκριτική Ανάλυση Μεθόδων Ανίχνευσης Ακμών

Η ανίχνευση ακμών αποτελεί κρίσιμο στάδιο στη επεξεργασία εικόνας και την υπολογιστική όραση, καθώς καθορίζει την ακρίβεια με την οποία προσδιορίζονται τα όρια και τα χαρακτηριστικά των αντικειμένων. Η απόδοση της διαδικασίας αυτής εξαρτάται σημαντικά από τον αλγόριθμο που θα χρησιμοποιηθεί, καθώς κάθε μέθοδος παρουσιάζει διαφορετική συμπεριφορά ως προς την ευαισθησία στον θόρυβο, την ακρίβεια αντοπισμού και το υπολογιστικό κόστος. Ο αλγόριθμος Canny έχει καθιερωθεί ως ένα από τα πιο αξιόπιστα εργαλεία, προσφέροντας καλή ισορροπία μεταξύ ευαισθησίας και ακρίβειας, καθώς και αντοχή σε θορυβώδη περιβάλλοντα. Ωστόσο, η πλήρης αξιολόγηση των δυνατοτήτων του απαιτεί σύγκριση με άλλες κλασικές και σύγχρονες προσεγγίσεις, όπως οι Sobel, Prewitt, Roberts Cross και Laplacian of Gaussian (LoG), αλλά και μεθόδους βασισμένες σε βαθιά μάθηση.

Ο τελεστής Sobel χρησιμοποιεί δύο μάσκες συνέλιξης για τον υπολογισμό των παραγώγων της εικόνας κατά τις κατευθύνσεις x και y. Οι μάσκε αυτές εφαρμόζονται μέσω της εξίσωσης $G = \sqrt{G_x^2 + G_y^2}$, όπου G_x και G_y αντιστοιχούν στις οριζόντιες και κατακόρυφες κλίσεις αντίστοιχα. Η μέθοδος αυτή είναι υπολογιστικά απλή και αρκετά γρήγορη, καθιστώντας την ιδανική για εφαρμογές σε πραγματικό χρόνο, ωστόσο παρουσιάζει αυξημένη ευαισθησία στον θόρυβο, γεγονός που μπορεί να οδηγήσει σε ψευδείς ανιχνεύσεις(Mande, Tripathy and Sarangi, 2015).

Ο Prewitt αποτελεί μια παραλλαγή του Sobel, με παρόμοια λογική υπολογισμού παραγώγων αλλά με διαφορετικές μάσκες, οι οποίες δίνουν ίση βαρύτητα σε όλα τα εικονοστοιχεία του παραθύρου. Αν και ο Prewitt είναι ακόμη πιο απλός στην υλοποίηση, η μειωμένη ακρίβεια αντοπισμού ακμών σε σχέση με τον Sobel τον καθιστά λιγότερο αποδοτικό σε εικόνες με λεπτές λεπτομέρειες. Παρόμοια, ο Roberts Cross χρησιμοποιεί μάσκες 2×2 για τον υπολογισμό της κλίσης, προσφέροντας υψηλή ακρίβεια σε λεπτές ακμές αλλά με σημαντική ευαισθησία στον θόρυβο, γεγονός που τον καθιστά κατάλληλο κυρίως για εικόνες χωρίς έντονα θορυβώδη στοιχεία(Mande, Tripathy and Sarangi, 2015).

Η μέθοδος Laplacian of Gaussian (LoG) βασίζεται στον υπολογισμό της δεύτερης παραγώγου της εικόνας, αφού πρώτα εφαρμοστεί Gaussian φίλτρο για την απομάκρυνση του θορύβου. Το LoG είναι ιδιαίτερα αποτελεσματικό στην ανίχνευση κλειστών περιγραμμάτων, ωστόσο η πολυπλοκότητα των υπολογισμών το καθιστά πιο αργό σε σύγκριση με τις απλούστερες μεθόδους. Από την άλλη πλευρά, οι μέθοδοι βαθιάς μάθησης αξιοποιούν συνελεκτικά νευρωνικά δίκτυα (CNNs) για να εκπαιδευτούν σε μεγάλα σύνολα δεδομένων, επιτυγχάνοντας υψηλή ακρίβεια και δυνατότητα ανίχνευσης ακμών ακόμη και σε δύσκολές συνθήκες φωτισμού ή θορύβου. Ωστόσο, απαιτούν σημαντικούς υπολογιστικούς πόρους και κατάλληλη εκπαίδευση για να αποδώσουν τα μέγιστα(*Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, 2009).

Συγκριτικά, ο Canny παραμένει σημείο αναφοράς λόγω της δυνατότητάς του να συνδυάζει χαμηλή πιθανότητα ψευδών ανιχνεύσεων με καλή ακρίβεια και αντοχή στον θόρυβο, ενώ οι Sobel και Prewitt επιλέγονται σε περιπτώσεις όπου η ταχύτητα είναι πιο σημαντική από την απόλυτη ακρίβεια. Ο Roberts Cross, παρότι απλός, έχει περιοσιμένη αντοχή στον θόρυβο, ενώ το LoG προσφέρει καλύτερα αποτελέσματα σε εφαρμογές όπου η αναγνώριση πλήρων περιγραμμάτων

είναι προτεραιότητα. Οι μέθοδοι βαθιάς μάθησης υπερέχουν σε ακρίβεια αλλά το αυξημένο υπολογιστικό κόστος περιορίζει τη χρήση τους σε συστήματα με ισχυρή υπολογιστική υποδομή.

2.5 Εισαγωγή στους Προσομοιωτές

Οι προσομοιωτές για τα αυτόνομα οχήματα αποτελούν βασικά εργαλεία για την ανάπτυξη και αξιολόγηση συστημάτων αυτόνομης οδήγησης. Επιτρέπουν την εκπαιδευτική διαδικασία και τη δοκιμή αλγορίθμων σε ελεγχόμενα, εικονικά περιβάλλοντα, αποφεύγοντας το κόστος και τους κινδύνους των φυσικών δοκιμών. Συγκεκριμένα, οι προσομοιωτές επιτρέπουν την αναπαραγωγή ρεαλιστικών σεναρίων, όπως ακραίες καιρικές συνθήκες, απρόβλεπτες αντιδράσεις άλλων οδηγών και σύνθετες καταστάσεις κυκλοφορίας, μειώνοντας τον κίνδυνο ατυχημάτων και εξασφαλίζοντας την ασφαλή αξιολόγηση των αλγορίθμων(Dosovitskiy *et al.*, 2017; Shah *et al.*, 2017).

Ένα από τα βασικά πλεονεκτήματα των προσομοιωτών είναι η δυνατότητα αναπαραγωγής των πειραμάτων. Αντίθετα με τον φυσικό κόσμο, όπου οι συνθήκες συχνά αλλάζουν, στους προσομοιωτές οι παράμετροι μπορούν να ρυθμιστούν με ακρίβεια, επιτρέποντας την επανάληψη των πειραμάτων με σταθερές συνθήκες και ακριβείς συγκρίσεις των αποτελεσμάτων.

Ωστόσο, παρά τα πλεονεκτήματα τους, παρουσιάζουν και ορισμένες προκλήσεις. Ένα από τα βασικά ζητήματα είναι η απόσταση μεταξύ των προσομοιωμένων και των πραγματικών συνθηκών (Sim2Real gap)(Dosovitskiy *et al.*, 2017; Li *et al.*, 2023). Η πλήρης αναπαράσταση του φυσικού κόσμου είναι εξαιρετικά δύσκολη, και κάποιες συνθήκες ή αντιδράσεις του περιβάλλοντος δεν μπορούν να προσομοιωθούν με ακρίβεια. Αυτή η διαφορά καθιστά την συνεχόμενη βελτίωση των προσομοιωτών απαραίτητη, προκειμένου να αυξηθεί η πιστότητά τους και να προσεγγιστούν καλύτερα οι πραγματικές συνθήκες οδήγησης(Li *et al.*, 2023)

Δυό από τους πιο προηγμένους και δημοφιλείς προσομοιωτές για αυτόνομα οχήματα είναι ο CARLA και ο AirSim. Αυτές οι πλατφόρμες προσφέρουν υψηλής ακρίβειας προσομοιώσεις και χρησιμοποιούνται ευρέως στην ακαδημαϊκή και βιομηχανική έρευνα(Dosovitskiy *et al.*, 2017; Shah *et al.*, 2017).

2.5.1 Προσομοιωτής CARLA

Ο προσομοιωτής CARLA (Car Learning to Act) είναι ένας προσομοιωτής ανοιχτού κώδικα, που σχεδιάστηκε για την ανάπτυξη, εκπαίδευση και δοκιμή συστημάτων αυτόνομης οδήγησης σε αστικά περιβάλλοντα. Ο προσομοιωτής προσφέρει ρεαλιστικό περιβάλλον προσομοίωσης με δωρεάν ψηφιακά δεδομένα, όπως τρισδιάστατες πόλεις, μοντέλα οχημάτων, πεζούς και σήματα κυκλοφορίας. Ο CARLA υποστηρίζει τη διαμόρφωση διαφόρων αισθητήρων, όπως κάμερες, LiDAR και GPS, προσφέροντας αναλυτικά δεδομένα σχετικά με την ταχύτητα, την επιτάχυνση, τις συγκρούσεις και τις παραβάσεις. Επιπλέον, παρέχει τη δυνατότητα προσαρμογής περιβαλλοντικών συνθηκών, όπως καιρικές αλλαγές και διαφορετικές ώρες της ημέρας. Οι καιρικές αλλαγές, όπως βροχή, ομίχλη ή ηλιοφάνεια, επηρεάζουν την απόδοση των αισθητήρων, ενώ οι διαφοροποιήσεις στη φωτεινότητα από την ημέρα στην νύχτα επηρεάζουν τη συμπεριφορά του οχήματος και των αισθητήρων του.

Ο προσομοιωτής CARLA υποστηρίζει τρεις κύριες προσεγγίσεις αξιολόγησης των συστημάτων αυτόνομης οδήγησης: την κλασική αρθρωτή προσέγγιση, που βασίζεται στην αντίληψη μέσω όρασης και στον σχεδιασμό πορείας, την εκμάθηση μέσω μίμησης (imitation

Εικόνα 2- Παράδειγμα περιβάλλοντος στον προσομοιωτή CARLA

learning), όπου ένα νευρωνικό δίκτυο μαθαίνει από δεδομένα ανθρώπινης οδήγησης, και την ενισχυτική μάθηση (reinforcement learning), που επιτρέπει στο σύστημα να μαθαίνει μέσω δοκιμών και λαμβάνει ανταμοιβές για σωστές ενέργειες. Η μηχανική προσομοίωση του CARLA στηρίζεται στην Unreal Engine 4 και λειτουργεί με αρχιτεκτονική πελάτη-εξυπηρετητή, επιτρέποντας πλήρη αλληλεπίδραση με το προσομοιωμένο περιβάλλον μέσω API υλοποιημένο σε Python.

Το περιβάλλον του CARLA αποτελείται από τρισδιάστατα μοντέλα που περιλαμβάνουν τόσο στατικά όσο και δυναμικά αντικείμενα. Στα στατικά συγκαταλέγονται κτίρια, σήματα κυκλοφορίας, βλάστηση και υποδομές, ενώ στα δυναμικά περιλαμβάνονται οχήματα και πεζούς. Οι αισθητήρες του συστήματος περιλαμβάνουν κάμερες RGB, αισθητήρες βάθους, αισθητήρες DVS (Dynamin Vision Sensor), και ψευδοαισθητήρες που παρέχουν σημασιολογική τμηματοποίηση του περιβάλλοντος. Αυτοί οι αισθητήρες παρέχουν δεδομένα για την κατάσταση του οχήματος και των γύρω αντικειμένων, όπως η θέση και η κατεύθυνση του οχήματος, οι κινήσεις του περιβάλλοντος, καθώς και η ανίχνευση εμποδίων και επικίνδυνων καταστάσεων(Dosovitskiy *et al.*, 2017).

2.5.2 Προσομοιωτής AirSim

Ο AirSim (Aerial Information and Robotics Simulation) είναι ένας προσομοιωτής ανοιχτού κώδικα για την ανάπτυξη και δοκιμή αυτόνομων οχημάτων, όπως drones και χερσαία ρομπότ. Χρησιμοποιεί τη μηχανή Unreal Engine για τη δημιουργία τρισδιάστατων ρεαλιστικών περιβαλλόντων, επιτρέποντας την εκπαίδευση αλγορίθμων μηχανικής μάθησης σε ελεγχόμενο και ασφαλές περιβάλλον.

Ο προσομοιωτής διαθέτει εξελιγμένα μοντέλα φυσικής που λαμβάνουν υπόψη αεροδυναμικές επιδράσεις, τριβή, καιρικές συνθήκες και την αλληλεπίδραση του οχήματος με το περιβάλλον. Επιπλέον, υποστηρίζει μια πληθώρα αισθητήρων, όπως κάμερες RGB, αισθητήρες βάθους, LiDAR, GPS, IMU και βαρόμετρο. Οι χρήστες μπορούν να προσαρμόσουν τις προσομοιώσεις, αλλάζοντας

περιβάλλοντα, προσθέτοντας αισθητήρες ή τροποποιώντας φυσικές παραμέτρους ώστε να ανταποκρίνονται σε πραγματικές εφαρμογές.

Εικόνα 3- Παράδειγμα περιβάλλοντος στον προσομοιωτή AirSim

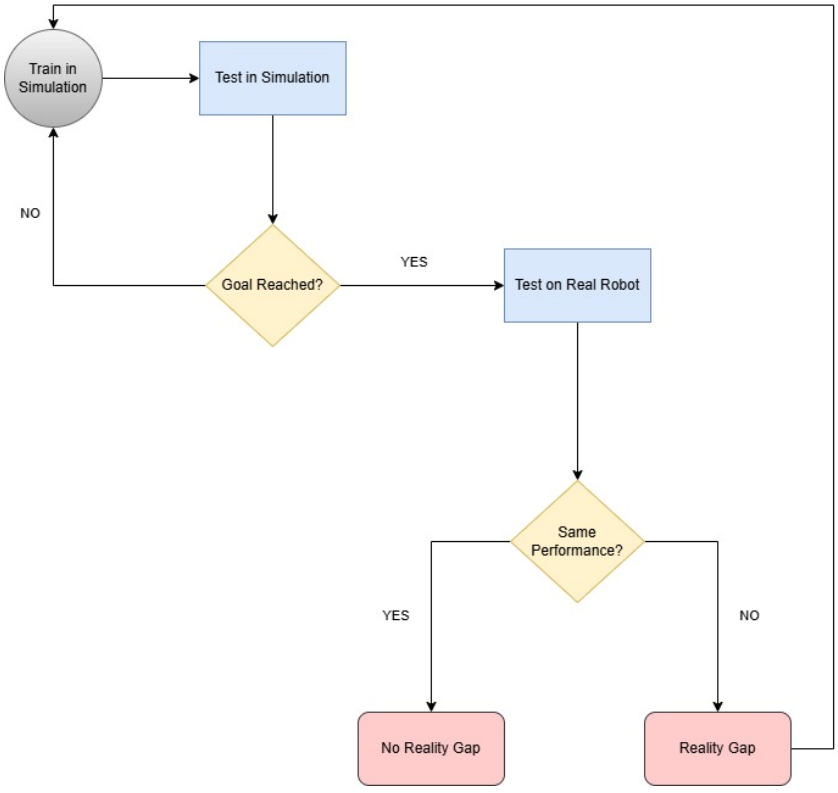
Επιπλέον, ο προσομοιωτής AirSim επιτρέπει τη δοκιμή αυτόνομων συστημάτων σε μεταβαλλόμενες συνθήκες, όπως διαφορετικοί τύποι εδάφους, εμπόδια και καιρικά φαινόμενα. Υποστηρίζει το πρωτόκολλο MavLnk και την ενσωμάτωσή του με το ROS (Robot Operating System), διευκολύνοντας τη διασύνδεση σε πραγματικά ρομπότ. Αυτό επιτρέπει τη μετάβαση από την προσομοίωση στην πραγματική δοκιμή με ελάχιστες αλλαγές στον κώδικα ελέγχου του οχήματος(Shah *et al.*, 2017).

Η προσομοίωση υποστηρίζει διαφορετικά είδη οχημάτων, όπως drones και αυτοκίνητα και παρέχει την δυνατότητα χειροκίνητου ελέγχου ή αυτοματοποιημένης πλοήγησης μέσω API σε C++ και Python. Ο προσομοιβτής περιλαμβάνει επίσης διάφορες οπτικές προβολές, όπως τρίτου προσώπου και πρώτου προσώπου (FPV).

2.6 Το Sim2Real Gap στην Αυτόνομη Οδήγηση

Το Sim2Real Gap αναφέρεται στη διαφορά απόδοσης που παρατηρείται όταν ένα σύστημα ελέγχου ή ένα μοντέλο, το οποίο έχει εκπαιδευτεί σε προσομοιωμένο περιβάλλον, εφαρμόζεται στον πραγματικό κόσμο. Το φαινόμενο αυτό είναι ιδιαίτερα κρίσιμο στη ρομποτική, όπου οι προσομοιωτές χρησιμοποιούνται για την εκπαίδευση ελεγκτών πριν από την εφαρμογή τους σε φυσικά ρομπότ.

Το κενό μεταξύ προσομοίωσης και πραγματικότητας προκύπτει από διάφορους παράγοντες. Πρώτον, τα ατελή φυσικά μοντέλα των προσομοιωτών συχνά απλοποιούν κρίσιμες φυσικές διεργασίες, όπως η τριβή, η παραμόρφωση και οι συγκρούσεις, με αποτέλεσμα μη ρεαλιστικές αναπαραστάσεις. Δεύτερον, ο θόρυβος στους αισθητήρες και του ενεργοποιητές προκαλεί ανακρίβειες στις μετρήσεις και καθυστερήσεις στις αντιδράσεις του συστήματος, κάτι που αποτυπώνεται επαρκώς στην προσομοίωση. Τέλος, η μεταβλητότητα του περιβάλλοντος είναι ένας σημαντικός παράγοντας, καθώς οι προσομοιωτές δυσκολεύονται να αναπαραγάγουν πραγματικέςε συνθήκες, όπως αλλαγές στον φωτισμό, τις επιφάνειες ή τις εξωτερικές παρεμβολές.



Εικόνα 4- Διαδικασία εντοπισμού του χάσματος προσομοίωσης-πραγματικότητας

Για την αξιολόγηση του Sim2Real Gap, ακολουθείται μια συγκεκριμένη διαδικασία, περιλαμβάνει την εκπαίδευση της πολιτικής ελέγχου σε προσομοιωμένο περιβάλλον, τη δοκιμή της στον ίδιο τον προσομοιωτή για επιβεβαίωση της αποτελεσματικότητάς της και εφόσον η απόδοση κρίνεται ικανοποοιητική, τη μεταφορά της στον πραγματικό ρομπότ. Στη συνέχεια, η απόδοση της αξιολογείται και συγκρίνεται με αυτή της προσομοίωσης, ώστε να εντοπιστούν τυχόν διαφορές. Αν παρατηρηθούν σημαντικές αποκλίσεις, είτε ο προσομοιωτής προσαρμόζεται ώστε να προσεγγίζει

καλύτερα την πραγματικότητα, είτε η πολιτική ελέγχου βελτιώνεται για να είναι πιο ανθεκτική στις πραγματικές συνθήκες(Salvato *et al.*, 2021).

Για να γεφυρωθεί το χάσμα μεταξύ προσομοίωσης και πραγματικότητας, έχουν αναπτυχθεί πολυδιάστατες προσεγγίσεις οι οποίες στοχεύουν στην αύξηση της γενικευσιμότητας και της ανθεκτικότητας των μοντέλων. Μία από τις πλέον καθιερωμένες στρατηγικές είναι η Τυχαιοποίηση Τομέα (Domain Randomization), η οποία εισάγει συστήματα στοχαστικές μεταβολές σε κρίσιμες παραμέτρους της προσομοίωσης, όπως η ένταση και κατεύθυνση φωτισμού, οι οπτικές υφές και η χωρική κατανομή των αντικειμένων. Μέσω αυτής της διαδικασίας, τα μοντέλα εκτίθενται σε ένα ευρύ φάσμα δυνατών καταστάσεων κατάτη διάρκεια της εκπαίδευσης, γεγονός που ενισχύει τη ικανότητά τους να αντιμετωπίζουν απρόβλεπτες συνθήκες σε πραγματικά περιβάλλοντα.

Συμπληρωματικά, η Μεταφορά Μάθησης (Transfer Learning - TL) αξιοποιεί τη γνώση που αποκτήθηκε σε προσομοιωμένα δεδομένα για να επιταχύνει και να βελτιώσει την προσαρμογή των μοντέλων σε πραγματικά δεδομένα. Η TL μειώνει σημαντικά την ανάγκη συλλογής και επεξεργασίας μεγάλων όγκων πραγματικών δεδομένων, τα οποία συχνά χαρακτηρίζονται από περιορισμένη διαθεσιμότητα και υψηλό κόστος. Στην πράξη, η TL λειτουργεί ως γέφυρα μεταξύ προσομοίωσης και πραγματικότητας, ενισχύοντας την αποτελεσματικότητα των μοντέλων και συνδέοντας άρρηκτα τις διαδικασίες DR κιαι την Προσαρμογή Δεδομένων (Domain Adaptation).

Η Domain Adaptation, επιδιώκει την ελαχιστοποίηση των στατιστικών και οπτικών διαφορών μεταξύ προσομοιωμένων και πραγματικών δεδομένων. Μέσω τεχνικών μετασχηματισμού και «μετάφρασης» των δεδομένων, επιτυγχάνεται η μείωση της απόκλισης στον χώρο χαρακτηριστικών, διευκολύνοντας την εφαρμογή των μοντέλων σε μη-αντιπροσωπευτικά πραγματικά σενάρια. Στο πλαίσιο αυτό, η Domain Adaptation λειτουργεί συμπληρωματικά προς τις DR και TL, καθώς η συστηματική τυχαιοποίηση και η μεταφορά γνώσης ενισχύονται όταν οι προσομοιωμένες αναπαραστάσεις πλησιάζουν περισσότερο τις πραγματικές.

Παράλληλα, η Συγχώνευση Αισθητήρων (Sensor Fusion) προσφέρει ένα επιπλέον επίπεδο αξιοπιστίας, ,καθώς ο συνδυασμός διαφορετικών πηγών πληροφορίας, όπως είναι οπτικά δεδομένα, LiDAR, radar και υπέρυθροι αισθητήρες, ελαχιστοποιεί τους περιορισμούς που συνδέονται με κάθε μεμονωμένο αισθητήρα. Η ενσωμάτωση πολυτροπικων δεδομένων βελτιώνει τη σταθερότητα των εκτιμήσεων και διευκολύνει την εφαρμογή των προηγούμενων τεχνικών, προσφέροντας ένα πλούσιο και διαφοροποιημένο σύνολο εισόδων για τη βελτιστοποίηση της γενικευσιμότητας.

Τέλος, η Ενισχυτική Μάθηση (Reinforcement learning - RL) αναδεικνύεται στη διεθνή βιβλιογραφία ως μια πολλά υποσχόμενη κατεύθυνση για την περαιτέρω ελαχιστοποίηση του χάσματος προσομοίωσης-πραγματικότητας. Μέσω πειραματισμού σε προσομοιωμένα περιβάλλοντα, οι αλγόριθμοι RL μπορούν να ανακαλύψουν στρατηγικές που έχουν υψηλή ικανότητα γενίκευσης στον πραγματικό κόσμο. Παρά το γεγονός ότι η παρούσα μελέτη δεν ενσωματώνει άμεσα αυτές τις προσεγγίσεις, η συνδυαστική εφαρμογή DR, TL, Domain Adaptation και Sensor Fusion μπορεί να αποτελέσει ένα ολοκληρωμένο πλαίσιο για τη βελτιστοποίηση της απόδοσης σε δυναμικά και αβέβαια περιβάλλοντα.

