

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας Design and Implementation of Self-navigated Vehicle with Lane Detection
Όνοματεπώνυμο Φοιτητή	Όλγα Σχίζα
Πατρώνυμο	Δημήτριος
Αριθμός Μητρώου	ΜΠΠΛ/ 18063
Επιβλέπων	Θεμιστοκλής Παναγιωτόπουλος, Καθηγητής

Ημερομηνία Παράδοσης

Νοέμβριος 2022

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

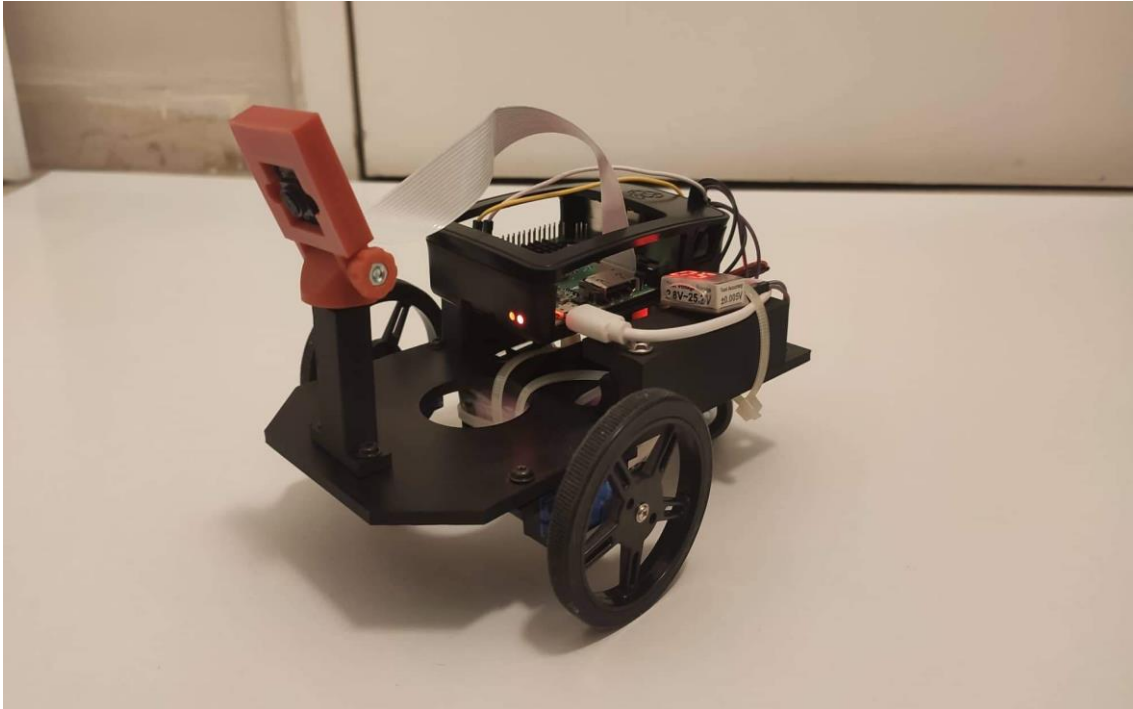
(υπογραφή)

(υπογραφή)

Θεμιστοκλής
Παναγιωτόπουλος
Καθηγητής

Διονύσιος Σωτηρόπουλος
Επίκουρος Καθηγητής

Ιωάννης Τασούλας
Επίκουρος Καθηγητής



Εικόνα 1. Το αυτοκατευθυνόμενο όχημα ολοκληρωμένο.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή μου κύριο Θεμιστοκλή Παναγιωτόπουλο για την υπομονή και την ευκαιρία που μου έδωσε να ασχοληθώ με αυτό το θέμα που αγαπώ και θέλω να εξελίξω. Επίσης θα ήθελα να ευχαριστήσω τους συμφοιτητές μου Συμεών Ζαπσής , Κωνσταντίνο Ζιώβα, Χρήστο Κούση , Ιωάννη Μουζάκη για την συνεισφορά τους. Τέλος θα ήθελα να ευχαριστήσω τον σύζυγο μου Παναγιώτη Τούρλα για την υπομονή και την υποστήριξη που μου προσέφερε όλο αυτό το διάστημα και τον Γιό μου Αριστομένη - Αναστάση Τούρλα που κατάφερε να κάνει κάθε δυσκολία πιο διασκεδαστική.

Περίληψη

Σκοπός της παρούσας εργασίας είναι η μελέτη της κατασκευής ενός αυτοκατευθυνόμενου οχήματος, που ανιχνεύει και ακολουθεί μία λωρίδα. Επί της ουσίας, αποτελεί μία προσπάθεια σύζευξης κατάλληλου υλικού και λογισμικού ούτως ώστε να κατασκευαστεί ένα μοντέλο οχήματος με δυνατότητα αυτόνομης κατεύθυνσης, η οποία ορίζεται μέσω του εντοπισμού λωρίδας. Για τον εντοπισμό λωρίδας χρησιμοποιήθηκε κάμερα συνδεδεμένη με υπολογιστή μονής πλακέτας και τοποθετήθηκαν σε τρισδιάστατα εκτυπωμένο σκελετό. Για την κίνηση του οχήματος χρησιμοποιήθηκαν δύο σερβομηχανές. Η διαδικασία κίνησης και εντοπισμού της λωρίδας και οι αντίστοιχοι αλγόριθμοι υλοποιήθηκαν σε γλώσσα προγραμματισμού python. Επιπλέον, χρησιμοποιήθηκε υπολογιστική όραση για τον εντοπισμό της λωρίδας. Η εργασία είναι χωρισμένη σε έξι θεματικές ενότητες :

- Στο πρώτο μέρος παρουσιάζεται η υπολογιστική όραση από θεωρητική σκοπιά και σε συνδυασμό με τα αυτοκατευθυνόμενα οχήματα, καθώς και άλλες εφαρμογές της.
- Στο δεύτερο τμήμα περιγράφεται το υλικό που χρησιμοποιήθηκε για την κατασκευή του οχήματος.
- Το τρίτο σκέλος της εργασίας αναφέρει το λογισμικό και τις βιβλιοθήκες που θεωρήθηκαν κατάλληλες για τις λειτουργίες του συγκεκριμένου οχήματος
- Το τέταρτο τμήμα αποτελεί την αναλυτική παρουσίαση της συνδεσμολογίας του αυτοκατευθυνόμενου οχήματος
- Στο πέμπτο μέρος παρουσιάζεται οι αλγόριθμοι της ανίχνευσης λωρίδας με κάθε λεπτομέρεια.
- Στο έκτο σκέλος υπάρχουν τα συμπεράσματα του εγχειρήματος καθώς και οι δυσκολίες που αντιμετωπίστηκαν κατά την κατασκευή , όπως και οι μελλοντικές βελτιώσεις που θα μπορούσαν να εφαρμοστούν.

Λέξεις – κλειδιά

Αυτοκατευθυνόμενο όχημα, ανίχνευση λωρίδας, υπολογιστική όραση, αλγόριθμος ανίχνευσης λωρίδας.

Abstract

The purpose of this thesis is to study the construction of a self-navigated vehicle which detects and follows a lane. This is an attempt to couple hardware and software in an appropriate way so that the resulting model vehicle has the ability to steer autonomously through lane detection. The core hardware consists of camera paired with a single board computer used for lane tracking, while both placed on a custom 3D-printed deck. Additionally, two servo motors are responsible for moving the vehicle. In terms of software, the vehicle is based on algorithms for locating the lane and applying the necessary corrections during movement so that the vehicle stays in course. These algorithms are implemented in the Python programming language. In addition, computer vision techniques are applied to locate the lane. The thesis is divided into six thematic sections:

- The first part where computer vision is presented from a theoretical aspect and in combination with self-navigated vehicles, as well as other applications which use computer vision.
- The second section describes the hardware that we used for the construction of the vehicle.
- The third section of the thesis mentions the suitable software and libraries of the operations of the specific vehicle.
- The fourth section presents in detail all the connections between the parts of the vehicle.
- In the fifth section, the algorithms for lane detection are presented.
- Lastly, the sixth section gathers the conclusions of the project as well as the difficulties that arose during the construction and opportunities for future improvements.

Keywords

Self-navigated vehicle, lane detection, computer vision, lane detection algorithm.

Περιεχόμενα

Περίληψη	5
Abstract	6
Κατάλογος Εικόνων.....	9
1. Υπολογιστική όραση	11
1.1 Ιστορική αναφορά στην υπολογιστικής όραση	12
1.2 Διαφορά μεταξύ Υπολογιστικής και Μηχανικής Όρασης	12
1.3 Υπολογιστική όραση και αυτόνομα οχήματα	14
1.3.1 Αυτοκατευθυνόμενα Οχήματα	14
1.4 Άλλες εφαρμογές υπολογιστικής όρασης	17
2 Το υλικό (hardware)	23
2.1 Ρόδες με ελαστικό από σιλικόνη 60x8 mm για σερβοκινητήρες .	23
2.2 Σερβοκινητήρας	23
2.2.1 Κινητήρας συνεχούς ρεύματος (Direct Current)	25
2.2.2 Βηματικός κινητήρας (Stepper Motor) :	26
2.2.3 Πλεονεκτήματα και μειονεκτήματα των DC , Σερβοκινητήρων και Βηματικών κινητήρων :	26
2.3 Σασί	27
2.4 Κάμερα για υπολογιστή μονής πλακέτας:	27
2.5 Μπαταρίες λιθίου συνδεδεμένες με ρυθμιστή τάσης και βολτόμετρο.	28
2.6 Υπολογιστής μονής πλακέτας	30
2.6.1 Γενικά χαρακτηριστικά του SBC που χρησιμοποιήθηκε :	30
3 Λειτουργικό σύστημα	36
3.1 Γλώσσα προγραμματισμού	36
3.2 Βιβλιοθήκες	38
3.2.1 OpenCV :	38
3.2.2 NumPy	38
3.2.3 Matplotlib	40
4 Συνδεσμολογία	41
5 Ανίχνευση λωρίδας σε πραγματικό χρόνο	43
5.1 Αλγόριθμος ανίχνευσης λωρίδας	43
5.1.1 Αλγόριθμος ανίχνευσης λωρίδας σε Python	56
5.2 Εναλλακτικός Αλγόριθμος	60

5.2.1	Εναλλακτικός Αλγόριθμος σε Python.	62
6	Συμπεράσματα	67
	Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές	679

Κατάλογος Εικόνων

Εικόνα 1. Το αυτοκατευθυνόμενο όχημα ολοκληρωμένο.....	3
Εικόνα 2. Υπολογιστική όραση σε πραγματικό χρόνο (Khillar, 2022).....	13
Εικόνα 3. Υπολογιστική Όραση vs Μηχανική όραση (Khillar, 2022)	14
Εικόνα 4. Τα επίπεδα αυτονομίας οδήγησης (Rambus Press, 2022)	17
Εικόνα 5. Ανίχνευση πεζών με υπολογιστική όραση (Kunanbaeva, 2020).....	18
Εικόνα 6. Κενές και κατειλημμένες θέσης στάθμευσης (Kunanbaeva, 2020)	19
Εικόνα 7. Ανάλυση ροής κυκλοφορίας (James Allen on September 4, 2019).....	20
Εικόνα 8. Τμηματοποίηση Όγκου εγκεφάλου με τεχνητή νοημοσύνη (Alarcon, 2018)	21
Εικόνα 9. Παρακολούθηση καλλιέργειας και απόδοσης (Rizzoli, 2022)	22
Εικόνα 10. Αυτόματο ταμείο (Rizzoli, 2022)	22
Εικόνα 11. Δύο ρόδες με ελαστικό από σιλικόνη 60x8 mm for FS90R servo (Pololu.com, n.d.)	23
Εικόνα 12. Δύο Feetech FS90R Continuous Servo Motor, σερβομηχανές.....	24
Εικόνα 13. Διάφορα μεγέθη συνεχούς σερβοκινητήρα (Pololu.com, n.d.)	25
Εικόνα 14. Κινητήρας συνεχούς ρεύματος (Anon., n.d.)	25
Εικόνα 15 Βηματικός κινητήρας (www.heason.com, χ.χ.).....	26
Εικόνα 16. Σασί από το πλάι	27
Εικόνα 17. Σύγκριση Εκδόσεων Κάμερας SBC (Raspberry Pi) (ΒΑΣΙΛΟΠΟΥΛΟΣ ΝΙΚΟΛΑΟΣ, n.d.).....	28
Εικόνα 18. Raspberry Pi Camera V2 (Anon., n.d.).....	28
Εικόνα 19. Μπαταρίες λιθίου συνδεδεμένες με ρυθμιστή τάσης και βολτόμετρο	30
Εικόνα 20. Υπολογιστής – Rpi 3 Model B+ (Anon., n.d.)	31
Εικόνα 21 Τα μέρη του Raspberry Pi 3b+ (eTechnophiles , n.d.).....	32
Εικόνα 22 Τα Pinout του Raspberry Pi 3b+ (eTechnophiles , n.d.)	33
Εικόνα 23 PWM και κύκλος λειτουργίας. (Stanislav Smith, Technical Transmission General Articles , 2022).....	35

Εικόνα 24. Λογισμικό Raspberry Pi OS. (Piltch, n.d.).....	36
Εικόνα 25. Λογότυπο της Python. (Anon., n.d.).....	37
Εικόνα 26.Εικόνα με RGB χρώμα με την χρήση πίνακα στην βιβλιοθήκη NumPy (Pramoditha, 2021).	39
Εικόνα 27. Συνδεσμολογία.....	41
Εικόνα 28. Βήματα ανίχνευσης λωρίδας (Geeks, Geeks for, 2022).....	43
Εικόνα 29. Το πρώτο βήμα.....	43
Εικόνα 30. Χρωματικό διάγραμμα (BenQ Color Technology Lab, 2019).	44
Εικόνα 31. Μετατροπή εικόνας σε κλίμακα του γκρι (Antoniadis, 2022).....	45
Εικόνα 32. Η εικόνα μετά την μετατροπή στη κλίμακα του Γκρι.....	45
Εικόνα 33. Καμπύλη της συνάρτησης του Gauss.	46
Εικόνα 34. Γραφική παράσταση συνάρτησης Gauss για 2 διαστάσεις (Sharda, 2021).47	
Εικόνα 35. Αποτέλεσμα της συνέλιξης (Sharda, 2021).	48
Εικόνα 36. Η εικόνα μετά την χρήση του Γκαουσιανού θολώματος.	49
Εικόνα 37. Ευθεία γραμμή εξίσωσης $y=mx+b$	53
Εικόνα 38. Αναπαράσταση Hesse.....	54
Εικόνα 39. Περιγραφή ευθείας σε πολικές συντεταγμένες.....	55
Εικόνα 40.Περιβάλλον μεταξύ ορίων. (etal, 1985)	61
Εικόνα 41. Η πάνω πλευρά του αυτοκατευθυνόμενου οχήματος.....	65
Εικόνα 42. Το αυτοκατευθυνόμενο όχημα.....	66

ΕΙΣΑΓΩΓΗ

Η αυτόνομη οδήγηση είναι σήμερα ένας από τους πιο απαιτητικούς αλλά και πιο εμπνευσμένους τομείς όπου βρίσκουν εφαρμογή οι σύγχρονες τεχνολογίες υπολογιστικής όρασης. Πολλά από τα οχήματα που χρησιμοποιούνται σήμερα θεωρούνται ημιαυτόνομα λόγω των χαρακτηριστικών ασφαλείας που διαθέτουν, για παράδειγμα συστήματα υποβοήθησης πέδησης. Επίσης, έχουν κάνει την εμφάνισή τους και αυτοκατευθυνόμενα οχήματα που βασίζουν τις δυνατότητες τους σε προηγμένα συστήματα ανίχνευσης ορίων λωρίδας, σημάτων και απροσδόκητων εμποδίων. Καθώς, λοιπόν ο αριθμός των αυτοκατευθυνόμενων οχημάτων αυξάνεται, παράλληλα αναπτύσσονται και μελέτες τόσο στην ακαδημαϊκή κοινότητα όσο και στη βιομηχανία σχετικά με την αυτόνομη οδήγηση που μπορούν να χωριστούν σε τρεις φάσεις.

Η πρώτη φάση είναι η αντίληψη και η λήψη αποφάσεων και ελέγχου. Δεύτερη φάση αποτελεί η αποτελεσματική λήψη πληροφοριών για το περιβάλλον μέσα στο οποίο κινείται το αυτοκατευθυνόμενο όχημα και η τρίτη φάση είναι ο έλεγχος που στοχεύει στον ορισμό των κατάλληλων μηχανικών παραμέτρων, ούτως ώστε το αυτοκατευθυνόμενο όχημα να κινείται ακολουθώντας μία προγραμματισμένη διαδρομή.

Η λήψη αποφάσεων επί της ουσίας προσπαθεί να καθορίσει μία τέλεια διαδρομή ώστε αυτού του είδους τα οχήματα να πλοηγηθούν με ασφάλεια προς τον προορισμό τους. Γίνεται, λοιπόν, αντιληπτό ότι η πρώτη φάση αποτελεί το βασικό συστατικό στα αυτοκατευθυνόμενα οχήματα, καθώς καθορίζει όλες τις ενέργειες του οχήματος, όπως την αλλαγή λωρίδας, τη χρήση του φρένου και την επιτάχυνση. Η λήψη αποφάσεων στα αυτοκινούμενα οχήματα γίνεται από τον συνδυασμό των μοντέλων λήψεων αποφάσεων με άλλα συστήματα, όπως αυτό της πλοήγησης.

Στην παρούσα εργασία θα ασχοληθούμε με το θεμελιώδες πρόβλημα στην αυτόνομη οδήγηση που είναι η ανίχνευση λωρίδας στα αυτοκατευθυνόμενα οχήματα. Πιο αναλυτικά, η παρούσα εργασία είχε ως στόχο την κατασκευή ενός ρομποτικού αυτοκατευθυνόμενου οχήματος το οποίο χρησιμοποιεί τον κατάλληλο αλγόριθμο ώστε να εντοπίζει την λωρίδα και να κινείται σε αυτή. Στην εργασία αυτή χρησιμοποιείται η υπολογιστική όραση μέσω της βιβλιοθήκης OpenCv για την ανίχνευση λωρίδας που αποτελεί, όπως έχει ειπωθεί το βασικότερο συστατικό για την καλύτερη και ασφαλή οδήγηση ώστε να κατανοούν με σαφήνεια το περιβάλλον πλοήγησης.

1. Υπολογιστική όραση

Η Υπολογιστική όραση είναι ένα διεπιστημονικό πεδίο που ασχολείται με το πώς οι υπολογιστές μπορούν να κατασκευαστούν για να αποκτήσουν κατανόηση υψηλού επιπέδου από ψηφιακές εικόνες ή βίντεο. Από μηχανικής σκοπιάς, επιδιώκει να αυτοματοποιήσει εργασίες που μπορεί να κάνει το ανθρώπινο οπτικό σύστημα. Η υπολογιστική όραση ασχολείται με την εξαγωγή, την ανάλυση και την κατανόηση χρήσιμων πληροφοριών από μία μεμονωμένη εικόνα ή μία ακολουθία εικόνων. Για την επίτευξη της αυτόματης οπτικής κατανόησης, βασίζεται στην ανάπτυξη αλγορίθμων. Ως επιστημονικός κλάδος, η υπολογιστική όραση ασχολείται με τη θεωρία πίσω από τεχνητά συστήματα που εξαγουν πληροφορίες από εικόνες. Ως τεχνολογικός κλάδος η υπολογιστική όραση επιδιώκει να εφαρμόσει τις θεωρίες και τα μοντέλα της για την κατασκευή συστημάτων υπολογιστικής όρασης. Η υπολογιστική όραση είναι ένα πεδίο τεχνητής νοημοσύνης που επιτρέπει σε υπολογιστές και συστήματα να αντλούν σημαντικές πληροφορίες από ψηφιακές εικόνες, βίντεο και άλλες οπτικές εισροές και να προβαίνουν σε ενέργειες με βάση τις πληροφορίες αυτές. Στην ουσία η υπολογιστική όραση εστιάζει στην αναπαραγωγή τμημάτων της πολυπλοκότητας του συστήματος ανθρώπινης όρασης και δίνει την δυνατότητα στους υπολογιστές να αναγνωρίζουν και να επεξεργάζονται αντικείμενα με τον ίδιο τρόπο που κάνουν οι άνθρωποι. (Wikipedia, n.d.)

1.1 Ιστορική αναφορά στην υπολογιστικής όραση

Η υπολογιστική όραση ξεκίνησε στα τέλη της δεκαετίας του 1960 σε πανεπιστήμια που ήταν πρωτοπόρα στην τεχνητή νοημοσύνη, με σκοπό την μίμηση του ανθρώπινου οπτικού συστήματος, ώστε να εφαρμοστεί στην δημιουργία έξυπνων ρομπότ. Το καλοκαίρι του 1966 έγινε η πρώτη απόπειρα σύνδεσης κάμερας με υπολογιστή ώστε να περιγραφεί το περιβάλλον όπου βρισκόταν. Βασική διαφορά της μηχανικής όρασης από την ψηφιακή επεξεργασία εικόνας, που είχε επικρατήσει εκείνη την εποχή, έγκειται στο ότι ιδανικό αποτέλεσμα για την μηχανική οπτική θα ήταν η εξαγωγή τρισδιάστατης δομής από εικόνες που θα έχουν κατανοήσει όλη την σκηνή που έχουν μπροστά τους.

Την δεκαετία του 1970 έγιναν πολλές μελέτες που έθεσαν τις βάσεις για πολλούς αλγόριθμους μηχανικής όρασης, που χρησιμοποιούνται ακόμα και σήμερα. Σε αυτές τις μελέτες συμπεριλαμβάνεται και η εξαγωγή ακμών από εικόνες, η επισήμανση γραμμών, η μη πολυεδρική και πολυεδρική μοντελοποίηση, η αναπαράσταση αντικειμένων ως διασύνδεση μικρότερων δομών της οπτικής ροής και της εκτίμησης της κίνησης. Οι μελέτες που ακολούθησαν την επόμενη δεκαετία ήταν βασισμένες σε αυστηρή μαθηματική ανάλυση και πτυχές ποσοτικές της υπολογιστικής όρασης, τέτοιες όπως η έννοια της κλίμακας του χώρου, συμπεράσματα σχημάτων προερχόμενα από διάφορα στοιχεία όπως η σκίαση, η υφή και η εστίαση και μοντέλα περιγράμματος γνωστά ως φίδια. Εν συνεχεία, οι ερευνητές κατάλαβαν ότι πολλές από τις μαθηματικές έννοιες που χρησιμοποιούν στην υπολογιστική όραση, μπορούν να αντιμετωπιστούν με την ίδια δομή βελτιστοποίησης όπως η ομαλοποίηση και τα τυχαία πεδία Markov.

Μέχρι την δεκαετία του 1990 ορισμένα θέματα ερευνών απασχόλησαν περισσότερο από άλλα, παραδείγματος χάρη η έρευνα για τις προβολικές τρισδιάστατες ανακατασκευές οδήγησε στην καλύτερη κατανόηση ρύθμιση της κάμερας. Η εμφάνιση των μεθόδων βελτιστοποίησης για τη ρύθμιση της κάμερα, βοήθησε στο να γίνει αντιληπτό ότι αρκετές από τις ιδέες είχαν ήδη ερευνηθεί από την θεωρία προσαρμογής δεσμίδων, από τον τομέα της φωτογραμμετρίας. Με τη σειρά του αυτό οδήγησε σε μεθόδους για αραιές τρισδιάστατες ανακατασκευές σκηνών από πολλαπλές εικόνες. Πρόοδος παρατηρήθηκε στο θέμα της πυκνής στερεοφωνικής συσχέτισης και στις περαιτέρω τεχνικές στερεοφωνικής πολλαπλής προβολής. Παράλληλα χρησιμοποιήθηκαν οι διακυμάνσεις των γραφικών διαχωρισμών για την κατάμηση της εικόνας. Επίσης εκείνη την δεκαετία σημειώνεται η χρήση τεχνικών στατιστικής εκμάθησης στην πράξη για την αναγνώριση προσώπων σε εικόνες.

Πλησιάζοντας τα τέλη του 1990 άλλαξε η αλληλεπίδραση μεταξύ των τομέων των γραφικών υπολογιστών και της μηχανικής όρασης και μέσα σε αυτή την αλλαγή συμπεριλαμβάνονται η απόδοση με βάση την εικόνα, η μορφοποίηση εικόνας, η παρεμβολή προβολής, η πανοραμική συρραφή εικόνας και τέλος η απόδοση του πρώιμου φωτεινού πεδίου. Οι πρόσφατες εργασίες χρησιμοποιούν μεθόδους που έχουν ως βάση τα χαρακτηριστικά, συνδυαστικά με τεχνικές μηχανικής μάθησης και πολύπλοκα πλαίσια βελτιστοποίησης. Μέσω της προόδου του τομέα Βαθιάς Μάθησης (Deep Learning), όπου περιλαμβάνονται αλγόριθμοι μεγάλης ακρίβειας ενισχύθηκε και ο τομέας της μηχανικής όρασης. (Wikipedia, n.d.)

1.2 Διαφορά μεταξύ Υπολογιστικής και Μηχανικής Όρασης

Η υπολογιστική όραση είναι η επιστήμη της αναπαραγωγής τμημάτων της πολυπλοκότητας του ανθρώπινου συστήματος όρασης προκειμένου να εξάγει χρήσιμες πληροφορίες από ψηφιακές εικόνες ή βίντεο. Δηλαδή, είναι ένα πεδίο της επιστήμης των υπολογιστών που ασχολείται με τεχνολογίες και εργαλεία για να κάνει τους υπολογιστές να βλέπουν αυτό ακριβώς που βλέπει ένας άνθρωπος και να ερμηνεύουν τον κόσμο γύρω τους. Χάρη στις τεχνολογικές εξελίξεις στον τομέα της τεχνητής νοημοσύνης και της μηχανικής μάθησης, η υπολογιστική όραση έχει κάνει τεράστιο άλμα προς το μέλλον. Στόχος είναι να δημιουργηθούν ψηφιακά συστήματα, τα οποία θα αναπαράγουν το σύστημα ανθρώπινης όρασης ώστε να επεξεργάζονται, να αναλύουν και να κατανοούν τις ψηφιακές εικόνες και να εξάγουν

ουσιαστικές γνώσεις από τον πραγματικό κόσμο. Σκοπός, λοιπόν, της υπολογιστικής όρασης είναι η αυτοματοποίηση των εργασιών που μπορεί να κάνει το ανθρώπινο οπτικό σύστημα, δηλαδή η διασφάλιση της κατανόησης μίας εικόνας από μηχανήματα έτσι ακριβώς όπως θα την κατανοούσε ένας άνθρωπος ή μερικές φορές και καλύτερα.



Εικόνα 2. Υπολογιστική όραση σε πραγματικό χρόνο (Khillar, 2022)


Η μηχανική όραση από την άλλη πλευρά είναι η χρήση ήδη υπαρχουσών τεχνολογιών και εργαλείων που βοηθούν τις μηχανές να μεταφέρουν με ακρίβεια συγκεκριμένα δεδομένα ή πληροφορίες χρησιμοποιώντας νέους τρόπους, με σκοπό να εφαρμοστούν αυτές οι πληροφορίες για να επιλυθούν προβλήματα στον πραγματικό κόσμο. Το πεδίο εφαρμογής της μηχανικής όρασης περιορίζεται στα βιομηχανικά περιβάλλοντα, όπου οι εργασίες είναι πολύ συγκεκριμένες και οι διαδικασίες απλοποιημένες και γνωστές. Η μηχανική όραση στις μέρες μας έχει αναπτυχθεί τόσο πολύ που συμπεριλαμβάνεται όχι μόνο σε όλες τις βιομηχανικές εφαρμογές αλλά και σε άλλες που δεν έχουν καμία σχέση με την βιομηχανική παραγωγή. Μία από τις πιο κοινές εφαρμογές του συστήματος μηχανικής όρασης στην πράξη είναι η επιθεώρηση προϊόντων σε μία παραγωγική διαδικασία, όπου θα πρέπει οι εργασίες και οι συνθήκες να έχουν αξιολογηθεί πριν τον σχεδιασμό του κατάλληλου συστήματος. Η μηχανική όραση χρησιμοποιείται επίσης και σε διάφορους τομείς επαγγελματικών εφαρμογών, όπως επιθεώρηση, καθοδήγηση, αναγνώριση και παρακολούθηση.

Συνεπώς, στόχος της υπολογιστικής όρασης είναι η δημιουργία ψηφιακών συστημάτων αντιγράφοντας το σύστημα ανθρώπινης όρασης προκειμένου να επεξεργαστεί, να αναλύσει και να κατανοήσει ψηφιακές εικόνες και να εξάγει ουσιαστικές γνώσεις από την ψηφιακή είσοδο. Άρα η ιδέα είναι να εξαχθούν χρήσιμες πληροφορίες από ψηφιακές εικόνες ή βίντεο. Η διαφορά, λοιπόν, έγκειται στο ότι η μηχανική όραση έχει ως στόχο να αντικαταστήσει τις δυνατότητες του ανθρώπινου οπτικού συστήματος με μία κάμερα και έναν υπολογιστή προκειμένου να συλλέξει όσο το δυνατόν περισσότερες χρήσιμες πληροφορίες για τον οπτικό κόσμο και να παραμελήσει τις περιττές πληροφορίες. Η μηχανική όραση επεκτείνει τις εφαρμογές της υπολογιστικής όρασης σε βιομηχανικό και παραγωγικό περιβάλλον. Παρόλο που η υπολογιστική όραση είναι αυτή που θα εστιάσει στην πλήρη κατανόηση της ψηφιακής πληροφορίας που θα εισαχθεί, καθώς και στην επεξεργασία και την ανάλυση της, υπάρχουν βιομηχανικές εφαρμογές όπου δεν απαιτούν τόσο προηγμένες δυνατότητες όπως αυτές που παρέχει η υπολογιστική όραση. Για αυτές τις περιπτώσεις υπάρχει η μηχανική όραση. Βέβαια, σύμφωνα με την Βικιπαίδεια η μηχανική όραση φαίνεται να είναι παρακλάδι της υπολογιστικής όρασης και έτσι θα θεωρηθεί στην παρούσα εργασία. (Wikipedia, n.d.)

Computer Vision VS Machine Vision

Comparison Chart

Computer Vision	Machine Vision
Focuses on replicating parts of the complexity of the human vision system.	Uses digital input captured by a camera or multiple cameras to determine action.
The goal is to extract meaningful information or insights from digital images or videos.	The processing of images delivered to the computer by sensors is the core of machine vision.
It aims to create digital systems by replicating the human vision system.	It substitutes capabilities of human vision and judgment with a camera and a computer.



Εικόνα 3. Υπολογιστική Όραση vs Μηχανική όραση (Khillar, 2022)

1.3 Υπολογιστική όραση και αυτόνομα οχήματα

Ο παγκόσμιος κλάδος της αυτοκινητοβιομηχανίας αντιμετωπίζει διάφορες προκλήσεις, η υπολογιστική όραση δίνει τη δυνατότητα στον τομέα της αυτοκινητοβιομηχανίας να γίνει πιο έξυπνος, ασφαλέστερος και πιο αποτελεσματικός. Έτσι στην αυτοκινητοβιομηχανία η υπολογιστική όραση χρησιμοποιείται για την επιθεώρηση κάθε λεπτομέρειας κατά την Παρασκευή

Η αυτοοδηγούμενη αυτοκινητοβιομηχανία έχει αναπτυχθεί σημαντικά τα τελευταία χρόνια, σε μεγάλο βαθμό η αυτονομία τέτοιου είδους οχημάτων επιτρέπεται μέσω της χρήσης τεχνικών υπολογιστικής όρασης τελευταίας τεχνολογίας. Η τεχνολογία της υπολογιστικής όρασης μπορεί να κάνει το αυτόνομο όχημα ασφαλέστερο, εφόσον μέσω αυτής γίνεται χρήση αυτής στην αντίληψη του περιβάλλοντος. (Neuromation, 2018) (Rafiaue, 2020)

1.3.1 Αυτοκατευθυνόμενα Οχήματα

Αυτοκατευθυνόμενο, αυτοοδηγούμενο ή ρομπωτικό όχημα, ονομάζεται ένα όχημα εδάφους που είναι σε θέση να αισθάνεται το περιβάλλον μέσα στο οποίο κινείται, μέσω ενσωματωμένων αυτοματισμών και να κινείται με ασφάλεια είτε με την ελάχιστη ανθρώπινη συνεισφορά, είτε χωρίς αυτήν. Αυτού του είδους τα οχήματα συνδυάζουν μία ποικιλία αισθητήρων για να αντιληφθούν το περιβάλλον στο οποίο κινούνται, όπως κάμερες (κυρίως θερμογραφικές), ραντάρ, GPS,

οδομετρία και μονάδες μέτρησης αδράνειας, sonar¹, lidar². Τα προηγμένα συστήματα ελέγχου ερμηνεύουν τις αισθητηριακές πληροφορίες για να εντοπίσουν τις κατάλληλες διαδρομές πλοήγησης, τις σημάσεις και τα εμπόδια. Οι μέθοδοι ελέγχου που βασίζονται στην Τεχνητή Νοημοσύνη μπορούν να χρησιμοποιηθούν για την εκμάθηση των αισθητηριακών πληροφοριών που έχουν συγκεντρωθεί, ώστε να υποστηρίξουν διαφορετικές εργασίες της αυτόνομης οδήγησης. Η αυτονομία των οχημάτων κατηγοριοποιείται σε έξι επίπεδα σύμφωνα με ένα σύστημα που αναπτύχθηκε από την SAE International³, αυτά τα επίπεδα είναι τα εξής:

Επίπεδο 0 : Χωρίς αυτοματισμό οδήγησης. Είναι το επίπεδο όπου ο οδηγός ελέγχει πλήρως το όχημα και είναι πάντα αφοσιωμένος στη λειτουργία του οχήματος που περιλαμβάνει το τιμόνι, την επιτάχυνση, το φρενάρισμα, τη στάθμευση και οποιουδήποτε άλλους απαραίτητους ελιγμούς ή ενέργειες για την οδήγηση ή την ακινητοποίηση του οχήματος. Τα οχήματα στερούνται εντελώς τεχνολογίες αυτοματισμού οδήγησης. Σε αυτό το επίπεδο δεν υπάρχει κανένας αυτοματισμός. Μπορεί το αυτοματοποιημένο σύστημα να εκδίδει προειδοποιήσεις και να παρέμβει στιγμιαία, αλλά σε καμία περίπτωση δεν έχει τον σταθερό έλεγχο του οχήματος, έτσι αυτές οι λειτουργίες δεν θεωρούνται αυτόνομες, επομένως οι οδηγοί πρέπει να παραμείνουν πλήρως προσεκτικοί και αφοσιωμένοι. Παράδειγμα περιορισμένης ή στιγμιαίας υποβοήθησης οδηγού περιλαμβάνουν το αυτόματο φρενάρισμα έκτακτης ανάγκης, την προειδοποίηση εμπρόσθιας σύγκρουσης και την προειδοποίηση αναχώρησης από την λωρίδα κυκλοφορίας.

Επίπεδο 1 : Βοήθεια οδηγού, σε αυτό το επίπεδο το σύστημα αυτοκινήτου παρέχει συνεχή υποβοήθηση με την επιτάχυνση, το φρενάρισμα ή το τιμόνι. Συγκεκριμένα παραδείγματα τεχνολογιών υποβοήθησης οδηγού επιπέδου 1 περιλαμβάνουν το προσαρμοστικό cruise control⁴ και την υποβοήθηση διατήρησης λωρίδας. Τα συστήματα υποβοήθησης οδηγού επιπέδου 1 που λειτουργούν μεμονωμένα δεν θεωρούνται αυτόνομη τεχνολογία, ωστόσο, τα προσαρμοστικά συστήματα υποβοήθησης Cruise control και διατήρησης λωρίδας που λειτουργούν ταυτόχρονα πληρούν τις προϋποθέσεις για αυτοματοποίηση επιπέδου 2. Παράδειγμα αυτοοδήγησης επιπέδου 1 αποτελεί το αυτόματο φρενάρισμα έκτακτης ανάγκης που ειδοποιεί τον οδηγό για σύγκρουση και επιτρέπει την πλήρη ικανότητα πέδησης. Τόσο η τεχνολογία επιπέδου 1 όσο και επιπέδου 2 απαιτούν από τους οδηγούς να παραμείνουν προσεκτικοί και πλήρως αφοσιωμένοι όταν χειρίζονται μηχανοκίνητο όχημα.

Επίπεδο 2 : Αυτό το επίπεδο χαρακτηρίζεται ως μερικός αυτοματισμός οδήγησης. Τα οχήματα παρέχουν μερικούς αυτοματισμούς βοηθώντας συνεχώς τους οδηγούς στην επιτάχυνση, το φρενάρισμα και το τιμόνι. Τα οχήματα επιπέδου 2 είναι συνήθως εξοπλισμένα με προηγμένα συστήματα υποβοήθησης οδηγού που μπορούν να αναλάβουν τον έλεγχο σε συγκεκριμένα σενάρια. Ο οδηγός σε αυτό το επίπεδο οφείλει να παρακολουθεί το οδόστρωμα και να είναι ανά πάσα στιγμή διαθέσιμος ώστε αν παραστεί η ανάγκη και το αυτοματοποιημένο σύστημα δεν ανταποκριθεί σωστά να πάρει τον έλεγχο του οχήματος. Οι εργασίες μοιράζονται μεταξύ οδηγού και του αυτοματοποιημένου συστήματος.

¹ sound navigation ranging / ηχητική πλοήγηση και εμβέλεια: το ενεργά συστήματα σόναρ εκπέμπουν ηχητικούς παλμούς και διαβάζουν την ηχώ που επιστρέφεται από φυσικές επιφάνειες.

² Light Detection And Ranging, βασίζεται στην εκπομπή παλμικής ακτινοβολίας λέιζερ στην ατμόσφαιρα και στην καταγραφή της οπισθοσκεδαζόμενης ακτινοβολίας λέιζερ.

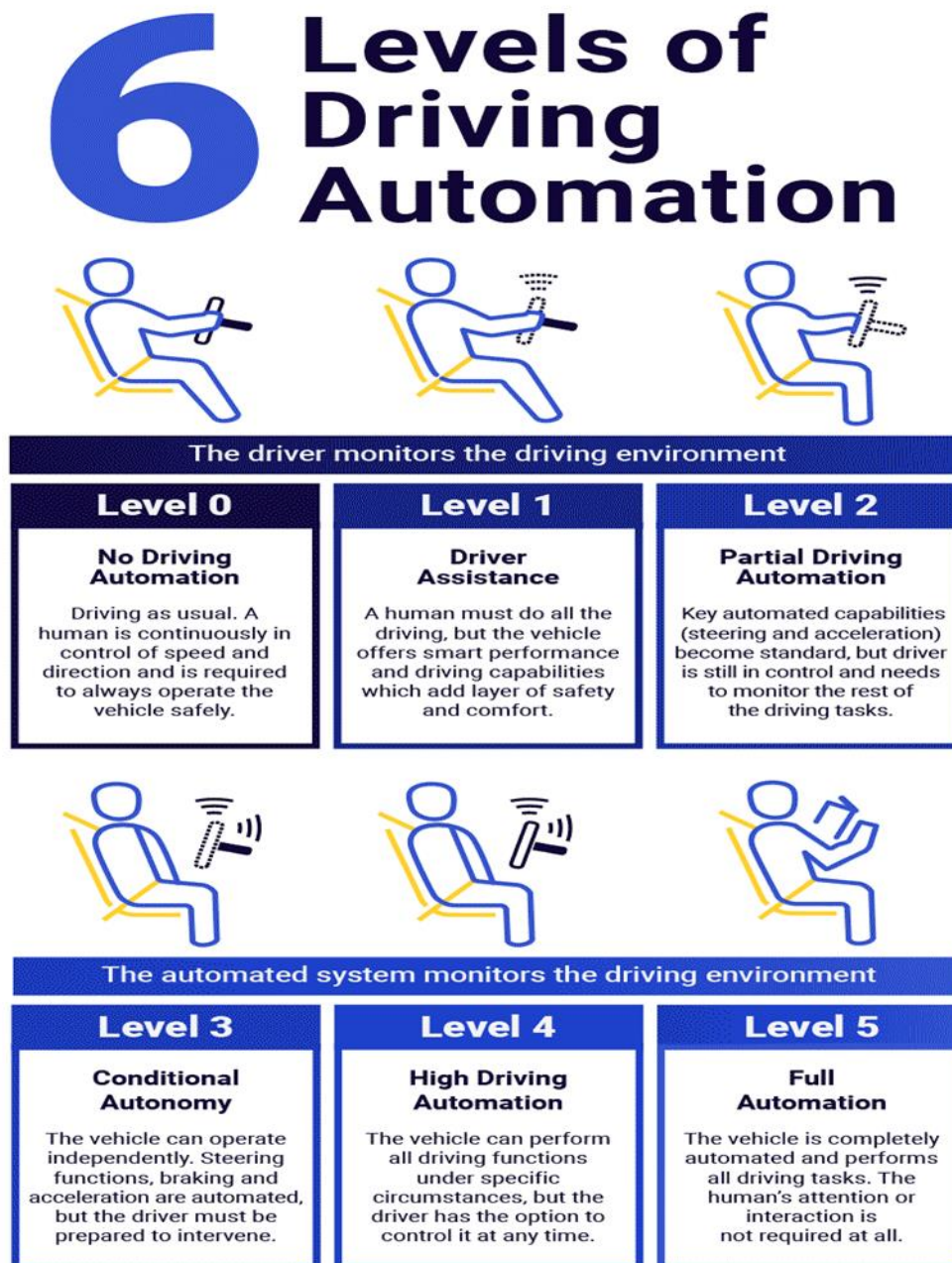
³ Η SAE International είναι μία παγκόσμια ενεργή επαγγελματική ένωση και οργάνωση προτύπων που εδρεύει στις Ηνωμένες Πολιτείες και αποτελείται από περισσότερους από 128000 μηχανικούς και τεχνικούς εμπειρογνώμονες στη μηχανική κινητικότητα.

⁴ Το cruise control είναι ένα σύστημα που ελέγχει αυτόματα την ταχύτητα ενός μηχανοκίνητου οχήματος. Το σύστημα είναι ένας σερβομηχανισμός που αναλαμβάνει το γκάζι του αυτοκινήτου για να διατηρεί μία σταθερή ταχύτητα όπως ορίζεται από τον οδηγό.

Επίπεδο 3 : Σε αυτό το επίπεδο τα οχήματα χειρίζονται αυτόνομα όλες τις εργασίες οδήγησης, ωστόσο οι οδηγοί πρέπει να είναι διαθέσιμοι πάντα για να αναλάβουν το τιμόνι σε περίπτωση που το προηγμένο σύστημα υποβοήθησης οδηγού σταματήσει ξαφνικά να λειτουργεί αποτελεσματικά.

Επίπεδο 4 : Το επίπεδο αυτό θεωρείται το επίπεδο υψηλού αυτοματισμού οδήγησης, όπου τα αυτόνομα συστήματα οχημάτων είναι πλήρως υπεύθυνα για όλες τις εργασίες οδήγησης και πλοήγησης. Τα οχήματα αυτά μπορούν να μεταφέρουν επιβάτες αυτόνομα, δηλαδή δεν χρειάζονται ούτε την εμπλοκή, ούτε την ετοιμότητα του οδηγού ώστε να πάρει τον έλεγχο του οχήματος. Όμως, το σύστημα αυτοματισμού επιπέδου 4 συνήθως περιορίζεται σε συγκεκριμένες γεωγραφικές τοποθεσίες και δεν μπορούν να ταξιδέψουν εκτός των καθορισμένων περιοχών εξυπηρέτησης ή κατά τη διάρκεια επικίνδυνων καιρικών συνθηκών. Ο αυτοματισμός οδήγησης επιπέδου 4 είναι κατάλληλος ιδιαίτερα για ταξί και λεωφορεία χωρίς οδηγό σε καθορισμένες διαδρομές, επίσης σε φορτηγά που μεταφέρουν εμπορεύματα εντός συγκεκριμένων γεωγραφικών ορίων, όπως και σε λεωφορεία επιβατών και φορτίων των αεροδρομίων.

Επίπεδο 5 : Ο πλήρης αυτοματισμός οδήγησης. Στο επίπεδο αυτό τα αυτόνομα οχήματα αναλαμβάνουν τον πλήρη έλεγχο όλων των καθηκόντων οδήγησης και πλοήγησης. Οι επιβάτες απλώς ορίζουν έναν προορισμό. Στο μέλλον, οχήματα εξοπλισμένα με συστήματα πλήρους αυτοματισμού οδήγησης επιπέδου 5 θα λειτουργούν ανεξάρτητα και καθολικά σε όλες τις καιρικές συνθήκες και δρόμους. (Rafique, 2020)

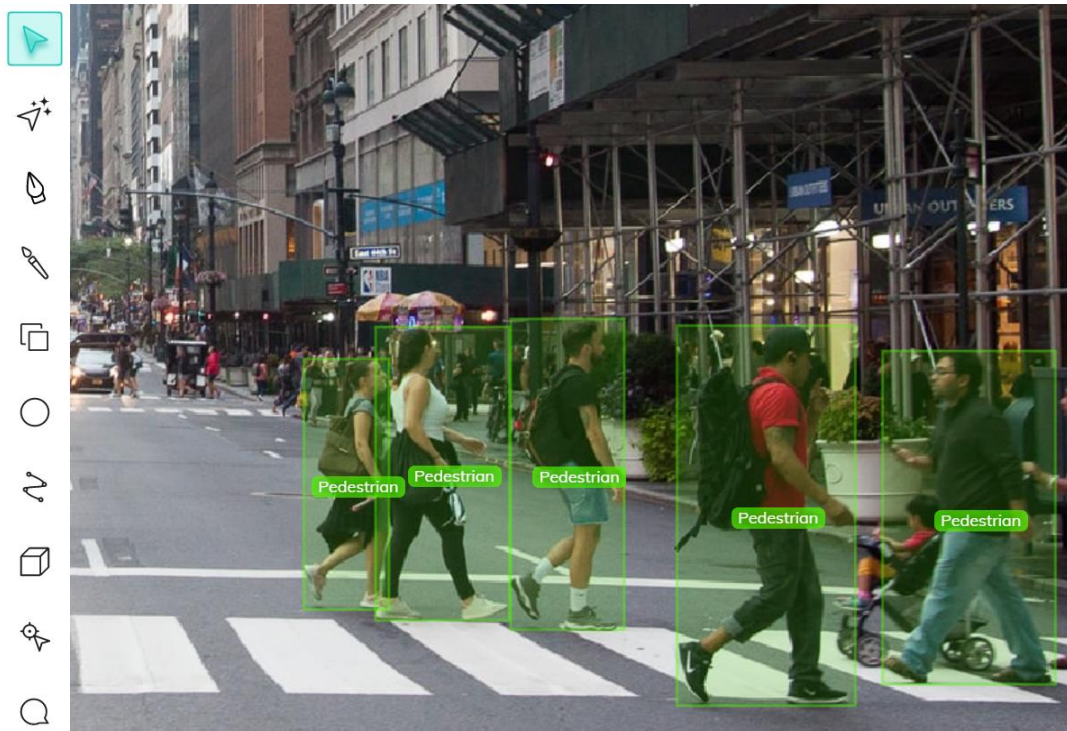


Εικόνα 4. Τα επίπεδα αυτονομίας οδήγησης (Rambus Press, 2022)

1.4 Άλλες εφαρμογές υπολογιστικής όρασης

Μεταφορές : Πέραν των αυτόνομων οχημάτων η υπολογιστική όραση εφαρμόζεται και σε άλλες περιπτώσεις αυτού του κλάδου.

- Η ανίχνευση και παρακολούθηση πεζών αποτελεί έναν από τους σημαντικότερους τομείς έρευνας για την υπολογιστική όραση, καθώς επιδρά στον σχεδιασμό συστημάτων προστασίας πεζών και έξυπνων πόλεων.



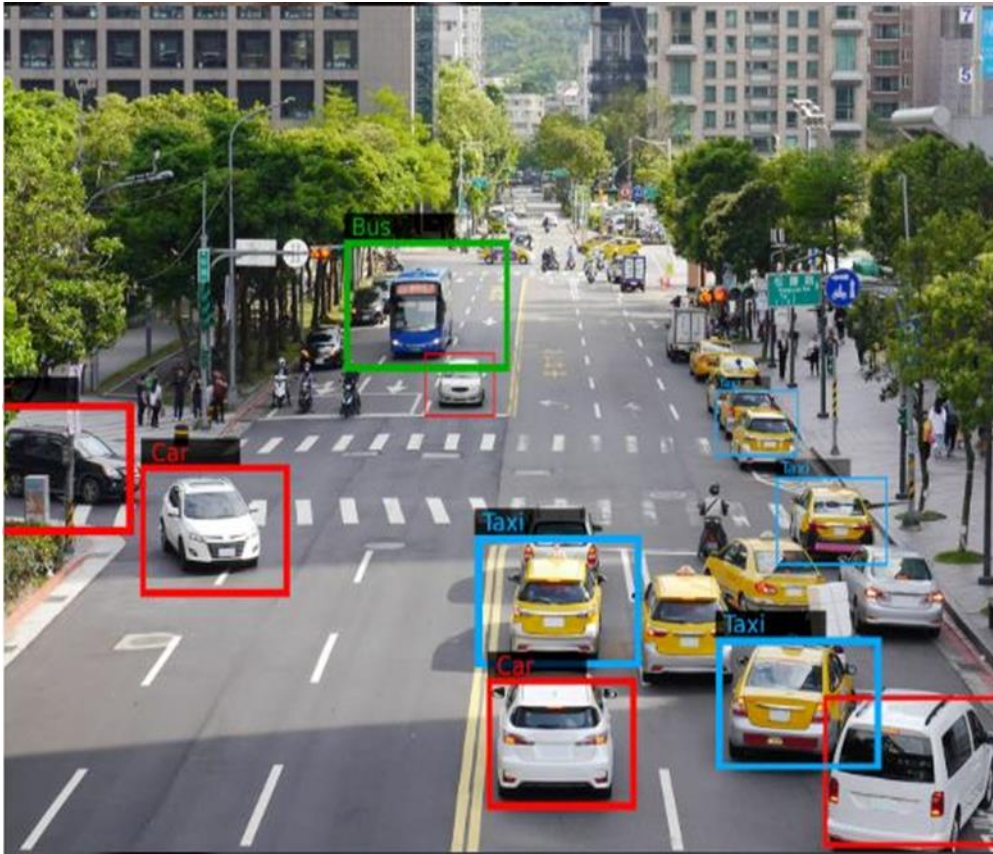
Εικόνα 5. Ανίχνευση πεζών με υπολογιστική όραση (Kunanbaeva, 2020)

- Η ανίχνευση κατάληψης στάθμευσης, όπου η υπολογιστική όραση χρησιμοποιείται για την ανίχνευση κατειλημμένων χώρων στάθμευσης στα συστήματα Καθοδήγησης και Πληροφοριών Στάθμευσης (PGI). Χάρη στην ανάπτυξη των Συνελκτικών Νευρωνικών Δικτύων (CNN), τα συστήματα ανίχνευσης κατάληψης στάθμευσης που βασίζονται σε κάμερα πέτυχαν σε σύντομο χρονικό διάστημα υψηλή ακρίβεια που δεν είναι επιρρεπής σε αλλαγές φωτισμού και καιρικών συνθηκών. Η ανίχνευση κατάληψης στάθμευσης μπορεί να συνδυαστεί με αναγνώριση πινακίδας κυκλοφορίας για να παρακολουθείται ανά πάσα στιγμή ποιο αυτοκίνητο καταλαμβάνει ποια θέση.



Εικόνα 6. Κενές και κατειλημμένες θέσης στάθμευσης (Kunanbaeva, 2020)

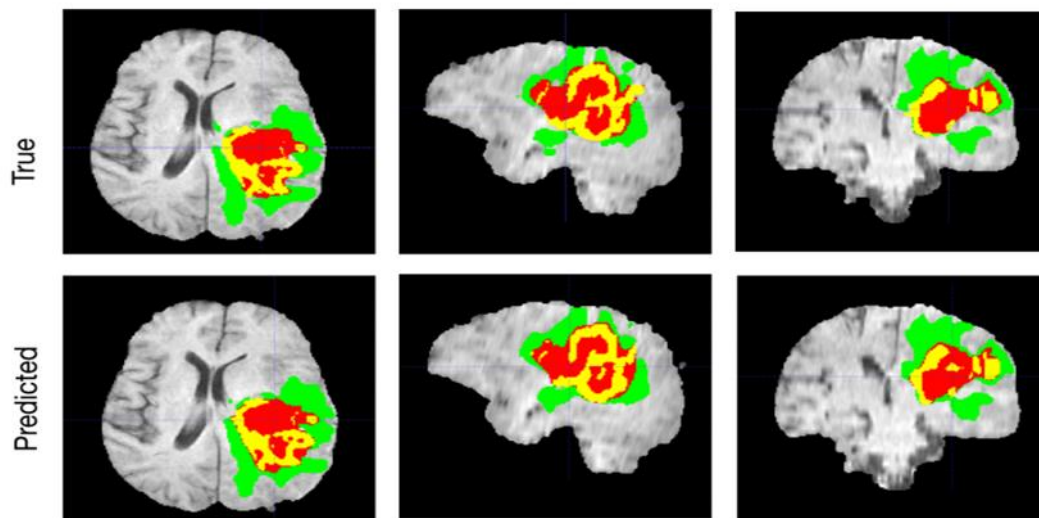
- Η ανάλυση ροής της κυκλοφορίας που γίνεται με την βοήθεια μη επανδρωμένου αεροσκάφους (drone) και την καταγραφή ροής της κυκλοφορίας μέσω κάμερας κατέστη δυνατή χάρης της εξέλιξης του τομέα υπολογιστικής όρασης. Μέσω αλγορίθμων γίνεται η παρακολούθηση και ανάλυση της πυκνότητας της κυκλοφορίας σε αστικές περιοχές με αποτέλεσμα τον σχεδιασμό καλύτερων συστημάτων διαχείρισης της κυκλοφορίας και την βελτίωση της οδικής ασφάλειας.



Εικόνα 7. Ανάλυση ροής κυκλοφορίας (James Allen on September 4, 2019)

- Η εποπτεία των συνθηκών των δρόμων, εδώ η υπολογιστική όραση χρησιμοποιείται για την ανίχνευση ελαττωμάτων στα οδοστρώματα καθώς και η ανίχνευση αλλαγών στο σκυρόδεμα και στην άσφαλτο. Η αυτόματη ανίχνευση κινδύνων οδοστρώματος έχει αποδώσει τα μέγιστα στην συντήρηση των δρόμων και στην αποφυγή ατυχημάτων. Τα δεδομένα που συλλέγονται από τις εικόνες υποβάλλονται σε επεξεργασία με σκοπό την δημιουργία συστημάτων ανίχνευσης και ταξινόμησης ρωγμών που επιτρέπουν την στοχευμένη αποκατάσταση και προληπτική συντήρηση των οδοστρωμάτων.

Ιατρική : Ένα από τα πιο σημαντικά πεδία εφαρμογής της υπολογιστικής όρασης, που χαρακτηρίζεται από την εξαγωγή πληροφοριών από δεδομένα εικόνας για την διάγνωση ενός ασθενούς, για παράδειγμα η ανίχνευση όγκων, αρτηριοσκλήρωσης ή μετρήσεις διαστάσεων οργάνων. Ενισχύει την ιατρική έρευνα παρέχοντας πληροφορίες για τον εγκέφαλο και για την ποιότητα των ιατρικών θεραπειών και τέλος, με την μείωση της επίδρασης του θορύβου, παρέχει βελτιωμένες εικόνες υπερήχων και εικόνες ακτινών Χ .



Εικόνα 8. Τμηματοποίηση Όγκου εγκεφάλου με τεχνητή νοημοσύνη (Alarcon, 2018)

Στρατιωτικές εφαρμογές : Οι στρατιωτικές εφαρμογές αποτελούν έναν από τους μεγαλύτερους τομείς της υπολογιστικής όρασης. Τα πιο προφανή παραδείγματα αποτελούν ο εντοπισμός εχθρικών οχημάτων ή στρατιωτών, καθώς και η καθοδήγηση πυραύλων. Πιο προηγμένα συστήματα καθοδήγησης πυραύλων στέλνουν το βλήμα σε μία περιοχή, χωρίς να είναι προκαθορισμένος ο στόχος και όταν ο πύραυλος φτάσει στην περιοχή συλλέγει δεδομένα τοπικών εικόνων και έπειτα επιλέγεται ο στόχος. Η έννοια της επίγνωσης του πεδίου μάχης υποδηλώνει την χρήση διαφόρων αισθητήρων εικόνας, οι οποίοι παρέχουν πλούσιο σύνολο πληροφοριών για μία σκληρή μάχη που μπορεί να χρησιμοποιηθεί για την υποστήριξη στρατηγικών αποφάσεων.

Στον κατασκευαστικό τομέα : όπου η υπολογιστική όραση χρησιμοποιείται στην ανίχνευση κινδύνων στο χώρο εργασίας ή στην επιθεώρηση στοιχείων υποδομών και στην προγνωστική συντήρηση για την αποφυγή ατυχημάτων και την βελτίωση υποδομών .

Στην γεωργία : Σε αυτόν τον τομέα έχουν χρησιμοποιηθεί αρκετά τα μοντέλα τεχνητής νοημοσύνης συμπεριλαμβανομένης και της υπολογιστικής όρασης σε τομείς όπως η παρακολούθηση των καλλιεργειών και της απόδοσης, η αυτοματοποιημένη συγκομιδή, η ανάλυση καιρικών συνθηκών, η παρακολούθηση της υγείας των ζώων ή την ανίχνευση ασθενειών των φυτών.



Εικόνα 9. Παρακολούθηση καλλιέργειας και απόδοσης (Rizzoli, 2022)

Στο λιανεμπόριο : Οι κάμερες που είναι εγκατεστημένες σε καταστήματα λιανικής επιτρέπουν στους πωλητές να συλλέγουν μεγάλοι όγκου οπτικών δεδομένων που βοηθούν στο σχεδιασμό καλύτερης λειτουργίας του λιανεμπορίου.