2.7 Συμπεράσματα

Το κεφάλαιο ανέδειξε τη σημασία της ανίχνευσης ακμών στην επεξεργασία εικόνας και την εφαρμογή της στα συστήματα αυτόνομης οδήγησης. Η ανίχνευση ακμών παρέχει κρίσιμες πληροφορίες για την αναγνώριση αντικειμένων, την παρακολούθηση λωρίδων κυκλοφορίας και τον εντοπισμό εμποδίων, αποτελώντας θεμελιώδη διαδικασία για αξιόπιστα οπτικά συστήματα. Ο αλγόριθμος Canny παραμένει σημείο αναφοράς, χάρη στην ικανότητά του να ισορροπεί μεταξύ ακριβείας, ανθεκτικότητας στον θόρυβο και σωστής χωρικής ανίχνευσης, ενώ οι φάσεις του, από το Gaussian smoothing μέχρι το edge tracking με hysteresis, προσφέρουν μια ολοκληρωμένη προσέγγιση που εξασφαλίζει αξιόπιστα αποτελέσματα ακόμα και σε θορυβώδεις εικόνες.

Παράλληλα, οι μεταβαλλόμενες συνθήκες φωτισμού, η χαμηλή αντίθεση και η παρουσία θορύβου επηρεάζουν σημαντικά την απόδοση των αλγορίθμων, γεγονός που καταδεικνύει την ανάγκη εφαρμογής τεχνικών προεπεξεργασίας και προσαρμοστικών κατωφλίων ώστε να διασφαλίζεται η σταθερότητα των αποτελεσμάτων σε διαφορετικά περιβάλλοντα. Η χρήση προσομοιωμένων περιβαλλόντων, όπως οι πλατφόρμες CARLA και AirSim, επιτρέπει ασφαλή και ελεγχόμενη ανάπτυξη αλγορίθμων ανίχνευσης ακμών, ενώ η μετάβαση από προσομοίωση σε πραγματικά δεδομένα, γνωστή ως Sim2Real Gap, απαιτεί ειδικές τεχνικές προσαρμογής ώστε τα μοντέλα να παραμένουν αξιόππιστα σε πραγματικές συνθήκες οδήγησης.

3

Μεθοδολογία Ανίχνευσης, Απόφασης και Καταγραφής Δεδομένων

3.1 Εισαγωγή

Η ανίχνευση εμποδίων αποτελεί κρίσιμο υποσύστημα σε αυτόνομα οχήματα, καθώς σχετίζεται άμεσα με την ασφάλεια και την ικανότητά τους να αντιδρούν σε δυναμικά περιβάλλοντα. Στην παρούσα εργασία, υιοθετείται μία ενιαία απλή αλλά αποδοτική μεθοδολογία ανίχνευσης βασισμένη σε επεξεργασία εικόνας και εντοπισμό ακμών μέσω του αλγορίθμου Canny. Η ίδια μεθοδολογία εφαρμόζεται και στις τρεις πλατφόρμες υλοποίησης – στους προσομοιωτές CARLA και AirSim, καθώς και στο πραγματικό σύστημα με Raspberry Pi 4, ώστε να επιτυγχάνεται άμεση συγκρισιμότητα των αποτελεσμάτων και αποτίμηση της επίδοσης υπό διαφορετικές συνθήκες φωτισμού και λειτουργίας.

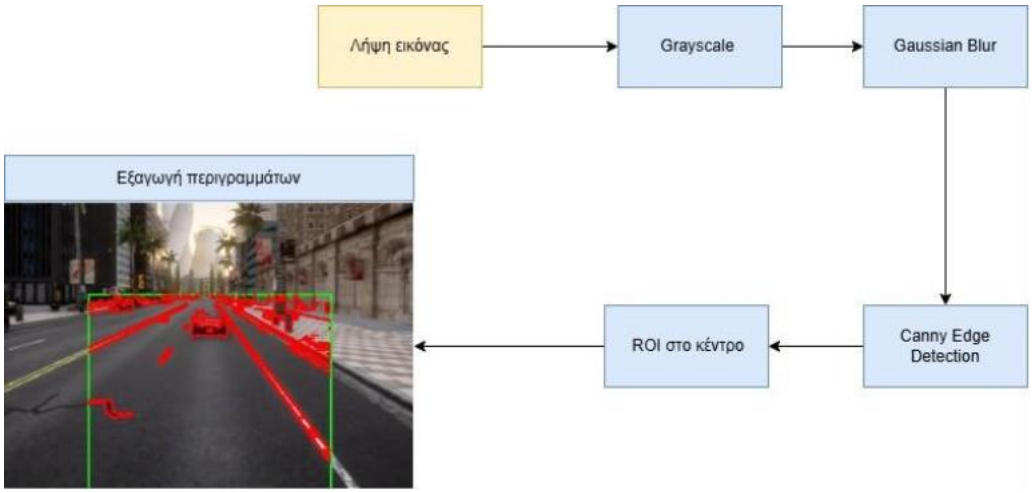
Η βασική αρχή είναι η εξαγωγή δοκιμών χαρακτηριστικών από την είσοδο εικόνας και η ερμηνεία τους για την απόφαση “STOP” ή “CONTINUE”. Η διαδικασία ανίχνευσης εκτελείται ανά καρέ (frame) και περιλαμβάνει στάδια προεπεξεργασίας, εντοπισμού ακμών, ανάλυσης περιοχής ενδιαφέροντος και απόφασης βάσει γεωμετρικών χαρακτηριστικών των ανιχνευθέντων στοιχείων.

3.2 Στάδια Επεξεργασίας Εικόνας

Η αλυσίδα επεξεργασίας ξεκινά με τη μετατροπή της εισερχόμενης εικόνα σε κλίμακα του γκρι (grayscale). Με αυτόν τον τρόπο εξαλείφονται τα χρωματικά κανάλια και διατηρείται μόνο η πληροφορία φωτεινότητας, που είναι κρίσιμη για την ανίχνευση ακμών. Στην συνέχεια, εφαρμόζεται φιλτράρισμα Gauss (Gaussian blur) για μείωση του θορύβου της εικόνας και εξομάλυνση, αποτρέποντας την παραγωγή ψευδών ακμών.

Το τρίτο στάδιο είναι η ανίχνευση ακμών μέσω του αλγορίθμου Canny. Ο αλγόριθμος Canny είναι ένας πολυσταδιακός αλγόριθμος, ο οποίος περιλαμβάνει υπολογισμό των τοπικών κλίσεων φωτεινότητας, καταστολή μη μέγιστων τιμών (non-maximum suppression), καθώς και διπλό κατώφλι (hysteresis thresholding) για την επιλογή των ισχυρότερων ακμών. Ο αλγόριθμος αυτός παρουσιάζει καλή ανθεκτικότητα στον θόρυβο και είναι κατάλληλος για εφαρμογές πραγματικού χρόνου σε ενσωματωμένα συστήματα.

Μετά την ανίχνευση των ακμών, ορίζεται μια περιοχή ενδιαφέροντος (Region of Interest - ROI) στο κατώτερο και κεντρικό τμήμα της εικόνας. Η επιλογή αυτής της περιοχής βασίζεται στην παραδοχή ότι τα εμπόδια που είναι εγγύτερα στο όχημα και χρήζουν άμεσης αντίδρασης θα εμφανιστούν πρώτα σε αυτό το οπτικό πεδίο. Η ανάλυση περιορίζεται σε αυτήν την περιοχή για μείωση του υπολογιστικού φόρτου και εστίαση σε κρίσιμες ζώνες.



Εικόνα 5- Ροή Σταδίων Επεξεργασίας Εικόνας

Μέσα στο ROI εφαρμόζεται ανάλυση πεεριγραμμάτων (contour detection), η οποία εντοπίζει κλειστά σύνολα ακμών. Για κάθε περίγραμμα που εντοπίζεται, υπολογίζεται το εμβαδόν του. Εάν διαπιστωθεί ότι το εμβαδόν ενός περιγράμματος υπερβαίνει το κατώφλι των 600 εικονοστοιχείων, το σύστημα καταγράφει την παρουσία εμποδίου (ObstacleDetected = True).

3.3 Απόσταση και Κριτήρια Απόφασης

Η τελική απόφαση για την αντίδραση του οχήματος βασίζεται όχι μόνο στην οπτική ένδειξη της ύπαρξης εμποδίου, αλλά καια στη μέτρηση της απόστασης από αυτό. Στους προσομοιωτές CARLA και AirSim, η απόσταση εξάγεται μέσω των εργαλείων API που παρέχουν τις συντεταγμένες των αντικειμένων ή της σκηνής. Στο Raspberry Pi, η απόσταση υπολογίζεται με τη χρήση υπερηχητικού αισθητήρα (HC-SR04), ο οποίος μετρά το χρόνο ανακλώμενου παλμού και επιστρέφει την απόσταση σε μέτρα.

Ο παρακάτω πίνακας συνοψίζει τον τρόπο υπολογισμού της απόστασης και την τιμή του κρίσιμου ορίου απόφασης d_crit ανά πλατφόρμα:

Πίνακας 1- Κριτήρια ανά πλατφόρμα

Πλατφόρμα	Μέθοδος απόστασης	Τιμή d_crit
CARLA	Συντεταγμένες μέσω API	10m
AirSim	Συντεταγμένες μέσω API	10m
Raspberry Pi	Αισθητήρας HC-SR04	0.15m

Η λογική λήψης απόφασης ακολουθεί τα παρακάτω κριτήρια:

- Αν ανιχνεύεται εμπόδιο (ObstacleDetected = True) και η απόσταση είναι μικρότερη από μία κρίσιμη τιμή d_crit, τότε το όχημα εισέρχεται σε κατάσταση STOP, με ενεργοποίηση φρεναρίσματος ή διακοπή του ελέγχου κίνησης.
- Αν δεν εντοπιστεί εμπόδιο ή η απόσταση υπερβαίνει το d_crit, τότε το όχημα συνεχίζει την πορεία του χωρίς τροποποίηση (κατάσταση CONTINUE).

Η τιμή d_crit επιλέγεται εμπειρικά και προσαρμόζεται στην κλίμακα και στο περιβάλλον λειτουργίας κάθε πλατφόρμας. Συγκεκριμένα, για τις προσομοιώσεις στους CARLA και AirSim ορίστηκε ως 10 μέτρα, ενώ στο πραγματικό σύστημα Raspberry Pi επιλέχθηκε τιμή 0.15 μέτρων, λόγω της μικρής κλίμακας του οχήματος και των περιορισμένων δυνατοτήτων του αισθητήρα.

3.4 Καταγραφή Δεδομένων για Επαλήθευση και Ανάλυση

Κάθε απόφαση του συστήματος συνοδεύεται από καταγραφή των σχετικών μεταβλητών σε αρχείο μορφής CSV. Τα πεδία περιλαμβάνουν: χρονική σήμανση, υπολογιζόμενη απόσταση, ύπαρξη εμποδίου, καθώς και την ενέργεια που εφαρμόστηκε (STOP ή CONTINUE). Επιπλέον, σε περιπτώσεις όπου εντοπίζεται εμπόδιο, αποθηκεύεται και το αντίστοιχο καρέ της εικόνας, ώστε να είναι δυνατή η μεταγενέστερη ανάλυση αποτυχιών ή επιτυχίων και η αξιολόγηση των επιδόσεων του συστήματος με τη χρήση δεικτών όπως το precision και το recall, όπως παρουσιάζεται αναλυτικά στο Κεφάλαιο 7.

1	Scenario	Distance	ObstacleDetected	Reaction
2	scenario_1on	15.037847518920898	False	FORWARD
3	scenario_1on	15.031586647033691	False	FORWARD
4	scenario_1on	15.027655601501465	False	FORWARD
5	scenario_1on	15.023375511169434	False	FORWARD
6	scenario_1on	15.018716812133789	False	FORWARD
7	scenario_1on	15.013895034790039	False	FORWARD
8	scenario_1on	15.00805950164795	False	FORWARD
9	scenario_1on	15.002678871154785	False	FORWARD
10	scenario_1on	14.99699592590332	False	FORWARD
11	scenario_1on	14.991458892822266	False	FORWARD
12	scenario_1on	14.985694885253906	False	FORWARD
13	scenario_1on	14.980377197265625	False	FORWARD
14	scenario_1on	14.975253105163574	False	FORWARD

Εικόνα 6 - Παράδειγμα καταγραφης δεδομένων σε CSV

Η μεθοδολογία αυτή έχει το πλεονέκτημα ότι είναι υπολογιστικά ελαφριά, επομένως εφαρμόσιμη σε συστήματα με περιορισμένους πόρους, όπως το Raspberry Pi, ενώ παράλληλα είναι αρκετά γενική και επεκτάσιμη, ώστε να προσαρμόζεται σε πιο πολύπλοκα σενάρια με χρήση επιπλέον αισθητήρων ή προηγμένων μοντέλων αναγνώρισης.

3.5 Συμπεράσματα

Στο κεφάλαιο αυτο παρουσιάστηκε μια ολοκληρωμένη μεθοδολογία για την ανίχνευση εμποδίων, τη λήψη αποφάσεων και την καταγραφή δεδομένων σε αυτόνομα οχήματα, η οποία συνδυάζει τεχνικές επεξεργασίας εικόνας με μετρήσεις απόστασης. Η ανάλυση των σταδίων επεξεργασίας δείχνει ότι ο περιορισμό της ανάλυσης σε μια καλά ορισμένη περιοχή ενδιαφέροντος μειώνει σημαντικά τον υπολογιστικό φόρτο χωρίς να χάνεται κρίσιμη πληροφορία για την ασφάλεια του οχήματος. Η καταγραφή των δεδομένων σε μορφή CSV, παράλληλα με τα αντίστοιχα καρέ εικόνας, δημιουργεί ένα πλήρες αρχείο για επαλήθευση, ανάλυση αποτυχιών και αξιολόγηση της επίδοσης του συστήματος με δείκτες όπως precision και recall. Η προτεινόμενη μεθοδολογία ειναι υπολογιστικά ελαφριά, γενική και επεκτάσιμη, καθιστώντας την κατάλληλη τόσο για ενσωματωμένα συστήματα χαμηλού κόστους, όπως το Raspberry Pi, όσο και για πιο σύνθετες εφαρμογές με πολλαπλούς αισθητήρες ή προηγμένα μοντέα αναγνώρισης.

4

Προσομοίωση στο CARLA

4.1 Εισαγωγή

Η πρώτη φάση αξιολόγησης της μεθόδου ανίχνευσης εμποδίων πραγματοποιήθηκε στον προσομοιωτή CARLA. Ο προσομοιωτής αυτός επιλέχθηκε λόγω της υψηλής πιστότητας που προσφέρει, επιτρέποντας την αναπαραγωγή ρεαλιστικών συνθηκών οδήγησης, όπως η παρουσία άλλων οχημάτων, πεζών και ποικίλων εμποδίων, χωρίς τον κίνδυνο που συνεπάγεται η δοκιμή σε πραγματικό περιβάλλον. Η χρίση προσομοίωσης δίνει τη δυνατότητα ελέγχου και απαναληψιμότητας, καθιστώντας δυνατή τη συστηματική αξιολόγηση της απόδοσης του αλγορίθμου υπό διαφορετικές συνθήκες φωτισμού, υγρασίας και σκιάς.

Στο πλαίσιο της παρούσας αξιολόγησης, η ανίχνευση εμποδίων εξετάστηκε σε ένα συγκεκριμένο χάρτη, τον Town03, με τη χρήση οχήματος τύπου Tesla Model 3 εξοπλισμένου με εμπρόσθια RGB κάμερα. Η επιλογή αυτή διασφαλίζει την ομοιομορφία των δεδομένων και περιορίζει την επίδραση τυχαίων παραγόντων στην αξιολόγηση. Δημιουργήθηκαν πέντε διαφορετικά σενάρια, τα οποία αναπαριστούν συνηθισμένες αλλά και ακραίες καταστάσεις οδήγησης, όπως υπερέκθεση φωτισμού, έντονο glare, βρεγμένο οδόστρωμα, έντονες σκιές και συνδυασμό πολλών παραγόντων ταυτόχρονα. Κάθε σενάριο επιτρέπει την ανάλυση της ευαισθησίας του αλγορίθμου σε περιβαλλοντικές μεταβολές και βοηθά στον εντοπισμό περιοχών που χρήζουν βελτίωσης.

Η αξιολόγηση μέσω προσομοίωσης παρέχει τόσο ποσοτικά όσο και ποιοτικά δεδομένα για την απόδοση της μεθόδου. Επιτρέπεθ τη μέτρηση ακρίβειας στην ανίχνευση εμποδίων και ταυτόχρονα την παρατήρηση συμπεριφοράς του αλγορίθμου σε συνθήκες που θα ήταν δύσκολο ή επικίνδυνο να αναπαραχθούν στον πραγματικό κόσμο. Μέσω αυτής της διαδικασίας, μπορούν να εντοπιστούν προβληματικά σενάρια, να προταθούν βελτιώσεις και να αυξηθεί ξ αξιοπιστίς του συστήματος πριν την εφαρμογή του σε πραγματικό όχημα.

4.2 Περιγραφή Περιβάλλοντος CARLA

Η πειραματική αξιολόγηση σε περιβάλλον προσομοίωσης πραγματοποιήθηκε με χρήση του CARLA Simulator, στην έκδοση 0.9.15, η οποία παρέχει προηγμένες δυνατότητες για την ανάπτυξη και δοκιμή αλγορίθμων αυτόνομης οδήγησης. Ως πεδίο δοκιμών επιλέγθηκε ο χάρτης Town03, ο οποίος διαθέτει εκτεταμένα ευθύγραμμα τμήματα οδού, ελαχιστοποιώντας την επιρροή παραμέτρων όπως στροφές ή κυκλοφορία, και επιτρέποντας την απομόνωση του προβλήματος ανίχνευσης εμποδίων.

Για τις ανάγκες της πειραματικής διάταξης, χρησιμοποιήθηκε το όχημα τύπου vehicle.tesla.model3, στο οποίο προσαρτήθηκε εικονικός αισθητήρας RGB κάμερας. Η κάμερα τοποθετήθηκε εμπρόσθια στο όχημα, σε ύψος περίπου 1.6 μέτρων από το οδόστρωμα, με οπτικό πεδίο (FOV) 90°, προσομοιώνοντας ρεαλιστικά την ανθρώπινη οπτική γωνία. Η ανάλυση των καρέ ορίστηκε στα 640×480 pixels, με ρυθμό λήψης 30 fps, τιμές που εξισορροπούν την ποιότητα εικόνας με την ανάγκη για υπολογιστική αποδοτικότητα.

Κατά τη διάρκεια των δοκιμών, το όχημα κινούνταν ευθύγραμμα με σταθερή ταχύτητα, χωρίς να εφαρμόζεται έλεγχος κατεύθυνσης μέσω συστήματος PID ή steering controller. Η απουσία ενεργού ελέγχου της πορείας εξασφαλίζει ότι τυχόν αλλαγές στην απόσταση από το εμπόδιο οφείλονται αποκλειστικά στην λειτουργία του συστήματος ανίχνευσης και όχι σε δυναμικές παρεμβάσεις στο τιμόνι.

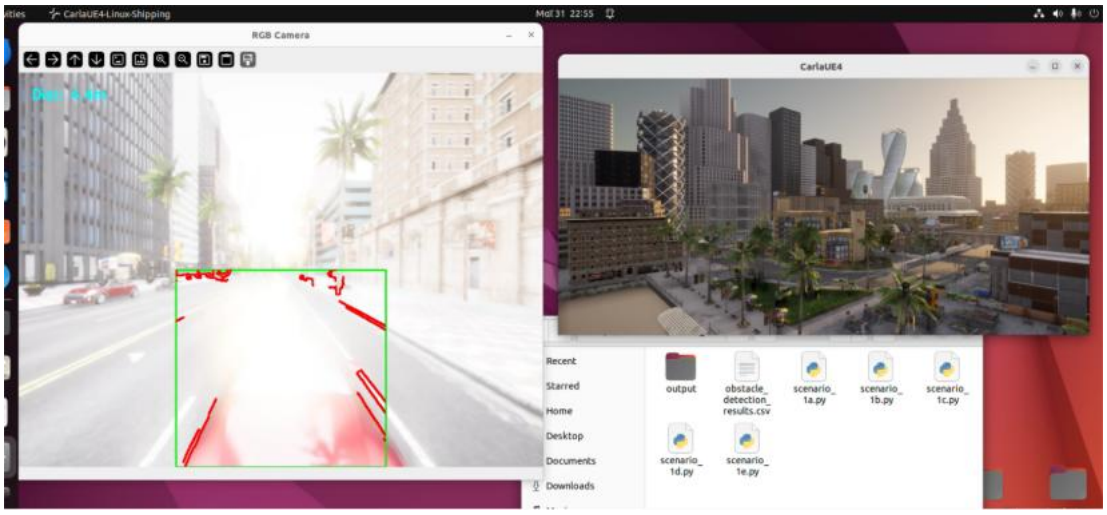
Για την προσομοίωση της παρουσίας εμποδίου, τοποθετήθηκε σταθερά μπροστά στο όχημα ένα αντικείμενο τύπου static.prop.streebarrier, το οποίο διαθέτει σαφή γεωμετρική μορφή και καθαρές ακμές, καθιστώντας το κατάλληλο για ανίχνευση μέσω επεξεργασίας εικόνας. Η μέθοδος ανίχνευσης εφαρμόστηκε σε πραγματικό χρόνο επί της εικόνας της κάμερας, με στόχο να αξιολογηθεί η ικανότητα εντοπισμού εμποδίου υπό μεταβαλλόμενες φωτιστικές συνθήκες και χωρίς εξωτερική πληροφορία θέσης ή βάθους.

4.3 Περιγραφή Σεναρίων

Για την αξιολόγηση της μεθόδου, σχεδιάστηκαν πέντε διαφορετικά σενάρια, καθένα από τα οποία διαφοροποιεί συγκεκριμένες παραμέτρους φωτισμού και περιβάλλοντος, με στόχο να μελετηθεί η ανθεκτικότητα της ανίχνευσης εμποδίου σε συνθήκες υψηλής ή χαμηλής φωτεινότητας.

4.3.1 Σενάριο 1 – Υπερβολικός φωτισμός & Ηλιακή έκθεση

Σε αυτό το σενάριο, το όχημα χρησιμοποιεί πλήρη φωτισμό: φώτα θέσης, μεσαίας και μεγάλης σκάλας, καθώς και φώτα φρένου. Η κάμερα λειτουργεί με υπερέκθεση (exposure 4.0), υψηλή ένταση στο φαινόμενο bloom και ενεργό lens flare, δημιουργόντας συνθήκες έντονης οπτικής παραμόφωσης. Ο ήλιος βρίσκεται πολύ χαμηλά στον ορίζοντα (10°) και πίσω από το όχημα, προκαλώντας ένα ισχυρό φαινόμενο glare, το οποίο συνοδεύεται από τεχνητή φωρεινότητα 100.00 lux.



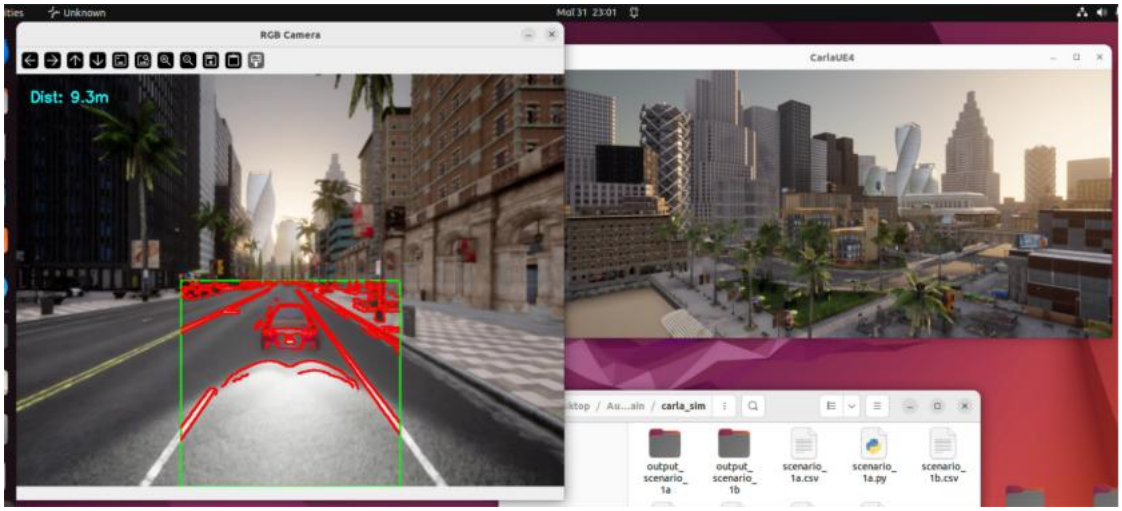
Εικόνα 7- Σενάριο 1. Υπερβολικός φωτισμός και ηλιακή έκθεση

Υπό αυτές τις συνθήκες, η αντίδραση του οχήματος εξαρτάται από την ικανότητα ανίχνευσης περιγραμμάτων σε κοντινή απόσταση. Αναμενόμενο αποτέλεσμα είναι η δυσκολία στην αναγνώριση του εμποδίου λόγω υπερέκθεσης της εικόνας με το glare να μειώνει σημαντικά την ορατότητα της κάμερας.

4.3.2 Σενάριο 2 – Μέτριος φωτισμός

Το σενάριο αυτό αφορά φωτισμό υπό κανονικές συνθήκες. Όλα τα φώτα του οχήματος είναι ενεργά, χωρίς όμως υπερέκθεση. Η κάμερα χρησιμοποιεί χαμηλή ένταση στο bloom και δεν υπάρχουν

πρόσθετα εφέ lens flare ή χειρισμοί στο exsposure. Οι καιρικές συνθήκες περιλαμβάνουν ήπια ηλιακή ακτινοβολία (περίπου 500 lux) με τον ήλιο χαμηλά στον ουρανό.

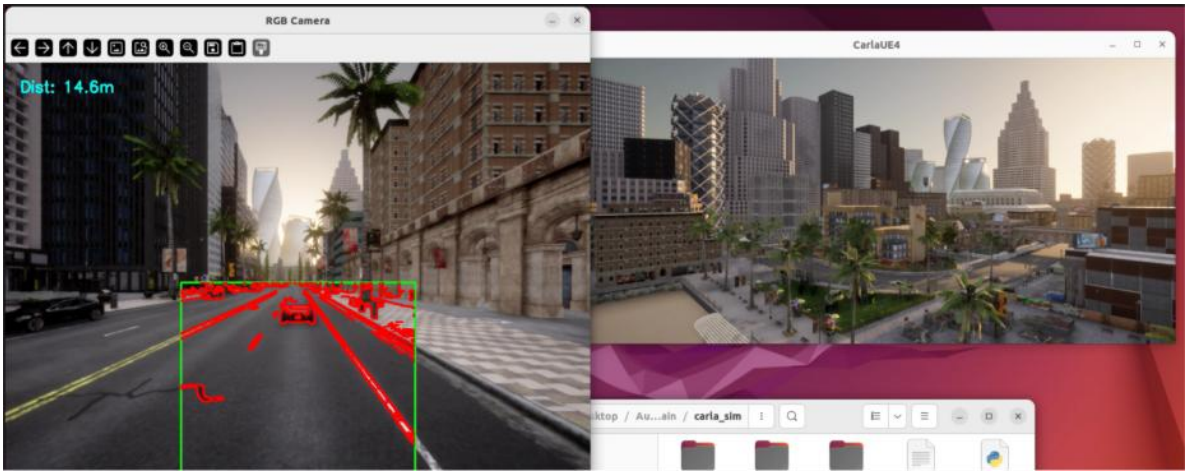


Εικόνα 8 - Σενάριο 2. Μέτριος φωτισμός.

Αυτό το περιβάλλον προσφέρει έναν ισορροπημένο φωτισμό, ο οποίος επιτρέπει στο σύστημα να ανιχνεύσει με μεγαλύτερη αξιοπιστία τα περιγράμματα των εμποδίων.

4.3.3 Σενάριο 3 – Κανονικές συνθήκες ημέρας

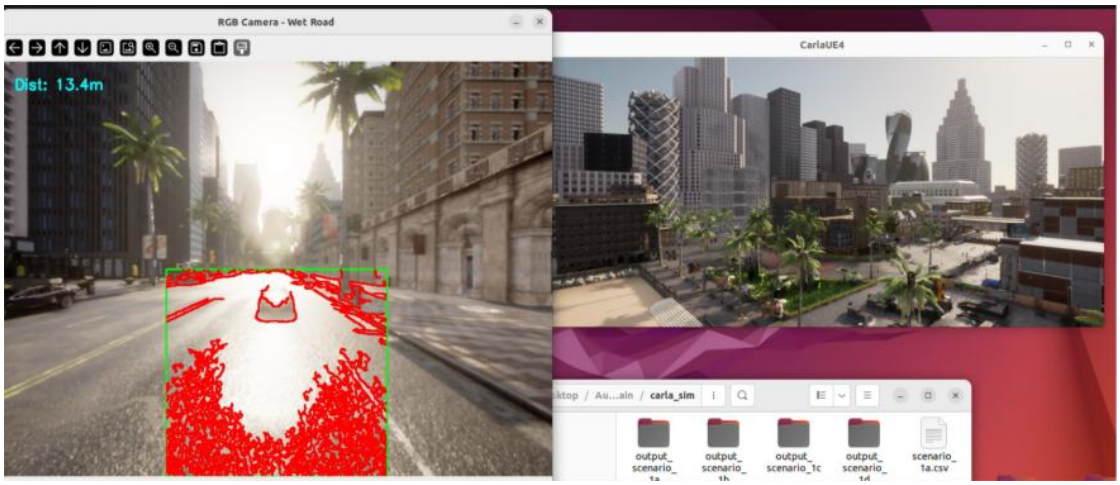
Σε αυτό το σενάριο, όλα τα φώτα του οχήματος είναι απενεργοποιημένα. Η κάμερα χρησιμοποιεί τις ίδιες ρυθμίσεις με το Σενάριο 2, χωρίς πρόσθετα εφέ, όπως lens flare ή αυξημένο exposure. Οι καιρικές συνθήκες είναι επίσης παρόμοιες, με φυσική φωτεινότητα και χαμηλό ήλιο. Παρά την απουσία τεχνητού φωτισμού, η φυσική ηλιακή ακτινοβολία θεωρείται επαρκής για την ανίχνευση του εμποδίου και το σύστημα αναμένεται να λειτουργεί με ακρίβεια, αφού δεν υπάρχουν παραμορφώσεις από τεχνητά εφέ.



Εικόνα 9 - Σενάριο 3. Κανονικές συνθήκες ημέρας.

4.3.4 Σενάριο 4 – Βρεγμένος δρόμος

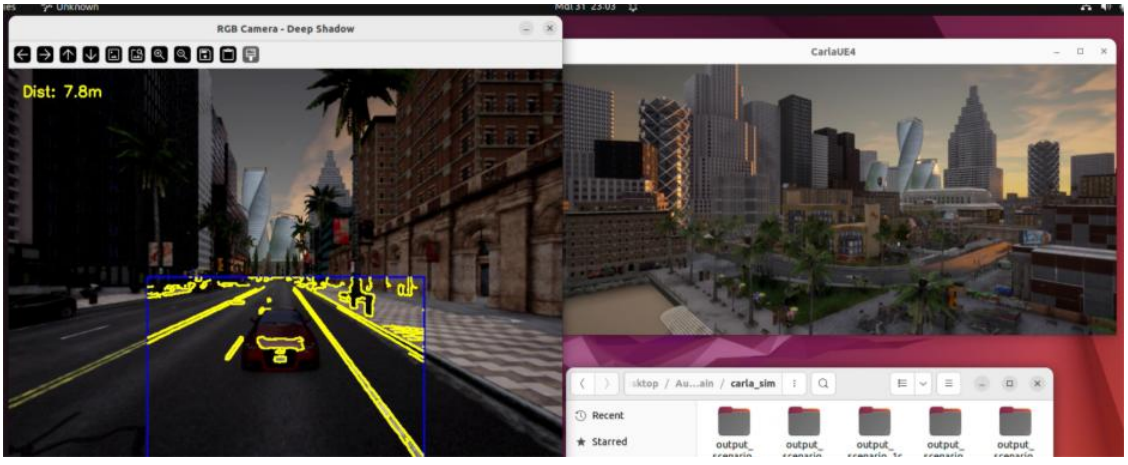
Το συγκεκριμένο σενάριο προσομοιώνει έντονα υγρό οδοστρωμα. Το επίπεδο υγρασίας (wetness) είναι υψηλό και συνοδεύεται από αυξημένο bloom, το οποίο προκαλεί έντονες αντανακλάσεις στο περιβάλλον. Παρ' όλες τις προκλήσεις, η ανίχνευση των εμποδίων εξακολουθεί να βασίζεται σε επεξεργασία άκρων με τον αλγόριθμο Canny. Οι εικόνες καταγράφονται συνεχώς για μεταγενέστερη ανάλυση, επιτρέποντας αξιολόγηση της ακρίβειας υπό συνθήκες με μειωμένη ορατότητα λόγω αντανακλάσεων.



Εικόνα 10 - Σενάριο 4. Βρεγμένος δρόμος.

4.3.5 Σενάριο 5 – Έντονη σκιά

Σε αυτό το σενάριο, ο ήλιος βρίσκεται σε εξαιρετικά χαμηλή θέση (μόλις 3°), δημιουργώντας έντονες και εκτεταμένες σκιές κατά μήκος του οδοστρώματος. Το εμπόδιο τοποθετείται στρατηγικά μέσα σε μία από αυτές τις σκιές, προσομοιώνοντας την περίπτωση όπου ένα αντικείμενο βρίσκεται σε πλήρη σκίαση. Παρα τις δυσμενείς φωτιστικές συνθήκες, το σύστημα καταφέρνει να ανιχνεύσει το εμπόδιο αξιοποιώντας τα διαθέσιμα περιγράμματα, επιδεικνύοντας ικανότητα αντίδρασης ακόμη και σε περιοχές χαμηλής φωτεινότητας.



Εικόνα 11 - Σενάριο 5. Έντονη σκιά.

4.4 Αποτελέσματα Ανίχνευσης

Τα πειράματα που πραγματοποιήθηκαν στο προσομοιωτικό περιβάλλον CARLA αποσκοπούν στην αξιολόγηση της απόδοσης του συστήματος ανίχνευσης εμποδίων μέσω επεξεργασίας ακμών σε διαφορετικές φωτιστικές και περιβαλλοντικές συνθήκες. Η ανίχνευση εμποδίων αποτελεί κρίσιμο παράγοντα για την ασφάλεια και την αποδοτικότητα των αυτόνομων οχημάτων, και η αποτελεσματικότητά της μπορεί να επηρεαστεί σημαντικά από παράγοντες όπως η ένταση του φωτισμού, η παρουσία glare από τον ήλιο, η υγρασία στο οδόστρωμα και η ύπαρξη σκιών

Στα αποτελέσματα που ακολουθούν, παρατηρούμε πώς το σύστημα ανταποκρίνεται σε κάθε σενάριο, με ιδιαίτερη έμφαση στην ικακότητα ανίχνευσης εμποδίων και στην απόσταση στην οποία το όχημα προχώρα πριν σταματήσει.

Πίνακας 2 - Αποτελέσματα ανίχνευσης εμποδίων στον προσομοιωτή CARLA.

Σενάριο	**Οπτικές Συνθήκες**	**Απόσταση Στάσης (m)**	**Εμπόδιο Ανιχνεύθηκε;**	**Αντίδραση Οχήματος**
Scenario_1	Υπερβολικός φωτισμός	~0.0	NO	FORWARD
Scenario_2	Μέτριος φωτισμός	~5.9	YES	STOP
Scenario_3	Κανονικές συνθήκες ημέρας	~9.7	YES	STOP
Scenario_4	Βρεγμένος Δρόμος	~10.0	YES	STOP
Scenario_5	Έντονη Σκιά	~5.4	YES	STOP

Τα αποτελέσματα δείχνουν ότι η μέθοδος ανίχνευσης είναι ιδιαίτερα αποτελεσματική σε περιβάλλοντα με φυσικό και σταθερό φωτισμό (Σενάρια 2, 3, 5), όπου το σύστημα μπορεί να αναγνωρίσει με ακρίβεια τα εμπόδια και να ενεργοποιήσει το φρενάρισμα εγκαίρως. Ωστόσο, όταν το σύστημα αντιμετωπίζει ακραίες συνθήκες, όπως υπερέκθεση και έντονο glare από τον ήλιο (Σενάριο 1), ή υγρό οδόστρωμα που προκαλεί αντανακλάσεις (Σενάριο 4), η ανίχνευση εμποδίων παρουσιάζει σημαντικές δυσκολίες.

Αυτές οι δυσκολίες έχουν ως αποτέλεσμα τη μη ανίχνευση εμποδίων, θέτοντας σε κίνδυνο την ασφάλεια του οχήματος και των επιβατών. Επιπλέον, η απόσταση στάση του οχήματος παρουσιάζει μικρές διαφοροποιήσεις ανάλογα με τις περιβαλλοντικές συνθήκες, κυμαινόμενη περίπου μεταξύ 8 με 11 μέτρων.

4.5 Συμπεράσματα

Η ανάλυση των αποτελεσμάτων που προέκυψαν από την προσομοίωση στο CARLA καταδεικνύει σημαντικές παρατηρήσεις σχετικά με την απόδοση της μεθόδου ανίχνευσης εμποδίων. Αρχικά, διαπιστώνεται ότι η ακρίβεια του αλγορίθμου επηρεάζεται σημαντικά από τις συνθήκες φωτισμού. Σε σενάρια με υπερέκθεση ή glare, η ικανότητα ανίχνευσης εμποδίων μειώνεται, γεγονός που υποδεικνύει την ανάγκη για περαιτέρω βελτιώσει στον αλγόριθμο ώστε να αντιμετωπιζονται τέτοιες ακραίες καταστάσεις. Αντίθετα, σε κανονικές ή μέτριες συνθήκες φωτισμού, η μέθοδος παρουσιάζει υψηλή ακρίβεια, αποδεικνύοντας τη σταθερότητα και αξιοπιστία της σε πιο τυπικά περιβάλλοντα οδήγησης.

Επιπλέον, η ανάλυση έδειξε ότι φυσικά φαινόμενα όπως βρεγμένο οδόστρωμα και έντονες σκιές δημιουργούν πρόσθετες προκλήσεις στην ανίχνευση. Οι αντανακλάσεις από το νερό και οι έντονες σκιές επηρεάζουν την ευκρίνεια των εικόνων, γεγονός που καθιστά δυσκολότερη την ακριβή αναγνώριση των εμποδίων. Παράλληλα, η επιλογή της σταθερής πορείας του οχήματος στην προσομοίωση επέτρεψε τον περιορισμό των παραγόντων που θα μπορούσαν να επηρεάσουν την αξιολόγηση, εξασφαλίζοντας ότι οι μεταβολές στα αποτελέσματα οφείλονται αποκλειστικά στις ιδιότητες του αλγορίθμου.

5

Προσομοίωση στο AirSim

5.1 Εισαγωγή

Η δεύτερη φάση αξιολόγησης της μεθόδου ανίχνευσης εμποδίων πραγματοποιήθηκε στον προσομοιωτή AirSim, με στόχο τη διερεύνηση της συμπεριφοράς του συστήματος σε περιβάλλοντα με μεταβαλλόμενες συνθήκες φωτισμού. Η επιλογή του AirSim βασίστηκε στη δυνατότητά του να παρέχει έναν υψηλό βαθμό παραμετροποίησης, προσφέροντας εργαλεία για τον πλήρη έλγχο της σκηνής και την εισαγωγή στοιχείων που επηρεάζουν άμεσα την οπτική αντίληψη. Σε αντίθεση με την πρώτη φάση, η οποία είχε πιο περιορισμένες δυνατότητες ως προς τη ρύθμιση εξωτερικών παραγόντων, η δευτερη φάση επέτρεψε μια πιο στοχευμένη μελέτη σε παραμέτρους που σχετίζονται με την πραγματική οδική εμπειρία.

Η οπτική ανίχνευση εμποδίων αποτελεί μία από τις πλέον κρίσιμες λειτουργίες στα αυτόνομα οχήματα, καθώς εξαρτάται σε μεγάλο βαθμό από την ποιότητα και τη σταθερότητα των οπτικών δεδομένων. Οι συνθήκες φωτισμού, είτε πρόκειται για φυσικές μεταβολές όπως η αλλαγή της ηλιοφάνειας, είτε για τεχνικούς παράγοντες όπως η ύπαρξη φώτων οδικού δικτύου, δημιουργούν καταστάσεις που μπορούν να περιορίσουν την ακρίβεια ανίχνευσης ή να οδηγήσουν σε ψευδείς συναγερμούς. Για τον λόγο αυτό, η διεξαγωγή ελεγχόμενων πειραμάτων στο AirSim αποτέλεσε κρίσιμο βήμα, καθώς επέτρεψε τη δημιουργία ρεαλιστικών σεναρίων που θα ήταν δύσκολο ή και αδύνατο να αναπαραχθούν σε πραγματικό περιβάλλον ή σε λιγότερο ευέλικτους προσομοιωτές.

5.2 Περιγραφή περιβάλλοντος AirSim

Για τις ανάγκες της προσομοίωσης χρησιμοποιήθηκε ο προκαθορισμένος χάρτης Blocks, ο οποίος περιέχει ευθύγραμμα τμήματα δρόμου και δυνατότητα προσαρμογής φωτιστικών στοιχείων μέσω του Unreal Editor. Το εικονικό όχημα που επιλέγχθηκε ήταν τύπου SimpleCar, με συνδεδεμένη εμπρόσθια κάμερα RGB, σε σταθερό ύψος και προσανατολισμό προς τον δρόμο. Η κάμερα απέδιδε εικόνες σε ανάλυση 640×480 pixels, με ρυθμό λήψης 30 fps.

Η μέθοδος επεξεργασίας εικόνας ήταν πανομοιότυπη με αυτή που εφαρμόστηκε στο CARLA, περιλάμβανε μετατροπή σε grayscale,φιλτράρισμα Gauss, ανίχνευση ακμών μέσω Canny Edge Detector και εφαρμογή περιοχής ενδιαφέροντος (ROI) στο κάτω και μεσαίο τμήμα της εικόνας. Οι ακμές αναλύονταν για την παρουσία περιγραμμάτων με εμβαδόν $> 600\, px^2$, και σε περίπτωση εντοπισμού και επαρκούς εγγύτητας, ενεργοποιείτο φρενάρισμα. Η πραγματική απόσταση του οχήματος από το εμπόδιο υπολογιζόταν από τις συντεταγμένες του αντικειμένου στο περιβάλλον Unreal Engine, εξασφαλίζοντας ακριβή μέτρηση ανεξάρτητα από τις παραμορφώσεις της εικόνας.

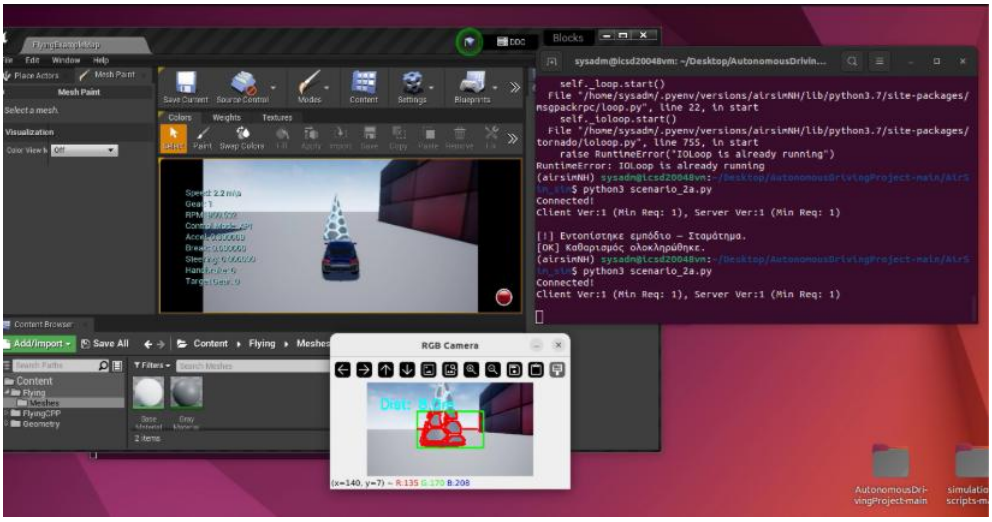
Οι φωτιστικές συνθήκες διαμορφώθηκαν μέσω πολλαπλών spotlights που τοποθετήθηκαν σε διαφορετικές θέσεις και γωνίες γύρω από το εμπόδιο, δημιουργώνταε σκιές, διαφοροποιημένη ένταση φωτισμού και περιοχές υψηλής αντίθεσης. Με αυτόν τον τρόπο προσομοιώθηκαν ρεαλιστικές καταστάσεις όπου η οπτική αντίληψη επηρεάζεται από εξωτερικούς παράγοντες, όπως η μεταβολή της ηλιοφάνειας ή η παρουσία τεχνητού φωτισμού.

5.3 Περιγραφή Σεναρίων

Για την δοκιμή της ανθεκτικότητας του συστήματος ανίχνευσης, δημιουργήθηκαν τέσσερα σενάρια φωτισμού, καθένα από τα οποία τροποποιεί την ένταση, κατεύθυνση ή κατανομή του φωτός σε σχέση με την κάμερα και το εμπόδιο.

5.3.1 Σενάριο 1 – Κανονικές Συνθήκες Φωτισμού

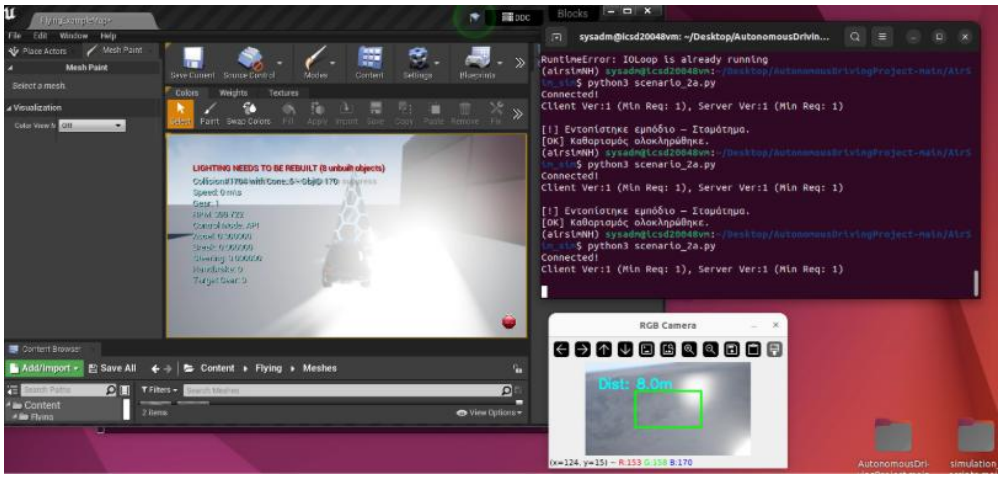
Το σενάριο αυτό προσομοιώνει ένα περιβάλλον χωρίς έντονο ή τεχνικό φωτισμό. Το εμπόδιο είναι ορατό μέσω φυσικού φωτισμού της σκηνής. Δεν υπάρχουν σκιές υψηλής έντασης ή φαινόμενα glare. Η κάμερα επεξεργάζεται την εικόνα με υψηλή καθαρότητα, διευκολύνοντας την ανίχνευση περιγραμμάτων. Σε αυτό το περιβάλλον, το σύστημα παρουσίασε σταθερή και αξιόπιστη ανίχνευση του εμποδίου, ενεργοποιώντας φρένο εγκαίρως.