Εικόνα 10. Αυτόματο ταμείο (Rizzoli, 2022)

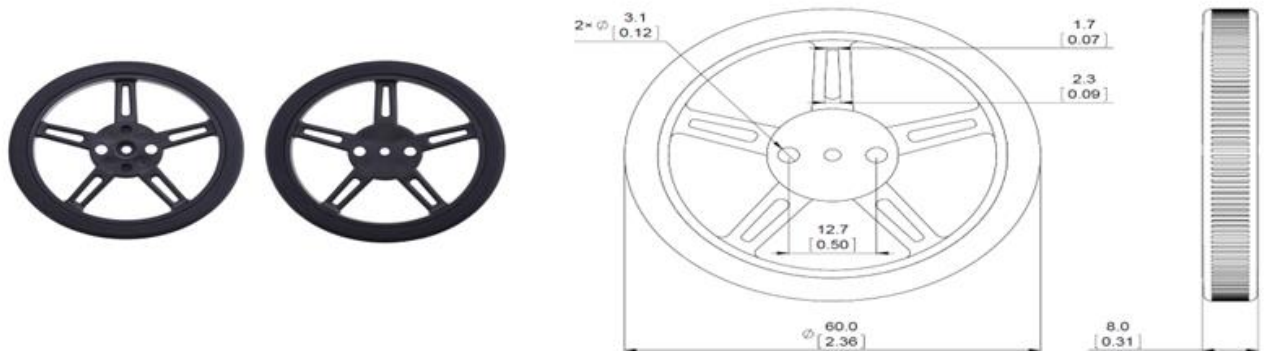
2 Το υλικό (hardware)

Το υλικό που χρησιμοποιήθηκε είναι το εξής :

2.1 Ρόδες με ελαστικό από σιλικόνη 60x8 mm για σερβοκινητήρες

Αυτές οι ρόδες είναι ειδικά σχεδιασμένες για τη χρήση από σερβομηχανισμούς μικρού συνεχούς περιστροφής FS90R και ψηφιακούς σερβομηχανισμούς μικροσυνεχούς περιστροφής FT90R της Feetech, παλαιότερα γνωστή ως Fitech. Ο συνδυασμός αυτός χρησιμεύει ως ένα μικρό, φθινό και εύκολο στον έλεγχο σύστημα κίνησης για μικρό ρομπότ. Οι τροχοί πιέζουν πάνω στο σερβοχέλιο και με την τοποθέτηση της μικρής βίδας που περιλαμβάνεται στο πακέτο εξαρτημάτων FS90R/FT90R συγκρατούνται στην θέση τους.

Οι τροχοί έχουν ελαστικά από σιλικόνη και η διάμετρος τους είναι 60mm (2,36"), επίσης αυτά τα ελαστικά διατίθενται και χωριστά. Οι δύο πρόσθετες οπές στερέωσης για βίδες τύπου 4 - 40 καθιστούν δυνατή την χρήση των τροχών για λήνες στερέωσης γενικής χρήσης για άξονες 3 mm, άξονες 4 mm και άξονες 5 mm. Οι υποδοχές που υπάρχουν στις πέντε ακτίνες επιτρέπουν την τοποθέτηση εξαρτημάτων στους τροχούς και ταιριάζουν στις βίδες #2 ή M2. Οι ακριβείς διαστάσεις τους παρουσιάζονται αναλυτικότερα στην εικόνα 12.



Εικόνα 11. Δύο ρόδες με ελαστικό από σιλικόνη 60x8 mm for FS90R servo (Pololu.com, n.d.)

2.2 Σερβοκινητήρας

Ο Σερβοκινητήρας δεν είναι τίποτε άλλο παρά ένας περιστροφικός ή γραμμικός ενεργοποιητής που επιτρέπει τον ακριβή έλεγχο της επιτάχυνσης και της ταχύτητας καθώς και τον έλεγχο της γωνιακής ή γραμμικής θέσης. Ο σερβοκινητήρας αποτελείται από ένα κατάλληλο κινητήρα συνδεδεμένο με ένα αισθητήρα για αναπληροφόρηση θέσης. Για την χρήση του απαιτείται ένας σχετικά εξελιγμένος ελεγκτής, συνήθως μια ειδική μονάδα που έχει σχεδιαστεί κυρίως για τον συνδυασμό με τους σερβοκινητήρες. (en.wikipedia.org, n.d.)

Παρ' όλο που ο όρος 'σερβοκινητήρας' παραπέμπει και συνήθως αναφέρεται σε κινητήρα κατάλληλο για χρήση σε ένα σύστημα ελέγχου κλειστού βρόχου, οι συγκεκριμένοι σερβοκινητήρες δεν αποτελούν κατηγορία κινητήρα. Οι σερβοκινητήρες χρησιμοποιούνται σε εφαρμογές όπως η ρομποτική, τα μηχανήματα CNC και οι αυτοματοποιημένες κατασκευές.



Εικόνα 12. Δύο Feetech FS90R Continuous Servo Motor, σερβομηχανές

Ένας σερβοκινητήρας είναι ένας σερβομηχανισμός κλειστού βρόχου που χρησιμοποιεί αναπληροφόρηση θέσης για να ελέγξει την κίνηση και την τελική θέση. (“Βοηθητικό μοτέρ”). Η είσοδος στον έλεγχο του είναι ένα σήμα, είτε ψηφιακό, είτε αναλογικό, το οποίο αντιπροσωπεύει τη θέση για την οποία έχει δοθεί εντολή να προσαρμόζει τον άξονα εξόδου.

Ο κινητήρας συνδυάζεται με κάποιου τύπου κωδικοποιητή θέσης για να παρέχει ανάδραση θέσης και ταχύτητας. Στην απλούστερη περίπτωση μετριέται μόνο η θέση. Έπειτα, η θέση που έχει μετρηθεί για την έξοδο συγκρίνεται με τη θέση εντολής και την εξωτερική είσοδο στον ελεγκτή. Εάν η θέση εξόδου διαφέρει από την απαιτούμενη, θα δημιουργηθεί ένα σήμα σφάλματος το οποίο με την σειρά του θα αναγκάσει τον κινητήρα να περιστραφεί προς οποιαδήποτε κατεύθυνση, ώστε να φέρει τον άξονα εξόδου στην κατάλληλη θέση. Όσο αρχίζουν να πλησιάζουν οι θέσεις, θα αρχίσει και το σήμα σφάλματος να μειώνεται στο μηδέν με αποτέλεσμα να σταματήσει ο κινητήρας. Στους απλούς σερβοκινητήρες η ανίχνευση θέσης πραγματοποιείται μόνο μέσω ελέγχου bang-bang του κινητήρα τους και ενός ποτενσιόμετρου. Ο κινητήρας είτε θα περιστρέφεται πάντα σε πλήρη ταχύτητα είτε θα είναι σταματημένος. Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι αυτού του τύπου σερβοκινητήρες δεν χρησιμοποιούνται ευρέως στον βιομηχανικό έλεγχο κίνησης, όμως αποτελούν την βάση των απλών και φθηνών σερβοκινητήρων που χρησιμοποιούνται για τα ραδιοελεγχόμενα μοντέλα. Οι πιο εξελιγμένοι σερβοκινητήρες χρησιμοποιούν οπτικούς περιστροφικούς κωδικοποιητές για τη μέτρηση της ταχύτητας του άξονα εξόδου και μια μεταβλητή ταχύτητας της κίνησης για τον έλεγχο της ταχύτητας του κινητήρα. (“Βοηθητικό μοτέρ”) Ο συνδυασμός αυτών των δύο βελτιώσεων με τον αλγόριθμο ελέγχου PID, επιτρέπουν στον σερβοκινητήρα να έρχεται στη θέση εντολής του πιο γρήγορα, με μεγαλύτερη ακρίβεια και με λιγότερη προσπάθεια (overshooting). (The Pi Hut, n.d.) (Wikipedia, n.d.)

Το FS90R είναι ένας μικρού μεγέθους σερβομηχανισμός που κατασκευάζεται από την εταιρεία Feetech, γνωστή παλιότερα ως Fitech, και χρησιμοποιείται ειδικά για συνεχή περιστροφή. Στα 6V έχει την μέγιστη ταχύτητα περιστροφής, δηλαδή περίπου 130 RPM (Revolution per minute, περιστροφή ανά λεπτό). Το RPM είναι μονάδα περιστροφικής ταχύτητας ή συχνότητας περιστροφής γύρω από συγκεκριμένο άξονα) χωρίς φορτίο και μπορεί να παράγει έως και 21 oz-in (1,5 kg-cm) ροπή. Το γεγονός ότι ο συγκεκριμένος σερβομηχανισμός μπορεί να ελεγχθεί από μικροελεγκτή με μια γραμμή εισόδου / εξόδου, χωρίς πρόσθετα ηλεκτρονικά, χρησιμοποιώντας μία απευθείας σύνδεση, τον καθιστά ιδιαίτερα εύχρηστο μικροσκοπικό ελεγκτή ακόμα και για την κατασκευή απλών ρομπότ. Ο σερβομηχανισμός συνεχούς περιστροφής FS90R μετατρέπει τους τυπικούς RC παλμούς θέσης σερβομηχανισμού σε ταχύτητα συνεχούς περιστροφής. Το προεπιλεγμένο σημείο ηρεμίας είναι 1,5 ms, αλλά αυτό μπορεί να ρυθμιστεί, περιστρέφοντας το ποτενσιόμετρο ρύθμισης μεσαίου μεγέθους με ένα κατσαβίδι. Τα πλάτη των παλμών πάνω από το σημείο ηρεμίας έχουν ως αποτέλεσμα την αριστερόστροφη περιστροφή, με την ταχύτητα να αυξάνεται καθώς αυξάνεται το πλάτος των παλμών. Τα πλάτη των παλμών που είναι κάτω από το σημείο ηρεμίας έχουν ως αποτέλεσμα την δεξιόστροφη περιστροφή, με την ταχύτητα να αυξάνεται καθώς μειώνεται το πλάτος των παλμών. Τέλος, ο συγκεκριμένος σερβομηχανισμός διαθέτει καλώδιο 10" (250 mm) που τερματίζεται με βύσμα τύπου JR. (Pololu.com, n.d.)

Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας



Εικόνα 13. Διάφορα μεγέθη συνεχούς σερβοκινητήρα (Pololu.com, n.d.)

2.2.1 Κινητήρας συνεχούς ρεύματος (Direct Current)

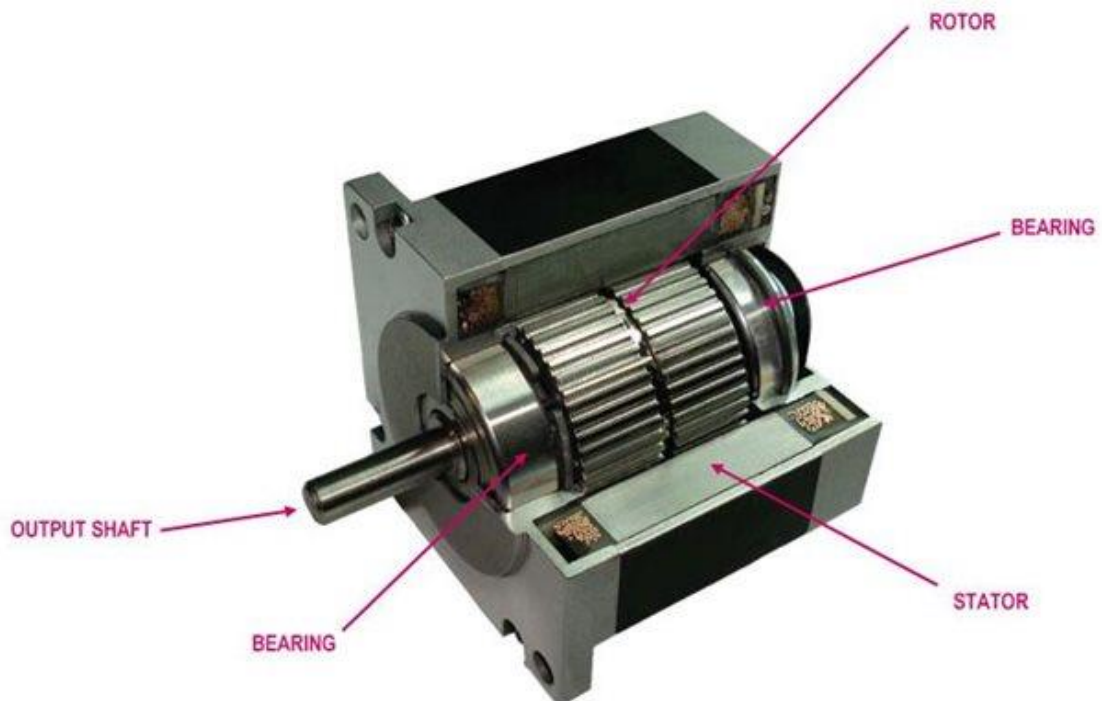
Οι κινητήρες συνεχούς ρεύματος είναι κινητήρες συνεχούς περιστροφής δυο καλωδίων, τροφοδοσίας και γείωσης. Στους DC κινητήρες όταν τα καλώδια συνδεθούν με το τροφοδοτικό αρχίζουν να περιστρέφονται μέχρι να αφαιρεθεί αυτή η ένταση. Οι περισσότεροι κινητήρες συνεχούς ρεύματος λειτουργούν με υψηλές στροφές ανά λεπτό. Οι κινητήρες συνεχούς ρεύματος μπορούν να χρησιμοποιηθούν για ανεμιστήρες ψύξης υπολογιστών, ραδιοελεγχόμενα αυτοκίνητα ή άλλες εφαρμογές αυτοκινήτων. Η ταχύτητα των συνεχούς ρεύματος κινητήρων ελέγχεται με την διαμόρφωση πλάτους παλμών (Pulse Width Modulation εν συντομία PWM), μια τεχνική γρήγορης παλμικής ενεργοποίησης και απενεργοποίησης. Το ποσοστό του χρόνου που αφιερώνεται για την εναλλαγή ενεργοποίησης/απενεργοποίησης καθορίζει την ταχύτητα του κινητήρα, για παράδειγμα εάν η τροφοδοσία εναλλάσσεται στο 50% (μισή ενεργοποιημένη και μισή απενεργοποιημένη), τότε η ταχύτητα που περιστρέφεται ο κινητήρας είναι η μισή από την μέγιστη ταχύτητα. Κάθε παλμός είναι τόσο γρήγορος που ο κινητήρας φαίνεται να περιστρέφεται συνεχώς χωρίς να κομπιάζει. (The Pi Hut, n.d.)



Εικόνα 14. Κινητήρας συνεχούς ρεύματος (Anon., n.d.)

2.2.2 Βηματικός κινητήρας (Stepper Motor) :

Ένας βηματικός κινητήρας στην ουσία είναι ένας σερβοκινητήρας που χρησιμοποιεί διαφορετική μέθοδο ελέγχου. Οι βηματικοί κινητήρες χρησιμοποιούν πολλαπλούς οδοντωτούς ηλεκτρομαγνήτες διατεταγμένους γύρω από ένα κεντρικό γρανάζι για να καθορίζουν τη θέση τους. Για τον έλεγχο του βηματικού κινητήρα, είναι απαραίτητο ένα εξωτερικό κύκλωμα ελέγχου ή ένας μικροελεγκτής (π.χ. Arduino ή Raspberry Pi) ούτως ώστε να ενεργοποιείται ξεχωριστά κάθε ηλεκτρομαγνήτης και να περιστρέφεται ο άξονας του κινητήρα. Όταν τροφοδοτούνται οι ηλεκτρομαγνήτες πρώτα ο "Α" και έπειτα ο "Β", ο άξονας του κινητήρα ευθυγραμμίζεται μεταξύ τους. Όταν ο "Α" είναι απενεργοποιημένος και ο "Β" είναι ενεργοποιημένος, το γρανάζι περιστρέφεται για να ευθυγραμμιστεί με τον μαγνήτη "Β". Κάθε ηλεκτρομαγνήτης είναι τοποθετημένος γύρω από το γρανάζι έτσι ώστε να δημιουργείται περιστροφή όταν αναβοσβήνει. Κάθε περιστροφή από τον ένα ηλεκτρομαγνήτη στον άλλο ονομάζεται << βήμα >>, και με αυτό τον τρόπο ο κινητήρας μπορεί να περιστραφεί με προκαθορισμένες γωνίες βημάτων μέσω μιας πλήρους περιστροφής 360 μοιρών. Ο σχεδιασμός του βηματικού κινητήρα παρέχει σταθερή ροπή συγκράτησης χωρίς να χρειάζεται ο κινητήρας να είναι ενεργοποιημένος. (The Pi Hut, n.d.)



Εικόνα 15 Βηματικός κινητήρας (www.heason.com, χ.χ.)

2.2.3 Πλεονεκτήματα και μειονεκτήματα των DC , Σερβοκινητήρων και Βηματικών κινητήρων :

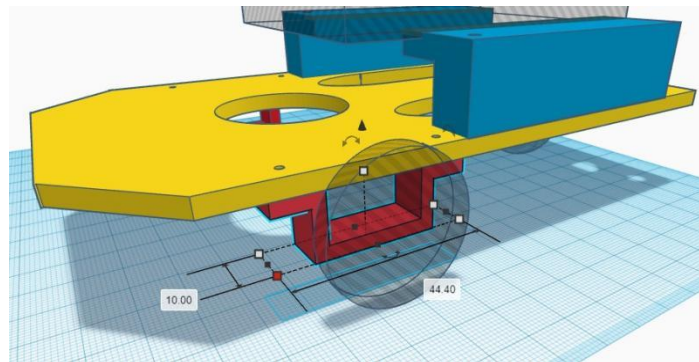
Οι κινητήρες συνεχούς ρεύματος (DC) είναι κινητήρες γρήγορης και συνεχούς περιστροφής που χρησιμοποιούνται κυρίως για οτιδήποτε χρειάζεται να περιστρέφεται με υψηλή περιστροφή ανά λεπτό (RPM), για παράδειγμα τροχούς αυτοκινήτου και ανεμιστήρες. Οι σερβοκινητήρες είναι υψηλής ροπής κινητήρες που περιστρέφονται με ακρίβεια σε περιορισμένη γωνία. Γενικά,

Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

αποτελούν μία εναλλακτική λύση υψηλής απόδοσης αντί των βηματικών κινητήρων (Stepper Motor), αλλά με πιο περίπλοκη ρύθμιση κατά τη διαμόρφωση του πλάτους παλμών (PWM) και συνήθως χρησιμοποιούνται στην ρομποτική. Τέλος οι βηματικοί κινητήρες είναι αργοί, εύκολη στη ρύθμιση, ακριβείς στην περιστροφή και στον έλεγχο. Πλεονεκτεί έναντι των άλλων κινητήρων, όπως και οι σερβοκινητήρες, στον έλεγχο μιας συγκεκριμένης θέσης. Οι βηματικοί κινητήρες ελέγχουν την θέση της κίνησης με την περιστροφή, χρησιμοποιώντας την πρόσθεση κλασμάτων και είναι κατάλληλοι για τρισδιάστατους εκτυπωτές. (The Pi Hut, n.d.)

2.3 Σασί

Το σασί είναι δύο επιπέδων φτιαγμένο από τρισδιάστατο εκτυπωτή, οι διαστάσεις του φαίνονται αναλυτικά στις φωτογραφίες. Στο πρώτο επίπεδο, που είναι και η βάση του οχήματος είναι τοποθετημένη η βάση της κάμερα και στην κάτω πλευρά είναι οι δύο σερβοκινητήρες, οι δύο ρόδες που είναι συνδεδεμένες με τους σερβοκινητήρες καθώς και μια τρίτη βοηθητική στο πίσω μέρος, στο κάτω μέρος του επιπέδου αυτού είναι τοποθετημένες οι δύο μπαταρίες λιθίου που τροφοδοτούν τις σερβομηχανές και τον υπολογιστή μονής πλακέτας (SBC). Στο δεύτερο επίπεδο υπάρχει ειδική εσοχή διαμορφωμένη στην οποία έχει τοποθετηθεί ο ρυθμιστής τάσης και στο πάνω μέρος της εσοχής βρίσκεται ένα βολτόμετρο και ο υπολογιστής μονής πλακέτας.



Εικόνα 16. Σασί από το πλάι

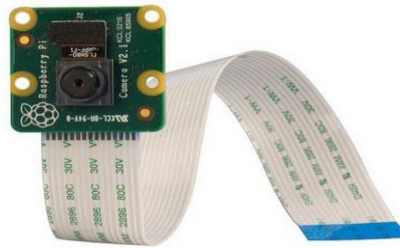
2.4 Κάμερα για υπολογιστή μονής πλακέτας:

Η κάμερα που χρησιμοποιήθηκε για την κατασκευή αυτού του οχήματος είναι η Raspberry Pi Camera Module V2 και είναι συμβατή με τον υπολογιστή μονής πλακέτας του οχήματος. Η Raspberry Pi Camera Module V2 αντικατέστησε την αρχική κάμερα (V1 - έτος έκδοσης 2013), τον Απρίλιο του 2016. Η κάμερα V2 διαθέτει αισθητήρα Sony IMX219 των 8 megapixel (σε σύγκριση με τον αισθητήρα OmniVision OV564T των 5 megapixel της αρχικής κάμερας). Η συγκεκριμένη κάμερα μπορεί να χρησιμοποιηθεί για την λήψη βίντεο υψηλής ευκρινείας καθώς και φωτογραφιών. Είναι αρκετά εύχρηστη με αποτέλεσμα να την χειρίζονται με άνεση και αρχάριοι, αλλά και οι προχωρημένοι χρήστες να επεκτείνουν τις γνώσεις τους. Υπάρχουν πολλά παραδείγματα χρήσης της κάμερας αυτής σε τεχνικές όπως time-lapse και slow-motion. Επίσης οι βιβλιοθήκες της μπορούν να συνδυαστούν με την κάμερα ώστε να δημιουργούν διάφορα εφέ. Ο αισθητήρας Sony IMX219 αποτελεί άλμα αναβάθμισης σε ποιότητα εικόνας, χρωμάτων και εκτέλεσης χαμηλού φωτισμού. Υποστηρίζει λειτουργίες βίντεο 1080p30, 720p60, VGA90 και ακίνητη λήψη. Η σύνδεσή της γίνεται μέσω καλωδίου κορδέλας 15 cm στη θύρα CSI του Raspberry Pi. Η κάμερα λειτουργεί με όλα τα μοντέλα Raspberry Pi 1,2,3 και 4, μόνο που σε όλα Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

τα μοντέλα του Raspberry Pi Zero απαιτείτε καλώδιο κάμερας Raspberry Pi Zero, διότι το τυπικό καλώδιο που παρέχεται με την κάμερα δεν είναι συμβατό με την υποδοχή της κάμερα στο Raspberry Pi Zero.

	Camera Module v1	Camera Module v2	HQ Camera
Net price	\$25	\$25	\$50
Size	Around 25 × 24 × 9 mm		38 × 38 × 18.4mm (excluding lens)
Weight	3g	3g	
Still resolution	5 Megapixels	8 Megapixels	12.3 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90	1080p30, 720p60 and 640 × 480p60/90	1080p30, 720p60 and 640 × 480p60/90
Linux integration	V4L2 driver available	V4L2 driver available	V4L2 driver available
C programming API	OpenMAX IL and others available	OpenMAX IL and others available	
Sensor	OmniVision OV5647	Sony IMX219	Sony IMX477
Sensor resolution	2592 × 1944 pixels	3280 × 2464 pixels	4056 × 3040 pixels
Sensor image area	3.76 × 2.74 mm	3.68 × 2.76 mm (4.6 mm diagonal)	6.287mm × 4.712 mm (7.9mm diagonal)
Pixel size	1.4 μm × 1.4 μm	1.12 μm × 1.12 μm	1.55 μm × 1.55 μm

Εικόνα 17. Σύγκριση Εκδόσεων Κάμερας SBC (Raspberry Pi) (ΒΑΣΙΛΟΠΟΥΛΟΣ ΝΙΚΟΛΑΟΣ, n.d.)



Εικόνα 18. Raspberry Pi Camera V2 (Anon., n.d.)

2.5 Μπαταρίες λιθίου συνδεδεμένες με ρυθμιστή τάσης και βολτόμετρο.

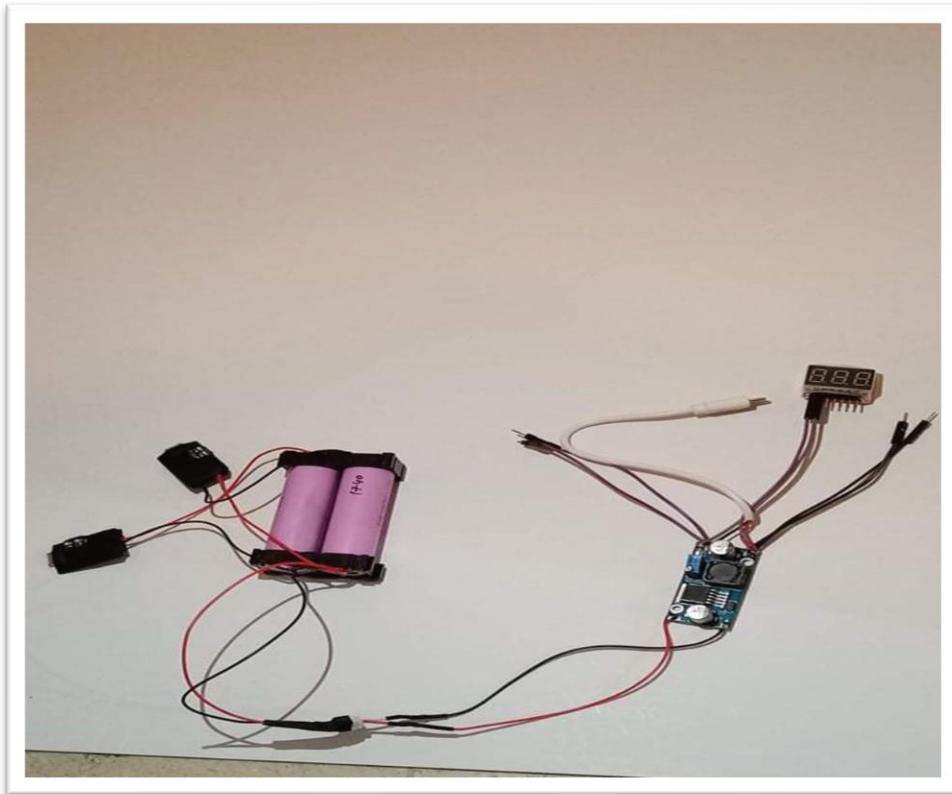
Βολτόμετρο : Το βολτόμετρο είναι ένα όργανο που χρησιμοποιείται για να μετρήσει τη διαφορά δυναμικού (ηλεκτρικής τάσης) μεταξύ δύο σημείων ενός ηλεκτρικού κυκλώματος. ("Βολτόμετρο - Βικιπαίδεια") Το βολτόμετρο λειτουργεί βάση είτε μαγνητικών είτε θερμικών Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

αποτελεσμάτων του ηλεκτρικού ρεύματος. Τα βολτόμετρα συνδέονται πάντα παράλληλα σε ένα κύκλωμα και για τον λόγο αυτό έχουν μεγάλη εσωτερική αντίσταση σε σχέση με τα άλλα στοιχεία του κυκλώματος ώστε να μην επηρεάζουν τα χαρακτηριστικά του. (“Βολτόμετρο - Wikiwand”) Το βολτόμετρο που χρησιμοποιήθηκε στην συγκεκριμένη πτυχιακή είναι ψηφιακό και είναι συνδεδεμένο με έναν ρυθμιστής τάσης.

Ρυθμιστής τάσης : Ο ρυθμιστής τάσης ή μετατροπέας τάσης είναι το εργαλείο το οποίο παρέχει ελεγχόμενη συνεχή τάση από μία πηγή που έχει ορισμένη τιμή συνεχούς τάσης, στην προκειμένη περίπτωση οι μπαταρίες λιθίου. Ο ρυθμιστής τάσης DC-DC⁵ έχει αρκετά πλεονεκτήματα όπως ότι έχει χαμηλό κόστος, υψηλό βαθμό απόδοσης, ταχύτατη δυναμική απόκριση και μικρό μέγεθος. Ο ρυθμιστής τάσης περιέχει τουλάχιστον δύο ημιαγωγούς, μία δίοδο και ένα τρανζίστορ, συχνά βέβαια οι πιο πρόσφατοι ρυθμιστές έχουν αντικαταστήσει τη δίοδο με ένα δεύτερο τρανζίστορ. Ακόμα περιέχει έναν πυκνωτή ή έναν επαγωγέα ή και τα δύο μαζί σε συνδυασμό. Στην έξοδο ενός ρυθμιστή προστίθενται φίλτρα κατασκευασμένα από πυκνωτές ώστε να μειώνεται ο κυματισμός τάσης, ένα τέτοιο φίλτρο υπάρχει και στην είσοδο από την πλευρά της τροφοδοσίας. Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε ο ρυθμιστής LM2596 DC-DC Buck Converter 4.5-40V to 3-35V Adjustable Step Down Module με τάση εισόδου 3,2V ~ 40V, τάση εξόδου 1,25V ~ 35V (ρυθμιζόμενο), ρεύμα εξόδου ≤ 3 A, αποδοτικότητα μετατροπής $\leq 92\%$, έξοδος κυματισμού < 30 mV, συχνότητα εναλλαγής 65 KHz, θερμοκρασία λειτουργίας $-45^{\circ}\text{C} \sim +85^{\circ}\text{C}$ και τέλος το μέγεθός του είναι 43,2 mm x 21,0 mm x 14,0 mm (Μήκος x Πλάτος x Ύψος. (Industrial Quick Search Directory , n.d.)