Εικόνα 12 - Σενάριο 1. Κανονικές συνθήκες φωτισμού.

5.3.2 Σενάριο 2 – Φωτισμός απο το εμπόδιο προς την κάμερα

Σε αυτό το σενάριο, ένας προβολέας υψηλής έντασης είναι τοποθετημένος πάνω στο εμπόδιο και κατευθύνεται προς την κάμερα του οχήματος, δημιουργώντας έντονο φώς στο οπτικό πεδίο.

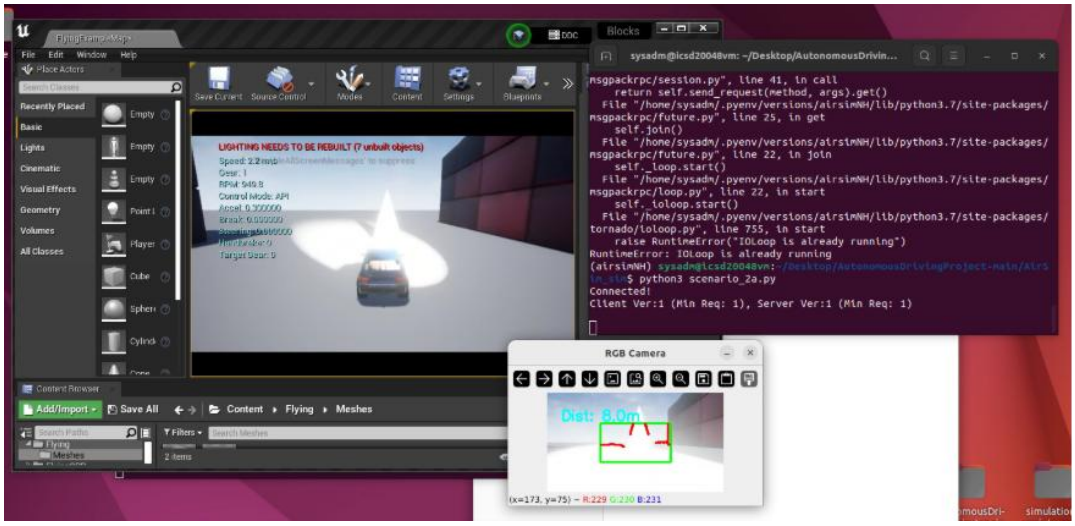


Εικόνα 13 - Σενάριο 2. Φωτισμός απο το εμπόδιο προς την κάμερα

Αυτό προκαλεί φωτεινά φαινόμενα όπως glare και μερική υπερέκθεση στην περιοχή γύρω από το εμπόδιο, γεγονός που επηρεάζει την αντίθεση και περιπλέκει την ανίχνευση των περιγραμμάτων. Παρά τις δυσκολίες, το σύστημα κατάφερε να εντοπίσει την ύπαρξη εμποδίου, με ελαφρώς καθυστερημένη απόκριση (λόγω χαμηλότερης ευκρίνειας των ακμών), και ενεργοποίησε φρενάρισμα, αποδεικνύοντας ανθεκτικότητα σε αντίξοες φωτιστικές συνθήκες.

5.3.3 Σενάριο 3 – Φωτισμός πάνω στο εμπόδιο

Σε αυτό το σενάριο, ένας προβολέας υψηλής έντασης είναι στραμμένος απευθείας πάνω στο εμπόδιο, φωτίζοντάς το έντονα από κοντινή απόσταση. Αν και το φώς δεν κατευθύνεται προς την κάμερα, όπως στο Σενάριο 2, η υπερφωτισμένη επιφάνεια του εμποδίου προκαλεί απώλεια τοπικής αντίθεσης και εξαφάνιση των περιγραμμάτων στις περιοχές υπερέκθεσης



Εικόνα 14 - Σενάριο 3. Φωτισμός πάνω στο εμπόδιο

Η κάμερα καταγράφει εικόνες χωρίς glare, αλλά το εμπόδιο εμφανίζεται ως ομοιογενής φωτεινή μάζα, χωρίς καθαρές ακμές. Αυτό επηρεάζει αρνητικά τον αλγόριθμο ανίχνευσης περιγραμμάτων (Canny), ο οποίος βασίζεται σε διαφορές φωτεινότητας για τον εντοπισμό ακμών.

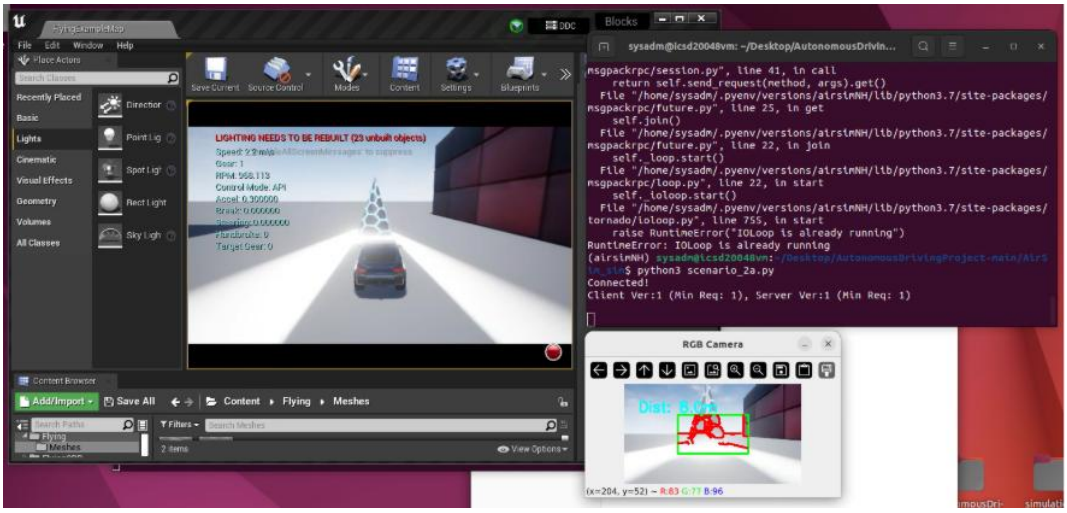
Ως αποτέλεσμα, το σύστημα δεν κατάφερε να εντοπίσει το εμπόδιο και συνέχισε να κινείται, χωρίς να ενεργοποιήσει φρενάρισμα. Το σενάριο αυτό αναδεικνύει έναν σημαντικό περιορισμό των οπτικώ συστημάτων σε περιπτώσεις έντονης τοπικής φωταγώγησης, όπου η ίδια η φωτεινότητα του αντικειμένου καθιστάται εμπόδιο στην αναγνώρισή του.

5.3.4 Σενάριο 4 – Φωτισμός πίσω από το εμπόσιο

Σε αυτό το σενάριο, ένας προβολέας τοποθετείται πίσω από το εμπόδιο, κατευθύνοντας τη φωτεινή δέσμη προε το πίσω μέρος του αντικειμένου. Η κάμερα του οχήματος βρίσκεται μπροστά από το εμπόδιο, καταγράφοντας την πρόσοψή του, η οποία δεν φωτίζεται άμεσα αλλά περιμετρικά, από το φως που διαχέεται δεξιά και αριστερά.

Αυτό δημιουργεί ένα φαινόμενο rim lighting, με τις άκρες του εμποδίου να αποκτούν ελαφρώς φωτεινότερο περίγραμμα, ενώ η κεντρική του επιφάνεια παραμένει σε σχετική σκιά. Παρότι η

αντίθεση ανάμεσα στο εμπόδιο και το φόντο δεν είναι υψηλή, η φωτεινή ενίσχυση των πλευρικών περιγραμμάτων αποδείχθηκε επαρκής για την επεξεργασία ακμών μέσω του αλγορίθμου.



Εικόνα 15 - Σενάριο 4. Φωτισμός πίσω από το εμπόδιο.

Κατά την εκτέλεση του σεναρίου, το σύστημα κατάφερε να εντοπ’ισει επιτυχώς το εμπόδιο, ενεργοποιώντας εγκαίρως την αντίδραση φρεναρίσματος. Το αποτέλεσμα αυτό επιβεβαιώνει ότι ο πλάγιος ή πίσω φωτισμός μπορεί να εισχύσει την αναγνωρισιμότητα περιγραμμάτων, ακόμα και χωρίς άμεσο φωτισμό της εμπρός πλευράς του αντικειμένου ή της κάμερας.

5.4 Αποτελέσματα Ανίχνευσης

Στο προσομοιωτικό περιβάλλον AirSim, εξετάστηκε η απόδοση του συστήματος ανίχνευσης εμποδίων, το οποίο βασίζεται σε επεξεργασία εικόνας από την εμπρόσθια κάμερα, υπό διάφορες φωτιστικές συνθήκες. Στόχος ήταν η αξιολόγηση της συμπεριφοράς του συστήματος σε σενάρια όπου οι συνθήκες φωτισμού επηρεάζουν άμεσα την ορατότητα και την ευκρίνεια των περιγραμμάτων του εμποδίου.

Πίνακας 3 - Αποτελέσματα ανίχνευσης εμποδίων στον προσομοιωτή AirSim

Σενάριο	**Απόσταση Στάσης(m)**	**Εμπόδιο Ανιχνεύθηκε;**	**Οπτικές Συνθήκες**	**Αντιδραση Οχήματος**
Scenario_1	~5.1	YES	Κανονικό φωτισμός	STOP
Scenario_2	~0.0	NO	Έντονο φως	FORWARD
Scenario_3	~0.0	NO	Υπερωτισμός στο εμπόδιο	FORWARD
Scenario_4	~3.9	YES	Περιμετρικός φωτισμός	STOP

Τα αποτελέσματα υποδεικνύουν ότι η απόδοση του συστήματος ανίχνευσης εμποδίων επηρεάζεται άμεσα από τις φωτιστικές συνθήκες του περιβάλλοντος. Στο Σενάριο 1, υπό φυσικό και σταθερό φωτισμό, το σύστημα λειτούργησε ορθά, εντοπίζοντας το εμπόδιο εγκαίρως και σταματώντας το όχημα σε απόσταση περίπου 5 μέτρων, δείχνοντας αποτελεσματική λειτουργία σε προβλέψιμες και σταθερές συνθήκες. Στο Σενάριο 2, η εντολή glare (λάμψη) από φωτισνή πηγή που “τύφλωσε” την κάμερα, οδήγησε σε πλήρη αποτυχία της ανίχνευσης, το σύστημα δεν αντιλήφθηκε το εμπόδιο και το όχημα συνέχισε χωρίς να σταματήσει, παρά την ύπαρξη εμποδίου σε πολύ μικρή απόσταση. Παρόμοια εικόνα παρατηρήθηκε και στο Σενάριο 3, όπου ο άμεσος φωτισμός του εμποδίου από προβολέα προκάλεσε υπερφωτισμό και απώλεια περιγραμμάτων, με αποτέλεσμα την αποτυχία ανίχνευσης και μη αντίδραση του οχήματος. Αντίθετα, στο Σενάριο 4, η ύπαρξη περιμετρικού φωτισμού (rim lighting) συνέβαλε θετικά στην ανάδειξη των περιγραμμάτων του εμποδίου, επιτρέποντας στο σύστημα να το εντόπίσει με επιτυχία και να σταματήσει το όχημα σε απόσταση περίπου 4 μέτρων.

5.5 Συμπεράσματα

Η δεύτερη φάση αξιολόγησης στον προσομοιωτή AirSim κατέδειξε με σαφήνεια ότι οι φωτιστικές συνθήκες αποτελούν κρίσιμο παράγοντα για την αξιοπιστία των μεθόδων ανίχνευσης εμποδίων βασισμένων σε επεξεργασία εικόνας. Η χρήση spotlights με διαφορετική ένταση και κατεύθυνση, είτε προς το εμπόδιο είτε απευθείας προς την κάμερα, επέτρεψε τη δημιουργία ποικιλίας σεναρίων που προσομοίαζαν πραγματικες προκλήσεις, όπως η εκτύφλωση αντανάκλασης, η απ'οτομη εναλλαγή φωτεινότητας και η παρουσία έντονων σκιών. Οι συνθήκες αυτές κατέστησαν εμφανή τα όρια της μεθόδου, καθ'ως σε αρκετές περιπτώσεις η εικόνα παρουσίασε φαινόμενα κορεσμού, με αποτέλεσμα την αλλοίωση των περιγραμμάτων και την εμφάνιση ψευδών ενδείξεων.

Παρά τις δυσκολίες αυτές, η μέθοδος διατήρησε σε πολλές δοκιμές τη λειτουργικότητά της, ενεργοποιώντας με συνέπεια τον μηχανισμό φραναρίσματος όταν το εμποδιο ήταν ορατό υπό επρακώς ισορροπημένες συνθήκες φωτισμού. Η συμπεριφορά αυτή επιβεβαίωσε ότι η προσέγγιση μπορεί να ανταποκριθεί αποτελεσματικά σε βασικά σενάρια, ωστόσο υπογράμμισε ότι η αξιοπιστία της μειώνεται σημαντικά σε ακραίες ή μη ομοιόμορφες φωτιστικές καταστάσεις, ιδιαίτερα, όταν το φως έπεφτε απευθείας στον φακό της κάμερας, παρατηρήθηκε μείωση της ακρίβειας αντοπισμού, γεγονός που θα μπορούσε να έχει σοβαρές επιπτώσεις σε πραγματικά περιβάλλοντα κυκλοφορίας.

6

Πειραματική Εκτέλεση σε Raspberry Pi 4

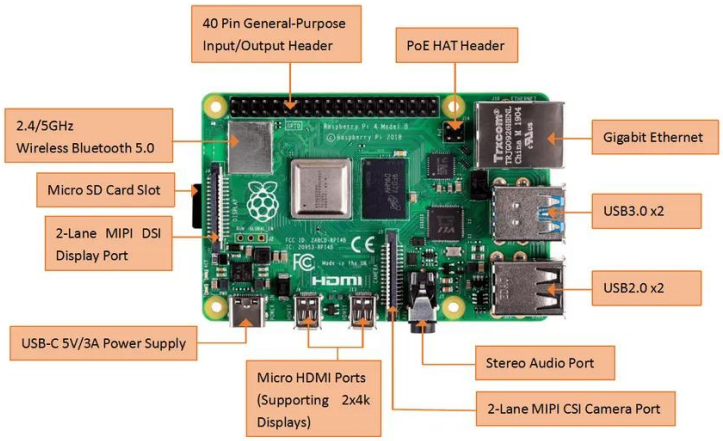
6.1 Εισαγωγή

Η παρούσα ενότητα εστιάζει στην εφαρμογή και αξιολόγηση ενός συστήματος ανίχνευσης εμποδίων σε πραγματικές συνθήκες, χρησιμοποιώντας ως υπολογιστική πλατφόρμα το Raspberry Pi 4. Στόχος είναι η μετάβαση από το ελεγχόμενο περιβάλλον των προσομοιωτών CARLA και AirSim σε ένα φυσικό ρομποτικό όχημα, προκειμένου να εξεταστεί η απόδοση των μεθόδων σε μη ιδανικές συνθήκες, όπου παράγοντες όπως η ανώμαλη επιφάνεια, οι μικρές αποκλίσεις στις ροπές των κινητήρων ή οι αλλαγές στο φωτισμό μπορούν να επηρεάσουν σημαντικά τη συμπεριφορά του οχήματος.

Η πειραματική φάση αναδεικνύει τόσο τις τεχνικές προκλήσεις που προκύπτουν κατά την υλοποίηση σε πραγματικό περιβάλλον όσο και την ανάγκη προσαρμογών σε επίπεδο υλικού και λογισμικού. Ιδιαίτερα σημαντική είναι η ενσωμάτωση ενός συστήματος διόρθωσης πορείας μέσω αναγνώρισης λωρίδας (Lane Detection), το οποίο δεν ήταν απαραίτητο στους προσομοιωτές, καθώς εκεί το όχημα κινούντας σε ιδανικές διαδρομές χωρίς αποκλίσεις. Στο φυσικό περιβάλλον, η ανίχνευση και διατήρηση της λωρίδας κίνησης ήταν κρίσιμη για την ομαλή και ασφαλή πορεία του οχήματος, εξασφαλίζοντας ότι το όχημα ακολουθεί σταθερά την επιθυμητή διαδρομή και μειώνοντας τον κίνδυνο συγκρούσεων ή απόκλισης από το δρόμο..

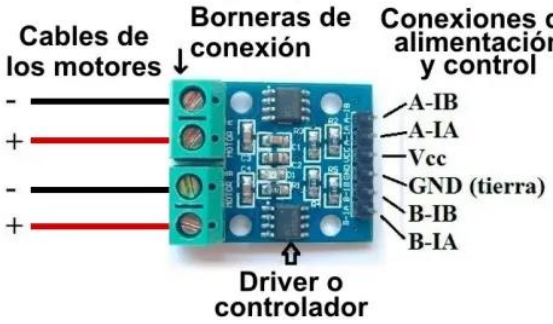
6.2 Περιγραφή Υλικού και Συνδεσμολογίας

Η πειραματική διάταξη στηρίχθηκε στην υπολογιστική πλατφόρμα Raspberry Pi 4 Model B, μια ιδιαίτερα διαδεδομένη επιλογή για εφαρμογές ρομποτικής χάρη στον συνδυασμό υπολογιστικής ισχύος, ευελιξίας και χαμηλού κόστους. Το συγκεκριμένο μοντέλο διαθέτει τετραπύρηνο επεξεργαστή ARM Cortex-A72 χρονισμένο στα 1.5 GHz, με δυνατότητα σύνδεσης με κάμερα USB ή CSI, καθώς και πλήθος εισόδων/εξόδων GPIO, που το καθιστούν ιδανικό για τον έλεγχο αισθητήρων και κινητήρων σε πραγματικό χρόνο. Επιπλέον, το Raspberry Pi 4 ενσωματώνει μνήμη RAM έως 8GB, διπλές θύρες micro-HDMI με υποστήριξη ανάλυσης 4K, καθώς και υποστήριξη Wi-Fi και Bluetooth, παρέχοντας τη δυνατότητα για ασύρματη επικοινωνία και απομακρυσμένο έλεγχο. Αυτά τα χαρακτηριστικά καθιστούν την πλατφόρμα ιδιαίτερα ευέλικτη για την ανάπτυξη αλγορίθμων επεξεργασίας εικόνας και real-time control σε ένα μικρό, χαμηλής κατανάλωσης πακέτο.



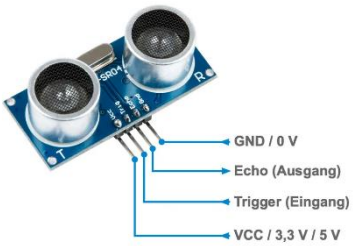
Εικόνα 16 - Raspberry Pi 4 Model B. Πηγή: DFRobot-"Raspberry Pi 4 Model B – 4 GB", DFRobot.com.

Το ρομποτικό όχημα κινείται με τη βοήθεια δύο κινητήρων συνεχούς ρεύματος (DC motors), οι οποίοι ελέγχονταν μέσω H-Bridge κυκλώματος οδήγησης (motor drive). Το H-Bridge L9110 είναι ένας από τους πιο δημοφιλείς οδηγούς κινητήρρων για μικρά ρομποτικά συστήματα, χάρη στη χαμηλή κατανάλωση ισχύος, την απλότητα και το μικρό του μέγεθος. Διαθέτει δύο κανάλια που επιτρέπουν τον ανεξάρτητο έλεγχο δύο κινητήρων, παρέχοντας ρεύμα έως περίπου 800mA ανά κανάλι, επαρκές για τυπικούς μικρούς DC κινητήρες. Η αρχιτεκτονική του L9110 επιτρέπει τόσο τον έλεγχο της κατεύθυνσης περιστροφής (προς τα εμπρός/πίσω) όσο και την ακριβή ρύθμιση της ταχύτητας σε σήματα PWM (Pulse Width Modulation). Η επιλογή του PWM κρίθηκε απαραίτητη, καθώς προσφέρει ομαλή επιτάχυνση/επιβράδυνση, μειώνει τις απότομες μεταβολές και συμβάλλει στη βελτίωση της σταθερότητας κινησης. Παράλληλα, ο σχεδιασμός του L9110 το καθιστά ανθεκτικό σε υπερφορτώσεις και παρέχει προστασία έναντι βραχυκυκλωμάτων, εξασφαλίζοντας αξιόπιστη και συνεχόμενη λειτουργία στο ρομποτικό όχημα.



Εικόνα 17 - L9110 H-Bridge Motor Driver. Πηγή: Robots Argentina-"Puente H: Placa controladora de motores L9110S", Robots Didacticos, Robots-Argentina.com.ar.

Για την αντίληψη του περιβάλλοντος, χρησιμοποιήθηκε ο υπερηχητικός αισθητήρας HC-SR04, ο οποίος αποτελεί μία από τις πιο αξιόπιστες και οικονομικές λύσεις για τη μέτρηση αποστάσεων μικρής κλίμακας (2m-400m). Ο αισθητήρας λειτουργεί με την εκπομπή υπερηχητικού παλμού και την καταγραφή του χρόνου που απαιτείται μέχρι να επιστρέψει η ηχώ από το εμπόδιο, υπολογίζοντας έτσι την απόσταση με βάση την ταχύτητα του ήχου στον αέρα.



Εικόνα 18 - Υπερηχητικός αισθητήρας HC-SR04. Πηγή: Elektronik-Kompendium- "HC-SR04-Ultrasonic Sensor (Ultraschallsensor)", Elektronik-kompendium.de.

Παράλληλα, για την οπτική αναγνώριση του χώρου, συνδέθηκε USB κάμερα στο Raspberry Pi, η οποία τοποθετήθηκε στο εμπρόσθιο τμήμα του οχήματος. Η θέση της κάμερας επιλέχθηκε ώστε να εξασφαλίζει ευρύ πεδίο λήψης προς την κατεύθυνση κίνησης, δίνοντας στο σύστημα τη δυνατότητα να ανιχνεύει τόσο εμπόδια όσο και οπτικά στοιχεία του περιβάλλοντος (π.χ. λωρίδες πορείας). Η ανάλυση βίντεο ορίστηλε στα 640×480 pixels με ρυθμό 30 καρέ ανά δευτερόλεπτο (fps), τιμές που προσφέρουν καλή ισορροπία μεταξύ ποιότητας εικόνας και ταχύτητας επεξεργασίας σε πραγματικό χρόνο, δεδομένων των περιορισμένων υπολογιστικών πόρων του Raspberry Pi.

Η τροφοδοσία του συστήματος πραγματοποιήθηκε μέσω φορητού powerbank 5V/3A, το οποίο εξασφάλισε επάρκεια ισχύος τόσο για τον υπολογιστή όσο και για τους κινητήρες. Η επιλογή powerbank αντί εξωτερικού τροφοδοτικού κατέστη δυνατή την πλήρως αυτόνομη λειτουργία του ρομπότ, χωρίς την ανάγκη καλωδίων ή σταθερή πηγής ρε΄ματος, προσομοιώνοντας έτσι πιο ρεαλιστικές συνθήκες χρήσης.

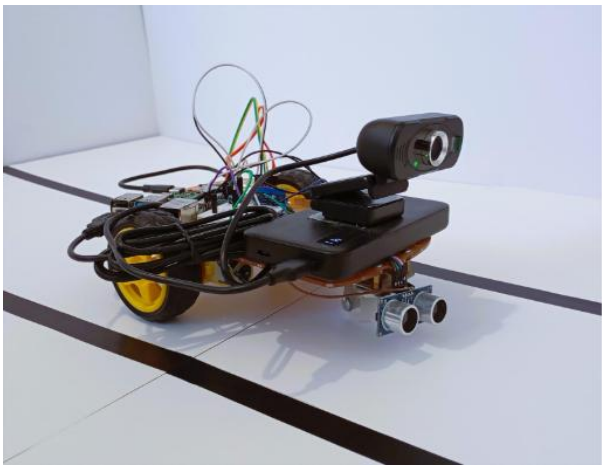
Ιδιαίτερη μέριμνα δόθηκε στη συνδεσμολογία του αισθητήρα HC-SR04, καθώς η έξοδος ECHO παρέχει παλμό 5V, ενώ τα GPIO pins του Raspberry Pi λειτουργούν με 3.3V logic και είναι εξαιτερικά ευαίσθητες σε υπερτάσεις. Για την προστασία της πλατφόρμα κατασκευάστηκε

διαιρέτης τάσης με δύο αντιστάσεις (R1=1kΩ, R2=2kΩ), ο οποίος μείωσε την τάση από 5V σε περίπου 3.3V.

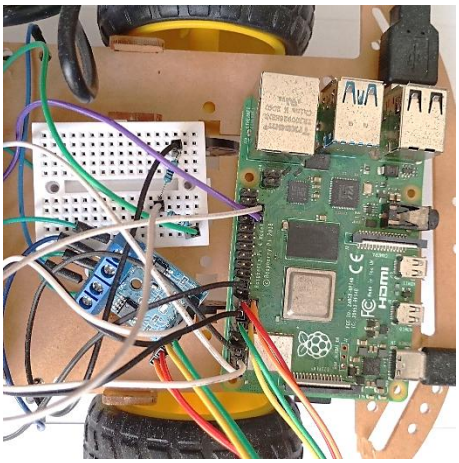
Η εξίσωση που διέπει τον διαιρέτη είναι:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2} = 5 \cdot \frac{2}{1+2} = 3.33V$$

Η εφαρμογή αυτής της προστατευτικής διάταξης ήταν καθοριστικής σημασίας για την αξιοπιστία και τη μακροχρόνια λειτουργία του συστήματος, αποτρέποντας πιθανή καταστοφή της υπολογιστικής πλατφόρμας.



Εικόνα 19 - Ρομποτικό Οχημα που χρησιμοποιήθηκε



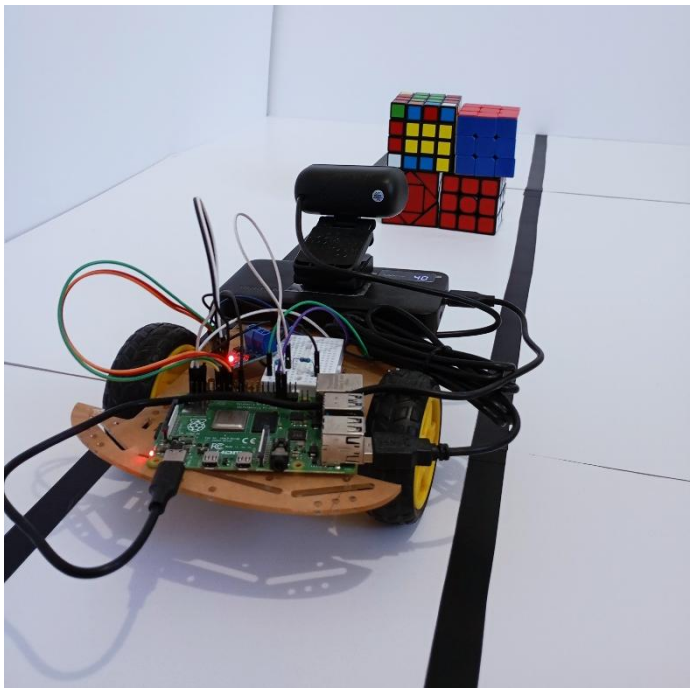
Εικόνα 20 - Συνδεσμολογία ρομποτικού οχήματος

6.3 Ανάγκη Ελέγχου Πορείας και Ενσωμάτωση Lane Detection

Κατά τις αρχικες δοκιμές παρατηρήθηκε ότι, παρά την αποστολή ίσων σημάτων PWM στους δύο κινητήρες, το όχημα δεν ακολουθούσε ευθύγραμμη πορεία. Το φαινόμενο αυτό αποδόθηκε σε μικροδιαφορές στην κατασκευή του οχήματος, αναμοιογένεια στο βάρος ανά τροχό, διαφορές στη ροπή των κινητήρων ή ακόμα και ανοχές στο γρανάζωμα. Ως αποτέλεσμα, το όχημα παρεξέκλινε σταθερά είτε αριστερά είτε δεξιά, γεγονός που καθιστούσε αδύνατη την ορθή αξιολόγηση της μεθόδου ανίχνευσης εμποδίων.

Για την αντιμετώπιση του προβλήματος, κρίθηκε απαραίτητη η ενσωμάτωση ενός συστήματος εντοπισμού λωρίδας (Lane Detection) το οποίο θα λειτουργεί ως μηχανισμός διόρθωσης πορείας σε πραγματικό χρόνο. Με την βοήθεια της κάμερας, το όχημα μπορούσε όχι μόνο να ανιχνεύει εμπόδιο αλλά και να προσαρμόζει τη διαδρομή του ώστε να διατηρείται στο κέντρο της λωρίδας κίνησης.

Η κάμερα τοποθετήθηκε στο εμπρόσθιο τμήμα του ρομπότ με μικρή κλίση προς το έδαφος, ώστε να καταγράφει την περιοχή μπροστά από το όχημα. Η επεξεργασία εστίαζε στην κατώτερη περιοχή του καρέ, όπου ήταν ορατές οι λωρίδες ή οι αντιθέσεις του εδάφους που όριζαν τα όρια του διαδρόμου κίνησης.



Εικόνα 21 - Παράδειγμα εικόνας του ρομποτικού οχήματος κατά την πορεία του στον διάδρομο κίνησης.

Η διαδικασία περιλάμβανε:

- Εφαρμογή binary thresholding στο επιλεγμένο τμήμα της εικόνας, ώστε να εντοπιστούν οι λευκές γραμμές ή αντιθέσεις (π.χ. μαυρο σε λευκό δάπεδο).
- Χωρισμό εικόνας σε δύο ημιπλάνα (αριστερό και δεξιό).
- Υπολογισμό κέντρου βάρους για κάθε ημιπλάνο με χρήση των χωρικών στιγμών.

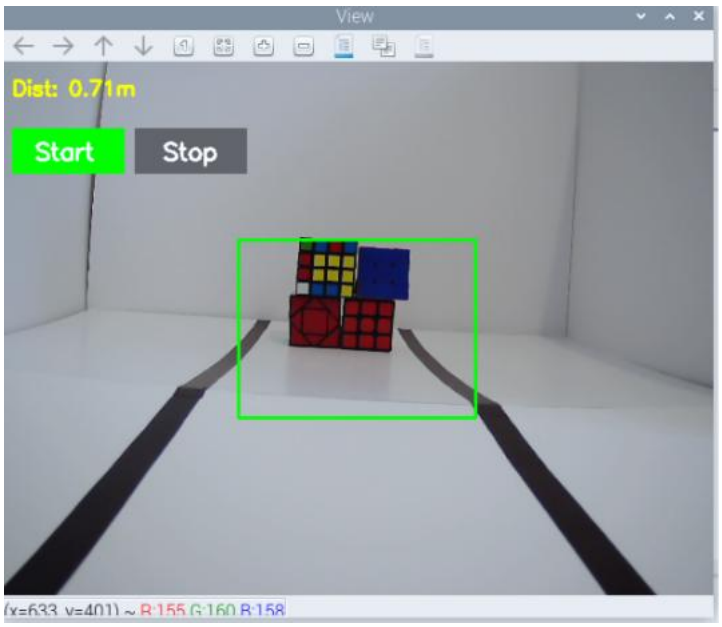
- Εκτίμηση του κέντρου της λωρίδας και σύγκριση με το κέντρο της εικόνας, ώστε να υπολογιστεί το σφάλμα κατεύθυνσης.

Το σφάλμα αυτό τροφοδοτεί ένα PID ελεγκτή (Proportional-Integral-Derivative controller), ο οποίος υπολογίζει δυναμικά τη διόρθωση που πρέπει να εφαρμοστεί. Ανάλογα με το πρόσημο και το μέγεθος της διόρθωσης, τροποποιούντας οι τιμές PWM των κινητήρων, ώστε το όχημα να στραφεί ελαφρώς αριστερά ή δεξιά μέχρι να συγκλίνει προς το κέντρο της λωρίδας.

Με τον τρόπο αυτο, επιτεύχθηκε σταθερή ευθυγράμμιση της πορεία, ανεξάρτητα από τις μικροανομοιογένεις του μηχανικού συστήματος. Η ενσωμάτωση Lane Detection αποτέλεσε ουσιαστικό βήμα για την αξιόπιστη εκτέλεση των πειραμάτων, καθώς εξάλειψε τον παράγοντα «παρέλλλιση οχήματος» που θα μπορούσε να οδηγήσει σε παραπλανητικά αποτελέσματα.

6.4 Γραφικό Περιβάλλον και Καταγραφή Συμβάντων

Αναπτύχθηκε ένα απλό, αλλά λειτουργικό, γραφικό περιβάλλον χρήστη (GUI) στο ίδιο παράθυρο προβολής της κάμερας χρησιμοποιώντας τη βιβλιοθήκη OpenCV. Το GUI περιλάμβανε βασικά εικονικά κουμπιά, όπως “Start” και “Stop”, τα οποία επέτρεπαν την ενεργοποίηση ή απενεργοποίηση της αυτόνομης λειτουργίας του οχήματος χρησιμοποιώντας το ποντίκι. Αυτή η απλή διεπαφή αποδείχθηκε ιδιαίτερα χρήσιμη για τις ανάγκες των πειραματικών δοκιμών, καθώς επέτρεπε τον έλεγχο και την παρακολούθηση του οχήματος από απόσταση, χωρίς να χρειάζεται φυσική παρέμβαση ή άμεση πρόσβαση στο όχημα.



Εικόνα 22 - Παράδειγμα εικόνας γραφικού περιβάλλοντος για απεικόνιση και καταγραφή συμβάντων

Κατά τη διάρκεια της πλοήγησης, το σύστημα κατέγραφε σε CSV αρχείο ανά frame τις εξής πληροφορίες:

- Timestamp: την ακριβή χρονική στιγμη του καρέ.
- Απόσταση εμποδίου: την αποσταση απο το εμπόδιο που μετρήθηκε απο τον υπερηχητικό αισθητήρα HC-SR04.
- Ύπαρξη εμποδίου εντός ROI(Region of Interest): δείκτης ανίχνευσης εμποδίου στο καθορισμένο τμήμα της εικόνας.
- Ληφθείσα ενέργεια: η ενέργεια που εκτελέστηκε από το όχημα στο συγκεκριμένο καρέ (π.χ. “STOP” ή “LANE FOLLOW”).

Επιπλέον, σε περιπτώσεις όπου ενεργοποιούνταν το φρενάρισμα λόγω ανίχνευσης εμποδίου, το αντίστοιχο καρέ αποθηκευόταν ως ξεχωριστό αρχείο εικόνας. Αυτό επέτρεπε τη μεταγενέστερη οπτική αξιολόγηση της αντίδρασης του συστήματος και τη διασταύρωση των μετρήσεων με τα πραγματικά γεγονότα στο περιβάλλον. Η καταγραφή των δεδομένων για κάθε καρέ ήταν πολύ σημαντική για την αξιολόγηση της απόδοσης του συστήματος, καθώς επέτρεπε την επεξεργασία των δεδομένων, την αναγνώριση επαναλαμβανόμενων μοτίβων και τη βελτίωση των αλγορίθμων ελέγχου.

6.5 Περιγραφή Σεναρίων

Η ανίχνευση εμποδίων μέσω επεξεργασίας εικόνας είναι μια μέθοδος που εξαρτάται έντονα από τις συνθήκες φωτισμού του περιβάλλοντος. Η ποιότητα των ακμών που εντοπίζονται μέσω του αλγορίθμου Canny και η συνοχή των περιγραμμάτων μπορεί να αλλοιωθούν σημαντικά όταν αλλάζει η κατεύθυνση, η ένταση ή η κατανομή του φωτός.

Στο πλαίσιο αυτό, σχεδιάστηκαν πέντε διαφορετικά σενάρια φωτισμού, με στόχο τη διερεύνηση της συμπεριφοράς του συστήματος υπό μεταβαλλόμενες εξωτερικές συνθήκες. Τα ίδια σενάρια εφαρμόστηκαν σε τρείς πλατφόρμες, στον προσομοιωτή CARLA, στον προσομοιωτή AirSim και στο πραγματικό ρομπότ με Raspberry Pi 4, ώστε να επιτευχθεί συγκρίσιμη αξιολόγηση της μεθόδου σε εικονικά και φυσικά περιβάλλοντα.

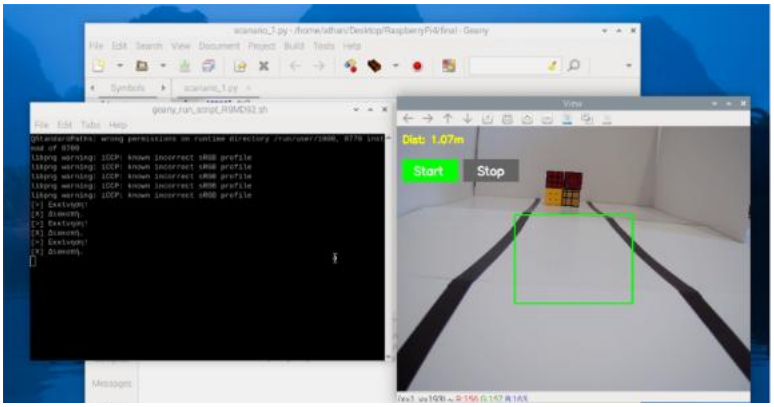
Η ρύθμιση των συνθυκών φωτισμού έγινε:

- Στον προσομοιωτή CARLA μέσω παραμέτρων του rendering engine (sun intensity, shadow intensity, directional lights).
- Στον προσομοιωτή AirSim με την χρήση τεχνητών “SpotLights” και επιλογή γωνίας πρόσπτωσης.
- Στο Raspberry Pi με χρήση πηγών φωτός (φακοί LED).

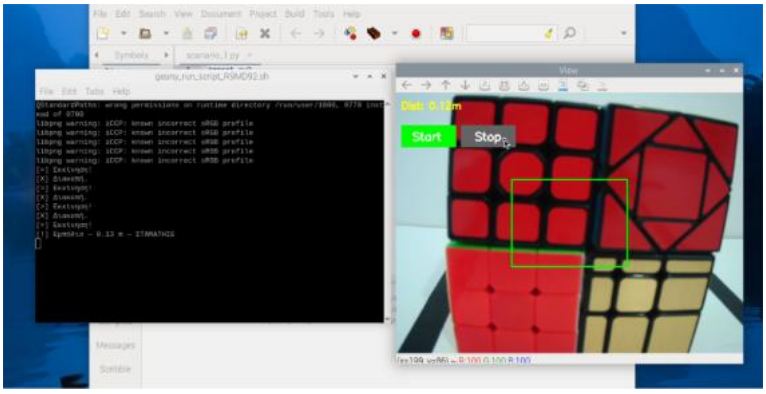
6.5.1 Σενάριο 1 – Φυσιολογικός Φωτισμός

Σε αυτό το βασικό σενάριο, η περιοχή γύρω από το εμπόδιο φωτίζεται ομοιόμορφα από διάχυτο φυσικό ή τεχνητό φως, χωρίς έντονες σκιές, κατευθυντική ακτινοβολία ή ανακλάσεις. Η κάμερα του Raspberry Pi κατέγραψε εικόνες υψηλής ποιότητας, με καθαρές ακμές και επαρκή αντίθεση, γεγονός που ευνόησε την απόδοση του Canny Edge Detector και την αξιόπιστη εξαγωγή περιγραμμάτων.

Η μέθοδος ανίχνευσης λειτούργησε με εξαιρετική συνέπεια, καταγράφοντας ποσοστό επιτυχίας 100% στις πέντε δοκιμές. Το όχημα φρέναρε εγκαίρως και σε όλες τις περιπτώσεις η απόσταση φρεναρίσματος κυμάνθηκε σε στενό εύρος (0.13m-0.15m).



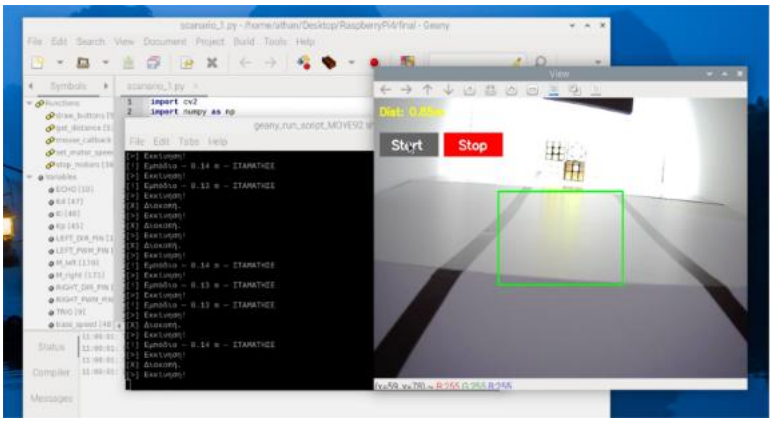
Εικόνα 23 - Σενάριο 1. Έναρξη Ανίχνευσης (Φυσιολογικός Φωτισμός).



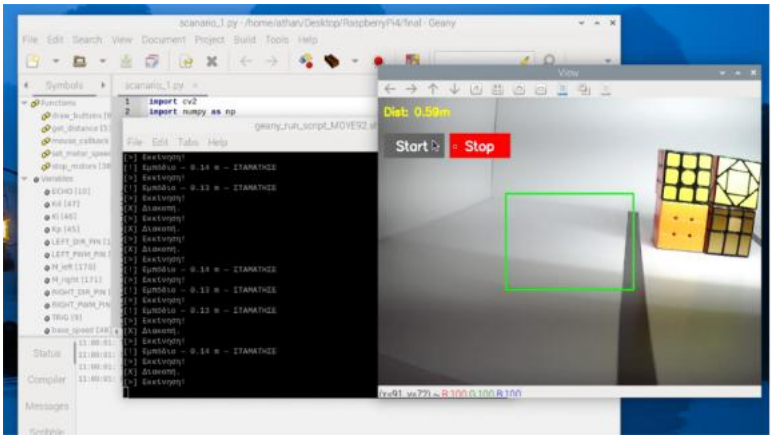
Εικόνα 24 - Σενάριο 1. Τέλος Ανίχνευσης (Φυσιολογικός Φωτισμός).

6.5.2 Σενάριο 2 – Φωτισμός προς το εμπόδιο από μπροστά

Σε αυτό το σενάριο, η φωτεινή πηγή τοποθετήθηκε στο άξονα της κάμερας και κατευθύνθηκε απευθείας προς το εμπόδιο. Το αποτέλεσμα ήταν η εμφάνιση έντονου φαινομένου υπερέκθεσης στην περιοχή ενδιαφέροντος, το οποίο προκάλεσε κορεσμό των pixels και απώλεια λεπτομερειών στις ακμές.



Εικόνα 26 - Σενάριο 2. κίτρινη επιφάνεια. Έναρξη Ανίχνευσης (Φωτισμός Μπροστά)



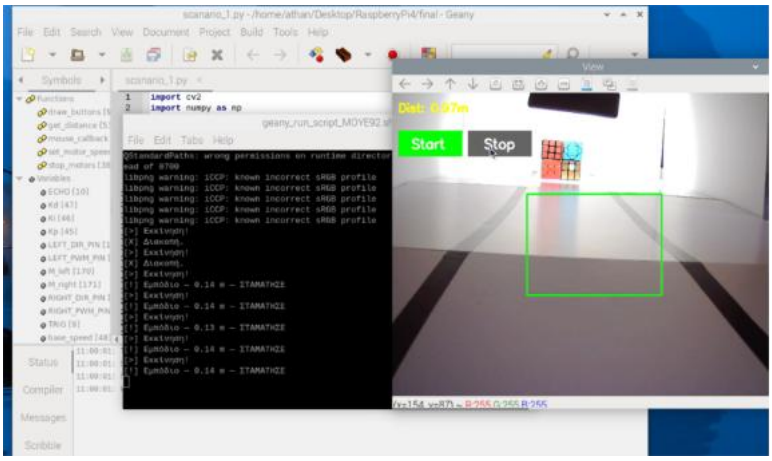
Εικόνα 25 - Σενάριο 2. κίτρινη επιφάνεια. Τέλος Ανίχνευσης (Φωτισμός Μπροστά),

Ο αλγόριθμος Canny δεν ήταν σε θέση να εντοπίσει επαρκή ακμοπληροφορία στις υπερέκθετες περιοχές, γεγονός που οδήγησε σε τρείς αποτυχίες εντοπισμού σε πέντε επαναλήψεις. Επιπλέον, σε μία από τις δύο επιτυχημένες επαναλήψεις η απόφαση “STOP” λήφθηκε με αισθητή καθυστέρηση, λόγω της καθυστερημένης σταθεροποίησης των ακμών στην περιοχή ενδιαφέροντο ς.

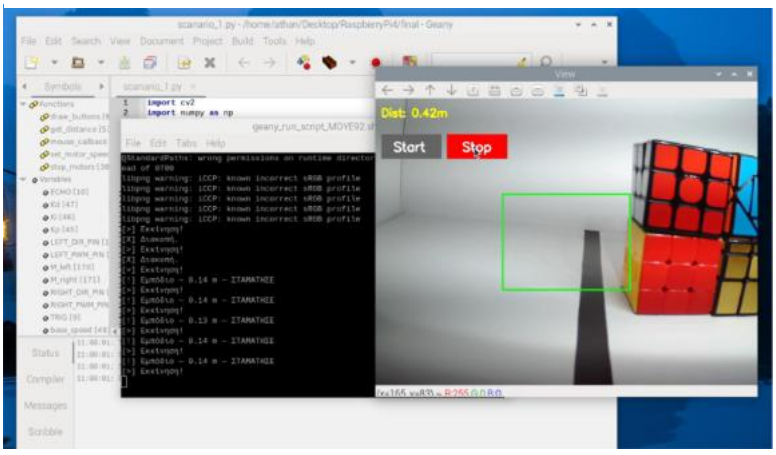
Προκειμένου να διερευνηθεί κατά πόσο το χρώμα του εμποδίου επηρεάζει την απόδοση υπό αυτές τις συνθήκες φωτισμού, πραγματοποιήθηκαν επιπλέον πειράματα, κρατώντας αμετάβλητα τα υπόλοιπα χαρακτηριστικά (μέγεθος, θέση, φωτιστική γωνία). Συγκεκριμένα, δοκιμάστηκε:

- Εμπόδιο με κόκκινη επιφάνεια (5 επαναλήψεις),
- Εμπόδιο με κίτρινη επιφάνεια (5 επαναλήψεις)

Τα αποτελέσματα έδειξαν ότι η κόκκινη επιφάνεια εντοπίστηκε σε 4 από τις 5 επαναλήψεις, ενώ η κίτρινη μόνο σε 2 από τις 5. Αυτό υποδηλώνει ότι ο συνδυασμός glare και υψηλής ανακλαστικότητας (ιδίως σε φωτεινά χρώματα όπως το κίτρινο) μπορεί να μειώσει περαιτέρω την ποιότητα της εικόνας και να δυσκολέψει τον εντοπισμό ακμών.



Εικόνα 28 - Σενάριο 2, κόκκινη επιφάνεια. Έναρξη Ανίχνευσης (Φωτισμός Μπροστά).