Μπαταρίες : Οι δύο μπαταρίες που χρησιμοποιήθηκαν είναι επαναφορτιζόμενες, τυπικής χωρητικότητας 3000 mAh και ελάχιστης τυπικής χωρητικότητας 2900 mAh , μέγιστο ρεύμα εκφόρτωσης 5c 15 A και μέγιστη τάση φόρτισης 4.2 V, με διάμετρο 18,33 mm, ύψος 64,85 mm και βάρος 48 g.

⁵ DC-DC είναι ο ρυθμιστής τάσης Συνεχούς ρεύματος – Συνεχούς ρεύματος.



Εικόνα 19. Μπαταρίες λιθίου συνδεδεμένες με ρυθμιστή τάσης και βολτόμετρο

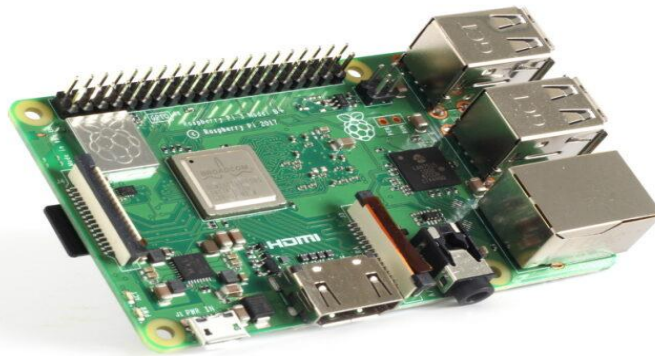
2.6 Υπολογιστής μονής πλακέτας

Ο υπολογιστής μονής πλακέτας (Single Board Computer ή SBC) είναι ένας υπολογιστής κατασκευασμένος σε μία πλακέτα κυκλώματος, με έναν ή πολλούς μικροεπεξεργαστές, μνήμη, είσοδο / έξοδο και τα υπόλοιπα χαρακτηριστικά που απαιτούνται ώστε να είναι ένας υπολογιστής λειτουργικός. Τέτοιου είδους υπολογιστές συνήθως κατασκευάζονται ως συστήματα επίδειξης ή ανάπτυξης, για εκπαιδευτικά συστήματα ή ως ενσωματωμένοι ελεγκτές υπολογιστών. Σε αντίθεση με τους επιτραπέζιους προσωπικούς υπολογιστές, ο υπολογιστής μονής πλακέτας συχνά δεν βασίζονται σε υποδοχές επέκτασης είτε για επέκταση είτε για περιφερειακές λειτουργίες. Χρησιμοποιείται ένα ευρύ φάσμα μικροεπεξεργαστών για την κατασκευή του υπολογιστή μονής πλακέτας. Τα απλά σχέδια συχνά έχουν στατική μνήμη RAM και χαμηλού κόστους επεξεργαστές 32 ή 64 bit όπως ο ARM. Στην εν λόγω εργασία χρησιμοποιείται ο υπολογιστής μονής πλακέτας Raspberry Pi 3 B+

2.6.1 Γενικά χαρακτηριστικά του SBC που χρησιμοποιήθηκε :

Όπως προαναφέρθηκε για την υλοποίηση του έργου χρησιμοποιήθηκε ο υπολογιστής μονής πλακέτας Raspberry Pi 3B+ που κυκλοφόρησε το 2018 για να αντικαταστήσει το Raspberry Pi 3, διαθέτει μία σειρά από βελτιώσεις σε σχέση με το προηγούμενο μοντέλο. Μέσα σε αυτές τις βελτιώσεις συμπεριλαμβάνεται η ταχύτερη CPU και ένα ταχύτερο Gigabit Ethernet (περιορίζεται στα 300 Mbit/s από την εσωτερική σύνδεση USB 2.0), η καρδιά του Pi 3B+ είναι το

BCM2837B0 που τρέχει με ταχύτητα ρολογιού 1,4 GHz, δηλαδή έναν γρήγορο και αποδοτικό επεξεργαστή. Αναλυτικότερα τα χαρακτηριστικά του συγκεκριμένου μοντέλου είναι :



Εικόνα 20. Υπολογιστής – Rpi 3 Model B+ (Anon., n.d.)

Επεξεργαστής : BCM237B0 είναι το συστατικό της συγκεκριμένης πλακέτας που βοηθάει στην εκτέλεση των συνόλων των οδηγιών που βασίζονται σε μαθηματικούς και λογικούς τύπους. Είναι τετραπύρηνος επεξεργαστής ARM CORTEX A53 ΣΤΑ 1,4 GHz 64 bit όπως έχει ειπωθεί και πιο πάνω.

RAM: Η μνήμη RAM που χρησιμοποιείται είναι 1 GB LPDDR2 SDRAM που είναι παρόμοια με την προηγούμενη έκδοση.

GPU: Αντιπροσωπεύει μονάδα επεξεργασίας γραφικών και χρησιμοποιείται για την εκτέλεση του υπολογισμού εικόνας. Η συγκεκριμένη GPU χρησιμοποιεί OpenGL ES έκδοση 2.0, API OpenVG με επιτάχυνση υλικού και αποκωδικοποίηση υψηλού προφίλ 1080p30 H.264. Μπορεί να παρέχει έως και 1 Gpixel/s, 1,5 Gtexels/s ή 24 GFLOP σε έναν υπολογιστή γενικής χρήσης.

Θύρες USB : Το μοντέλο αυτό αποτελείται από 4 θύρες USB όπως και στο προηγούμενο μοντέλο.

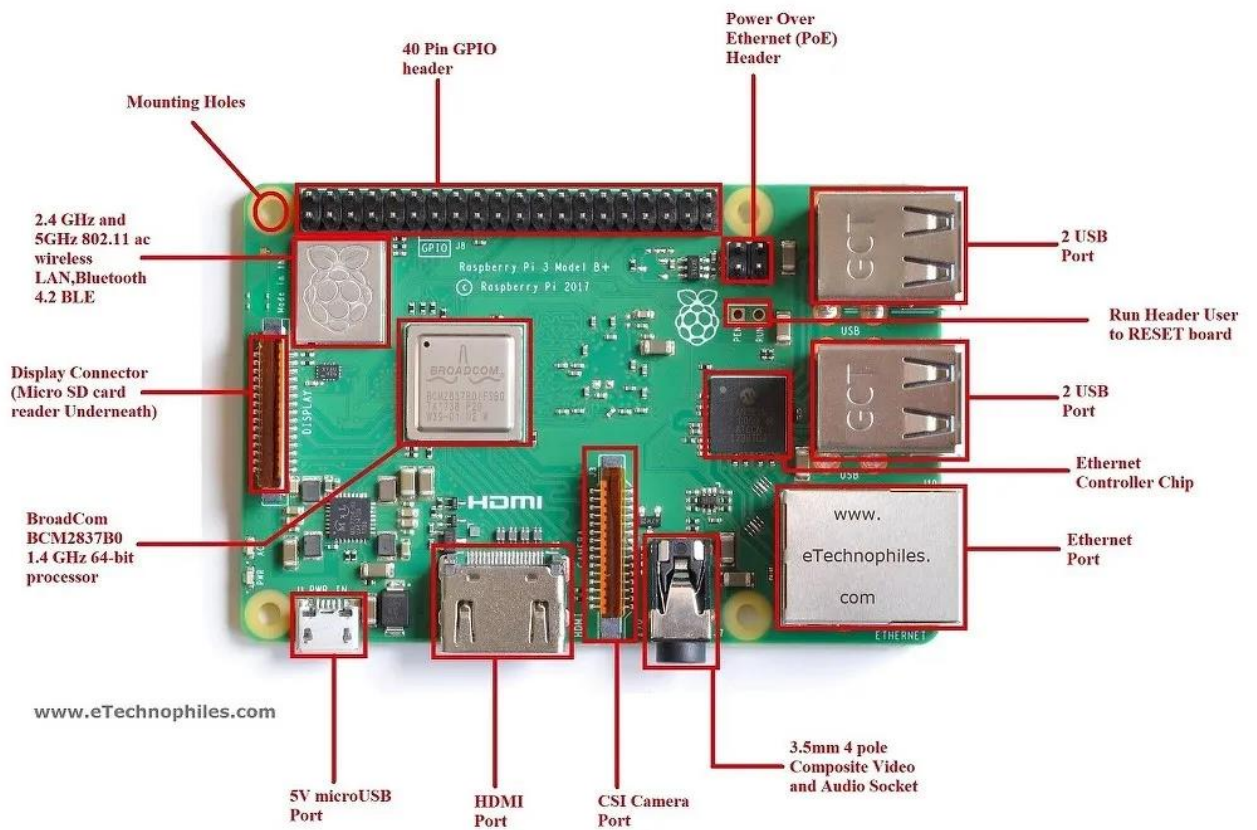
Υποδοχή πηγής τροφοδοσίας Micro USB : Η υποδοχή αυτή χρησιμοποιείται για την παροχή ρεύματος 5V στην πλακέτα και καταναλώνει περίπου 170 έως 200 mA περισσότερη ισχύς από το προηγούμενο μοντέλο. Η υποδοχή της τροφοδοσίας τοποθετείται δίπλα στην υποδοχή HDMI.

Σύνδεση HDMI και μετατροπή (Composite) : Τόσο η υποδοχή εξόδου ήχου, όσο και η υποδοχή μετατροπέα βίντεο (video composite) βρίσκονται σε μία ενιαία τετραπολική υποδοχή 3,5 mm, η οποία βρίσκεται κοντά στην θύρα HDMI, και τώρα όλη η υποδοχή τροφοδοσίας και μετατροπής ήχου-βίντεο τοποθετούνται στη μία πλευρά του πίνακα.

Σκληρός δίσκος USB : Η πλακέτα έχει την δυνατότητα να χρησιμοποιήσει έναν εξωτερικό σκληρό δίσκο USB.

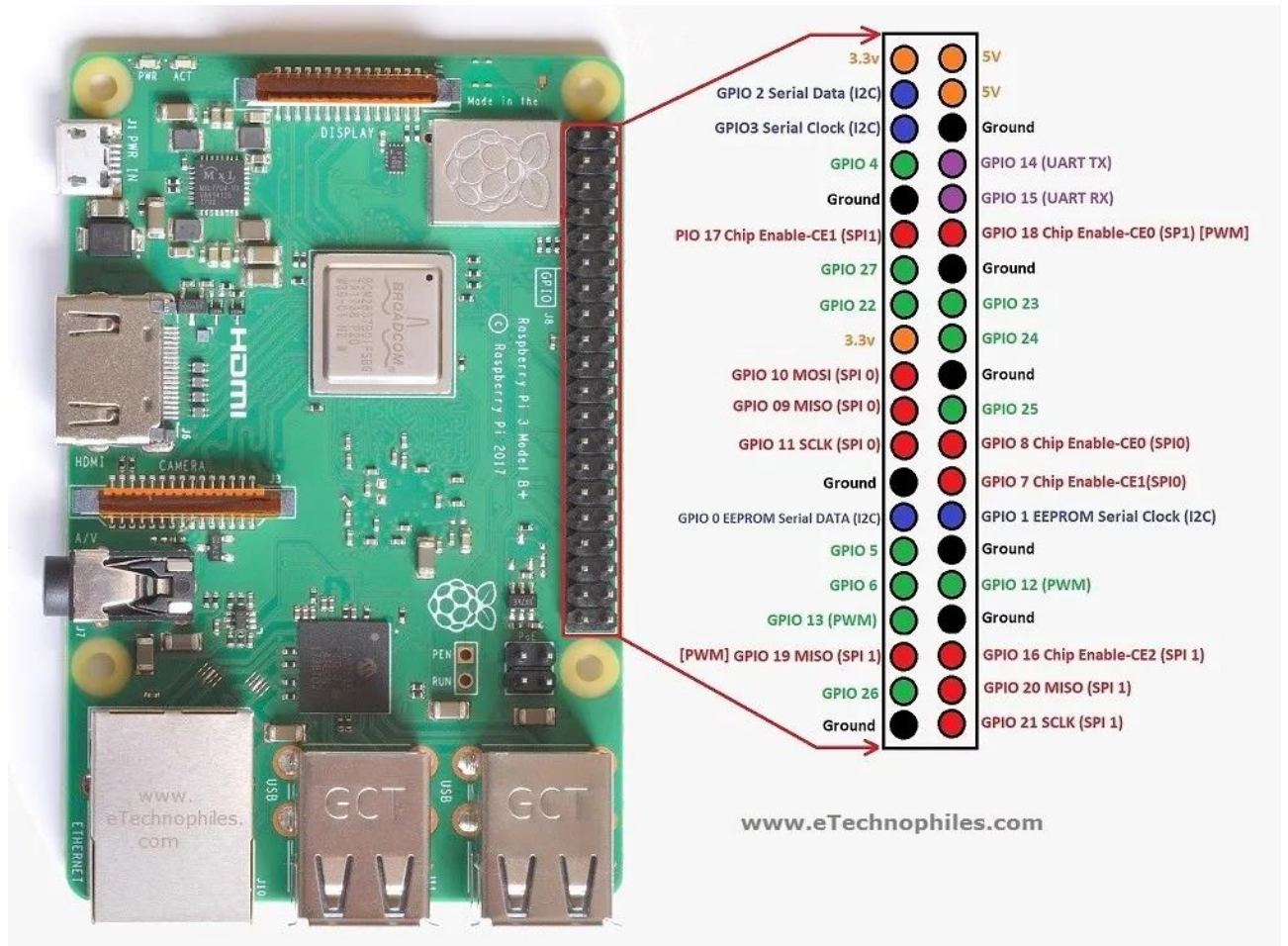
Power over Ethernet: Η δυνατότητα αυτή επιτρέπει την τροφοδοσία της πλακέτας χρησιμοποιώντας τα καλώδια ethernet.

Τέλος έχει αντικατασταθεί η υποδοχή μνήμης SD από μία υποδοχή κάρτας μνήμης micro SD.



Εικόνα 21 Τα μέρη του Raspberry Pi 3b+ (eTechnophiles , n.d.)

Τα Pinout του συγκεκριμένου μοντέλου με λειτουργία GPIO δίνονται παρακάτω :



Εικόνα 22 Τα Pinout του Raspberry Pi 3b+ (eTechnophiles , n.d.)

Το GPIO αντιπροσωπεύει τους ακροδέκτες εισόδου γενικής χρήσης, αυτοί οι ακροδέκτες χρησιμοποιούνται για τη σύνδεση της συγκεκριμένης πλακέτας με εξωτερικές συσκευές εισόδου/εξόδου. Το μοντέλο αυτό αποτελείται από ένα GPIO 40 ακροδεκτών. Μία τυπική διεπαφή για να συνδέσει μία SBC ή έναν μικροεπεξεργαστή με άλλες συσκευές χρησιμοποιεί τους ακροδέκτες γενικής χρήσης (GPIO) εισόδου/εξόδου. Τέλος εφόσον αυτοί οι ακροδέκτες δεν έχουν συγκεκριμένη λειτουργία, μπορούν να προσαρμοστούν σύμφωνα με το λογισμικό που θα χρησιμοποιηθεί.

Η συγκεκριμένη πλακέτα αποτελείται από δύο ακροδέκτες 5V, δύο ακροδέκτες 3.3V και 9 ακροδέκτες γείωσης (0V) οι οποίοι δεν μπορούν να παραμετροποιηθούν.

5V : Οι ακροδέκτες των 5V χρησιμοποιούνται για την απευθείας παροχής της τροφοδοσίας 5V που προέρχεται από τον προσαρμογέα ηλεκτρικού δικτύου. Αυτός ο ακροδέκτης μπορεί να χρησιμοποιηθεί και για την τροφοδοσία του Raspberry Pi, καθώς και για την τροφοδοσία άλλων συσκευών 5V.

3,3V : Ο ακροδέκτης 3V είναι αυτός που χρησιμοποιείται για την παροχή σταθερής τροφοδοσίας 3,3V σε εξωτερικά εξαρτήματα και για την δοκιμή LED.

GND: Η γείωση αναφέρεται συνήθως ως GND και είναι ζωτικής σημασίας για κάθε ηλεκτρονικό έργο.

Ακροδέκτες εισόδου / εξόδου : Ένας ακροδέκτης GPIO που έχει οριστεί ως είσοδος διαβάζει το σήμα που έχει ληφθεί από το Raspberry Pi και έχει σταλεί από τη συσκευή που είναι συνδεδεμένη σε αυτόν τον ακροδέκτη. Οποιαδήποτε τάση μεταξύ των 1,8 V και 3,3 V διαβάζεται ως Υψηλή τάση και η τάση χαμηλότερη από 1,8 V ως Χαμηλή από το Raspberry Pi. Ένας Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

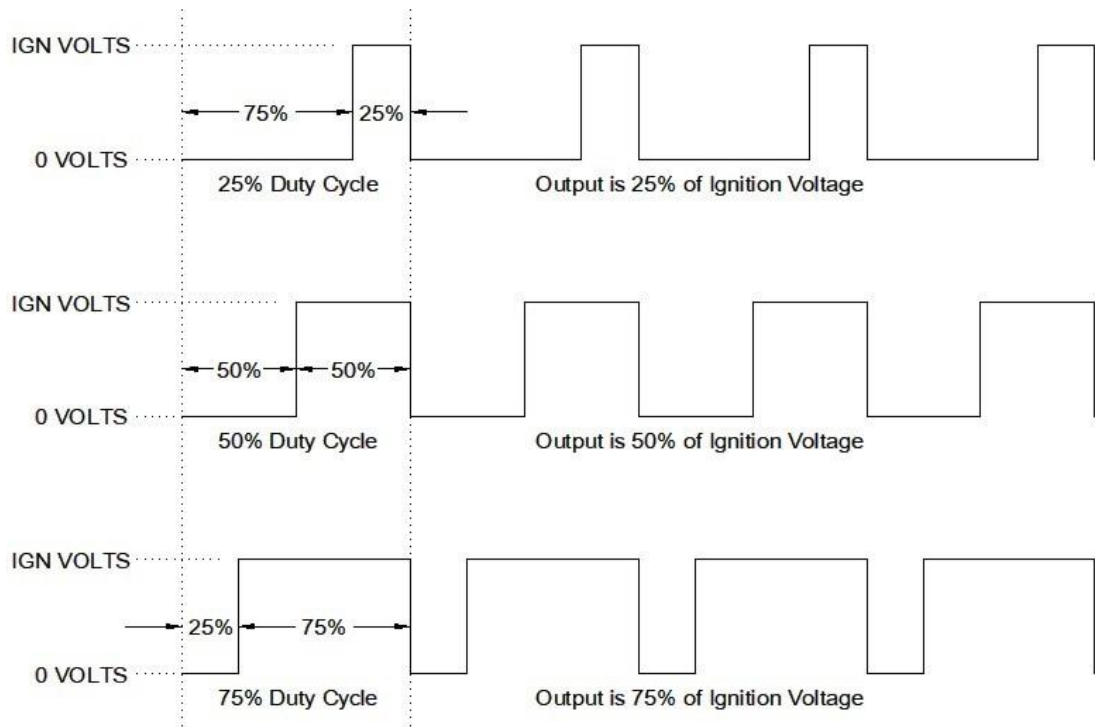
ακροδέκτης GPIO που έχει οριστεί ως ακροδέκτης εξόδου στέλνει το σήμα τάσης ως υψηλό σήμα στα 3,3 V και ως χαμηλό σήμα στα 0 V. Όταν ο ακροδέκτης έχει ρυθμιστεί σε Υψηλό, η τάση στην έξοδο είναι 3,3 V και όταν ρυθμιστεί σε Χαμηλό, η τάση στην έξοδο είναι 0 V.

Μαζί με την απλή λειτουργία των ακροδεκτών εισόδου / εξόδου, οι ακροδέκτες GPIO μπορούν να εκτελέσουν και κάποιες άλλες λειτουργίες, μία από αυτές τις λειτουργίες είναι η διαμόρφωση πλάτους παλμών πιο γνωστή ως PWM (Pulse Width Modulation).

Τι είναι η διαμόρφωση πλάτους παλμών PWM : Η διαμόρφωση πλάτους παλμών είναι μία τεχνική για τον έλεγχο αναλογικών κυκλωμάτων με τις ψηφιακές εξόδους ενός επεξεργαστή και χρησιμοποιείται σε ευρύ φάσμα εφαρμογών που έχουν να κάνουν με την μέτρηση και τις επικοινωνίες μέχρι τον έλεγχο και την μετατροπή ισχύος. Στην πιο βασική περίπτωση, το PWM χρησιμοποιείται για την προσομοίωση ενός αναλογικού σήματος χρησιμοποιώντας μία ψηφιακή πηγή, στην προκειμένη περίπτωση τον ακροδέκτη GPIO. Τα ψηφιακά σήματα έχουν δύο θέσεις, είτε ενεργοποίηση είτε απενεργοποίηση που ερμηνεύεται σε 1 και 0. Από την άλλη πλευρά το αναλογικό σήμα μπορεί να είναι ενεργοποιημένο σε άπειρο αριθμό θέσεων μεταξύ του 0 και του 1, είτε πλησιάζει το 1 είτε κατεβαίνει στο μηδέν. (Barr, n.d.) (Torrens, n.d.)

Το PWM είναι μία μέθοδος μείωσης της μέσης ισχύος που παρέχεται από ένα ηλεκτρικό σήμα χωρίζοντάς το σε διακριτά μέρη. Στην ουσία είναι η προσπάθεια αλλαγής του πλάτους του παλμού που έχουμε χωρίς όμως να αλλαχτεί η συχνότητα. Παλμός είναι ένα ηλεκτρικό σήμα όπου η παρουσία του δεν είναι απεριόριστα μεγάλη. Ένας παλμός έχει αρχή, μέση και τέλος και ο χρόνος από την αρχή μέχρι το τέλος μπορεί να μετρηθεί. Οι παλμοί στους οποίους αναφέρεται η συγκεκριμένη εργασία είναι πολύ μικρότεροι, ένας παλμός έχει ένα πεπερασμένο μήκος στο χρόνο και τότε είναι ενεργοποιημένο (ON), βρίσκεται σε αυτή την κατάσταση όσο υπάρχει ηλεκτρικό ρεύμα. Όταν αφαιρεθεί το ηλεκτρικό σήμα περνάει στην κατάσταση του απενεργοποιημένου (OFF). Εάν το ηλεκτρικό ρεύμα αλλάξει από ενεργοποιημένο (ON) σε απενεργοποιημένο (OFF) με συνεχή και ίσα διαστήματα μεταξύ των δύο καταστάσεων, τότε ο παλμός λέγεται ότι είναι περιοδικός και η περίοδος του είναι ίση με το χρονικό διάστημα που ήταν ενεργοποιημένο (ON) συν το ποσό όταν δεν υπήρχε ηλεκτρικό ρεύμα (OFF). (Heath, n.d.) Όσο περισσότερο βρίσκεται στην ενεργοποιημένη κατάσταση σε σύγκριση με τις περιόδους απενεργοποίησης, τόσο μεγαλύτερη είναι η συνολική ισχύς που παρέχεται στο φορτίο. Για το PWM είναι σημαντικό η περίοδος να παραμένει σταθερή. Ακόμα ένα βασικό συστατικό του PWM είναι η συχνότητα. Συχνότητα είναι πόσες φορές ο παλμός άνοιξε και έσβησε κάθε δευτερόλεπτο και μπορεί να υπολογιστεί παίρνοντας το αντίστροφο της περιόδου. Ένας παλμός δημιουργείται σε συνεχή βάση, αν όμως διαφοροποιηθεί ο χρόνος που το ηλεκτρικό σήμα είναι ενεργοποιημένο και διατηρώντας σταθερή την περίοδο, θα επηρεαστεί ο χρόνος απενεργοποίησης και μέσω αυτής της διαδικασίας υπολογίζεται ο κύκλος λειτουργίας. Αναλυτικότερα :

Κύκλος λειτουργίας (Duty Cycle) : ο όρος αυτός περιγράφει την αναλογία του χρόνου ενεργοποίησης προς το κανονικό διάστημα ή περίοδο του χρόνου. Όταν η τροφοδοσία είναι απενεργοποιημένη για τις περισσότερες φορές, ο χαμηλός κύκλος αντιστοιχεί σε χαμηλή ισχύ. Ο κύκλος λειτουργίας έχει ποσοστιαία έκφραση και το 100% είναι το πλήρως ενεργοποιημένο. Σε περίπτωση που το ψηφιακό σήμα είναι τις μισές φορές ενεργοποιημένο και τις άλλες μισές απενεργοποιημένο, τότε το ψηφιακό σήμα έχει κύκλο λειτουργίας 50% και μοιάζει με κύμα «τετράγωνο». Σε περίπτωση το ψηφιακό σήμα διαθέτει περισσότερο χρόνο στην ενεργοποίηση από ό,τι στην κατάσταση απενεργοποίησης, έχει κύκλο λειτουργίας >50%. Σε αντίστροφη περίπτωση, δηλαδή το ψηφιακό σήμα διαθέτει περισσότερο χρόνο στην κατάσταση απενεργοποίησης από ό,τι στην ενεργοποίησης, έχει κύκλο λειτουργίας <50%. (avayan, n.d.) (en.wikipedia.org, 2009)



Εικόνα 23 PWM και κύκλος λειτουργίας. (Stanislav Smith, Technical Transmission General Articles , 2022).

Είναι σύνηθες οι σύγχρονοι μικροελεγκτές να έχουν ενσωματωμένους ελεγκτές PWM και όπως έχει ειπωθεί στην περιγραφή της συγκεκριμένης πλακέτας που χρησιμοποιήθηκε στην παρούσα εργασία, εκτίθενται σε εξωτερικούς ακροδέκτες ως περιφερειακές συσκευές υπό τον έλεγχο υλικολογισμικού μέσω εσωτερικών διεπαφών προγραμματισμού. Συνήθως χρησιμοποιούνται για τον έλεγχο κινητήρων συνεχούς ρεύματος στην ρομποτική, στην παρούσα εργασία χρησιμοποιείται για τον έλεγχο σερβοκινητήρων.

3 Λειτουργικό σύστημα

Το λειτουργικό σύστημα που χρησιμοποιείται στα Raspberry Pi είναι το Raspberry Pi OS (πρώην Raspbian). Είναι ένα λειτουργικό σύστημα που βασίζεται στο Debian (ελεύθερο λογισμικό ανοιχτού κώδικα, που αναπτύχθηκε υπό την υποστήριξη της κοινότητας Debian Project και θεμελιώθηκε από τον Ian Murdock στις 16 Αυγούστου το 1993). Από το 2013 παρέχεται επίσημα από το ίδρυμα Raspberry Pi ως το κύριο λειτουργικό σύστημα για τους συμπαγείς υπολογιστές μιας πλακέτας. Για πρώτη φορά το Raspberry Pi OS αναπτύχθηκε από τους Mike Thompson και Peter Green ως Raspbian, μια ανεξάρτητα και ανεπίσημη σύνδεση του Debian με το Raspberry Pi.

Η πρώτη έκδοση κυκλοφόρησε στις 15 Ιουλίου του 2012. Εφόσον το Raspberry Pi εκείνη την εποχή δεν είχε κάποιο επίσημο λειτουργικό σύστημα, το Ίδρυμα Raspberry Pi αποφάσισε να βασιστεί στην δουλειά που είναι το Raspbian και άρχισε να παράγει και να κυκλοφορεί τη δική του έκδοση λογισμικού. Η πρώτη κυκλοφορία του Raspbian από το Ίδρυμα ανακοινώθηκε στις 10 Σεπτεμβρίου του 2013. Στις 28 Μαΐου του 2020 το Ίδρυμα Raspberry Pi κυκλοφόρησε μια beta έκδοση 64-bit του επίσημου λειτουργικού τους συστήματος, ωστόσο αυτή η έκδοση δεν βασίστηκε στο Raspbian. Από την στιγμή που το Ίδρυμα Raspberry Pi δεν ήθελε να χρησιμοποιήσει το όνομα Raspbian για να αναφέρεται σε λογισμικό που δεν βασίζεται στο έργο του Raspbian, άλλαξε το όνομα του παρεχόμενου λειτουργικού συστήματος σε Raspberry Pi OS. Η αλλαγή αυτή έγινε και στην 32-bit έκδοση, αν και συνεχίζει να βασίζεται στο Raspbian. Η 64-bit έκδοση του Raspberry Pi OS επίσημα κυκλοφόρησε στις 2 Φεβρουαρίου του 2022. Το καινούριο λογισμικό αξιοποιεί με τον καλύτερο τρόπο τα Raspberry Pi με επεξεργαστή ARM και λειτουργεί σε κάθε πλακέτα εκτός από την Pico. (Wikipedia, n.d.) (Piltch, n.d.)



Εικόνα 24. Λογισμικό Raspberry Pi OS. (Piltch, n.d.)

3.1 Γλώσσα προγραμματισμού

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε σε αυτό το εγχείρημα είναι Python. Η python είναι μία γλώσσα προγραμματισμού υπολογιστών που χρησιμοποιείται συχνά για την κατασκευή ιστοσελίδων και λογισμικού, την αυτοματοποίηση εργασιών και τη διεξαγωγή ανάλυσης δεδομένων. Η python είναι μία γλώσσα γενικής χρήσης και υψηλού επιπέδου, που σχεδιάσμος και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

σημαίνει ότι μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας ποικιλίας διαφορετικών προγραμμάτων και δεν είναι εξειδικευμένη για συγκεκριμένα προβλήματα. Είναι δυναμική γλώσσα προγραμματισμού και υποστηρίζει συλλογή απορριμμάτων, ανήκει στις γλώσσες προστακτικού προγραμματισμού και υποστηρίζει και το διαδικαστικό προγραμματιστικό υπόδειγμα και το αντικειμενοστρεφές. Η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation και είναι ανοιχτού λογισμικού γλώσσα. Το όνομα της προέρχεται από την ομάδα των Άγγλων κωμικών Monty Python. Σαν γλώσσα προγραμματισμού είναι εύκολη στην χρήση της και το συντακτικό της δίνει την δυνατότητα στους προγραμματιστές να εκφράζουν έννοιες σε λιγότερες γραμμές κώδικα. Έχει πολλές βιβλιοθήκες με αποτέλεσμα να πραγματοποιούνται ευκολότερα συνηθισμένες εργασίες και είναι αρκετά εύκολη η εκμάθησή της. Πολλά λειτουργικά συστήματα έχουν την δυνατότητα εγκατάστασης των διερμηνέων (interpreter) της Python με αποτέλεσμα να εκτελεί κώδικα σε μία ευρεία γκάμα συστημάτων, επίσης μερικά από τα πιο δημοφιλή λειτουργικά συστήματα εισάγουν τον κώδικα της Python σε αυτόνομα εκτελέσιμα προγράμματα. Αυτό επιτρέπει τη διανομή για χρήση του λογισμικού βασισμένο σε Python χωρίς να είναι απαραίτητη η εγκατάσταση του διερμηνέα της Python. Βασικό της μειονέκτημα αποτελεί το γεγονός ότι σαν γλώσσα δεν μπορεί να χρησιμοποιηθεί για την γραφή λειτουργικών συστημάτων, καθώς είναι γλώσσα που χρησιμοποιεί διερμηνέα, πράγμα που την κάνει πιο αργή σε σχέση με τις γλώσσες που χρησιμοποιούν μεταγλωττιστές όπως η C και C++.



Εικόνα 25. Λογότυπο της Python. (Anon., n.d.)

Η ιστορία της Python: Η Python επινοήθηκε στα τέλη της δεκαετίας του 1980 από τον Guido van Rossum Centrum Wiskunde & Informatica (CWI) στην Ολλανδία για να διαδεχτεί την γλώσσα προγραμματισμού ABC και στην αρχή ήταν γλώσσα σεναρίων που χρησιμοποιήθηκε στο λειτουργικό σύστημα Amoeba. Η εφαρμογή της ξεκίνησε τον Δεκέμβριο του 1989 και ο Van Rossum ανέλαβε την αποκλειστική ευθύνη για το έργο, ως κύριος προγραμματιστής μέχρι τις 12 Ιουλίου του 2018 οπότε σταμάτησε. Τον Ιανουάριο του 2019, οι προγραμματιστές του πυρήνα της Python εξέλεξαν ένα πενταμελές Διοικητικό Συμβούλιο για να ηγηθεί του έργου. (“Python (γλώσσα προγραμματισμού)”) Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου του 2000, με πολλά και σημαντικά νέα χαρακτηριστικά, στις 3 Δεκεμβρίου του 2008 κυκλοφόρησε η Python 3.0. Το 2021 τα Python 3.9.2 και 3.8.8 επιταχύνθηκαν. Στις 7 Σεπτεμβρίου του 2022 πραγματοποιήθηκαν τέσσερις νέες εκδόσεις οι 3.10.7, 3.9.14, 3.8.14 και 3.7.14. (Wikipedia, n.d.)

3.2 Βιβλιοθήκες

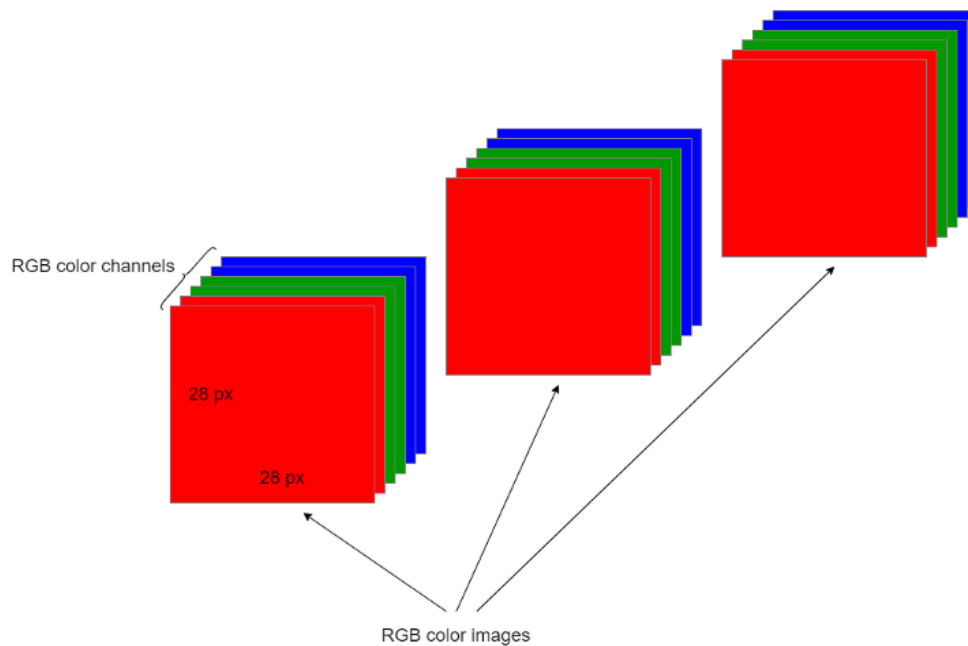
3.2.1 OpenCV :

Η OpenCv (Open Source Computer Vision Library) είναι μία βιβλιοθήκη λογισμικού ανοιχτού υπολογιστικής όρασης, όπως λέει και το όνομά της, και μηχανικής μάθησης. Η συγκεκριμένη βιβλιοθήκη δημιουργήθηκε για να παρέχει μία κοινή υποδομή για εφαρμογές υπολογιστικής όρασης και να επιταχύνει τη χρήση της μηχανικής αντίληψης στα εμπορικά προϊόντα. Η βιβλιοθήκη αυτή περιέχει πάνω από 2500 βελτιστοποιημένους αλγόριθμους αποτελούμενους από ένα σύνολο κλασικών και υπερσύγχρονων αλγορίθμων υπολογιστικής όρασης και μηχανικής μάθησης.

Οι αλγόριθμοι αυτοί μπορούν να χρησιμοποιηθούν για την ανίχνευση και αναγνώριση προσώπων, αναγνώριση αντικειμένων, ταξινόμηση ανθρώπινων ενεργειών σε βίντεο, την παρακολούθηση κινήσεων της κάμερας, την παρακολούθηση κινουμένων αντικειμένων, την εξαγωγή τρισδιάστατων μοντέλων αντικειμένων, την παραγωγή τρισδιάστατων σημείων από στερεοφωνικές κάμερες, τη συρραφή εικόνων για την παραγωγή υψηλής ανάλυσης εικόνας μίας ολόκληρης σκηνής, την εύρεση παρόμοιων εικόνων από βάση δεδομένων εικόνων, την αφαίρεση κόκκινων ματιών από εικόνες που τραβήχτηκαν με φλας. Η βιβλιοθήκη αυτή χρησιμοποιείται εκτενώς σε εταιρείες, ερευνητικές ομάδες και από κυβερνητικούς φορείς. Διαθέτει διεπαφές C++, Python, Java και MATLAB και υποστηρίζει Windows, Linux, Android και Mac OS. Η OpenCv κλίνει κυρίως σε εφαρμογές όρασης σε πραγματικό χρόνο, είναι γραμμένη εγγενώς σε C++ και έχει ένα υπόδειγμα διεπαφής που λειτουργεί άψογα με STL (Standard Template Library) κοντέινερ. (OpenCV team , 2022)

3.2.2 NumPy

Το NumPy είναι μία βιβλιοθήκη για την γλώσσα προγραμματισμού Python, όπου υποστηρίζει απλούς και πολυδιάστατους πίνακες. Περιλαμβάνει μια μεγάλη συλλογή μαθηματικών συναρτήσεων υψηλού επιπέδου για λειτουργία σε των πινάκων, όπως ρουτίνες για γρήγορες λειτουργίες σε πίνακες συμπεριλαμβανομένης της μαθηματικής λογικής, χειρισμούς σχημάτων, ταξινόμησης, επιλογής εισόδου/εξόδου, διακριτούς μετασχηματισμούς Fourier, βασική γραμμική άλγεβρα, βασικές στατιστικές πράξεις και πάρα πολλά άλλα. Στην ουσία είναι το βασικό πακέτο για επιστημονικούς υπολογισμούς στην Python.



Εικόνα 26.Εικόνα με RGB χρώμα με την χρήση πίνακα στην βιβλιοθήκη NumPy (Pramoditha, 2021).

Ιστορικά στοιχεία για την NumPy

Αρχικά η γλώσσα προγραμματισμού Python δεν σχεδιάστηκε για αριθμητικούς υπολογισμούς, παρ'όλα αυτά η επιστημονική και η μηχανική κοινότητα έδειξε ενδιαφέρον. Το 1995 ιδρύεται η ομάδα ειδικών συμφερόντων (SIG) matrix-sig με στόχο τον καθορισμό ενός πακέτου υπολογισμών με πίνακες. Ανάμεσα στα μέλη της ομάδας ήταν και ο σχεδιαστής και συντηρητής της Python, Guido van Rossum, ο οποίος επέκτεινε τη σύνταξη της Python για να διευκολύνει τον υπολογισμό πινάκων.

Η πρώτη υλοποίηση, που έγινε το 1995, ονομάστηκε Numeric, γνωστό και ως Αριθμητικές επεκτάσεις Python. Βασικά το Numeric είχε πολλά ψευδώνυμα, για παράδειγμα το όνομα του έργου του SourceForge για αυτήν την βιβλιοθήκη είναι numpy, ενώ ο Konrad Hinsen ονόμασε το πακέτο του ScientificPython ως μνεία στο Numerical Python. Βασίστηκε στην υλοποίηση ενός πακέτου πινάκων, που ολοκληρώθηκε από τον Jim Fulton και γενικεύτηκε από τον τότε μεταπτυχιακό φοιτητή στο Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT) Jim Hugunin. Συνεισφέροντες ήταν και οι David Ascher, Paul DuBois και Konrad Hinsen.

Η τελευταία έκδοση του Numeric (v24.2) κυκλοφόρησε στις 11 Νοεμβρίου του 2005. Αρκετοί επιστήμονες χρησιμοποίησαν το Numeric σαν βάση για τον επιστημονικό τους κώδικα, με αποτέλεσμα να αναπτύξουν τις δικές τους ενότητες. Το 2001 οι Travis Oliphant, Eric Jones και Pearu Peterson ένωσαν τις ενότητες τους σε ένα επιστημονικό πακέτο που το ονόμασαν SciPy.

Ωστόσο, οι επιστήμονες ήθελαν την επέκταση του Numeric με τρόπο που δεν επιτρεπόταν από τον κώδικα των τότε βάσεων, και ενώ υπήρχε η επιθυμία να εισαχθεί το Numeric ή έστω κάτι παρόμοιο στην τυπική βιβλιοθήκη, αλλά ο Guido van Rossum ξεκαθάρισε ότι ο κώδικας δεν μπορούσε να διατηρηθεί στην τότε κατάστασή του. Αποτέλεσμα όλων αυτών ήταν να γραφτεί, από τους Perry Greenfield, Todd Miller και Rick White, ένα νέο πακέτο που ονομάστηκε Numarray και ήταν μία πιο ευέλικτη αντικατάσταση του Numeric. Επειδή οι δημιουργοί του Numarray εργάζονταν στο Επιστημονικό Ινστιτούτο του Διαστημικού Τηλεσκοπίου, σκόπευαν να χρησιμοποιήσουν το Numarray για επεξεργασίες αστρονομικών εικόνων.

Το Numarray είχε πιο γρήγορες λειτουργίες για μεγάλους πίνακες, αλλά για τους μικρούς πίνακες ήταν πιο αργός, έτσι και οι δύο βιβλιοθήκες χρησιμοποιήθηκαν παράλληλα σε Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

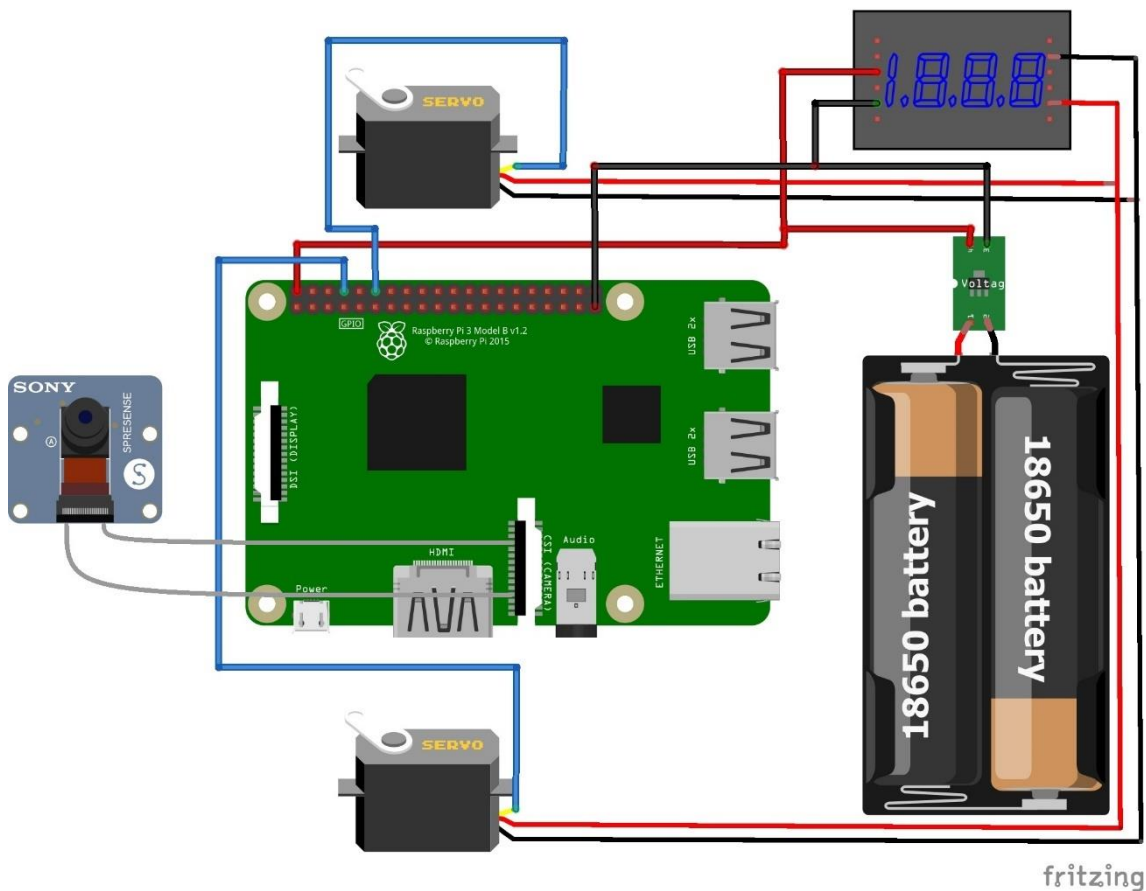
διαφορετικές περιπτώσεις. Η τελευταία έκδοση του Numarray (v1.5.2) κυκλοφόρησε στις 24 Αυγούστου του 2006. Και η βιβλιοθήκη Numeric και η βιβλιοθήκη Numarray έχουν πλέον καταργηθεί.

Στις αρχές του 2005 ο Travis Oliphant ήθελε να ενοποιήσει την κοινότητα γύρω από ένα πακέτο πίνακα που θα ήταν ενιαίο, οπότε μετέφερε τις δυνατότητες του Numarray στο Numeric και κυκλοφόρησε το αποτέλεσμα το 2006 με το όνομα NumPy 1.0. Το NumPy ήταν μέρος του SciPy, όμως διαχωρίστηκε από αυτό για να αποφευχθεί η εγκατάσταση του μεγάλου πακέτου SciPy για την λήψη ενός πίνακα αντικειμένου. Το 2011 με την έκδοση NumPy 1.5.0 προστέθηκε η υποστήριξη για Python 3. (Charles R. Harris¹, 2020) (NumPy developers, n.d.)

3.2.3 Matplotlib

Το Matplotlib είναι μία βιβλιοθήκη που γράφτηκε από τον John D. Hunter (1968 - 2012) με την συνεισφορά από έναν αυξανόμενο αριθμό προγραμματιστών και χρηστών. Σύμφωνα με το εισαγωγικό κείμενο που γράφτηκε από τον ίδιο τον John D. Hunter το 2008, το Matplotlib είναι μια βιβλιοθήκη για τη δημιουργία δισδιάστατων γραφημάτων για πίνακες στην Python. Παρ'όλο που έχει τις ρίζες του στην εξομίωση των εντολών γραφικών MATLAB, είναι ανεξάρτητο από το MATLAB και μπορεί να χρησιμοποιηθεί με αντικειμενοστραφή τρόπο στο Pythonic. Το Matplotlib είναι γραμμένο σε καθαρή Python, όμως κάνει χρήση του NumPy και άλλου κώδικα επέκτασης για την παροχή καλής απόδοσης ακόμη και μεγάλων πινάκων. ((1968-2012), 2008)

4 Συνδεσμολογία



Εικόνα 27. Συνδεσμολογία.

Η συνδεσμολογία του συγκεκριμένου οχήματος έγινε ως εξής :

Οι δύο σερβοκινητήρες είναι τοποθετημένοι δεξιά και αριστερά από το σασί. Σε αυτούς τους σερβοκινητήρες έχουν τοποθετηθεί οι δύο ρόδες σιλικόνης, το σήμα ή PWM των κινητήρων έχει συνδεθεί

με το Raspberry Pi μέσω των ακροδεκτών 14 και 18.

Τα καλώδια της γείωσης και της τροφοδοσίας είναι συνδεδεμένα με τον ρυθμιστή τάσης. Ο ρυθμιστής τάσης με την σειρά του έχει συνδεθεί με το βολτόμετρο. Επίσης συνδεδεμένο είναι και το Raspberry με τον ρυθμιστή τάσης και τέλος οι μπαταρίες είναι συνδεδεμένες με τον ρυθμιστή τάσης τροφοδοτώντας και την πλακέτα και τους σερβοκινητήρες.

Οι μπαταρίες είναι συνδεδεμένες σε σειρά και το σύνολο της τάσης τους είναι 7,20 V. Όμως το Raspberry πρέπει να τροφοδοτείται με μόνο 5V, αλλιώς θα καεί η πλακέτα, γι' αυτό ακριβώς τον λόγο έχει τοποθετηθεί ο ρυθμιστής τάσης ώστε να ρυθμίσει την τάση στο θεμιτό επίπεδο, τροφοδοτώντας επαρκώς και την πλακέτα, αλλά και τους δύο σερβοκινητήρες.

Το βολτόμετρο έχει τοποθετηθεί ώστε να είναι ευκολότερος ο έλεγχος της τάσης τροφοδοσίας τόσο για την πλακέτα όσο και για τους σερβοκινητήρες.

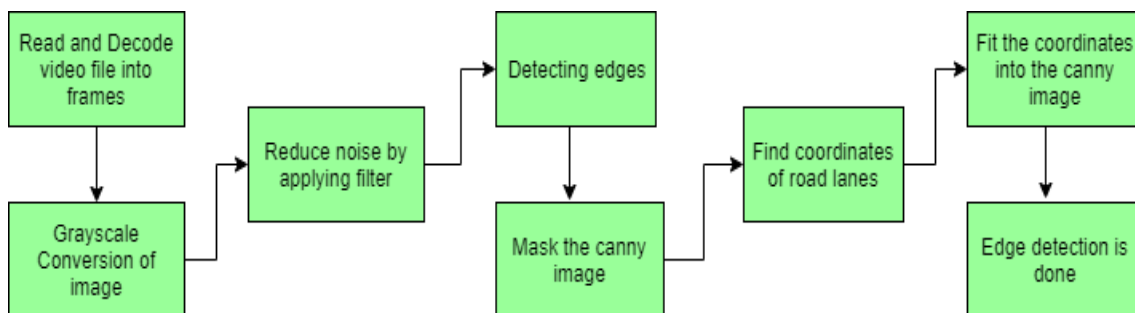
Τέλος, η κάμερα έχει συνδεθεί με τον υπολογιστή μονής πλακέτας στην κατάλληλη θήρα. Σε αυτό το σημείο θα πρέπει να ειπωθεί ότι στην πρώτη προσπάθεια σύνδεσης της πλακέτας με τροφοδοτικό χρησιμοποιήθηκε power bank, όμως κατά τη διάρκεια της εκκίνησης οι σερβοκινητήρες καταναλώνουν περισσότερο ρεύμα με αποτέλεσμα να μην είναι επαρκές το Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

ρεύμα για την εκκίνηση του Raspberry. Αυτό αντιμετωπίστηκε με την τοποθέτηση των μπαταριών, οι οποίες τροφοδοτούν επαρκώς και την πλακέτα αλλά και τους σερβοκινητήρες. Η παραπάνω εικόνα αντιπροσωπεύει την συνδεσμολογία του οχήματος.

5 Ανίχνευση λωρίδας σε πραγματικό χρόνο

5.1 Αλγόριθμος ανίχνευσης λωρίδας

Μία από τις σημαντικότερες καινοτομίες στην Τεχνητή Νοημοσύνη είναι τα αυτοκατευθυνόμενα οχήματα, τα οποία αξιοποιούν αλγόριθμους Βαθιάς Μάθησης (Deep Learning) και εξελίσσουν την κοινωνία δημιουργώντας νέες ευκαιρίες στον τομέα της μετακίνησης. Για να κατασκευαστεί ένα αυτοκινούμενο όχημα είναι αναγκαίο να χρησιμοποιηθούν μοντέλα τεχνητής νοημοσύνης που θα είναι κατάλληλα εκπαιδευμένα. Το αρχικό στάδιο από τα πολλά βήματα που περιλαμβάνονται στην εκπαίδευση των μοντέλων ενός τέτοιου οχήματος, είναι η ανίχνευση λωρίδας. Η ανίχνευση λωρίδας περιέχει συγκεκριμένα βήματα, τα οποία χρησιμοποιούνται είτε για υπολογιστική όραση σε πραγματικό χρόνο, όπως και στην συγκεκριμένη εργασία, είτε για ανίχνευση λωρίδας σε βίντεο. Τα βήματα είναι τα εξής:

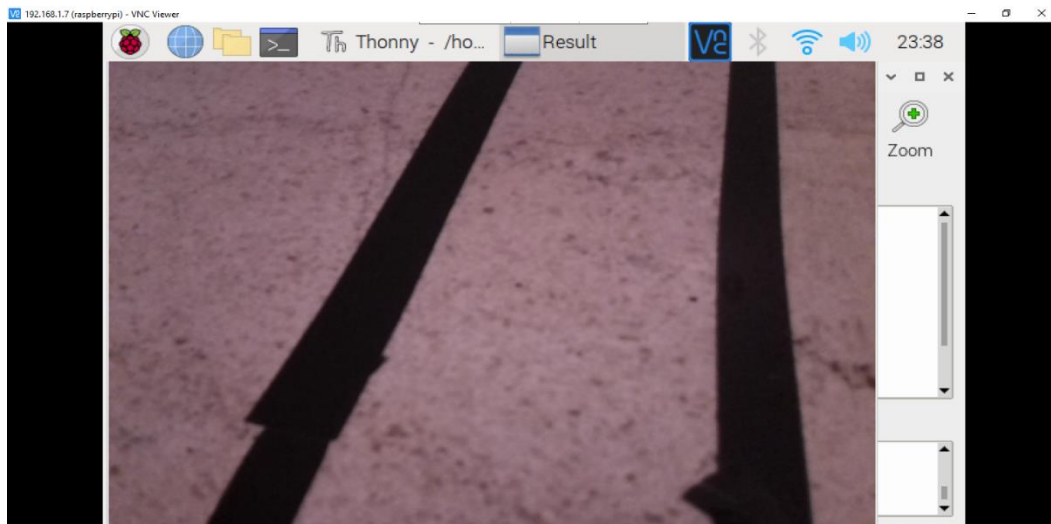


Εικόνα 28. Βήματα ανίχνευσης λωρίδας (Geeks, Geeks for, 2022).

Αναλυτικότερα :

- Πρώτο βήμα :

Σκοπός αυτού του βήματος είναι η λήψη ενός βίντεο (αυτό γίνεται και με την ανίχνευση λωρίδας σε πραγματικό χρόνο, όπως έχει προαναφερθεί), η αρχικοποίηση της λήψης και η μετατροπή του συγκεκριμένου βίντεο σε μία ακολουθία εικόνων, δηλαδή η αποκωδικοποίηση κάθε καρέ του βίντεο.



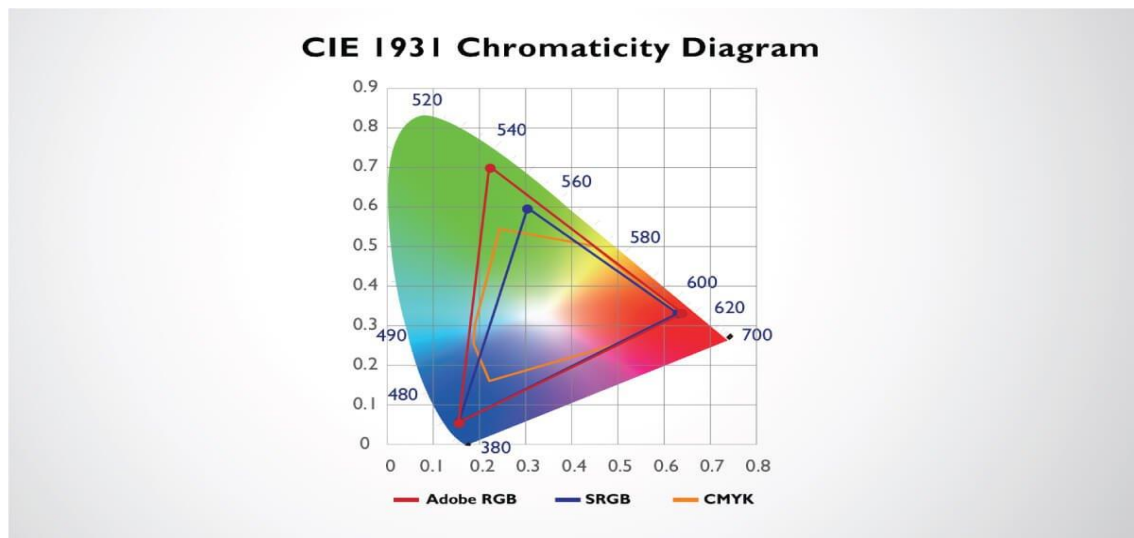
Εικόνα 29. Το πρώτο βήμα.

Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

- Δεύτερο βήμα :

Σε αυτό το βήμα περιλαμβάνεται η μετατροπή εικόνας σε κλίμακα του γκρι. Συγκεκριμένα κάθε εικόνα έχει τρεις βασικές ιδιότητες :

1. Το μέγεθος : Αυτό είναι το ύψος και το πλάτος εικόνας και αναπαρίσταται σε εκατοστά, ίντσες ή εικονοστοιχεία (pixels).
2. Ο χρωματικός χώρος : Οι χώροι μέσα στους οποίους γίνεται η διαχείριση των χρωμάτων ονομάζονται χρωματικοί. Μέσα στους χώρους αυτούς υπάρχουν τα χρωματικά πρότυπα. Παραδείγματα χρωματικών προτύπων αποτελούν το RGB (Red, Green, Blue) και το HSV (Hue, Saturation, Value).



Εικόνα 30. Χρωματικό διάγραμμα (BenQ Color Technology Lab, 2019).

3. Τα κανάλια : Τα κανάλια αποτελούν χαρακτηριστικό του χρωματικού προτύπου. Για παράδειγμα το RGB χρωματικό πρότυπο έχει τρία είδη χρωμάτων ή χαρακτηριστικών, το κόκκινο (Red), το πράσινο (Green) και το μπλε (Blue) από όπου πήρε και το όνομα του.

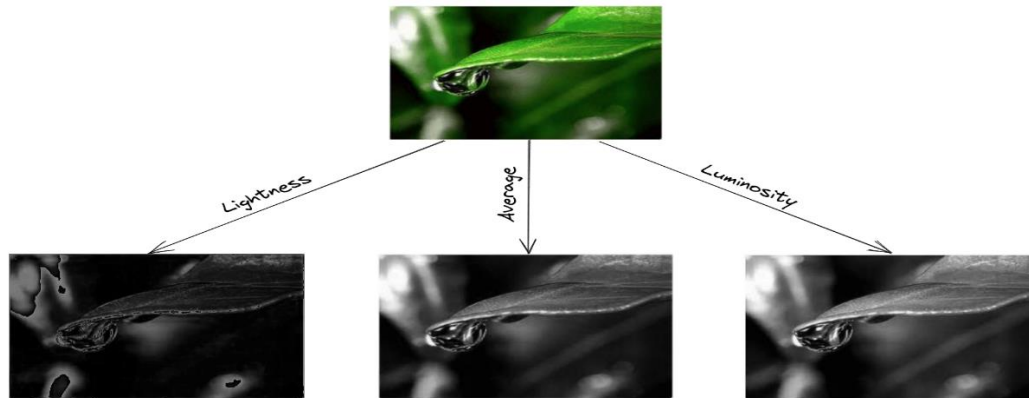
Η διαφορά μεταξύ RGB εικόνας και εικόνας σε κλίμακα του γκρι έγκειται στο ότι η RGB εικόνα (en.wikipedia.org, 13 October 2022) χρησιμοποιεί τρία κανάλια, ενώ η εικόνα σε κλίμακα του γκρι μόνο ένα. Αυτό σημαίνει ότι μια εικόνα της κλίμακας του γκρι είναι μία 8-bit εικόνα, με κάθε εικονοστοιχείο (pixel) να περιέχει 256 διαφορετικές αποχρώσεις του γκρι, από την τιμή εικονοστοιχείου 0 που αντιπροσωπεύει το μαύρο μέχρι και την τιμή 255 που αντιπροσωπεύει το λευκό. Η μετατροπή στην κλίμακα του γκρι είναι εξαιρετικά σημαντικό βήμα, καθώς μέσω της κλίμακας αυτής είναι πιο εύκολος ο εντοπισμός των βασικών ακμών. Τα χρώματα, που προσλαμβάνονται από την κάμερα, πολλές φορές δίνουν πληροφορίες που μπερδεύουν περισσότερο, παρά βοηθούν στην ανίχνευση λωρίδας.