Εικόνα 27 - Σενάριο 2, κόκκινη επιφάνεια. Τέλος Ανίχνευσης (Φωτισμός Μπροστά).

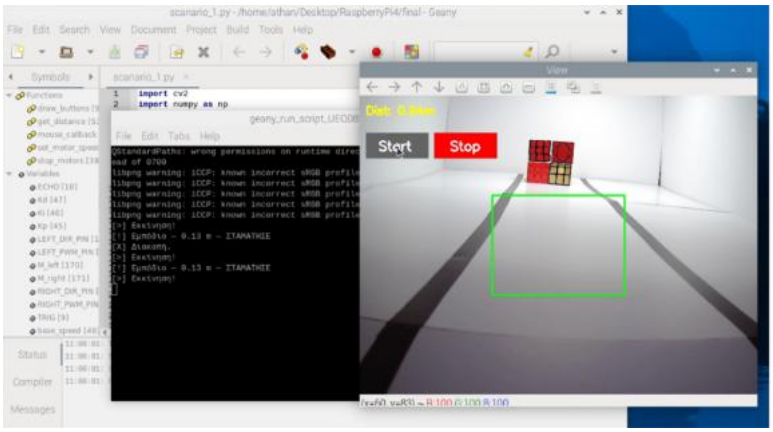
Η συνολική ακρίβεια για το Σενάριο 2 καταγράφηκε μεταβλητή, με εύρος επιτυχίας 40% έως 80% ανάλογα με τα χαρακτηριστικά του αντικειμένου. Η διαφορά στην απόδοση ενδέχεται να σχετίζεται με τη φασματική απόκριση της κάμερας, καθώς οι κάμερες RGB χαμηλού κόστους είναι

συνήθως πιο ευαίσθητες στο κόκκινο φάσμα και λιγότερο ικανές να αποδώσουν με ακρίβεια φωτεινά και πολύ ανακλαστικά χρώματα όπως το κίτρινο, ειδικά υπό έντονο φωτισμό. Το glare και η υπερέκθεση μειώνουν το τοπικό contrast, γεγονός που επηρεάζει την ικανότητα του αλγορίθμου Canny να εντοπίσει ακμές.

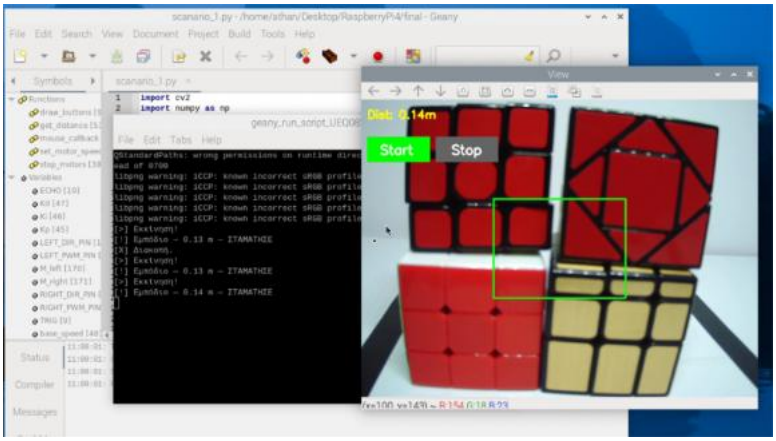
6.5.3 Σενάριο 3 – Φωτισμός από πάνω στο εμπόδιο

Η πηγή φωτός τοποθετήθηκε κατακόρυφα πάνω από το εμπόδιο, δημιουργώντας έντονο φωτισμό στο άνω μέρος και σκιά στη βάση του. Καθώς η περιοχή ενδιαφέροντος αντιστοιχεί κυρίως στο κατώτερο τμήμα της εικόνας, η σκιά αλλοίωσε την ποιότητα των ακμών στην κρίσιμη ζώνη.

Ο αλγόριθμος εντόπισε μερικώς το εμπόδιο στις περισσότερες δοκιμές, με μικρές διακυμάνσεις στην τιμή της απόστασης STOP. Παρατηρήθηκαν 1-2 αποτυχημένες εντοπίσεις, κυρίως όταν η σκιά ήταν έντονη και συγχώνευε το εμπόδιο με το φόντο. Η μέθοδος έδειξε μέτρια ανοχή στις συγκεκριμένες συνθήκες.



Εικόνα 30 - Σενάριο 3. Έναρξη Ανίχνευσης (Φωτισμός από πανω)

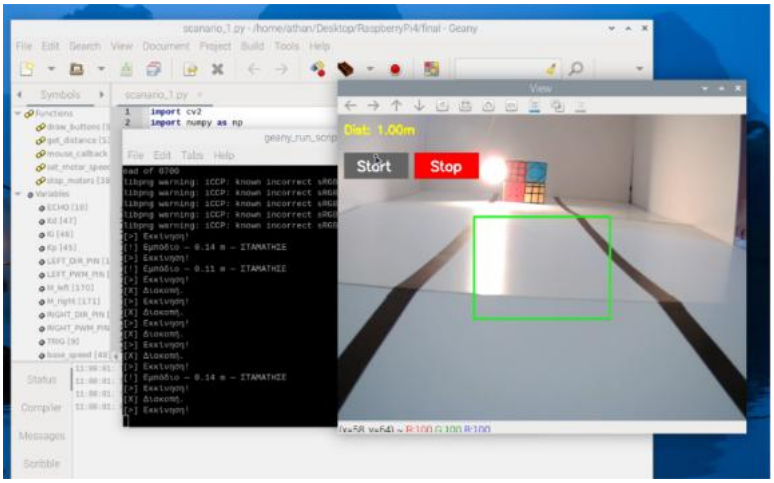


Εικόνα 29 - Σενάριο 3. Τέλος Ανίχνευσης (Φωτισμός απο πάνω)

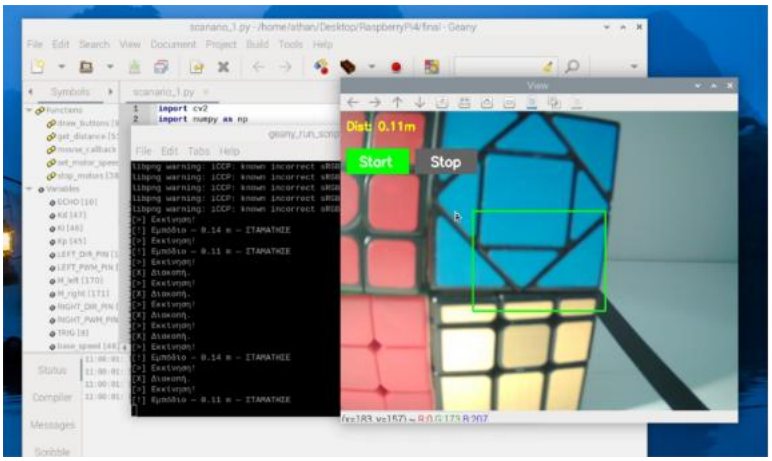
6.5.4 Σενάριο 4 – Φωτισμός από πλάγια προς την κάμερα

Η φωτεινή πηγή τοποθετήθηκε υπό πλάγια γωνία ($\sim 60^{\circ}$) ως προς την κατεύθυνση κάμερας με το εμπόδιο, τόσο από αριστερά όσο και από δεξιά του εμποδίου, ώστε να εξεταστεί η επίδραση του πλάγιου φωτισμού από διαφορετικές κατευθύνσεις.

Το εμπόδιο παρουσίασε πλευρικές σκιές και περιοχές έντονης αντανάκλασης, ιδιαίτερα σε επιφάνειες με γυαλάδα ή ανομοιογενή ύφή. Σε ορισμένες επαναλήψεις , το όχημα φάνηκε να επιβραδύνει ή να σταματά προσωρινά, παρότι δεν καταγράφηκε εντολή “STOP” στο CSV και δεν πληρούνταν τα τυπικά κριτήρια ανίχνευσης.



Εικόνα 32 - Σενάριο 4. Έναρξη Ανίχνευσης (Πλάγιος Φωτισμός)



Εικόνα 31 - Σενάριο 4. Τέλος Ανίχνευσης (Πλάγιος Φωτισμός)

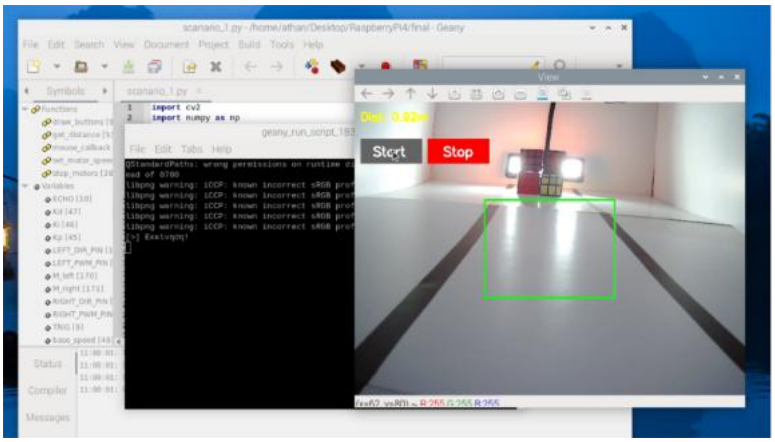
Το φαινόμενο αυτό δεν τεκμηριώθηκε ποσοτικά, αλλά ενδέχεται να οφείλεται σε προσωρινές λανθασμένες ανιχνεύσεις περιγραμμάτων, λόγω των ανακλάσεων ή των σκιών, οι οποίες δημιούργησαν οπτική αίσθηση ψευδώς θετικών STOP (false positives). Η μεταβλητότητα στην αντίδραση του οχήματος ήταν υψηλή, με διακυμάνσεις ανά επανάληψη, γεγονός που υποδεικνύει την ευαισθησία της μεθόδου σε συνθήκες πλάγιου και ανομοιογενούς φωτισμού.

6.5.5 Σενάριο 5 – Φωτισμός από πίσω προς την κάμερα

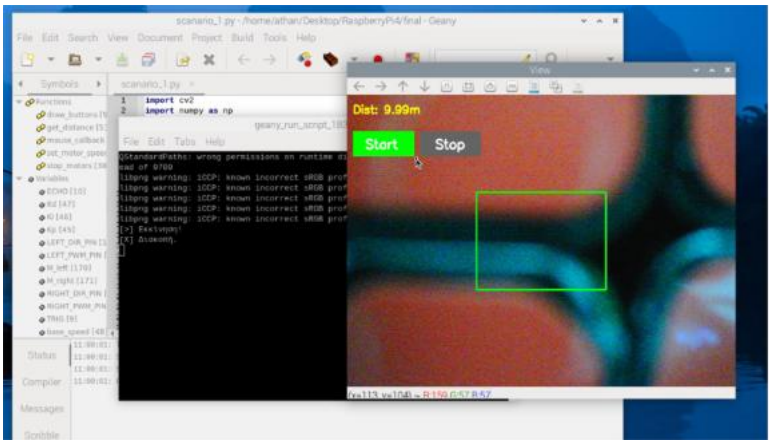
Σε αυτό το σενάριο, η κάμερα βλέπει το εμπόδιο με έντονη φωτεινή πηγή στο παρασκήνιο. Το αντικείμενο προβάλλεται σκοτεινό σε έντονα φωτεινό φόντο, με χαμηλή αντίθεση και ελάχιστες διαφορές έντασης μεταξύ ROI και background.

Το αποτέλεσμα ήταν πλήρης αποτυχία ανίχνευσης σε αρκετές επαναλήψεις, καθώς δεν εντοπίζονταν επαρκείς ακμές. Σε λίγες περιπτώσεις η μέθοδος ανταποκρίθηκε οριακά. Συνολικά,

το σενάριο αυτό αποδείχθηκε το πλέον απαιτητικό, με ποσοστό αποτυχίας 100% στο Raspberry Pi,επιβεβαιώνοντας την ευαισθησία του αλγορίθμου σε συνθήκες χαμηλού contrast.



Εικόνα 33 - Σενάριο 5. Έναρξη Ανίχνευσης (Φωτισμός απο πίσω)



Εικόνα 34 - Σενάριο 5. Τέλος Ανίχνευσης (Φωτισμός απο πίσω)

6.6 Αποτελέσματα Ανίχνευσης

Η ποσοτική αξιολόγηση της απόδοσης του συστήματος ανίχνευσης εμποδίων στο Raspberry Pi πραγματοποιήθηκε με βάση πέντε επαναλήψεις για κάθε σενάριο φωτισμού. Σε κάθε επανάληψη καταγράφηκε η αντίδραση του οχήματος, η επιτυχής ή αποτυχημένη ανίχνευση, καθώς και η απόσταση στην οποία ενεργοποιήθηκε το φρενάρισμα, εφόσον αυτό πραγματοποιήθηκε

Στο Σενάριο 1, η μέθοδος εμφάνισε εξαιρετική σταθερότητα. Και στις πέντε επαναλήψεις εντοπίστηκε σωστά το εμπόδιο, με άμεση απόκριση και πολύ μικρές διακυμάνσεις στην απόσταση φρεναρίσματος (0.12m), καταδεικνύοντας τη βέλτιστη συμπεριφορά του συστήματος σε ομοιογενείς συνθήκες φωτισμού.

Το Σενάριο 2 παρουσίασε αυξημένη πολυπλοκότητα, ιδιαίτερα όσον αφορά τον χρωματισμό του εμποδίου. Με το κόκκινο αντικείμενο, το σύστημα ανταποκρίθηκε επιτυχώς σε 4 από τις 5 επαναλήψεις. Με το κίτρινο αντικείμενο, όμως, η επιτυχία μειώθηκε στις 2/5, γεγονός που υποδεικνύει ότι το glare σε συνδυασμό με την αυξημένη ανακλαστικότητα του φωτεινού χρώματος επηρεάζει αρνητικά την ικανότητα εντοπισμού.

Στο Σενάριο 3, η μέθοδος λειτουργούσε σωστά και στις πέντε επαναλήψεις, αν και παρατηρήθηκαν μικρές αποκλίσεις στις αποστάσεις STOP, πιθανώς λόγω της σκιάς που σχηματιζόταν στη βάση του εμποδίου.

Αντίθετα, το Σενάριο 4, ανέδειξε τη δυσκολία του συστήματος σε περιπτώσεις ανομοιόμορφου φωτισμού και σκιών. Μόνο 3 από τις 5 επαναλήψεις κατέληξαν σε επιτυχή αναγνώριση του εμποδίου, ενώ στις υπόλοιπες η πορεία συνεχίστηκε με βάση το Lane Detection.

Το πλέον απαιτητικό ήταν το Σενάριο 5, όπου σε καμία από τις πέντε επαναλήψεις δεν κατεγράφη επιτυχής εντοπισμός. Το εμπόδιο εμφανιζόταν σκοτεινό με χαμηλό contrast σε φωτεινό υπόβαθρο, καθιστώντας αδύνατη την εξαγωγή σταθερών ακμών από τον αλγόριθμο Canny Edge Detector. Παρακάτω ακολουθεί ο συγκεντρωτικός πίνακας των αποτελεσμάτων.

Πίνακας 4 - Συγκεντρωτικά αποτελέσματα ανίχνευσης εμποδίων στο Raspberry Pi.

Σενάριο	**Δοκιμή**	**Απόσταση STOP(m)**	**Ανίχνευση Εμποδιού**	**Ενέργεια Οχήματος**
Scenario_1	1	0.12	YES	STOP
	2	0.12	YES	STOP
	3	0.12	YES	STOP
	4	0.12	YES	STOP
	5	0.12	YES	STOP
Scenario_2_red	1	-	NO	LANE FOLLOW
	2	0.14	YES	STOP
	3	0.14	YES	STOP
	4	0.13	YES	STOP
	5	0.14	YES	STOP

Scenario_2_yellow	1	0.14	YES	STOP
	2	-	NO	LANE FOLLOW
	3	0.11	YES	STOP
	4	-	NO	LANE FOLLOW
	5	-	NO	LANE FOLLOW
Scenario_3	1	0.13	YES	STOP
	2	0.13	YES	STOP
	3	0.14	YES	STOP
	4	0.11	YES	STOP
	5	0.12	YES	STOP
Scenario_4	1	0.11	YES	STOP
	2	0.14	YES	STOP
	3	0.11	YES	STOP
	4	-	NO	LANE FOLLOW
	5	-	NO	LANE FOLLOW
Scenario_5	1	-	NO	LANE FOLLOW
	2	-	NO	LANE FOLLOW
	3	-	NO	LANE FOLLOW
	4	-	NO	LANE FOLLOW
	5	-	NO	LANE FOLLOW

Η ανάλυση του πίνακα αποκαλύπτει ότι το σύστημα παρουσιάζει υωηλή απόδοση σε συνθήκες κανονικού φωτισμού, αλλά η ευαισθησία του σε υπερέκθεση, σκιάσεις ή πίσω φωτισμό οδηγεί σε σημαντική μείωση της αξιοπιστίας. Οι αποκλίσεις αυτές τονίζουν την ανάγκη περαιτέρω ενίσχυσης της μεθόδου μέσω πιο ανθεκτικών τεχνικών εντοπισμού ακμών ή χρήσης πολυτροπικών αισθητήρων.

Σημειώνεται ότι κατά την εκτέλεση των δοκιμών έγινε καταγραφή βίντεο της πορείας τπυ ρομποτικού οχήματος, με στόχο την επιβεβαίωση της πραγματικής του συμπεριφοράς σε περιπτώσεις όπου το CSV δεν κατέγραψε εντολή STOP. Η οπτική ανασκόπηση των βίντεο αποκάλυψε περιπτώσεις στις οποίες το όχημα επιβράδυνε ή σταματούσε προσωρινά, ενδεχομένως λόγω παροδικών ανιχνεύσεων εντός περιοχής ενδιαφέροντος, οι οποίες όμως δεν συνοδεύονταν από πλήρη καταγραφή στο σύστημα logging.

6.7 Συμπεράσματα

Η πειραματική εφαρμογή του συστήματος σε πραγματικό ρομποτικό όχημα επιβεβαίωσε ότι η μετάβαση από προσομοιωτικό σε φυσικό περιβάλλον απαιτεί λεπτομερή σχεδιασμό και σημαντικές προσαρμογές τόσο στο υλικό όσο και στο λογισμικό. Η χρήση του Raspberry Pi 4 αποδείχθηκε επαρκής για την εκτέλεση αλγορίθμων επεξεργασίας εικόνας σε πραγματικό χρόνο, ενώ οι κινητήρες DV σε συνδυασμό με τον οδηγό L9110 και τα σήματα PWM εξασφάλισαν ομαλή και ακριβή κίνηση του οχήματος.

Η ενσωμάτωση του υπερηχητικού αισθητήρα HC-SR04 επέτρεψε αξιόπιστη μετρηση αποστασεων από εμπόδια, ενώ η κάμερα χρησιμοποιήθηκε για ανίχνευση εμποδίων και παρακολούθηση της πορείας μέσω του συστήματος Lane Detection. Η προστασία των GPIO pins με διαιρέτη τάσης εξασφάλισε την ασφαλή λειτουργία της πλατφόρμας. Η εφαρμογή του συστήματος Lane Detection ήταν ακθοριστικής σημασίας, καθώς αντιμετώπισε αποκλίσεις στην πορεία λόγω κατασκευαστικών ανωμαλιών, διαφορών στη ροπή των κινητήρων ή ακανόνιστων χαρακτηριστικών της επιφάνειας κίνησης, γεγονός που δεν υπήρχε στους προσομοιωτές.

Συνολικά, τα αποτελέσματα δείχνουν ότι το σύστημα μπορεί να λειτουργεί αξιόπιστα σε πραγματικό περιβάλλον, διατηρώντας την ασφάλεια και την ομαλή πορεία του οχήματος. Παράλληλα, η εμπειρία αυτή παρέχει σημαντική γνώση για την επέκταση του συστήματος σε πιο σύνθετες εφαρμογές ρομποτικής και αυτόνομης πλοήγησης, όπως η διαχείριση πολύπλοκων διαδρομών, η αποφυγή δυναμικών εμποδίων και η συνεργασία με άλλα οχήματα σε πραγματικό χρόνο.

7

Αξιολόγηση Απόδοσης Ανίχνευσης Εμποδίων

7.1 Εισαγωγή

Η αξιολόγηση της απόδοσης ενός συστήματος ανίχνευσης εμποδίων αποτελεί το πιο κρίσιμο στάδιο για την επιβεβαίωση της αξιοπιστίας και της λειτουργικότητας του. Ο στόχος της διαδικασίας αυτής είναι να διασφαλιστεί ότι το σύστημα μπορεί να αναγνωρίσει εμπόδια με ακρίβεια και ταχύτητα, αποτρέποντας ενδεχόμενες ατυσκρορύσεις σε περιβάλλοντα προσομοίωσης ή στον πραγματικό κόσμο.

Η αξιολόγηση οραγματοποιήθηκε σε δύο επίπεδα. Πρώτον,σε περιβάλλον προσομοίωσης, χρησιμοποιώντας τους προσομοιωτές CARLA και AirSim, οι οποίοι επιτρέπουν τον έλεγχο και τη μεταβολή των συνθηκών φωτισμού, της υφής του εδάφους και της θέσης των εμποδίων με μεγάλη ακρίβεια. Δεύτερον, σε πραγματική υλοποίηση, με χρήση ενός Raspberry Pi 4, ώστε να μελετηθεί η συμπεριφορά του συστήματος σε πραγματικές συνθήκες, όπου οι παράγοντες περιβάλλοντος είναι λιγότερο ελεγχόμενοι.

Η μέτρηση της απόδοσης δεν περιορίστηκε στην καταγραφή ποσοστών επιτυχίας, αλλά περιέλαβε την ποσοτική ανάλυση των αποτελεσμάτων με τη χρήση τυποποιημένων δεικτών, όπως Precision, Recall. Επιπλέον, καταγράφηκαν οι βασικοί δείκτες True Positives (TP), False Positives (FP) και False Negatives (FN), οι οποίοι επιτρέπουν τη λεπτομερή ανάλυση της απόδοσης ανά σενάριο.

Η συγκριτική ανάλυση ανάμεσα σε προσομοίωση και πραγματικό υλικό επιτρέπει τον εντοπισμό του Sim2Real gap, δηλαδη τηε απόστασης μεταξύ της συμπεριφοράς του συστήματος στον εικονικό κόσμο και στον πραγματικό κόσμο. Αυτό καθιστά δυνατή την εξαγωγή συμπερασμάτων σχετικά με όριο και τις δυνατότητες βελτίωσης της μεθόδου.

7.2 Μετρικές Αξιολόγησης

Η αξιολόγηση της απόδοσης βασίστηκε στον υπολογισμό των δεικτών Precision και Recall οι οποίοι παρέχουν μια ποσοτική εικόνα για την ακρίβεια και την πληρότητα των ανιχνεύσεων του συστήματος

Οι βασικοί ορισμοί που χρησιμοποιήθηκαν είναι οι εξής:

- True Positives (TP): πρόκειται για περιπτώσεις όπου το σύστημα ανίχνευσε σωστά ένα εμπόδιο που βρισκόταν σε κρίσιμο σημείο, δηλαδή σε απόσταση μικρότερη των 10 μέτρων και παρήγαγε την κατάλληλη ένδειξη" STOP". Αυτό σημαίνει ότι το σύστημα λειτούργησε σωστά, αποτρέποντας πιθανή σύγκρουση.
- False Positives (FP): είναι οι περιπτώσεις στις οποίες το σύστημα εξέδωσε ένδειξη "STOP" ενώ στην πραγματικότητα δεν υπήρχε εμπόδιο. Τα ψευδώς θετικά σήματα μπορούν να επηρεάσουν αρνητικά τη ροή της λειτουργίας του συστήματος, δημιουργώντας περιττές διακοπές ή ανησυχία στον χρήστη.
- False Negatives (FN): αφορούν τις περιπτώσεις όπου υπήρχε πραγματικό εμπόδιο σε απόσταση μικρότερη των 10m, αλλά το σύστημα δεν το ανίχνευσε. Τα ψευδώς αρνητικά αποτελέσματα θεωρούνται πιο επικίνδυνα, καθώς μπορούν να οδηγήσουν σε ατυχήματα ή συγκρούσεις λόγω μη ανίχνευσης των εμποδίων.

Με βάση αυτούς τους δείκτες, οι δύο βασικές μετρικές αξιολόγησης ορίζονται ως εξής:

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

Το Precision εκφράζει την ακρίβεια των προβλέψεων του συστήματος, δηλαδή το ποσοστό των ειδοποιήσεων "STOP" που ήταν σωστές σε σχέση με το σύνολο των ειδοποιήσεων που δόθηκαν. Ένα υψηλό Precision υποδεικνύει ότι το σύστημα αποφεύγει τα περιττά σήματα και μειώνει τον αριθμό των ψευδών θετικών αποτελεσμάτων, ενισχύοντας έτσι την αξιοπιστία του και την εμπιστοσύνη του χρήστη.

Αντίθετα, το Recall μετρά την πληρότητα της ανίχνευσης, δηλαδή την ικανότητα του συστήματος να εντοπίζει όλα τα πραγματικά εμπόδια. Ένα υψηλό Recall σημαίνει ότι το σύστημα χάνει ελάχιστα εμπόδια, μειώνοντας τον κίνδυνο ατυχημάτων και εξασφαλίζοντας υψηλή ασφάλεια κατά τη λειτουργία.

Για μια πιο ολοκληρωμένη και ισορροπημένη εκτίμηση της απόδοσης, χρησιμοποιήθηκε και ο δείκτεις F1-score, ο οποίος συνδυάζει Precision και Recall σε μία ενιαία μετρική ως εξής:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Ο F1-score είναι ιδιαίτερα χρήσιμος όταν υπάρχει ασυμμετρία μεταξύ ψευδώς θετικών και ψευδώς αρνητικών αποτελεσμάτων, καθώς δίνει μια συνολική εικόνα της απόδοσης του συστήματος. Η χρήση του επιτρέπει να αξιολογείται το σύστημα με τρόπο που λαμβάνει υπόψη

τόσο την ακρίβεια όσο και την πληρότητα, προσφέροντας μια πιο ρεαλιστική εκτίμηση της ικανότητας του να λειτουργεί αξιόπιστα σε πραγματικές συνθήκες.

7.3 Ανάλυση Αποτελεσμάτων

Η προσέγγιση που βασίζεται στη λογική Canny + ROI + Contours > 600 px αποδείχθηκε απλή και αποδοτική, ιδιαίτερα σε σενάρια με καθαρό διαχωρισμό αντικειμένων και φωτεινές συνθήκες. Ωστόσο, παρουσιάστηκαν ψευδώς θετικές και ψευδώς αρνητικές ανιχνεύσεις σε περιβάλλοντα με σκιές ή ετερόφωτο φόντο.

Για την αξιολόγηση της απόδοσης, παραθέτουμε τον παρακάτω πίνακα Precision / Recall για κάθε σενάριο:

Πίνακας 5 - Ανάλυση Απόδοσης Αλγορίθμου Ανίχνευσης Εμποδίων ανά Σενάριο και Προσομοιωτή.

Σενάριο	Προσομοιωτής	Precision	Recall	F1-score	Παρατηρήσεις
Scenario_1-Full Lights	CARLA	0.00	0.00	0.00	Υπερέκθεση, μηδενική ανίχνευση
Scenario_2-Normal Lights	CARLA	1.00	0.02	0.03	Χαμηλή ανάκτηση
Scenario_3-No Lights	CARLA	1.00	0.05	0.09	Σταθερή ακρίβεια, μειωμένη ανάκτηση
Scenario_4-Wet Road	CARLA	0.006	0.20	0.01	Αντανακλάσεις, υψηλά false positives
Scenario_5-Deep Shadow	CARLA	1.00	0.009	0.01	Περιορισμός λόγω σκιών
Scenario_A-Normal Lighting	AirSim	1.00	0.013	0.02	Χαμηλή ανάκτηση
Scenario_B-Spotlight to Camera	AirSim	0.00	0.00	0.00	Glare, πλήρης αποτυχία
Scenario_C-Spotlight on Object	AirSim	0.00	0.00	0.00	Υπερέκθεση του εμποδίου
Scenario_D-Backlight	AirSim	1.00	0.012	0.02	Rim lighting βελτίωσε ανίχνευση

Η αξιολόγηση δείχνει πως η απόδοση του συστήματος εξαρτάται σημαντικά από τις συνθήκες φωτισμού και την ποιότητα των ακμών στο οπτικό πεδίο. Σε περιπτώσεις υπερέκθεσης, σκιών ή θορύβου, η μέθοδος Canny αντιμετωπίζει δυσκολίες στην έγκαιρη και σωστή ανίχνευση εμποδίων.

Παράλληλα, στο Raspberry Pi εκτελέστηκαν πέντε σενάρια, αντίστοιχα με εκείνα του AirSim. Η συνολική απόδοση για κάθε σενάριο παρουσιάζεται στον Πινακα , με καταγραφή του αριθμού επιτυχών εντοπισμών ανά 5 επαναλήψεις.

Πίνακας 6 - Συνολική Επιτυχία Ανίχνευσης Εμποδίων στο Raspberry Pi 4 ανά Σενάριο.

Σενάριο	**CARLA**	**AirSim**	**Raspberry Pi 4**
Normal Ambient Light	5/5	5/5	5/5
Glare (φωτισμός από κάμερα)	5/5	5/5	4/5(κόκκινο),2/5(κίτρινο)
Overhead illumination (φως από πάνω)	-	5/5	5/5
Side Light (πλάγιος φωτισμός)	-	5/5	3/5
Backlight (φως από πίσω)	-	5/5	0/5

Σημαντικό στοιχείο της αξιολόγησης είναι ότι τα σενάρια που χρησιμοποιήθηκαν στον προσομοιωτή CARLA περιλάμβαναν ειδικές συνθήκες, όπως το "υγρός δρόμος" και η "βαθία σκιά", οι οποίες δεν εφαρμόστηκαν αυτούσια στη φυσική υλοποίηση με το Raspberry Pi. Αυτές οι συνθήκες δημιουργούν ιδιαίτερα απαιτητικά περιβάλλοντα για την ανίχνευση εμποδίων και η απουσία τους στο Raspberry Pi περιορίζει την άμεση σύγκριση. Αντίθετα, τα σενάρια που χρησιμοποιήθηκαν στον προσομοιωτή AirSim ήταν ίδια με αυτά που εκτελέστηκαν στο Raspberry Pi, επιτρέποντας μια πιο άμεση και συγκρίσιμη αξιολόγηση της απόδοσης του αλγορίθμου υπό παρόμοιες συνθήκες φωτισμού.

Τα αποτελέσματα δείχνουν σαφή σταθερότητα και υψηλή ακρίβεια στους προσομοιωτές, τόσο στο CARLA όσο και στο AirSim. Αντίθετα, η εκτέλεση στο Raspberry Pi παρουσίασε μεγαλύτερες αποκλίσεις και μειωμένη αξιοπιστία σε ορισμένα σενάρια. Οι κύριες αιτίες αυτών των διαφορών σχετίζονται με τον περιορισμό της συναμικής εμβέλειας της κάμερας USB, τη μη ιδανική ισορροπία λευκού και τη λειτουργία αυτόματης έκθεσης, καθώς και τον ετερογενή φωτισμό στο φυσικό περιβάλλον. Ιδιαίτερα στο σενάριο Backlight, παρατηρήθηκε ότι το φαινόμενο rim lighting στο AirSim συνέβαλε στη βελτίωση της ανίχνευσης, ενώ στο Raspberry Pi η ίδια συνθήκη οδήγησε σε πλήρη αποτυχία (0/5 επιτυχημένες ανιχνεύσεις), λόγω χαμηλής αντίθεσης και έντονου θορύβου στην εικόνα.

7.4 Συμπεράσματα

Η ανάλυση των αποτελεσμάτων ανέδειξε ότι η μέθοδος μπορεί να λειτουργήσει αποτελεσματικά σε ιδανικές συνθήκες φωτισμού και με καθαρό οπτικό πεδίο, παρέχοντας υψηλά ποσοστά ακρίβειας στους προσομοιωτές CARLA και AirSim. Ωστόσο, η ευαισθησία της μεθόδου σε παράγοντες όπως οι σκιές, η υπερέκθεση και οι αντανακλάσεις περιορίζει την αξιοπιστία της σε πιο απαιτητικά περιβάλλοντα, οδηγώντας σε αυξημένα ποσοστά ψευδώς θετικών και ψευδώς αρνητικών ανιχνεύσεων.

Η σύγκριση μεταξύ των προσομοιωτών και της πραγματικής υλοποίησης στο Raspberry Pi αποκάλυψε το χάσμα Sim2Real, το οποίο αποδίδεται σε περιορισμούς του υλικού αλλά και στην πολυπλοκότητα των φυσικών συνθηκών φωτισμού. Ιδιαίτερα το σενάριο Backlight ανέδειξε τη δυσκολία μετάβασης από το ελεγχόμενο περιβάλλον του AirSim, όπου το rim lighting βοήθησε την ανίχνευση, σε πραγματικές συνθήκες όπου το ίδιο σενάριο απέτυχε πλήρως.

Συνολικά, τα αποτελέσματα δείχνουν ότι ενώ οι προσομοιωτές αποτελούν ένα πολύτιμο εργαλείο για την αρχική αξιολόγηση αλγορίθμων, η μετάβαση σε πραγματικές συνθήκες παραμένει κρίσιμη πρόκληση. Για να βελτιωθεί η αξιοπιστία του συστήματος, απαιτούνται πιο ανθεκτικές μέθοδοι ανίχνευσης που να ενσωματώνουν πληροφορίες πέραν των απλών ακμών, όπως συνδυασμό μεθόδων μηχανικής μάθησης ή χρήση επιπλέον αισθητήρων. Η εμπειρία από την παρούσα υλοποίηση καταδεικνύει τη σημασία της δοκιμής σε πραγματικά περιβάλλοντα, ώστε να εντοπίζονται έγκαιρα οι περιορισμοί και να σχεδιάζονται στρατηγικές βελτίωσης.

8

Συμπεράσματα

8.1 Εισαγωγή

Το παρόν κεφάλαιο συνοψίζει τα βασικά αποτελέσματα και τα συμπεράσματα που προέκυψαν από την εκπόνηση της διπλωματικής εργασίας, εστιάζοντας στην αξιολόγηση της προτεινόμενης μεθοδολογίας ανίχνευσης εμποδίων για αυτόνομα οχήματα. Η μελέτη αυτή επικεντρώθηκε στην ανάπτυξη ενός συστήματος που συνδυάζει τεχνικές επεξεργασίας εικόνας από εμπρόσθια κάμερα με αισθητήρες απόστασης, προκειμένου να επιτευχθεί αξιόπιστη ανίχνευση εμποδίων σε διαφορετικά περιβάλλοντα. Η εισαγωγή αυτού του κεφαλαίου αποσκοπεί στο να περοσφέρει μια συνολική εικόνα και των περιορισμών που αναδείχθηκαν κατά την εφαρμογή της μεθοδολογίας.

Η ανάλυση οργανώνεται σε τρείς βασικές ενότητες. Πρώτον, παρουσιάζονται τα συνοπτικά αποτελέσματα που καταγράφηκαν τόσο σε προσομοιωτικά περιβάλλοντα όσο και σε πραγματικές συνθήκες, δίνοντας έμφαση στην αξιοπιστία και την ακρίβεια του συστήματος. Δεύτερον, αναλύονται οι περιορισμοί της μεθοδολογίας, οι οποίοι σχετίζονται με παράγοντες όπως οι μεταβαλλόμενες συνθήκες φωτισμού, η απουσία δυναμικών εμποδίων και οι περιορισμένοι υπολογιστικοί πόροι της πλατφόρμας υλοποίησης. Τέλος, παρουσιάζονται προτάσεις για μελλοντική έρευνα, που επικεντρώνονται στην ενίσχυση της ανθεκτικότητας, στη δυνατότητα παρακολούθησης κινητών εμποδίων, στην πολυτροπικότητα αισθητήρων και στην αξιοποίηση τεχνικών βαθίας μάθησης και ενισχυτικής μάθησης.

8.2 Συνοπτικά Αποτελέσματα

Η παρούσα διπλωματική εργασία είχε ως βασικό στόχο την ανάπτυξη, υλοποίηση και αξιολόγηση μιας μεθοδολογίας ανίχνευσης εμποδίων για αυτόνομα οχήματα, βασισμένη κυρίως στην επεξεργασία εικόνας από εμπρόσθια κάμερα, σε συνδυασμό με αισθητήρες απόστασης. Η μελέτη εστίασε στην κατανόηση της συμπεριφοράς του συστήματος υπό διαφορετικές συνθήκες, τόσο σε προσομοιωτικά περιβάλλοντα όσο και σε πραγματικές δοκιμές.

Στα αρχικά κεφάλαια έγινε αναλυτική αναφορά στη σημασία της οπτικής ανίχνευσης στην αυτόνομη οδήγηση. Τονίστηκε ότι οι κλασικές τεχνικές επεξεργασίας εικόνας, όπως η ανίχνευση ακμών μέσω του αλγορίθμου Canny, παραμένουν ιδιαίτερα σημαντικές λόγω της χαμηλής υπολογιστικής τους απαίτησης και της δυνατότητας εφαρμογής τους σε πλατφόρμες με περιορισμένους πόρους. Η εισαγωγή περιοχής ενδιαφέροντος (ROI) στο κάτω και μεσαίο τμήμα της εικόνας αποδείχθηκε καθοριστική, καθώς επέτρεψε τη μείωση του όγκου δεδομένων που επεξεργάζεται ο αλγόριθμος, χωρίς να χαθεί η κρίσιμη πληροφορία για την ανίχνευση εμποδίων.

Η πρώτη φάση αξιολόγησης πραγματοποιήθηκε στο προσομοιωτικό περιβάλλον CARLA, όπου ελέγχθηκε η ορθή λειτουργία του συστήματος σε ελεγχόμενες αυνθήκες κυκλοφορίας. Τα πειράματα έδειξαν ότι το όχημα μπορούσε να εντοπίζει με ακρίβεια την παρουσία την παρουσία στατικών εμποδίων και να λαμβάνει την κατάλληλη απόφαση ακινητοποίησης. Παράλληλα, η καταγραφή δεδομένων σε επίπεδο frame διευκόλυνε την ανάλυση της συμπεριφοράς του συστήματος και επέτρεψε τη συστηματική τεκμηρίωση των αποτελεσμάτων.

Η δεύτερη βάση αξιολόγησης έλαβε χώρα στο AirSim, όπου το βάρος δόθηκε στη μελέτη της επίδρασης των συνθηκών φωτισμού στην ανίχνευση αμποδίων. Σχεδιάστηκαν και υλοποιήθηκαν τέσσερα διαφορετικά σενάρια φωτισμού, με ελεγχόμενη χρήση spotlights. Τα αποτελέσματα αποκάλυψαν ότι το σύστημα επιτυγχάνει ικανοποιητική απόδοση σε κανονικές συνθήκες αλλά παρουσιάζει σημαντικές αδυναμίες σε περιπτώσεις έντονου φωτισμού ή glare, όπου το εμπόδιο δεν ανιχνεύθηκε. Αντίθετα, όταν υπήρχε έμμεσος φωτισμός (rim lighting), η ενίσχυση των περιγραμμάτων συνέβαλε θετικά στην αναγνώριση.

Τέλος, το σύστημα δοκιμάστηκε σε πραγματικές συνθήκες, με υλοποίηση σε ρομποτικό όχημα βασισμένο σε Raspberry Pi 4. Παρά τους περιορισμένους υπολογιστικούς πόρους, η μεθοδολογία κατάφερε να λειτουργήσει σε πραγματικό χρόνο, να ανιχνεύσει εμπόδια με χρήση κάμερας και υπερηχητικού αισθητήρα, και να οδηγήσει σε έγκαιρη ακινητοποίηση του οχήματος. Τα αποτελέσματα αυτής της φάσης ενισχύουν την πρακτική αξία της μεθόδου και αποδεικνύουν τη δυνατότητα εφαρμογής της σε συστήματα χαμηλού κόστους.

8.3 Περιορισμοί

Παρά τα ενθαρρυντικά αποτελέσματα και τη συμβολή της παρούσας εργασίας στην κατανόηση και αξιολόγηση μεθόδων ανίχνευσης εμποδίων, υπάρχουν συγκεκριμένοι περιορισμοί που πρέπει να αναγνωριστούν. Η κατανόηση αυτών των περιορισμών είναι απαραίτητη για να εκτιμηθεί σωστά η αξία της μελέτης και να καθοριστούν ρεαλιστικές κατευθύνσεις για μελλοντική έρευνα.

Ένας πρώτος και ιδιαίτερα εμφανής περιορισμός αφορά την ευαισθησία στις συνθήκες φωτισμού. Τα πειράματα στο AirSim ανέδειξαν ότι η μεθοδολογία, η οποία βασίζεται αποκλειστικά στην ανάλυση ακμών μέσω του αλγορίθμου Canny, αποτυγχάνει σε περιπτώσεις έντονου υπερφωτισμού ή glare. Σε τέτοιες περιπτώσεις, το εμπόδιο «εξαφανίζεται» οπτικά, καθιστώντας αδύνατη την ανίχνευσή του. Στο πλαίσιο μιας πραγματικής οδήγησης, το πρόβλημα αυτό μπορεί να εμφανιστεί όταν ο ήλιος βρίσκεται χαμηλά στον ορίζοντα, όταν η αντανάκλαση από υγρές επιφάνειες είναι έντονη ή όταν φώτα από διερχόμενα οχήματα τυφλώνουν προσωρινά τον αισθητήρα. Η ευαισθησία αυτή περιορίζει την αξιοπιστία του συστήματος σε πραγματικές συνθήκες, όπου ο φωτισμός μεταβάλλεται συνεχώς και δεν είναι δυνατό να ελεγχθεί.

Δεύτερος περιορισμός είναι η εστίαση αποκλειστικά σε στατικά εμπόδια. Παρότι η μελέτη αποδεικνύει ότι η προτεινόμενη μεθοδολογία μπορεί να λειτουργήσει σε βασικές περιπτώσεις, η πραγματικότητα είναι πολύ πιο περίπλοκη. Στη καθημερινή οδική εμπειρία, τα εμπόδια δεν είναι πάντοτε ακίνητα, μπορεί να είναι πεζοί, ποδηλάτες, ζώα ή άλλα οχήματα που κινούνται απρόβλεπτα, δημιουργώντας σενάρια που απαιτού όχι μόνο ανίχνευση αλλά και πρόβλεψη της μελλοντικής τους θέσης. Η απουσία δυναμικών εμποδίων από την αξιολόγηση περιορίζει σημαντικά τη δυνατότητα γενίκευσης των αποτελεσμάτων.

Τρίτος περιορισμός συνδέεται με την υπολογιστική ισχύ της πλατφόρμα Raspberry Pi 4. Αν και το Raspberry Pi αποτελεί μια οικονομική και ευέλικτη λύση, οι υπολογιστικοί του πόροι δεν επαρκούν για την υλοποίηση πιο προηγμένων μεθόδων, όπως μοντέλα βαθιάς μάθησης για ανίχνευση αντικειμένων σε πραγματικό χρόνο. Αυτό σημαίνει ότι το σύστημα, στην παρούσα μορφή, είναι περιορισμένο σε απλές τεχνικές επεξεργασίας εικόνας.

Τέλος, σημαντικός περιορισμός αποτελεί και η απλότητα της διαμόρφωσης των δρόμων που εξετάστηκαν. Τα πειράματα πραγματοποιήθηκαν σε ευθύγραμμα τμήματα δρόμου χωρίς περίπλοκες διασταυρώσεις, κυκλικούς κόμβους, έντονες στροφές ή ποικιλία αστικών συνθηκών. Αντίστοιχα, δεν εξετάστηκαν περιβάλλοντα με πολλαπλά εμπόδια που ενδέχεται να κρύβουν το ένα το άλλο. Αυτή η απλοποίηση ήταν αναγκαία για να αξιολογηθεί μεμονωμλενα η μεθοδολογία, ωστοσο περιορίζει τη δυνατότητα εξαγωγής συμπερασμάτων για πιο σύνθετες και ρεαλιστικές καταστάσεις.

8.4 Μελλοντικές Κατευθύνσεις

Η παρούσα εργασία μπορεί να αποτελέσει αφετηρία για περαιτέρω έρευνα και βελτίωση σε πολλαπλά επίπεδα. Οι μελλοντικές κατευθύνσεις που προτείνονται εδώ σενδέονται άμεσα με τους περιορισμούς και σε νέες τεχνολογικές προοπτικές.

Μια πρώτη και άμεση κατεύθυνση αφορά τη βελτίωση της ανθεκτικότητας σε διαφορετικές φωτιστικές συνθήκες. Η ενσωμάτωση μεθόδων βαθιάς μάθησης, όπως τα μοντέλα YOLO (You Only Look Once) ή SSD (Single Shot Detector), μπορεί να ξεπεράσει τα όρια των κλασικών αλγορίθμων επεξεργασίας εικόνας, καθώς μαθαίνουν να αναγνωρίζουν αντικείμενα ακόμη και σε εικόνες με θόρυβο, σκιάσεις ή μερική απίκρυψη. Παράλληλα, τεχνικές όπως το data augmentation (ενίσχυση δεδομένων μέσω τεχνητής παραλλαγής) και domain adaption (προσαρμογή μοντέλων από το εικονικό στο πραγματικό περιβάλλον – Sim2Real) μπορούν να βοηθήσουν ώστε τα μοντέλα να λειτουργούν με μεγαλύτερη αξιοπιστία εκτός του πλαισίου της προσομοίωσης(Hu *et al.*, 2023).

Μια δεύτερη προοπτική αφορά την ανίχνευση και παρακολούθηση δυναμικών εμποδίων. Η εφαρμογή τεχνικών object tracking, όπως Kalman Filter, SORT ή DeepSORT, μπορεί να επιτρέψει στο σύστημα όχι μόνο να ανιχνεύει αλλά και να προβλέπει την κίνηση ενός αντικειμένου. Σε συνδυασμό με αλγορίθμους πρόβλεψης τροχίας, το όχημα θα είναι σε θέση να λαμβάνει πιο σύνθετες αποφάσεις, όπως αλλαγή λωρίδας ή ελιγμούς αποφυγής, γεγονός που θα το φέρει πιο κοντά σε επίπεδα πραγματικής αυτόνομης οδήγησης(*2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance : 27-30 November 2018, Auckland, New Zealand*, 2018).

Τρίτη σημαντική κατεύθυνση είναι η πολυτροπικότητα αισθητήρων, η εξάρτηση από μία μονο κάμερα RGB αποτελεί περιορισμό, καθώς καθιστά το σύστημα ευάλωτο σε φαινόμενα glare ή κακές καιρικές συνθήκες. Ο συνδυασμός κάμερας με αισθητήρες LiDAR, radar ή ακόμη και IMU μέσω τεχνικών sensor fusion μπορεί να ενισχύσει σημαντικά την ακρίβεια και την ανθεκτικότητα. Η συγχώνευση πληροφοριών από πολλαπλές πηγές επιτρέπει την κάλυψη των αδυναμιών κάθε αισθητήρα ξεχωριστά και οδηγεί σε πιο αξιόπιστες αποφάσεις.