Η μετατροπή στην κλίμακα του γκρι γίνεται ως εξής :

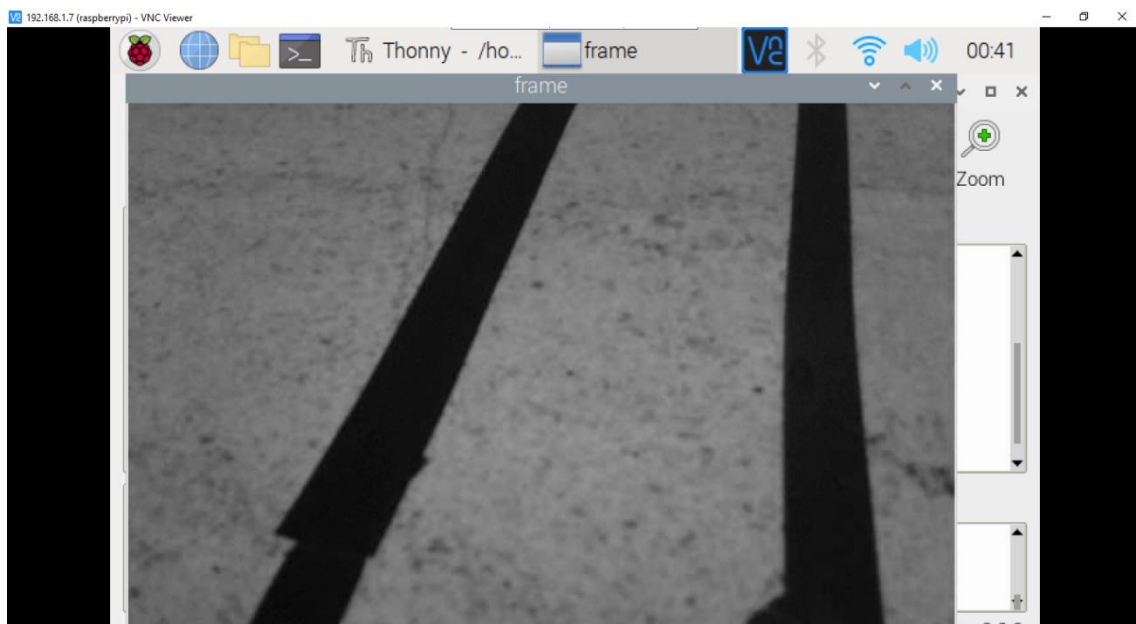
Για να γίνει η μετατροπή ενός χρώματος σε κλίμακα του γκρι θα πρέπει να υπολογιστεί η ένταση της φωτεινότητας αυτού του χρώματος. Ένας γρήγορος τρόπος υπολογισμού της έντασης αυτής είναι η λήψη της μέσης τιμής των κόκκινων, πράσινων και μπλε καναλιών, με τον παρακάτω τύπο:

$$I = \frac{R + G + B}{3}$$

Παρ'όλο που η μέθοδος αυτή δεν είναι τέλεια, διότι το ανθρώπινο μάτι δεν αντιλαμβάνεται στο ίδιο επίπεδο έντασης το κόκκινο, το πράσινο και το μπλε, παράγει λογικά αποτελέσματα.



Εικόνα 31. Μετατροπή εικόνας σε κλίμακα του γκρι (Antoniadis, 2022).



Εικόνα 32. Η εικόνα μετά την μετατροπή στη κλίμακα του Γκρι

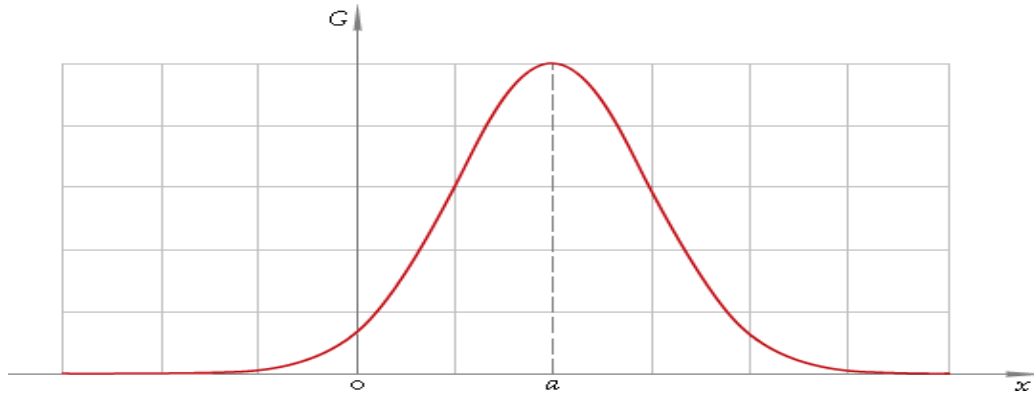
- Τρίτο βήμα :

Το βήμα αυτό στοχεύει στη μείωση του θορύβου. Θόρυβος σε μία εικόνα ονομάζεται οτιδήποτε δεν μας ενδιαφέρει σε αυτήν, αυτό μπορεί να είναι είτε διακυμάνσεις φωτός, είτε επιδράσεις κβαντοποίησης, είτε θορύβους αισθητήρα. Επίσης, υπάρχουν τρεις τύποι θορύβου σε μία εικόνα, ο παλμικός θόρυβος που περιλαμβάνει τυχαία συμβάντα από λευκά εικονοστοιχεία (pixel), ο θόρυβος αλάτι και πιπέρι που περιλαμβάνει την εμφάνιση ασπρόμαυρων εικονοστοιχείων και τέλος ο Γκαουσιανός θόρυβος που είναι διακυμάνσεις στην ένταση που ακολουθούν την κανονική κατανομή του Gauss. Ο θόρυβος μπορεί να δημιουργήσει ψευδείς ακμές, επομένως αποτελεί επιτακτική ανάγκη η εξομάλυνση της εικόνας από τους ήχους. Για να επιτευχθεί η εξομάλυνση εικόνων θα πρέπει να χρησιμοποιηθεί το φίλτρο Gauss, το οποίο αποτελεί στάδιο προεπεξεργασίας στους αλγόριθμους υπολογιστικής όρασης, καθώς και στα μοντέλα Τεχνητής Νοημοσύνης και Βαθιάς Μάθησης, προκειμένου να βελτιωθούν οι δομές εικόνας με βάση το ανθρώπινο οπτικό σύστημα αντίληψης. Μάλιστα, έχει διαπιστωθεί ότι οι νευρώνες δημιουργούν παρόμοιο φίλτρο με αυτό του Gauss. Το φίλτρο αυτό είναι ένα χαμηλό περατό φίλτρο που αφαιρεί τα ακραία εικονοστοιχεία ή τα στοιχεία υψηλής συχνότητας για μείωση του θορύβου και χρησιμοποιεί τα παράθυρα φιλτραρίσματα ή μάσκες φίλτρου, δηλαδή μία περιοχική επιλογής,

γραμμικά. Πήρε το όνομά του από τον διάσημο επιστήμονα Carl Friedrich Gauss, επειδή τα συμμετρικά βάρη που περιέχονται στο φίλτρο υπολογίζονται σύμφωνα με την Γκαουσιανή συνάρτηση κατανομής. Η συνάρτηση κατανομής του Gauss έχει σχήμα καμπάνας και στην (DigitalSreeni, χ.χ.) (Wikipedia, n.d.) γενική της μορφή είναι :

$$G(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\alpha)^2}{2\sigma^2}}$$

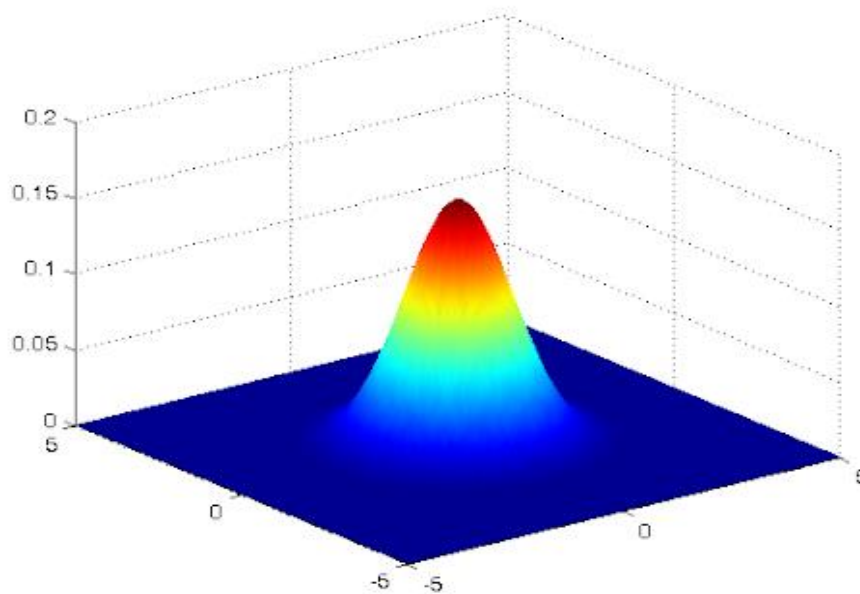
Όπου το α είναι η μέση τιμή του x και σ η τυπική του απόκλιση. Η τυπική απόκλιση στο τετράγωνο, σ^2 , ονομάζεται διακύμανση του x και η καμπύλη της συνάρτησης είναι η εξής :



Εικόνα 33. Καμπύλη της συνάρτησης του Gauss.

Το φίλτρο του Gauss αναθέτει διαφορετικά μεγέθη στα εικονοστοιχεία μέσα σε κάθε περιοχή του πυρήνα και αντικαθιστά το κεντρικό στοιχείο με το Γκαουσιανό σταθμισμένο μέσο όρο. Ο πυρήνας δεν είναι τίποτε άλλο παρά ένας μικρός πίνακας που χρησιμοποιείται για την όξυνση, την θόλωση ή την ανίχνευση ακμών. Μερικές φορές ο πυρήνας ονομάζεται μάσκα ή φίλτρο. Σε κάθε πυρήνα, τα εικονοστοιχεία που βρίσκονται στο κέντρο γενικά τείνουν να έχουν μεγαλύτερο βάρος και μειώνονται όσο πλησιάζουμε στην άκρη του πυρήνα στις κατευθύνσεις x και y . Σε αυτό το σημείο χρησιμοποιείται η συνάρτηση κατανομής του Gauss για δύο διαστάσεις ώστε να δημιουργηθεί ένας πίνακας $n \times m$ και να ξεκινήσει η Γκαουσιανή εξομάλυνση (Mark S. Nixon and Alberto S. Aguado, 2002). Ο τύπος της συνάρτησης κατανομής του Gauss στις 2 διαστάσεις είναι :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}$$



Εικόνα 34. Γραφική παράσταση συνάρτησης Gauss για 2 διαστάσεις (Sharda, 2021).

Οι τιμές που παράγονται από την συνάρτηση αυτή, δημιουργούν τον λεγόμενο πίνακα συνέλιξης ή πυρήνα, που εφαρμόζεται σε κάθε εικονοστοιχείο της αρχικής εικόνας. Ο πυρήνας συνήθως είναι αρκετά μικρός καθώς όσο μεγαλύτερος είναι τόσο περισσότεροι υπολογισμοί πρέπει να γίνουν για κάθε εικονοστοιχείο. Τα x και y καθορίζουν το δέλτα από το κεντρικό εικονοστοιχείο $(0,0)$, εάν για παράδειγμα η επιλεγμένη ακτίνα για τον πυρήνα ήταν 3, τα x και y θα κυμαίνονταν από το -3 έως το 3 (τα άκρα συμπεριλαμβάνονται). Το σ είναι η τυπική απόκλιση και επιδρά στο πόσο σημαντικά τα γειτονικά εικονοστοιχεία των κεντρικών εικονοστοιχείων επηρεάζουν το αποτέλεσμα των υπολογισμών. Θα μπορούσε να υποστηριχθεί ότι θα έπρεπε να ληφθούν υπόψη όλα τα εικονοστοιχεία στην εικόνα για να είναι σωστό το αποτέλεσμα αφού η συνάρτηση κατανομής εκτείνεται άπειρα, αλλά στην πράξη τα εικονοστοιχεία πέρα του 3σ (δηλαδή με απόσταση από το εξεταζόμενο στοιχείο μεγαλύτερη από τρεις τυπικές αποκλίσεις) έχουν πολύ μικρό αντίκτυπο στις τιμές που προκύπτουν.

Κατά την εκτέλεση χρειάζεται να δημιουργηθεί μία ξεχωριστή εικόνα εισόδου. Η εικόνα προέλευσης δεν μπορεί να τροποποιηθεί απευθείας, διότι η αλλαγή των τιμών των εικονοστοιχείων θα μπερδέψει τους υπολογισμούς για το γειτονικό εικονοστοιχείο στην επόμενη προσέγγιση. Σε αυτό το σημείο επεξεργαζόμαστε τις ακμές. Εάν παρατηρηθεί το πρώτο εικονοστοιχείο σε μία εικόνα, ο πυρήνας του θα εκτεινόταν πέρα από τα όρια της εικόνας. Αυτό έχει ως αποτέλεσμα, κατά την υλοποίηση να αγνοείται το εξωτερικό σύνολο εικονοστοιχείων και να αντιγράφονται οι ακμές ή να δημιουργείται καμπύλη στην εικόνα.

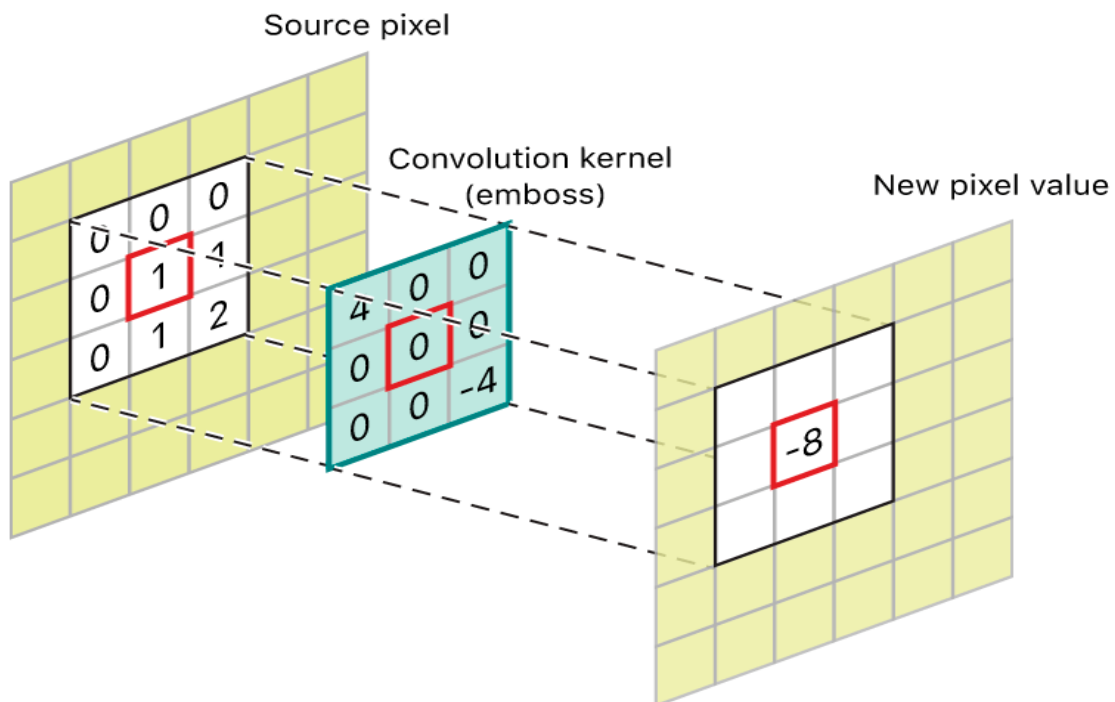
Για την εφαρμογή της συνάρτησης κατανομής του Gauss πρέπει να προσδιοριστούν λογικές τιμές για τα x , y και σ . Ο πυρήνας τεχνικά μπορεί να έχει αυθαίρετο μέγεθος, αυτό σημαίνει ότι το σ θα πρέπει να κλιμακώνεται αναλογικά με το μέγεθος του πυρήνα. Στην περίπτωση που η ακτίνα του πυρήνα είναι μεγάλη, αλλά το σ μικρό, τότε όλα τα νέα εικονοστοιχεία που εισάγονται με την μεγαλύτερη ακτίνα, δεν επηρεάζουν πραγματικά τον υπολογισμό. Για να ολοκληρωθεί η εκτέλεση, θα πρέπει να κανονικοποιηθούν οι τιμές στον πυρήνα. Σε διαφορετική περίπτωση το αποτέλεσμα της εικόνας δεν θα είναι το βέλτιστο, καθώς οι τιμές θα αθροίζονται σε ελαφρά μικρότερη τιμή από το ένα.

Τέλος, το μέγεθος του πυρήνα θα πρέπει να είναι περιττός αριθμός ούτως ώστε να διασφαλιστεί ότι υπάρχει πραγματικό κεντρικό εικονοστοιχείο. Όταν υπολογιστεί ένας κατάλληλος πυρήνας, τότε εκτελείται η Γκαουσιανή εξομάλυνση με την χρήση συνέλιξης (convolution) (Rein Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

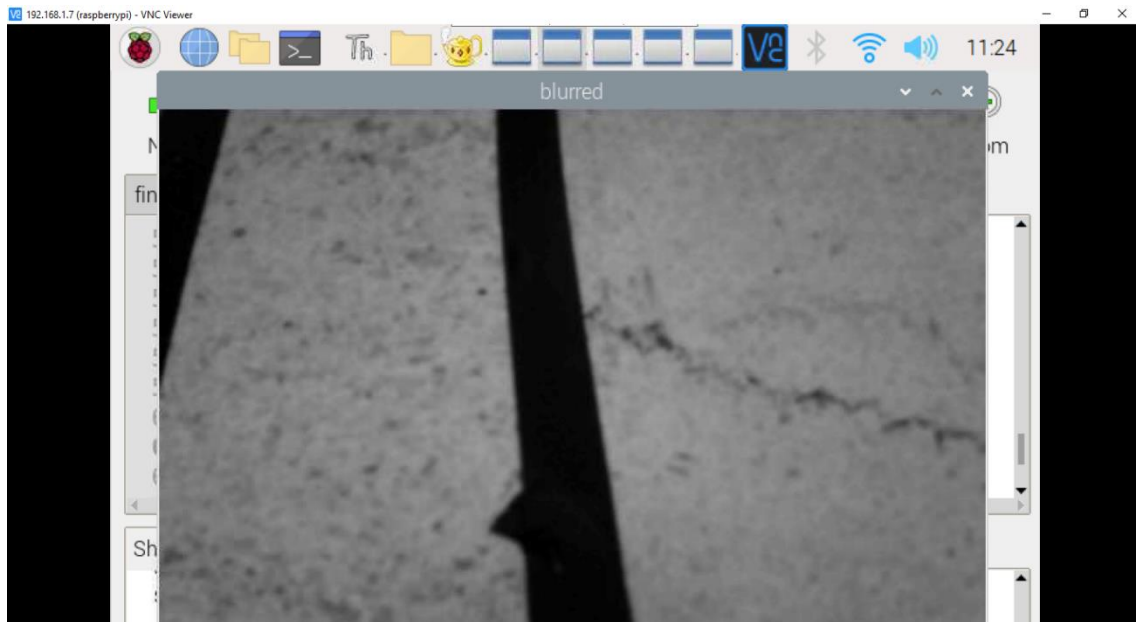
van den Boomgaard, 2021). Μαθηματικά, η συνέλιξη είναι μία πράξη που εκτελείται με δύο συναρτήσεις (f και g), για να παραχθεί μία τρίτη συνάρτηση. Στην επεξεργασία εικόνας, η συνέλιξη είναι η διαδικασία μετασχηματισμού μίας εικόνας με την εφαρμογή ενός πυρήνα πάνω από κάθε εικονοστοιχείο και τους τοπικούς γείτονες του σε ολόκληρη την εικόνα. Ο πυρήνας είναι ένας πίνακας τιμών του οποίου το μέγεθος και οι τιμές καθορίζουν το αποτέλεσμα του μετασχηματισμού στην διαδικασία της συνέλιξης.

Η διαδικασία της συνέλιξης έχει τρία βήματα, το πρώτο είναι η τοποθέτηση του πυρήνα συνέλιξης πάνω από κάθε εικονοστοιχείο της εικόνας και τον πολλαπλασιασμό κάθε τιμής του πυρήνα με το αντίστοιχο εικονοστοιχείο. Το δεύτερο βήμα είναι το άθροισμα των πολλαπλασιασμένων τιμών που προκύπτουν και στη συνέχεια επιστρέφεται η προκύπτουσα τιμή ως νέα τιμή του κεντρικού εικονοστοιχείου. Το τρίτο βήμα είναι η επανάληψη της διαδικασίας σε ολόκληρη την εικόνα. Στην παρακάτω εικόνα φαίνεται ένας πυρήνας 3×3 που περιπλέκεται πάνω από μία εικόνα πηγής 6×6 . Το κεντρικό στοιχείο του πυρήνα τοποθετείται πάνω από το εικονοστοιχείο πηγής και στη συνέχεια το εικονοστοιχείο πηγής αντικαθίσταται με ένα σταθμισμένο άθροισμα του ίδιου και των γύρω εικονοστοιχείων.

Η έξοδος τοποθετείται στην τιμή του εικονοστοιχείου προορισμού. Στο συγκεκριμένο παράδειγμα στην πρώτη θέση έχουμε το 0 στο εικονοστοιχείο πηγής και 4 στον πυρήνα, πολλαπλασιάζοντας 4×0 έχει αποτέλεσμα 0, οπότε πηγαίνει στο επόμενο εικονοστοιχείο όπου στο εικονοστοιχείο πηγής είναι το 0 όπως και στις επόμενες δύο θέσεις. Πολλαπλασιάζοντας 0×0 βγαίνει πάλι 0. Συνεχίζοντας, στο κέντρο της εικόνας πηγής υπάρχει το 1 και στην αντίστοιχη θέση του πυρήνα το 0, πολλαπλασιάζοντας 0×1 βγαίνει 0 και συνεχίζεται αυτή η διαδικασία μέχρι που καταλήγουμε στην τελευταία θέση όπου το εικονοστοιχείο πηγής είναι 2 και -4 το αντίστοιχο εικονοστοιχείο στον πυρήνα. Πολλαπλασιάζοντας 2×-4 βγαίνει το αποτέλεσμα -8. Σε αυτή την φάση αθροίζουμε όλα τα αυτά τα αποτελέσματα και παίρνουμε ως απάντηση το -8, οπότε το αποτέλεσμα της συνέλιξης θα είναι το -8 και το αποτέλεσμα αυτό ενημερώνεται στην εικόνα προορισμού. Αλλάζοντας τις τιμές στον πυρήνα αλλάζουν και το εφέ στην εικόνα. Να σημειωθεί εδώ ότι αυτή η διαδικασία ονομάζεται μετασχηματισμός Weierstrass.



Εικόνα 35. Αποτέλεσμα της συνέλιξης (Sharda, 2021).



Εικόνα 36. Η εικόνα μετά την χρήση του Γκαουσιανού θολώματος.

- Τέταρτο βήμα :

Canny Edge Detection, ανίχνευση ακμών Canny. Για την ανίχνευση ενός φάσματος ακμών στις εικόνες, χρησιμοποιείται ένας αλγόριθμος πολλαπλών σταδίων που αναπτύχθηκε από τον John F. Canny το 1986. Είναι μία τεχνική για την εξαγωγή χρήσιμων δομικών πληροφοριών από διάφορα αντικείμενα, που προσλαμβάνονται μέσω της μηχανικής όρασης, και τη μείωση του όγκου δεδομένων προς επεξεργασία. Έχει εφαρμοστεί ευρέως σε διάφορα συστήματα υπολογιστικής όρασης καθώς ο Canny ανακάλυψε ότι οι απαιτήσεις για την εφαρμογή ανίχνευσης ακμών σε διαφορετικά συστήματα όρασης είναι σχετικά παρόμοιες, έτσι μια Canny εφαρμογή ανίχνευσης ακμών μπορεί να χρησιμοποιηθεί για πολλές και διαφορετικές καταστάσεις. Για να θεωρηθεί επιτυχής η ανίχνευση ακμών με την εφαρμογή Canny θα πρέπει να ικανοποιούνται οι εξής απαιτήσεις:

1. Ανίχνευση με χαμηλό ποσοστό σφάλματος, αυτό σημαίνει ότι η ανίχνευση θα πρέπει να πιάσει με ακρίβεια όσο το δυνατό περισσότερες από τις ακμές που φαίνονται στην εικόνα.
2. Το σημείο ακμής που ανιχνεύεται από τον χειριστή, θα πρέπει να εντοπίζεται με ακρίβεια στο κέντρο της ακμής.
3. Μία δεδομένη ακμή στην εικόνα θα πρέπει να επισημαίνεται μόνο μία φορά και στο σημείο που είναι πιο έντονη. Επίσης ο θόρυβος της εικόνας δεν πρέπει να δημιουργεί ψευδείς ακμές.

Για να πετύχει τα παραπάνω ο Canny χρησιμοποίησε τεχνικές από το πεδίο του Λογισμού Μεταβολών (Calculus of Variations), δηλαδή εργαλεία που βελτιστοποιούν μια δοσμένη συνάρτηση. Η βέλτιστη συνάρτηση στον ανιχνευτή του Canny περιγράφεται από το άθροισμα τεσσάρων εκθετικών όρων, αλλά μπορεί να προσεγγιστεί ικανοποιητικά και από την πρώτη παράγωγο της συνάρτησης του Gauss. (“Canny ανιχνευτής άκρων”) Μεταξύ των μεθόδων ανίχνευσης που έχουν αναπτυχθεί ως τώρα, ο αλγόριθμος ανίχνευσης Canny είναι μια από τις πιο αυστηρά καθορισμένες μεθόδους που παρέχει αξιόπιστα αποτελέσματα. Μάλιστα είναι ένας από τους πιο δημοφιλείς αλγόριθμους, λόγω της αυστηρότητας των τριών προϋποθέσεων που πρέπει να πληρούνται κατά την ανίχνευση και λόγω της απλότητας κατά τη διαδικασία υλοποίησης. Ο αλγόριθμος ανίχνευσης ακμών Canny αναλύεται σε πέντε διαφορετικά βήματα.

- Εφαρμογή του φίλτρου του Gauss για την εξομάλυνση της εικόνας και αφαίρεση των θορύβων,

- Εύρεση της έντασης κλίσεων της εικόνας (κλίση εικόνας είναι μία αλλαγή κατεύθυνσης στην ένταση ή το χρώμα σε μία εικόνα και αποτελεί είναι από τα θεμελιώδη δομικά στοιχεία επεξεργασίας εικόνας). Επί της ουσίας ο αλγόριθμος αυτός υπολογίζει όλες τις κατευθύνσεις της κλίσης στην θολή εικόνα που δημιουργήθηκε απο την χρήση του φίλτρου Gauss και στην συνέχεια εντοπίζει τις ακμές με μεγάλες αλλαγές έντασης. Μία ακμή σε μία εικόνα μπορεί να δείχνει προς διάφορες κατευθύνσεις, επομένως ο αλγόριθμος Canny χρησιμοποιεί τέσσερα φίλτρα ανίχνευσης, ώστε να ανιχνεύονται οι οριζόντιες, οι κάθετες και διαγώνιες ακμές στην θολή εικόνα. Ο τελεστής ανίχνευσης ακμών επιστρέφει μία τιμή για την πρώτη παράγωγο στην οριζόντια κατεύθυνση (G_x) και στην κατακόρυφη κατεύθυνση (G_y). Σε αυτό το σημείο, θα πρέπει να προσδιοριστεί η κατεύθυνση της κλίσης της ακμής με τις παρακάτω συναρτήσεις :

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \text{atan2}(G_y, G_x)$$

Όπου το G μπορεί να υπολογιστεί με Πυθαγόρεια πρόσθεση και το atan2 είναι η συνάρτηση του τόξου με δύο ορίσματα. Η γωνία της ακμής στρογγυλοποιείται σε μία γωνία από της τέσσερις γωνίες που αντιπροσωπεύουν την κατακόρυφη, την οριζόντια και τις δύο διαγώνιες (0° , 45° , 90° και 135°). Μία κατεύθυνση ακμής που πέφτει σε κάθε περιοχή χρώματος, ορίζεται σε συγκεκριμένη τιμή γωνίας, για παράδειγμα, η θ στα σημεία $[0^\circ, 22.5^\circ]$ ή $[157.5^\circ, 180^\circ]$ αντιστοιχίζεται στο μηδέν.

- Εφαρμογή ορίου στο μέγεθος της κλίσης ή περιορισμός του κατώτερου ορίου ώστε να απαλλαχθεί τελείως η εικόνα από τα ψευδή αποτελέσματα κατά την ανίχνευση ακμών. Είναι μία τεχνική απίσχνασης ακμών που χρησιμοποιείται για την αφαίρεση ανεπιθύμητων και αναληθών σημείων στις άκρες μίας εικόνας, χρησιμοποιείται μετά την εφαρμογή του φίλτρου του Gauss στην εικόνα και στην ουσία εφαρμόζεται ο περιορισμός του κατώτερου ορίου για την εύρεση των θέσεων με την μεγαλύτερη αλλαγή τιμής της έντασης. Ο αλγόριθμος για κάθε εικονοστοιχείο στην κλίση εικόνας περιλαμβάνει πρώτα την σύγκριση της έντασης της ακμής του τρέχοντος εικονοστοιχείου, με την ισχύ ακμής του εικονοστοιχείου στις θετικές και αρνητικές κατευθύνσεις κλίσης. Έπειτα, εάν η ένταση της ακμής του τρέχοντος εικονοστοιχείου είναι μεγαλύτερη σε σύγκριση με τα υπόλοιπα εικονοστοιχεία του πυρήνα που έχουν την ίδια κατεύθυνση (για παράδειγμα ένα εικονοστοιχείο που δείχνει προς την κατεύθυνση y , θα συγκριθεί με το εικονοστοιχείο που βρίσκεται πάνω και κάτω από αυτό στον κατακόρυφο άξονα), η τιμή θα διατηρηθεί. Σε διαφορετική περίπτωση, η τιμή θα καταργηθεί. Σε ορισμένες υλοποιήσεις, ο αλγόριθμος κατηγοριοποιεί σε ένα μικρό σύνολο διακριτών κατευθύνσεων τις συνεχείς κατευθύνσεις κλίσης. Στη συνέχεια μετακινεί ένα φίλτρο 3×3 πάνω στο αποτέλεσμα του προηγούμενου βήματος (δηλαδή, την ένταση των ακμών και τις κατευθύνσεις της κλίσης). Σε κάθε εικονοστοιχείο, συγκρατεί την ένταση της ακμής του κεντρικού εικονοστοιχείου, ρυθμίζοντας την τιμή σε μηδέν, εάν το μέγεθός της δεν είναι μεγαλύτερο από το μέγεθος των δύο γειτόνων στην κατεύθυνση κλίσης. Για παράδειγμα στην περίπτωση που η στρογγυλοποιημένη γωνία κλίσης είναι 0° , δηλαδή μεταξύ της βόρειας και νότιας κατεύθυνσης, το σημείο θεωρείται ότι βρίσκεται στην ακμή εάν το μέγεθος της κλίσης είναι μεγαλύτερο από τα μεγέθη των εικονοστοιχείων που έχουν κατεύθυνση ανατολική και δυτική. Σε άλλη περίπτωση, όταν η στρογγυλοποιημένη γωνία κλίσης είναι 90° , δηλαδή η ακμή είναι σε κατεύθυνση μεταξύ ανατολής και δύσης, το σημείο θεωρείται ότι βρίσκεται στην ακμή εάν το μέγεθος της κλίσης είναι μεγαλύτερο από τα μεγέθη στο εικονοστοιχεία που βρίσκονται μεταξύ της βόρειας και νότιας κατεύθυνσης. Στην περίπτωση που η στρογγυλοποιημένη γωνία κλίσης είναι 135° , δηλαδή η ακμή έχει βορειοανατολική και νοτιοδυτική κατεύθυνση, το σημείο θεωρείται ότι βρίσκεται στην ακμή εάν το μέγεθος της κλίσης είναι μεγαλύτερο από τα μεγέθη στα εικονοστοιχεία που βρίσκονται στις κατευθύνσεις βορειοδυτικά και νοτιοανατολικά. Τέλος, στην περίπτωση που η στρογγυλοποιημένη γωνία κλίσης είναι 45° , δηλαδή η ακμή έχει βορειοδυτική -

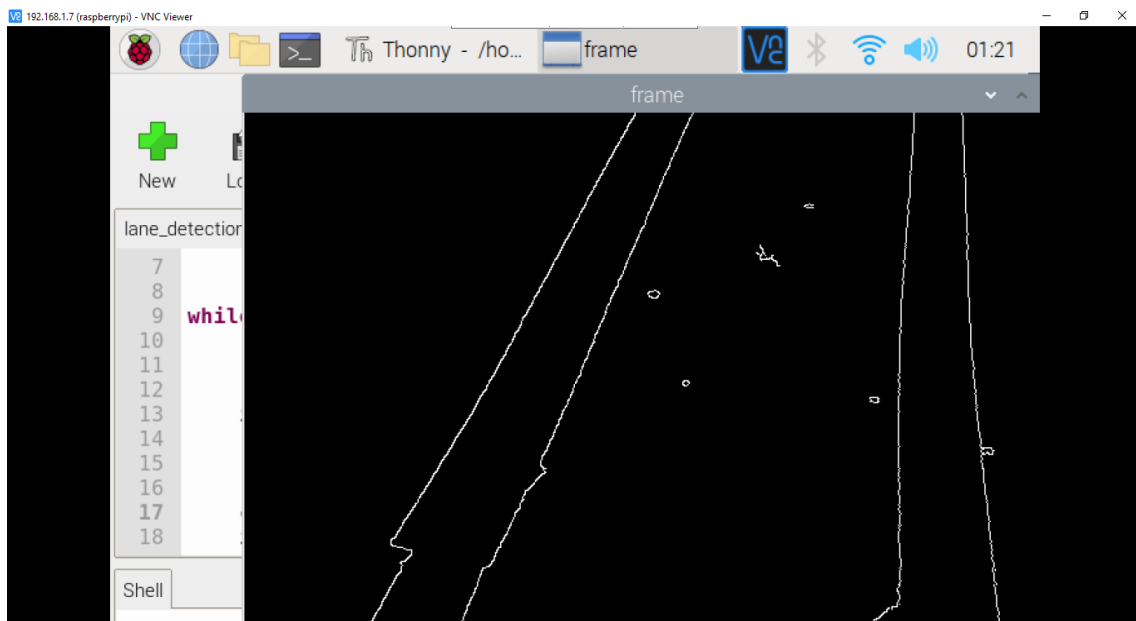
νοτιοανατολική κατεύθυνση, το σημείο θεωρείται ότι βρίσκεται στην ακμή εάν το μέγεθος της κλίσης είναι μεγαλύτερο από τα μεγέθη στα εικονοστοιχεία με βορειοανατολικές και νοτιοδυτικές κατευθύνσεις. Σε πιο ακριβείς υλοποιήσεις χρησιμοποιείται γραμμική παρεμβολή μεταξύ των δύο γειτονικών εικονοστοιχείων που διασχίζουν την κατεύθυνση της κλίσης. Η γραμμική παρεμβολή στα μαθηματικά είναι μία μέθοδος προσαρμογής καμπύλης χρησιμοποιώντας γραμμικά πολυώνυμα (δηλαδή πολυώνυμα πρώτου βαθμού) για την κατασκευή νέων σημείων δεδομένων εντός του εύρους ενός διακριτού συνόλου γνωστών σημείων δεδομένων. Επί παραδείγματι, όταν η γωνία κλίσης είναι μεταξύ 89° και 180° , η παρεμβολή μεταξύ των κλίσεων στα βόρεια και βορειοανατολικά εικονοστοιχεία θα δώσει μία τιμή παρεμβολής και η παρεμβολή μεταξύ των νότιων και νοτιοδυτικών εικονοστοιχείων θα δώσει την άλλη. Το μέγεθος της κλίσης στο κεντρικό εικονοστοιχείο πρέπει να είναι μεγαλύτερο και από τα δύο για να επισημανθεί ως ακμή. Να σημειωθεί ότι το πρόσημο της κατεύθυνσης είναι άσχετο, δηλαδή Βορράς - Νότος είναι το ίδιο με Νότος - Βορράς και ούτω καθεξής.

- Εφαρμογή διπλού ορίου για τον περιορισμό των ακμών. Μετά την εφαρμογή της μη μέγιστης συγκράτησης, τα εναπομείναντα εικονοστοιχεία ακμών παρέχουν μία πιο ακριβή αναπαράσταση των πραγματικών ακμών σε μία εικόνα. Εν τούτοις, ορισμένα εικονοστοιχεία που προκαλούν θόρυβο και χρωματικές διαφορές παραμένουν. Για να εξηγηθούν αυτές οι ψευδείς αντιδράσεις, είναι απαραίτητο να φιλτράρονται οι ακμές των εικονοστοιχείων με χαμηλή τιμή κλίσης και να διατηρούνται οι ακμές των εικονοστοιχείων με υψηλή τιμή κλίσης. Αυτό επιτυγχάνεται με την συλλογή υψηλών και χαμηλών τιμών ορίου. Εάν η τιμή κλίσης της ακμής ενός εικονοστοιχείου είναι υψηλότερη από την τιμή του ορίου, επισημαίνεται ως εικονοστοιχείο ισχυρής ακμής. Εάν η τιμή κλίσης της ακμής ενός εικονοστοιχείου είναι μικρότερη από την τιμή του υψηλού ορίου και μεγαλύτερη από την τιμή του χαμηλού ορίου, επισημαίνεται ως εικονοστοιχείο ασθενούς ακμής. Εάν η τιμή κλίσης της ακμής ενός εικονοστοιχείου είναι μικρότερη από την τιμή του χαμηλού ορίου, θα καταργηθεί. Οι τιμές του διπλού ορίου καθορίζονται εμπειρικά και ο ορισμός τους εξαρτάται από περιεκτικότητα (περιεχόμενο content) της εικόνας εισόδου.
- Ανίχνευση ακμών με την χρήση της υστέρησης. Με αυτό το βήμα ολοκληρώνεται η διαδικασία ανίχνευσης ακμών, αποκρύπτοντας όλες τις περιττές ακμές που είναι αδύναμες και δεν συνδέονται με τις ισχυρές. Οι ισχυρές ακμές των εικονοστοιχείων θα πρέπει σίγουρα να βρίσκονται στην ακμή της τελικής εικόνας, εφόσον είναι απόρροια πραγματικών ακμών της εικόνας. Ωστόσο, υπάρχουν και οι αδύναμες ακμές των εικονοστοιχείων που μπορεί να έχουν εξαχθεί από πραγματική ακμή ή από διακυμάνσεις χρώματος ή θορύβου, αυτές οι αδύναμες ακμές θα πρέπει να αφαιρεθούν ώστε να επιτευχθεί ένα ακριβές αποτέλεσμα. Είναι συνηθέστερο το εικονοστοιχείο με ασθενής ακμή, η οποία έχει προκληθεί από πραγματικές ακμές, να συνδέεται με ένα εικονοστοιχείο ισχυρής ακμής, ενώ οι αντανάκλασεις θορύβου να μην συνδέονται. Για την παρακολούθηση της σύνδεσης των ακμών, εφαρμόζεται η ανάλυση κηλίδων (blob analysis) αναλύοντας το εικονοστοιχείο μιας αδύναμης ακμής και τα οχτώ συνδεδεμένα γειτονικά εικονοστοιχεία. Οι κηλίδες είναι ομάδες εικονοστοιχείων σε μία εικόνα με ορισμένες κοινές ιδιότητες, για παράδειγμα τιμή στην κλίμακα του γκρι. Οποιαδήποτε ανάλυση κηλίδας ξεκινά με τον εντοπισμό αυτών των περιοχών στην εικόνα και χρησιμοποιεί εργαλεία που ονομάζονται ανιχνευτές κηλίδων. Ο ανιχνευτής κηλίδων λειτουργεί μετατρέποντας την εικόνα σε ένα σύνολο δυαδικών εικόνων. Όταν η εικόνα είναι σε κλίμακα του γκρι, η διαδικασία είναι απλά να τεθεί ένα όριο στην εικόνα, όπως ακριβώς γίνεται στην παρούσα εργασία. Στην περίπτωση που η εικόνα είναι έγχρωμη κάθε κανάλι θα έχει το δικό του όριο ανεξάρτητα. Στο σύνολο των εικόνων, κάθε μία από αυτές αντιπροσωπεύει μια λειτουργία κατωφλιού σε διαφορετικό όριο. Οι συνδεδεμένες περιοχές προσδιορίζονται σε κάθε δυαδική εικόνα συνόλου. Οι συντεταγμένες κηλίδων προσδιορίζονται και ομαδοποιούνται σύμφωνα με το εύρος του ορίου και όταν εντοπιστούν οι κηλίδες, μπορούν να αναλυθούν με τον υπολογισμό βαθμωτών μεγεθών όπως περιοχή, περίμετρο, κυρτότητα, κυκλικότητα και πόσο συμπαγές είναι. Εφόσον, λοιπόν, υπάρχει ένα εικονοστοιχείο με ισχυρή ακμή που εμπλέκεται στην κηλίδα, αυτό το αδύναμο σημείο ακμής μπορεί να αναγνωρισθεί ως αυτό που πρέπει να διατηρηθεί.

Στην συγκεκριμένη φάση θα πρέπει να ειπωθεί ότι η ανίχνευση λωρίδας Canny αποτελεί βασικό κομμάτι της παρούσας εργασίας. Ωστόσο, ο παραδοσιακός αυτός αλγόριθμος έχει και κάποια ελαττώματα που παρατηρούνται σε πιο απαιτητικές εργασίες ανίχνευσης ακμών. Τα ελαττώματα αυτά συνοψίζονται ως εξής :

1. Το φίλτρο του Gauss που εφαρμόζεται για την εξομάλυνση του θορύβου, θα εξομαλύνει και την ακμή, η οποία θεωρείται ως το χαρακτηριστικό της υψηλής συχνότητας. Αυτό θα έχει ως αποτέλεσμα την πιθανότητα απώλειας αδύναμων ακμών και την εμφάνιση μεμονωμένων ακμών στο αποτέλεσμα.
2. Για τον υπολογισμό του πλάτους της κλίσης, ο παραδοσιακός αλγόριθμος ανίχνευσης ακμών Canny χρησιμοποιεί το κέντρο σε ένα 2x2 παράθυρο για να υπολογίσει την πεπερασμένη διαφορά της μέσης τιμής που αναπαριστά το πλάτος της κλίσης. Αυτή η μέθοδος είναι ευαίσθητη στον θόρυβο και μπορεί εύκολα να ανιχνεύσει ψευδείς ακμές και να χάσει τις πραγματικές.
3. Στον παραδοσιακό αλγόριθμο ακμών Canny, θα υπάρχουν δύο σταθερές καθολικές τιμές ορίου για την αφαίρεση των ψευδών ακμών. Όμως, όταν η εικόνα γίνεται πολύπλοκη, οι διαφορετικές τοπικές περιοχές θα χρειαστούν πολύ διαφορετικές τιμές ορίου ώστε να εντοπισθούν με ακρίβεια οι πραγματικές ακμές. Επιπλέον, στην παραδοσιακή μέθοδο οι καθολικές τιμές ορίου προσδιορίζονται χειροκίνητα μέσω πειραμάτων, αυτό οδηγεί σε πολύπλοκους υπολογισμούς στην περίπτωση χειρισμού μεγάλου αριθμού διαφορετικών εικόνων.
4. Τέλος το αποτέλεσμα της παραδοσιακής ανίχνευσης δεν μπορεί να φτάσει σε ικανοποιητικά υψηλή ακρίβεια μίας αντανάκλασης για κάθε ακμή πολλαπλών σημείων που θα εμφανιστεί.

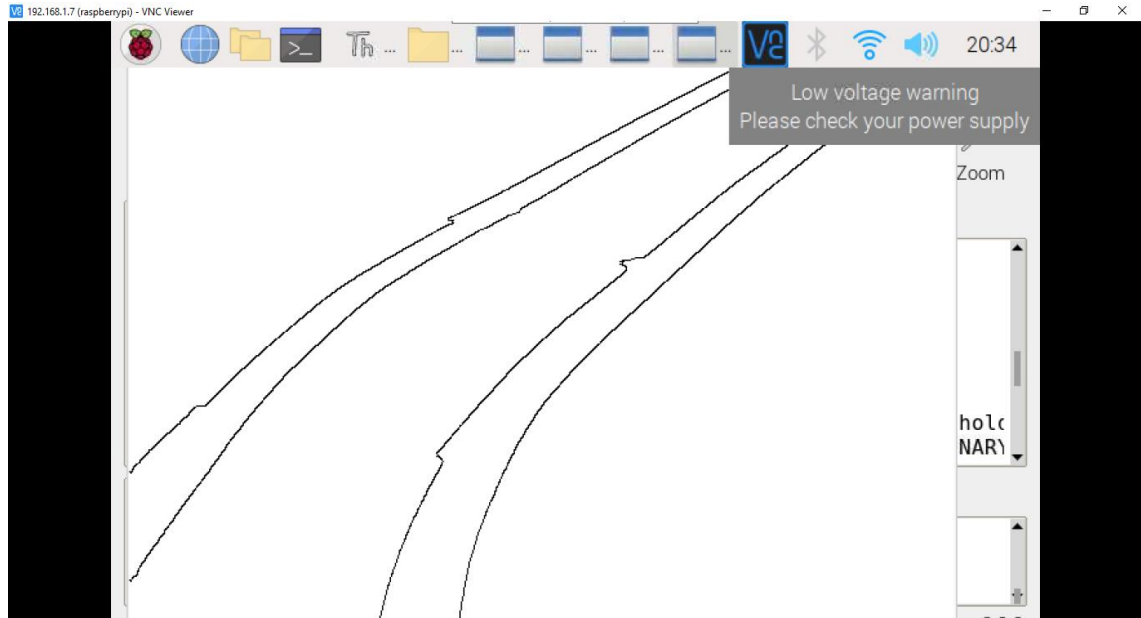
Η εικόνα όπως



- Πέμπτο βήμα :

Περιοχή ενδιαφέροντος. Σε αυτό το βήμα γίνεται ο υπολογισμός μόνο της περιοχής που καλύπτεται από την λωρίδα του δρόμου. Δημιουργείται μία μάσκα που έχει την ίδια

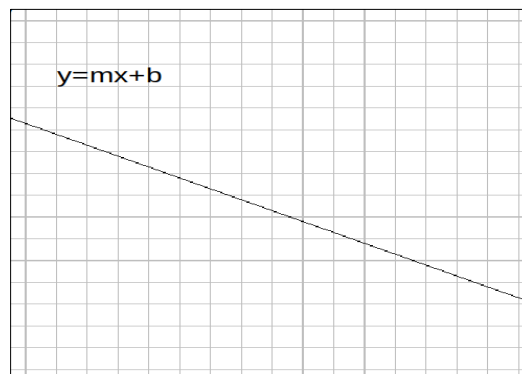
διάσταση με αυτή της εικόνας του δρόμου που μας ενδιαφέρει και εκτελείται ο δυαδικός τελεστής AND μεταξύ κάθε εικονοστοιχείου της εικόνας ,που έχει παραχθεί μετά την ανίχνευση ακμών Canny, και αυτής της μάσκας. Τελικά η μάσκα καλύπτει την εικόνα Canny και δείχνει την περιοχή ενδιαφέροντος που εντοπίζεται από το πολυγωνικό περίγραμμα της μάσκας. (Wikipedia, n.d.). Στην συγκεκριμένη περίπτωση όμως



- Έκτο βήμα :

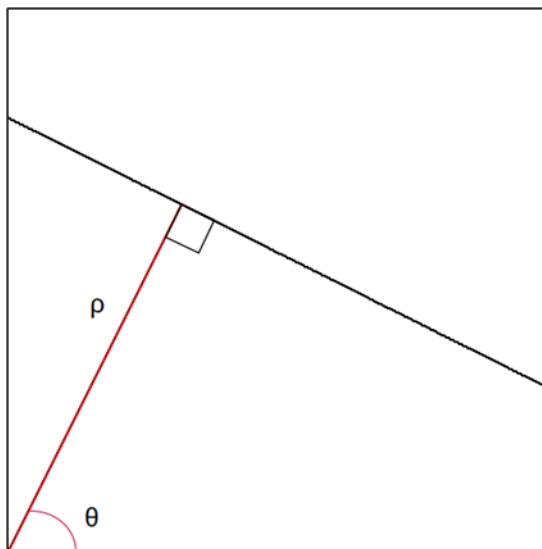
Μετασχηματισμός Hough.

Σκοπός του μετασχηματισμού Hough είναι η εύρεση στιγμιότυπων μιας κλάσης σχημάτων (όπως κύκλος ή ευθεία γραμμή) μέσα από μια διαδικασία ψηφοφορίας. Η ανάγκη για τον μετασχηματισμό αυτόν προκύπτει λόγω των ατελειών που μπορεί να παρουσιάζει μια ψηφιακή εικόνα. Για παράδειγμα, μπορεί να απουσιάζουν pixels από την εικόνα καθιστώντας δυσδιάκριτο ένα σχήμα ή μπορεί να εισαχθεί θόρυβος με την εφαρμογή ενός αλγορίθμου ανίχνευσης ακμών (σ.σ.. η "ανίχνευση ακμών" αποτελεί ένα σύνολο τεχνικών που επιτρέπουν την ανίχνευση απότομων αλλαγών στην φωτεινότητα μιας εικόνας). Η εφαρμογή του μετασχηματισμού Hough για την ανίχνευση μιας ευθείας γραμμής σε μια εικόνα αποτελεί την απλούστερη περίπτωση είναι όμως διαφωτιστική όσον αφορά τις βασικές αρχές του αλγορίθμου. Ας υποθέσουμε ότι έχουμε μια ευθεία γραμμή που περιγράφεται από την εξίσωση $y=mx+b$



Εικόνα 37. Ευθεία γραμμή εξίσωσης $y=mx+b$.

Αν οι παράμετροι m , b λάβουν σταθερές τιμές, π.χ. $y=2x+5$, τότε έχουμε έναν πλήρη χαρακτηρισμό της γραμμής, δηλαδή μπορούμε να βρούμε κάθε σημείο της. Αυτό σημαίνει πως μπορούμε να "συμπυκνώσουμε" μια ευθεία γραμμή σε ένα ζεύγος της μορφής (m, b) . Το πρόβλημα αυτής της μεθόδου συνίσταται στο αδυνατούμε να περιγράψουμε γραμμές κάθετες στον άξονα των x καθώς η κλίση m απειρίζεται. Για να ξεπεράσουμε αυτό το εμπόδιο, μπορούμε να χρησιμοποιήσουμε πολικές αντί για καρτεσιανές συντεταγμένες ώστε να περιγράψουμε την ζητούμενη ευθεία μέσα από το μοναδικό ευθύγραμμο τμήμα που ξεκινά από την αρχή των αξόνων, καταλήγει στην ζητούμενη ευθεία και είναι κάθετο σε αυτήν



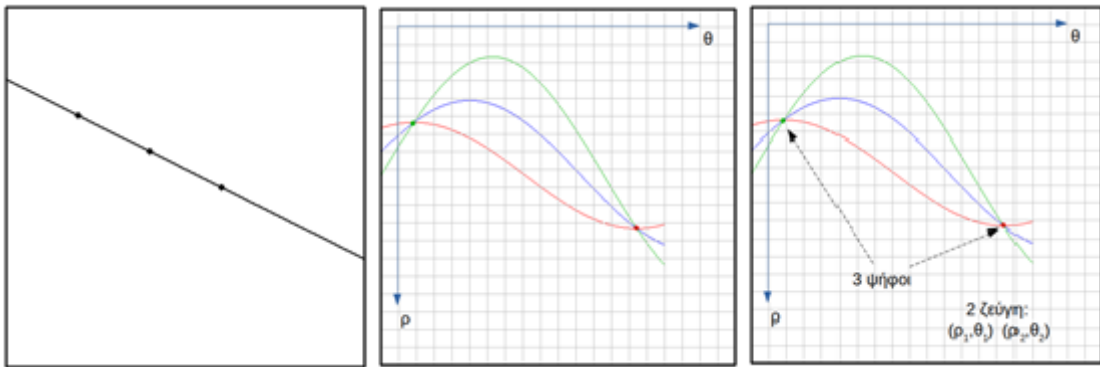
Εικόνα 38. Αναπαράσταση Hesse.

Η παραπάνω αναπαράσταση για μια ευθεία γραμμή είναι γνωστή και ως κανονική μορφή ή αναπαράσταση Hesse. Με την κανονική μορφή μπορούμε να περιγράψουμε πλήρως μια ευθεία στο επίπεδο μέσα από ένα ζεύγος παραμέτρων (ρ, θ) , ισοδύναμο με το ζεύγος (m, b) . Ο τύπος που μας δίνει την αρχική ευθεία είναι πλέον $x \cdot \cos\theta + y \cdot \sin\theta = \rho$, περιγράφοντας έτσι την γραμμή που διέρχεται από το σημείο (x, y) του Καρτεσιανού επιπέδου και είναι κάθετη στην ημιευθεία με αρχή το σημείο $(0,0)$ και γωνία θ . Το σύνολο των ζευγών (ρ, θ) συνθέτουν τον λεγόμενο χώρο Hough.

Η ουσία του μετασχηματισμού συνίσταται στην κατασκευή ενός δισδιάστατου πίνακα που παίζει το ρόλο του χώρου Hough αλλά με διακριτές τιμές μέσα σε ένα συγκεκριμένο πεδίο τιμών για τα ρ και θ . Η ανάγκη για έναν τέτοιο περιορισμό είναι προφανής απόρροια της πεπερασμένης μνήμης που έχει ένας υπολογιστής. Ακόμα, μια ευθεία σε μια ψηφιακή εικόνα μπορεί να θεωρηθεί ως ένα σύνολο από φωτεινά σημεία. Αυτός είναι και ο λόγος που ο μετασχηματισμός Hough εφαρμόζεται για ευκολία μετά από κάποια τεχνική ανίχνευσης ακμών όπως η Canny edge detection. Επιπλέον, οι παράμετροι που αφορούν την κατασκευή του πίνακα και τα διαθέσιμα (ρ, θ) καθορίζουν την ταχύτητα και την ακρίβεια με την οποία ανιχνεύονται οι ζητούμενες ευθείες, γι' αυτό και χρειάζεται προσοχή κατά την επιλογή τους και μερικές δοκιμές μέχρις ότου καταλήξουμε στην παραμετροποίηση του αλγορίθμου με την καλύτερη απόδοση.

Ακολουθεί ένα απλό αλλά πλήρες παράδειγμα όπου εφαρμόζουμε τα βήματα του μετασχηματισμού για την εύρεση των παραμέτρων μιας ευθείας σε μια δοσμένη εικόνα:

Έστω ότι μας δίνεται η παρακάτω εικόνα, στην οποία θέλουμε να βρούμε αν υπάρχουν ευθείες και με ποια ζεύγη παραμέτρων θα μπορούσαμε να τις περιγράψουμε. Για ευκολία έχουμε ήδη σημειώσει τρία σημεία της για τα οποία θα εφαρμόσουμε τον μετασχηματισμό.



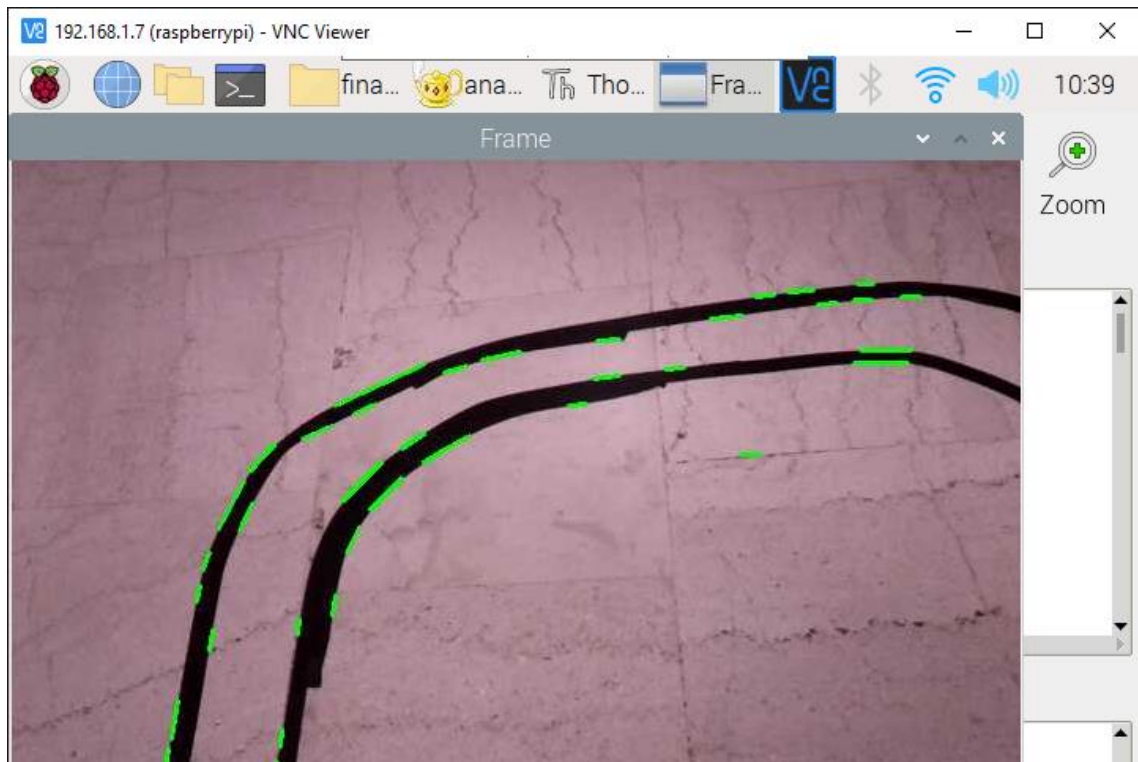
Εικόνα 39. Περιγραφή ευθείας σε πολικές συντεταγμένες.