Σε επίπεδο υλικού, η μετάβαση σε πιο ισχυρές υπολογιστικές πλατφόρμες, όπως οι μονάδες NVIDIA jetson ή ακόμα και ειδικά edge AI accelerators, θα δώσει τη δυνατότητα εκτέλεσης πιο απαιτητικών αλγορίθμων σε πραγματικό χρόνο. Αυτό θα επιτρέψει την υλοποίηση μοντέλων βαθιάς μάθησης ή ενισχυτικής μάθησης σε συσκευές ενσωματωμένων συστημάτων, χωρίς να απαιτείται πανίσχυρη υποδομή.

Ιδιαίτερα σημαντική προοπτική για μελλοντική έρευνα αποτελεί η αξιοποίηση της Ενισχυτικής Μάθησης (Reinforcement Learning). Μέσω της RL, το σύστημα μπορεί να μάθει στρατηγικές πλοήγησης και αποφυγής εμποδίων μέσα από δοκιμή και σφάλμα, λαμβάνοντας ανταμοιβές για τις επιτυχημένες αποφάσεις. Σε συνδυασμό με προσομοιωτές όπως CARLA και AirSim, μπορούν να δημιουργηθούν ρεαλιστικά περιβάλλοντα όπου ένας «πράκτορα» θα εκπαιδεύεται αυτόνομα να αντιδρά σε δυναμικές και απρόβλεπτες συνθήκες. Η ενισχυτική μάθηση έχει τη δυνατότητα να ξεπεράσει περιορισμούς κλασικών αλγορίθμων, καθώς δεν βασίζεται αποκλειστικά στην απτική αναγνώριση, αλλά στην ενεργή λήψη αποφάσεων με βάση το εκάστοτε περιβάλλον.

Τέλος, μια ακόμη μελλοντική κατεύθυνση είναι η αξιοποίηση σενεργατικών συστημάτων μέσω επικοινωνίας V2V (Vehicle-to-Vehicle) και V2X (Vehicle-to-Everything). Σε τέτοια σενάρια,

πολλαπλά οχήματα μοιράζονται δεδομένα αισθητήρων και πληροφορίες για το περιβάλλον, με αποτέλεσμα ένα πιο ολοκληρωμένο και ακριβές μοντέλο της κυκλοφοριακής κατάστασης(Abbas, Fan and Khan, 2019). Η προσέγγιση αυτή συδέεται με την ανάπτυξη έξυπνων πόλεων και θα μπορούσε να ενισχύσει σημαντικά την ασφάλεια και την αποδοτικότητα των αυτόνομων μεταφορών.

Βιβλιογραφία

2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance : 27-30 November 2018, Auckland, New Zealand (2018). Institute of Electrical and Electronics Engineers.

Abbas, F., Fan, P. and Khan, Z. (2019) 'A Novel Low-Latency V2V Resource Allocation Scheme Based on Cellular V2X Communications', IEEE Transactions on Intelligent Transportation Systems, 20(6), pp. 2185–2197. Available at: https://doi.org/10.1109/TITS.2018.2865173.

Bouchner, P., Novotný, S. and Piekník, R. (2007) In-cockpit dynamic lighting system for driving simulators and its impact on quality of driving immersion.

Canny, J. (1986) A Computational Approach to Edge Detection, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.

Dosovitskiy, A. et al. (2017) 'CARLA: An Open Urban Driving Simulator'. Available at: http://arxiv.org/abs/1711.03938.

Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE (2009).

Hu, X. et al. (2023) 'How Simulation Helps Autonomous Driving:A Survey of Sim2real, Digital Twins, and Parallel Intelligence'. Available at: http://arxiv.org/abs/2305.01263.

Li, Y. et al. (2023) 'Choose Your Simulator Wisely: A Review on Open-source Simulators for Autonomous Driving'. Available at: https://doi.org/10.1109/TIV.2024.3374044.

Mande, Sudhakar., Tripathy, Amiya. and Sarangi, R.K.. (2015) Proceedings : 2015 International Conference on Technology for Sustainable Development (ICTSD-2015) : by Don Bosco Institute of Technology (DBIT), Mumbai, India, 4th to 6th February, 2015. IEEE.

Salvato, E. et al. (2021) 'Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning', IEEE Access, 9, pp. 153171–153187. Available at: https://doi.org/10.1109/ACCESS.2021.3126658.

Sekehravani, E.A., Babulak, E. and Masoodi, M. (2020) 'Implementing canny edge detection algorithm for noisy image', Bulletin of Electrical Engineering and Informatics, 9(4), pp. 1404–1410. Available at: https://doi.org/10.11591/eei.v9i4.1837.

Shah, S. et al. (2017) 'AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles'. Available at: http://arxiv.org/abs/1705.05065.

Yuan, Z. et al. (2023) 'CLAHE-Based Low-Light Image Enhancement for Robust Object Detection in Overhead Power Transmission System', IEEE Transactions on Power Delivery, 38(3), pp. 2240–2243. Available at: https://doi.org/10.1109/TPWRD.2023.3269206.

Παράρτημα Ι: Επιλεγμένα Τμήματα Κώδικα

Κώδικας 1: CARLA – Σενάριο "Wet Road"

Το σενάριο αυτό προσομοιώνει ένα όχημα σε υγρό δρόμο με έντονο φως, ελέγχοντας την ανίχνευση εμποδίων και την αντίδραση STOP όταν ανιχνεύεται εμπόδιο. Καταγράφονται επίσης εικόνες και δεδομένα σε CSV.

```
    import glob
    import os
    import sys
    import time
    import csv
    import numpy as np
    import cv2
    import math
    import carla

    try:
    sys.path.append(glob.glob('CARLA_0.9.15/PythonAPI/carla/dist/carla-
*%d.%d-%s.egg' % (
    sys.version_info.major,
    sys.version_info.minor,
    'win-amd64' if os.name == 'nt' else 'linux-x86_64'))[0])
    except IndexError:
    pass

    def process_image(image, image_data_dict):
    array = np.frombuffer(image.raw_data, dtype=np.uint8)
    array = np.reshape(array, (image.height, image.width, 4))
    image_data_dict['image'] = array

    client = carla.Client('localhost', 2000)
    client.set_timeout(10.0)
    world = client.get_world()
    blueprint_library = world.get_blueprint_library()

    vehicle_bp = blueprint_library.filter('vehicle.tesla.model3')[0]
    spawn_point = world.get_map().get_spawn_points()[10]
    vehicle = world.spawn_actor(vehicle_bp, spawn_point)

    camera_bp = blueprint_library.find('sensor.camera.rgb')
    camera_bp.set_attribute('image_size_x', '800')
```

```
    camera_bp.set_attribute('image_size_y', '600')
    camera_bp.set_attribute('fov', '110')
    camera_bp.set_attribute('enable_postprocess_effects', 'true')
    camera_bp.set_attribute('bloom_intensity', '5.0')  # έντονη λάμψη

    camera_transform = carla.Transform(carla.Location(x=1.5, y=0.0,
z=2.4))
    camera = world.spawn_actor(camera_bp, camera_transform,
attach_to=vehicle)

    camera_data = {'image': np.zeros((600, 800, 4), dtype=np.uint8)}
    camera.listen(lambda image: process_image(image, camera_data))

    obstacle_bp = blueprint_library.filter('vehicle.audi.tt')[0]
    spawn_loc = spawn_point.location
    obstacle_location = carla.Location(
    x=spawn_loc.x + 15 * math.cos(math.radians(spawn_point.rotation.yaw)),
    y=spawn_loc.y + 15 * math.sin(math.radians(spawn_point.rotation.yaw)),
    z=spawn_loc.z + 0.5
    )

    camera_loc = camera.get_transform().location
    vector_to_camera = camera_loc - obstacle_location
    yaw_deg = math.degrees(math.atan2(vector_to_camera.y,
vector_to_camera.x))
    pitch_deg = math.degrees(math.atan2(vector_to_camera.z,
math.sqrt(vector_to_camera.x**2 + vector_to_camera.y**2)))
    obstacle_rotation = carla.Rotation(pitch=pitch_deg, yaw=yaw_deg,
roll=0)
    obstacle_transform = carla.Transform(obstacle_location,
obstacle_rotation)
    obstacle_vehicle = world.spawn_actor(obstacle_bp, obstacle_transform)

    # Ρύθμιση καιρού για γυαλιστερό δρόμο
    weather = world.get_weather()
    weather.sun_altitude_angle = 30.0
    weather.sun_azimuth_angle = (spawn_point.rotation.yaw + 180) % 360
    weather.sun_angular_diameter = 30.0
    weather.direct_light_intensity = 800.0
    weather.cloudiness = 0.0
    weather.precipitation = 0.0
    weather.fog_density = 0.0
    weather.wetness = 100.0         # πολύ υγρός δρόμος
    weather.bloom_intensity = 5.0   # έντονη λάμψη
    world.set_weather(weather)

    output_folder = 'output_scenario_1d'
    os.makedirs(output_folder, exist_ok=True)
```

```
    csv_path = 'scenario_1d.csv'
    with open(csv_path, 'a', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(['Scenario', 'Distance', 'ObstacleDetected',
'Reaction'])

    try:
    vehicle.set_transform(spawn_point)
    vehicle.apply_control(carla.VehicleControl(throttle=0.0, brake=1.0))
    time.sleep(1)
    obstacle_vehicle.set_transform(obstacle_transform)

    while True:
    world.wait_for_tick()
    frame = camera_data['image']
    if frame.sum() == 0:
    continue

    image = frame.copy()
    h, w = image.shape[:2]

    gray = cv2.cvtColor(image, cv2.COLOR_BGRA2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    edges = cv2.Canny(blur, 30, 90)

    y_start = int(h * 0.5)
    y_end = h
    x_start = int(w * 0.3)
    x_end = int(w * 0.7)
    roi = edges[y_start:y_end, x_start:x_end]

    contours, _ = cv2.findContours(roi, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    obstacle_detected = any(cv2.contourArea(c) > 600 for c in contours)

    v_loc = vehicle.get_location()
    o_loc = obstacle_vehicle.get_location()
    dist = v_loc.distance(o_loc)

    if obstacle_detected and dist < 10:
    vehicle.apply_control(carla.VehicleControl(throttle=0.0, brake=1.0))
    reaction = "STOP"
    timestamp = time.strftime("%H%M%S")
    filename = f"{output_folder}/wetroad_img_{timestamp}.jpg"
    cv2.imwrite(filename, image)
    writer.writerow(['wet_road', dist, True, reaction])
```

```
    break
    else:
    vehicle.apply_control(carla.VehicleControl(throttle=0.3, brake=0.0,
steer=0.0))
    writer.writerow(['wet_road', dist, obstacle_detected, 'FORWARD'])

    cv2.rectangle(image, (x_start, y_start), (x_end, y_end), (0, 255, 0),
2)
    cv2.drawContours(image[y_start:y_end, x_start:x_end], contours, -1,
(0, 0, 255), 2)
    cv2.putText(image, f"Dist: {dist:.1f}m", (20, 40),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 0), 2)
    cv2.imshow("RGB Camera - Wet Road", image)

    if cv2.waitKey(1) & 0xFF == ord('q'):
    raise KeyboardInterrupt()

    except KeyboardInterrupt:
    print("\n[!] Τερματισμός από χρήστη.")
    finally:
    camera.stop()
    vehicle.destroy()
    obstacle_vehicle.destroy()
    camera.destroy()
    cv2.destroyAllWindows()
print("[OK] Καθαρισμός ολοκληρώθηκε.")
```

Κώδικας 2: AirSim

Το σενάριο στο AirSim χρησιμοποιεί RGB και Depth εικόνες για ανίχνευση εμποδίων και καταγραφή αποστάσεων. Το όχημα αντιδρά αυτόματα (STOP ή FORWARD) ανάλογα με την παρουσία εμποδίων και την απόσταση τους.

```
    import airsim
    import numpy as np
    import cv2
    import os
    import time
    import math
    import csv

    # Σύνδεση με τον client
    client = airsim.CarClient()
    client.confirmConnection()
    client.enableApiControl(True)
    car_controls = airsim.CarControls()
```

```
    # Δημιουργία φακέλου εξόδου
    output_folder = 'scenario_D'
    os.makedirs(output_folder, exist_ok=True)

    # CSV αρχείο
    csv_path = 'scenario_D.csv'
    with open(csv_path, 'a', newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(['Scenario',    'Distance',    'ObstacleDetected',
'Reaction'])

        try:
            # Σταμάτημα και αρχική θέση
            car_controls.throttle = 0.0
            car_controls.brake = 1.0
            client.setCarControls(car_controls)
            time.sleep(1)

            while True:
                # Λήψη RGB και Depth εικόνων
                responses = client.simGetImages([
                    airsim.ImageRequest("0",       airsim.ImageType.Scene,
False, False),
                    airsim.ImageRequest("0", airsim.ImageType.DepthPlanar,
True)
                ])

                if responses[0].width == 0 or responses[1].width == 0:
                    continue

                # Επεξεργασία RGB εικόνας
                img1d    =    np.frombuffer(responses[0].image_data_uint8,
dtype=np.uint8)
                img_rgb        =        img1d.reshape(responses[0].height,
responses[0].width, 3)
                image = img_rgb.copy()
                h, w = image.shape[:2]

                gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
                blur = cv2.GaussianBlur(gray, (5, 5), 0)
                edges = cv2.Canny(blur, 30, 90)

                # Περιοχή ενδιαφέροντος (ROI)
                roi_height = int(h * 0.4)
                roi_width = int(w * 0.4)
                y_center = h // 2
                x_center = w // 2
```

```
                y_start = y_center - roi_height // 2
                y_end = y_center + roi_height // 2
                x_start = x_center - roi_width // 2
                x_end = x_center + roi_width // 2

                roi = edges[y_start:y_end, x_start:x_end]

                # Περιγράμματα
                contours, _ = cv2.findContours(roi, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
                obstacle_detected = any(cv2.contourArea(c) > 600 for c in
contours)

                # Ανάλυση depth εικόνας για υπολογισμό απόστασης
                depth_data = np.array(responses[1].image_data_float,
dtype=np.float32)
                depth_img = depth_data.reshape(responses[1].height,
responses[1].width)
                roi_depth = depth_img[y_start:y_end, x_start:x_end]
                valid_depths = roi_depth[np.isfinite(roi_depth)]
                dist = np.median(valid_depths) if valid_depths.size > 0
else 999.0

                # Αντίδραση
                if obstacle_detected and dist < 10:
                    car_controls.throttle = 0.0
                    car_controls.brake = 1.0
                    reaction = "STOP"
                    client.setCarControls(car_controls)

                    timestamp = time.strftime("%H%M%S")
                    filename =
f"{output_folder}/scenario1on_img_{timestamp}.jpg"
                    cv2.imwrite(filename, image)
                    writer.writerow(['scenario_1on', dist, True,
reaction])
                    print("[!] Εντοπίστηκε εμπόδιο — Σταμάτημα.")
                    break
                else:
                    car_controls.throttle = 0.3
                    car_controls.brake = 0.0
                    client.setCarControls(car_controls)
                    writer.writerow(['scenario_1on', dist,
obstacle_detected, 'FORWARD'])

                # Οπτικοποίηση
                cv2.rectangle(image, (x_start, y_start), (x_end, y_end),
(0, 255, 0), 2)
                roi_display = image[y_start:y_end, x_start:x_end].copy()
```

```
                cv2.drawContours(roi_display, contours, -1, (0, 0, 255), 2)
                image[y_start:y_end, x_start:x_end] = roi_display
                cv2.putText(image,   f"Dist:   {dist:.1f}m",   (20,   40),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 0), 2)
                cv2.imshow("RGB Camera", image)

                if cv2.waitKey(1) & 0xFF == ord('q'):
                    raise KeyboardInterrupt()

        except KeyboardInterrupt:
            print("\n[!] Τερματισμός από χρήστη.")
        finally:
            client.enableApiControl(False)
            cv2.destroyAllWindows()
        print("[OK] Καθαρισμός ολοκληρώθηκε.")
```

Κώδικα 3: Raspberry Pi – Ανίχνευση εμποδίων και lane following

Αυτός ο κώδικας ελέγχει ένα ρομποτικό όχημα συνδυάζοντας αισθητήρα υπερήχων και κάμερα για την ανίχνευση εμποδίων, καθώς και lane following με PID έλεγχο. Οι αποστάσεις και οι ενέργειες καταγράφονται σε CSV.

```
import cv2
import numpy as np
import RPi.GPIO as GPIO
import time
import os
import csv

# === Πινάκια ===
TRIG = 13
ECHO = 19
LEFT_PWM_PIN = 18
LEFT_DIR_PIN = 17
RIGHT_PWM_PIN = 23
RIGHT_DIR_PIN = 22

# === Ρυθμίσεις GPIO ===
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
GPIO.setup(LEFT_PWM_PIN, GPIO.OUT)
GPIO.setup(LEFT_DIR_PIN, GPIO.OUT)
GPIO.setup(RIGHT_PWM_PIN, GPIO.OUT)
```

```
GPIO.setup(RIGHT_DIR_PIN, GPIO.OUT)

# === PWM ===
left_pwm = GPIO.PWM(LEFT_PWM_PIN, 100)
right_pwm = GPIO.PWM(RIGHT_PWM_PIN, 100)
left_pwm.start(0)
right_pwm.start(0)

def set_motor_speed(left_speed, right_speed):
    GPIO.output(LEFT_DIR_PIN, left_speed >= 0)
    GPIO.output(RIGHT_DIR_PIN, right_speed >= 0)
    left_pwm.ChangeDutyCycle(min(abs(left_speed), 100))
    right_pwm.ChangeDutyCycle(min(abs(right_speed), 100))

def stop_motors():
    left_pwm.ChangeDutyCycle(0)
    right_pwm.ChangeDutyCycle(0)
    GPIO.output(LEFT_DIR_PIN, GPIO.LOW)
    GPIO.output(RIGHT_DIR_PIN, GPIO.LOW)

# === PID Ρυθμίσεις ===
Kp = 0.4
Ki = 0.0
Kd = 0.2
base_speed = 60
last_error = 0
integral = 0

# === Απόσταση ===
def get_distance():
    GPIO.output(TRIG, False)
    time.sleep(0.01)
    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    timeout_start = time.time()
    while GPIO.input(ECHO) == 0:
        if time.time() - timeout_start > 0.03:
            return 9.99
    start_time = time.time()

    timeout_start = time.time()
    while GPIO.input(ECHO) == 1:
        if time.time() - timeout_start > 0.03:
            return 9.99
    stop_time = time.time()
```

```
    elapsed = stop_time - start_time
    distance = (elapsed * 34300) / 2
    return round(distance / 100, 2)

# === Καταγραφή CSV ===
output_folder = 'output'
os.makedirs(output_folder, exist_ok=True)
csv_path = os.path.join(output_folder, 'obstacle_log.csv')
new_file = not os.path.exists(csv_path) or os.path.getsize(csv_path) == 0
csvfile = open(csv_path, 'a', newline='')
writer = csv.writer(csvfile)
if new_file:
    writer.writerow(['Timestamp',    'Distance(m)',    'ObstacleDetected',
'Action'])

# === Κατάσταση λειτουργίας ===
robot_running = False

# === Συντεταγμένες κουμπιών ===
button_start = (10, 60, 100, 40)
button_stop = (120, 60, 100, 40)

def draw_buttons(frame):
    color_start = (0, 255, 0) if not robot_running else (100, 100, 100)
    color_stop = (0, 0, 255) if robot_running else (100, 100, 100)
    cv2.rectangle(frame, (button_start[0], button_start[1]),
                         (button_start[0]+button_start[2],
button_start[1]+button_start[3]), color_start, -1)
    cv2.putText(frame, "Start", (button_start[0]+20, button_start[1]+28),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,255,255), 2)

    cv2.rectangle(frame, (button_stop[0], button_stop[1]),
                         (button_stop[0]+button_stop[2],
button_stop[1]+button_stop[3]), color_stop, -1)
    cv2.putText(frame,  "Stop",  (button_stop[0]+25,  button_stop[1]+28),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255,255,255), 2)

# === Συνάρτηση για χειρισμό κλικ ===
def mouse_callback(event, x, y, flags, param):
    global robot_running
    if event == cv2.EVENT_LBUTTONDOWN:
        if (button_start[0] <= x <= button_start[0]+button_start[2] and
            button_start[1] <= y <= button_start[1]+button_start[3]):
            print("[>] Εκκίνηση!")
            robot_running = True
        elif (button_stop[0] <= x <= button_stop[0]+button_stop[2] and
              button_stop[1] <= y <= button_stop[1]+button_stop[3]):
```

```
            print("[X] Διακοπή.")
            robot_running = False
            stop_motors()

# === Εκκίνηση κάμερας ===
cap = cv2.VideoCapture(0, cv2.CAP_V4L2)
time.sleep(2)

if not cap.isOpened():
    print("[ERROR] Δεν ανοίγει η κάμερα.")
    cap.release()
    csvfile.close()
    GPIO.cleanup()
    exit()

cv2.namedWindow("View")
cv2.setMouseCallback("View", mouse_callback)

try:
    while True:
        ret, frame = cap.read()
        if not ret:
            continue

        draw_buttons(frame)

        if robot_running:
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            blur = cv2.GaussianBlur(gray, (5, 5), 0)
            edges = cv2.Canny(blur, 30, 90)

            h, w = edges.shape
            roi_obstacle = edges[h//3:h*2//3, w//3:w*2//3]
            contours, _ = cv2.findContours(roi_obstacle, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
            obstacle_detected = any(cv2.contourArea(c) > 600 for c in
contours)

            dist = get_distance()
            timestamp = time.strftime("%H:%M:%S")

            if obstacle_detected and dist < 0.15:
                stop_motors()
                filename = os.path.join(output_folder,
f"obstacle_{timestamp.replace(':','')}.jpg")
                cv2.imwrite(filename, frame)
                writer.writerow([timestamp, dist, True, 'STOP'])
```

```
                csvfile.flush()
                print(f"[!] Εμπόδιο — {dist:.2f} m — ΣΤΑΜΑΤΗΣΕ")
                robot_running = False
            else:
                # === Lane Following ===
                roi_lane = frame[h - 200:h, :]
                gray_lane = cv2.cvtColor(roi_lane, cv2.COLOR_BGR2GRAY)
                _, binary = cv2.threshold(gray_lane, 100, 255,
cv2.THRESH_BINARY_INV)

                left_half = binary[:, :binary.shape[1] // 2]
                right_half = binary[:, binary.shape[1] // 2:]

                M_left = cv2.moments(left_half)
                M_right = cv2.moments(right_half)

                if M_left["m00"] > 0 and M_right["m00"] > 0:
                    cx_left = int(M_left["m10"] / M_left["m00"])
                    cx_right = int(M_right["m10"] / M_right["m00"]) +
binary.shape[1] // 2
                    lane_center = (cx_left + cx_right) // 2
                    frame_center = binary.shape[1] // 2
                    error = lane_center - frame_center

                    # PID correction
                    integral += error
                    derivative = error - last_error
                    correction = Kp * error + Ki * integral + Kd *
derivative
                    last_error = error

                    left_speed = base_speed - correction
                    right_speed = base_speed + correction
                    set_motor_speed(left_speed, right_speed)
                    writer.writerow([timestamp, dist, False, 'LANE
FOLLOW'])
                    csvfile.flush()
                else:
                    stop_motors()
        else:
            stop_motors()

        # Εμφάνιση ROI και απόστασης
        h, w = frame.shape[:2]
        cv2.rectangle(frame, (w//3, h//3), (w*2//3, h*2//3), (0,255,0), 2)
        dist_display = get_distance()
        cv2.putText(frame, f"Dist: {dist_display:.2f}m", (10,30),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0,255,255), 2)
```

```
        cv2.imshow("View", frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

except KeyboardInterrupt:
    print("\n[!] Τερματισμός από χρήστη με Ctrl+C.")

finally:
    print("[OK] Καθαρισμός πόρων...")
    stop_motors()
    left_pwm.stop()
    right_pwm.stop()
    if cap.isOpened():
        cap.release()
    cv2.destroyAllWindows()
    csvfile.close()
    GPIO.cleanup()
```