Αρχικά κατασκευάζουμε τον δισδιάστατο πίνακα όπου θα σημειώνουμε τα (ρ, θ) που προκύπτουν από την λύση της εξίσωσης $x \cdot \cos\theta + y \cdot \sin\theta = \rho$ όπου το ζεύγος (x, y) είναι γνωστό. Για να λύσουμε την προηγούμενη εξίσωση, επιλέγουμε διαδοχικές τιμές για το θ και λύνουμε ως προς ρ . Τα σημεία που προκύπτουν από τις διαδοχικές λύσεις της εξίσωσης σχηματίζουν μια ημιτονοειδή καμπύλη ενώ οι καμπύλες που προκύπτουν από συνευθειακά σημεία έχουν σημεία τομής. Με αυτόν τον τρόπο μετατρέπουμε το πρόβλημα εύρεσης συνευθειακών σημείων σε πρόβλημα εύρεσης τοπικών μεγίστων σε έναν δισδιάστατο πίνακα. Στο σχήμα φαίνονται τα σημεία τομής των καμπυλών για τα συνευθειακά σημεία της αρχικής εικόνας και από αυτά μπορούμε να εξάγουμε 2 ζεύγη παραμέτρων (ρ, θ) που περιγράφουν την ευθεία σε πολικές συντεταγμένες.

Αναφέρονται ενδεικτικά οι κυριότερες παράμετροι που επηρεάζουν την απόδοση του μετασχηματισμού Hough:

1. διαβάθμιση του ρ : ελάχιστη μεταβολή της απόστασης (μετριέται σε pixels)
2. διαβάθμιση του θ : ελάχιστη μεταβολή της γωνίας (μετριέται σε radians)
3. όριο ψήφων: το ελάχιστο πλήθος ψήφων ώστε ένα (ρ, θ) να θεωρηθεί ως περιγραφή ευθείας
4. ελάχιστο μήκος ευθείας: κάθε ευθεία με μήκος μικρότερο από μια δοσμένη τιμή απορρίπτεται
5. μέγιστο κενό μεταξύ τμημάτων μιας ευθείας: επιτρέπει να "ενώσουμε" τμήματα που ανήκουν στην ίδια ευθεία

Μια σημαντική παρατήρηση είναι και το γεγονός ότι δεν είναι γνωστός κάποιος μαθηματικός τύπος που να μας δίνει την ιδανική παραμετροποίηση a priori, παρά μόνο εμπειρικές προσεγγίσεις που δουλεύουν καλύτερα για το εκάστοτε πρόβλημα. (madmortigan, n.d.) (Nayar, n.d.) (en.wikipedia.org, n.d.).

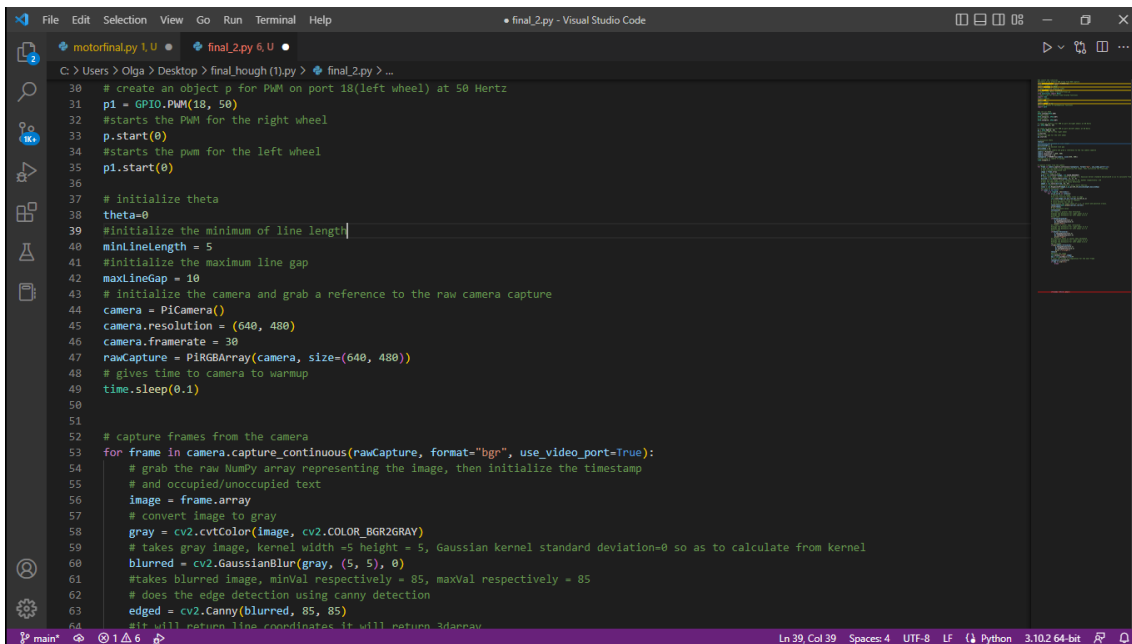


5.1.1 Αλγόριθμος ανίχνευσης λωρίδας σε Python

```

File Edit Selection View Go Run Terminal Help
final_2.py - Visual Studio Code
motorfinal.py 1 U • final_2.py 5 U X
C:\Users\Olga\Desktop> final_hough (1).py > final_2.py > ...
1 #we insert the libraries
2 #picamera.array produce RGB array from RGB capture
3 from picamera.array import PiRGBArray
4 #import raspberry's GPIO
5 import RPi.GPIO as GPIO
6 # import picamera captures video
7 from picamera import PiCamera
8 # import class Motor from motorfinal.py
9 from motorfinal import Motor
10 # import time for various time-related functions
11 import time
12 #import opencv
13 import cv2
14 #import numpy
15 import numpy as np
16 #provides access to mathematical functions
17 import math
18
19
20 #we set our GPIO
21 GPIO.setmode(GPIO.BCM)
22 # right wheel
23 GPIO.setup(14, GPIO.OUT)
24 # left wheel
25 GPIO.setup(18, GPIO.OUT)
26
27 # create an object p for PWM on port 14(right wheel) at 50 Hertz
28 p = GPIO.PWM(14, 50)
29
30 # create an object p1 for PWM on port 18(left wheel) at 50 Hertz
31 p1 = GPIO.PWM(18, 50)
32 #starts the PWM for the right wheel
33 p.start(0)
34 #starts the pwm for the left wheel
35 p1.start(0)

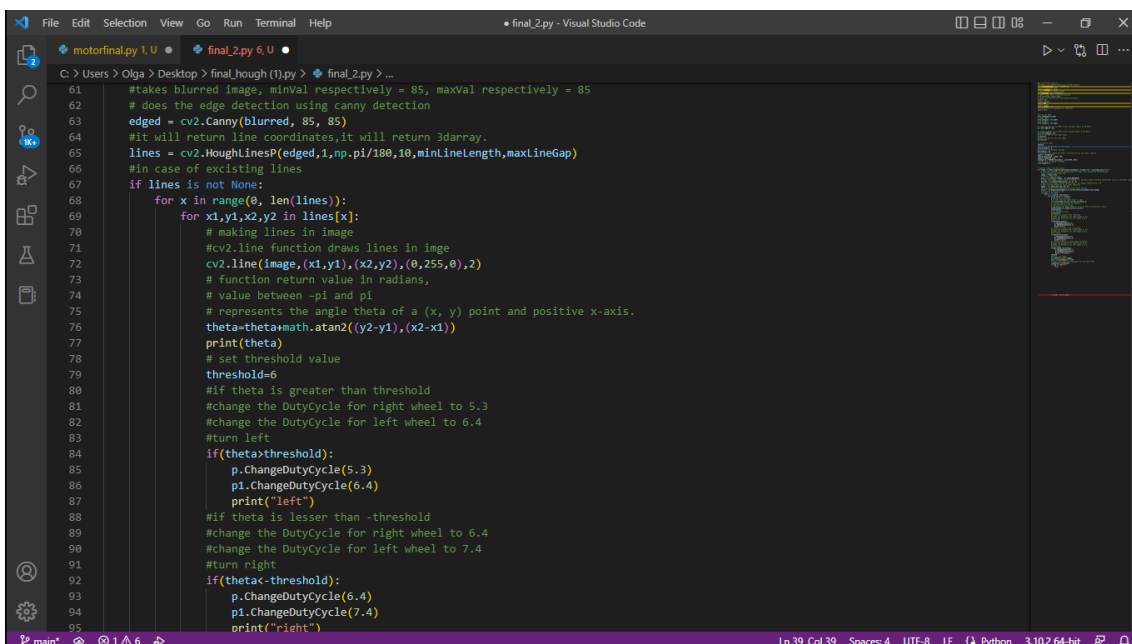
```



```

File Edit Selection View Go Run Terminal Help
motorfinal.py 1,U final_2.py 6,U
C:\Users\Olga\Desktop>final_hough(1).py > final_2.py > ...
30 # create an object p for PWM on port 18(left wheel) at 50 Hertz
31 p1 = GPIO.PWM(18, 50)
32 #starts the PWM for the right wheel
33 p.start(0)
34 #starts the pwm for the left wheel
35 p1.start(0)
36
37 # initialize theta
38 theta=0
39 #initialize the minimum of line length
40 minlinelength = 5
41 #initialize the maximum line gap
42 maxlineGap = 10
43 # initialize the camera and grab a reference to the raw camera capture
44 camera = PiCamera()
45 camera.resolution = (640, 480)
46 camera.framerate = 30
47 rawCapture = PiRGBArray(camera, size=(640, 480))
48 # gives time to camera to warmup
49 time.sleep(0.1)
50
51
52 # capture frames from the camera
53 for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
54     # grab the raw NumPy array representing the image, then initialize the timestamp
55     # and occupied/unoccupied text
56     image = frame.array
57     # convert image to gray
58     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
59     # takes gray image, kernel width =5 height = 5, Gaussian kernel standard deviation=0 so as to calculate from kernel
60     blurred = cv2.GaussianBlur(gray, (5, 5), 0)
61     #takes blurred image, minVal respectively = 85, maxVal respectively = 85
62     # does the edge detection using canny detection
63     edged = cv2.Canny(blurred, 85, 85)
64     #it will return line coordinates, it will return 3darray

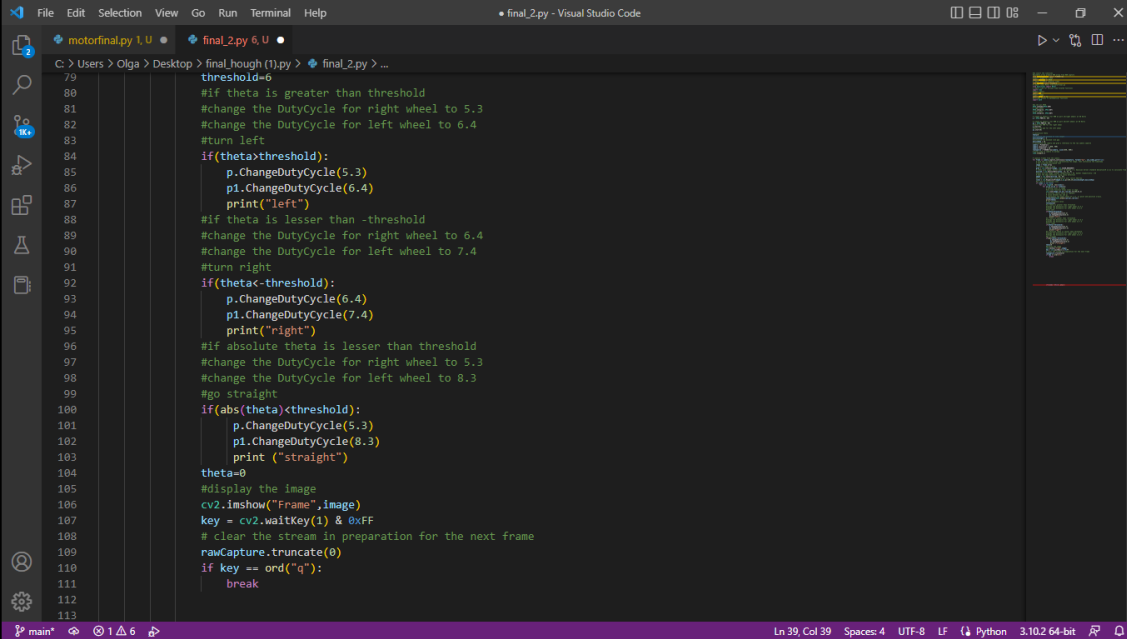
```



```

File Edit Selection View Go Run Terminal Help
motorfinal.py 1,U final_2.py 6,U
C:\Users\Olga\Desktop>final_hough(1).py > final_2.py > ...
61 #takes blurred image, minVal respectively = 85, maxVal respectively = 85
62 # does the edge detection using canny detection
63 edged = cv2.Canny(blurred, 85, 85)
64 #it will return line coordinates, it will return 3darray.
65 lines = cv2.HoughLinesP(edged,1,np.pi/180,10,minLineLength,maxLineGap)
66 #in case of existing lines
67 if lines is not None:
68     for x in range(0, len(lines)):
69         for x1,y1,x2,y2 in lines[x]:
70             # making lines in image
71             #cv2.line function draws lines in image
72             cv2.line(image, (x1,y1), (x2,y2), (0,255,0), 2)
73             # function return value in radians,
74             # value between -pi and pi
75             # represents the angle theta of a (x, y) point and positive x-axis.
76             theta=theta+math.atan2((y2-y1),(x2-x1))
77             print(theta)
78             # set threshold value
79             threshold=6
80             #if theta is greater than threshold
81             #change the DutyCycle for right wheel to 5.3
82             #change the DutyCycle for left wheel to 6.4
83             #turn left
84             if(theta>threshold):
85                 p.ChangeDutyCycle(5.3)
86                 p1.ChangeDutyCycle(6.4)
87                 print("left")
88             #if theta is lesser than -threshold
89             #change the DutyCycle for right wheel to 6.4
90             #change the DutyCycle for left wheel to 7.4
91             #turn right
92             if(theta<-threshold):
93                 p.ChangeDutyCycle(6.4)
94                 p1.ChangeDutyCycle(7.4)
95                 print("right")

```



```
File Edit Selection View Go Run Terminal Help • final_2.py - Visual Studio Code
motorfinal.py 1, U • final_2.py 6, U
C:\Users\Olga\Desktop> final_rough (1).py > final_2.py > ...
79 threshold=6
80 #if theta is greater than threshold
81 #change the DutyCycle for right wheel to 5.3
82 #change the DutyCycle for left wheel to 6.4
83 #turn left
84 if (theta>threshold):
85     p.ChangeDutyCycle(5.3)
86     p1.ChangeDutyCycle(6.4)
87     print("left")
88 #if theta is lesser than -threshold
89 #change the DutyCycle for right wheel to 6.4
90 #change the DutyCycle for left wheel to 7.4
91 #turn right
92 if (theta<-threshold):
93     p.ChangeDutyCycle(6.4)
94     p1.ChangeDutyCycle(7.4)
95     print("right")
96 #if absolute theta is lesser than threshold
97 #change the DutyCycle for right wheel to 5.3
98 #change the DutyCycle for left wheel to 8.3
99 #go straight
100 if (abs(theta)<threshold):
101     p.ChangeDutyCycle(5.3)
102     p1.ChangeDutyCycle(8.3)
103     print ("straight")
104 theta=0
105 #display the image
106 cv2.imshow("Frame",image)
107 key = cv2.waitKey(1) & 0xFF
108 # clear the stream in preparation for the next frame
109 rawCapture.truncate(0)
110 if key == ord("q"):
111     break
112
113
main* 1 6 Ln 39, Col 39 Spaces: 4 UTF-8 LF Python 3.10.2 64-bit
```

(abhinav, χ.χ.)

Για τους σερβοκινητήρες ο κώδικας σε python είναι:

```

1
2 # imports Rpi.GPIO module
3 import RPi.GPIO as GPIO
4 #have a delay
5 from time import sleep
6 #choose BCM
7 GPIO.setmode(GPIO.BCM)
8 #disable warnings, GPIO already in use
9 GPIO.setwarnings(False)
10
11 # create a class to initialize
12 # set up GPIO
13 #to create object for PWM
14 #on port that will take from fina_2.py
15 #at 50 Hz
16 class Motor():
17     def __init__(self,EnaA,EnaB):
18         self.EnaA = EnaA
19         self.EnaB = EnaB
20         GPIO.setup(self.EnaA,GPIO.OUT)
21         GPIO.setup(self.EnaB,GPIO.OUT)
22         self.pwmA = GPIO.PWM(self.EnaA, 50)
23         self.pwmA.start(0)
24         self.pwmB = GPIO.PWM(self.EnaB, 50)
25         self.pwmB.start(0)
26
27     def move(self, speed=0, turn=0, t=0):
28         # We scale the input speed from 0:10 to a range required from the pwm module
29         # We scale the input turn from -10:10 to a range of -1:1
30         # Adjust default calculation so that when speeds are equal the robot moves forward
31         if speed<10:
32             speed=10
33         elif speed>10:
34             speed= 0
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

35
36
37     if turn <-10:
38         turn=-10
39     elif turn >10:
40         turn=10
41
42     leftspeed =6.4 - (speed/10)
43     rightspeed =7.3 + (speed/10)
44
45     turn /= 10
46     leftspeed -= turn
47     rightspeed -= turn
48
49     if leftspeed <5.4: leftspeed = 5.4
50     elif leftspeed > 6.4: leftspeed = 6.4
51
52     if rightspeed <7.3: rightspeed = 7.3
53     elif rightspeed > 8.3: rightspeed = 8.3
54
55
56
57     self.pwmA.ChangeDutyCycle(leftspeed)
58     self.pwmB.ChangeDutyCycle(rightspeed)
59
60     sleep(t)
61     #create a function to stop robot move
62     def stop(self,t=0):
63         self.pwmA.ChangeDutyCycle(0)
64         self.pwmB.ChangeDutyCycle(0)
65         sleep(t)
66
67     #we create a main so as to check if servo work correctly
68     def main():
69         motor1.move(5,0,0)
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

```

61 #create a function to stop robot move
62 def stop(self,t=0):
63     self.pwmA.ChangeDutyCycle(0)
64     self.pwmB.ChangeDutyCycle(0)
65     sleep(t)
66
67 #we create a main so as to check if servo work correctly
68 def main():
69     motor1.move(5,0,0)
70     #motor1.stop(0)
71
72     #motor1 takes the gpio port
73     #makes a loop and run
74 if __name__ == '__main__':
75     motor1 = Motor(14,18)
76     rep = 1
77     while rep<10000:
78         main()
79         rep+=1
80
81
82
83
84
85

```

5.2 Εναλλακτικός Αλγόριθμος

Ο εναλλακτικός αλγόριθμος που χρησιμοποιήθηκε είναι η παρακολούθηση συνόρων που αποτελεί μία από τις θεμελιώδεις τεχνικές για την επεξεργασία ψηφιοποιημένων δυαδικών εικόνων. Επί της ουσίας, εξάγει μία ακολουθία συντεταγμένων μεταξύ μίας συνδεδεμένης συνιστώσας 1-εικονοστοιχείων και από ένα συνδεδεμένο 0-εικονοστοιχείο. Η πληροφορία που εξάγεται είναι η σχέση του περιβάλλοντος μεταξύ των δύο περιγραμμάτων, του εξωτερικού περιγράμματος και του περιγράμματος των οπών.

Εξωτερικό περίγραμμα ορίζεται ως το σύνολο των σημείων περιγράμματος σε ένα τυχαίο 1-στοιχείο και 0-στοιχείο τα οποία το περιβάλλουν άμεσα. Με τον ίδιο τρόπο ορίζονται το περίγραμμα οπής για τα σημεία περιγράμματος μεταξύ μίας οπής και του 1-στοιχείου, το οποίο περιβάλλει άμεσα το περίγραμμα οπής. Να σημειωθεί ότι για το τυχαίο 1-στοιχείο μιας δυαδικής εικόνας κάθε εξωτερικό περίγραμμα είναι ένα και μοναδικό. Το ίδιο ισχύει και για την οπή, κάθε περίγραμμα οπής είναι επίσης μοναδικό.

Εάν υπάρχει αντιστοιχία ένα προς ένα μεταξύ του εξωτερικού περιγράμματος και ενός 1-στοιχείου, και μεταξύ των περιγραμμάτων οπής και ενός 0-στοιχείου, τότε μπορεί να καθοριστεί η τυπολογική δομή μίας δεδομένης δυαδικής εικόνας. Αυτό ο αλγόριθμος αποτελεί μία εκτεταμένη έκδοση του αλγορίθμου ακολουθίας συνόρων, που στην ουσία ξεχωρίζει τα εξωτερικά περιγράμματα από τα περιγράμματα οπών. Με αυτό τον αλγόριθμο

1. μπορεί να τεθεί ένα μοναδικό σημάδι που θα ακολουθείται πάνω σε κάθε περίγραμμα αντί του να υιοθετείται η ίδια διαδικασία για κάθε περίγραμμα ξεχωριστά.
2. και να προστεθεί μία διαδικασία απόκτησης γονικού περιγράμματος που θα διεξαχθεί από τρέχον περίγραμμα. Γονικό περίγραμμα ενός εξωτερικού περιγράμματος μεταξύ ενός 1-στοιχείου S_1 και το 0-στοιχείο S_2 , το οποίο περιβάλλει το S , ορίζεται ως :
 - Το περίγραμμα οπής μεταξύ του S_2 και του 1-στοιχείου που περιβάλλει το S_2 άμεσα, εφόσον το S_2 είναι οπή.
 - Τα εικονοστοιχεία της εικόνας , εφόσον S_2 είναι το φόντο.

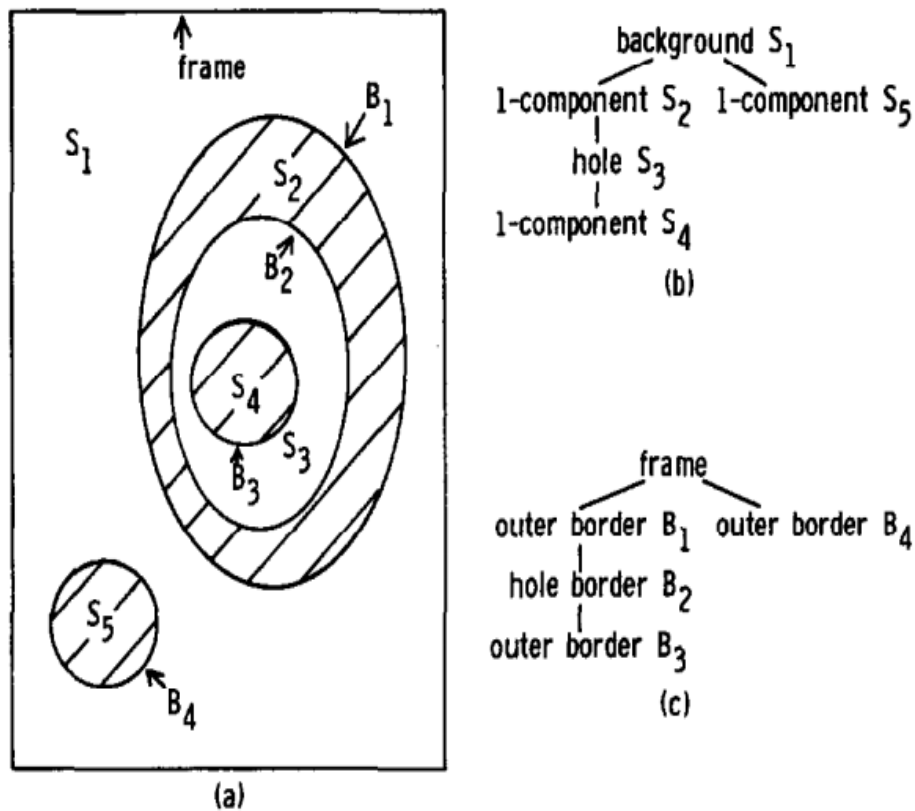
Τα εικονοστοιχεία της δυαδικής εικόνας αποτελούνται από την γραμμή (uppermost row), την κάτω γραμμή (lowermost row), την αριστερή στήλη (leftmost column), τη δεξιά στήλη Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

(rightmost column). Τα εικονοστοιχεία με πυκνότητες 0 και 1, ονομάζονται 0-εικονοστοιχεία και 1-εικονοστοιχεία. Υποτίθεται ότι τα 0-εικονοστοιχεία γεμίζουν το πλαίσιο μίας δυαδικής εικόνας και επίσης μπορεί να εκχωρηθεί οποιαδήποτε ακέραια τιμή σε ένα εικονοστοιχείο κατά την διάρκεια της επεξεργασίας.

Στη συνέχεια το εικονοστοιχείο που βρίσκεται στην i -οστή σειρά και η j -οστή στήλη θα αντιπροσωπευτεί από τον αριθμό της σειράς και τον αριθμό της στήλης (i, j) και ακολουθείται το σύστημα συντεταγμένων με τον αριθμό i να αυξάνεται από πάνω προς το κάτω μέρος και αντίστοιχα ο αριθμός στήλης j αυξάνεται από αριστερά προς τα δεξιά, έτσι η εικόνα θα έχει τιμή πυκνότητας f_{ij} και ορίζεται ως $F = f\{i j\}$. Ένα 1-στοιχείο και ένα 0-στοιχείο, τα συνδετικά στοιχεία των 1-εικονοστοιχείων και 0-εικονοστοιχείων αντίστοιχα. Εάν ένα 0-στοιχείο S περιέχει το πλαίσιο (frame) της εικόνας, τότε το S ορίζεται ως φόντο, διαφορετικά ορίζεται ως σπή.

Περιβάλλον μεταξύ συνδεδεμένων στοιχείων, για δύο δοθέντα συνδεδεμένα στοιχεία S_1 και S_2 σε μία δυαδική εικόνα, εάν υπάρχει ένα εικονοστοιχείο που να ανήκει στο S_2 για οποιοδήποτε 4-μονοπάτι από ένα εικονοστοιχείο που ανήκει στο S_1 για ένα εικονοστοιχείο του πλαισίου, θεωρείται ότι το S_2 περιβάλλει το S_1 . Εάν το S_2 περιβάλλει το S_1 τότε υπάρχει οριακός σημείο μεταξύ του και τότε λέμε ότι το S_2 περιβάλλει το S_1 .

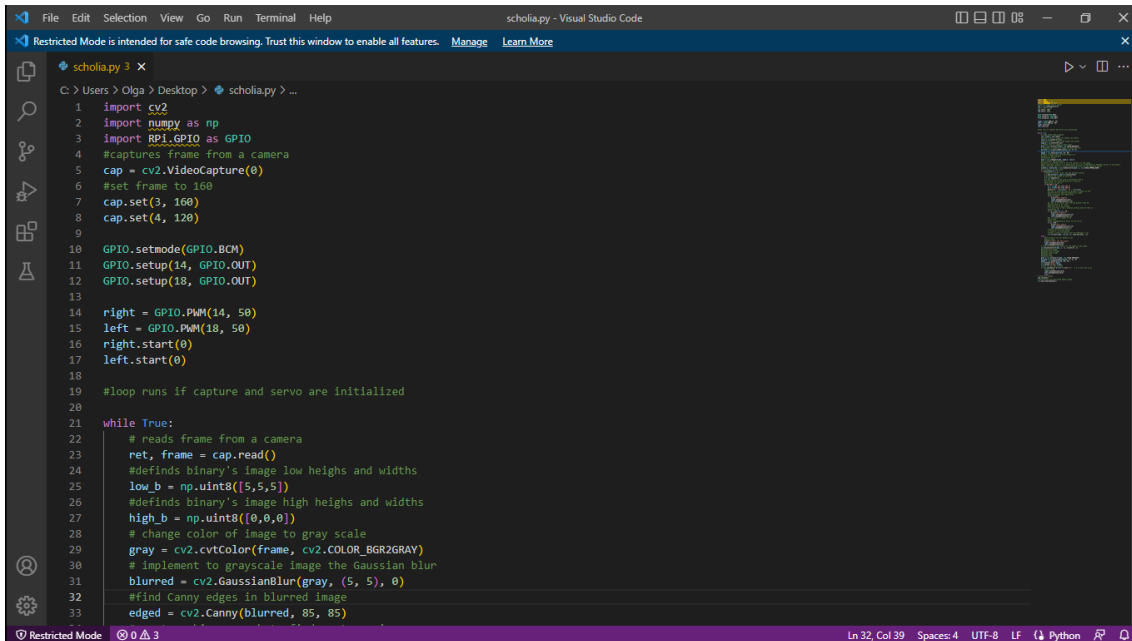
Περιβάλλον μεταξύ ορίων. Όταν το B_n περιβάλλει το B_0 και υπάρχει μία συνέχεια ορίων B_0, B_1, \dots, B_n .



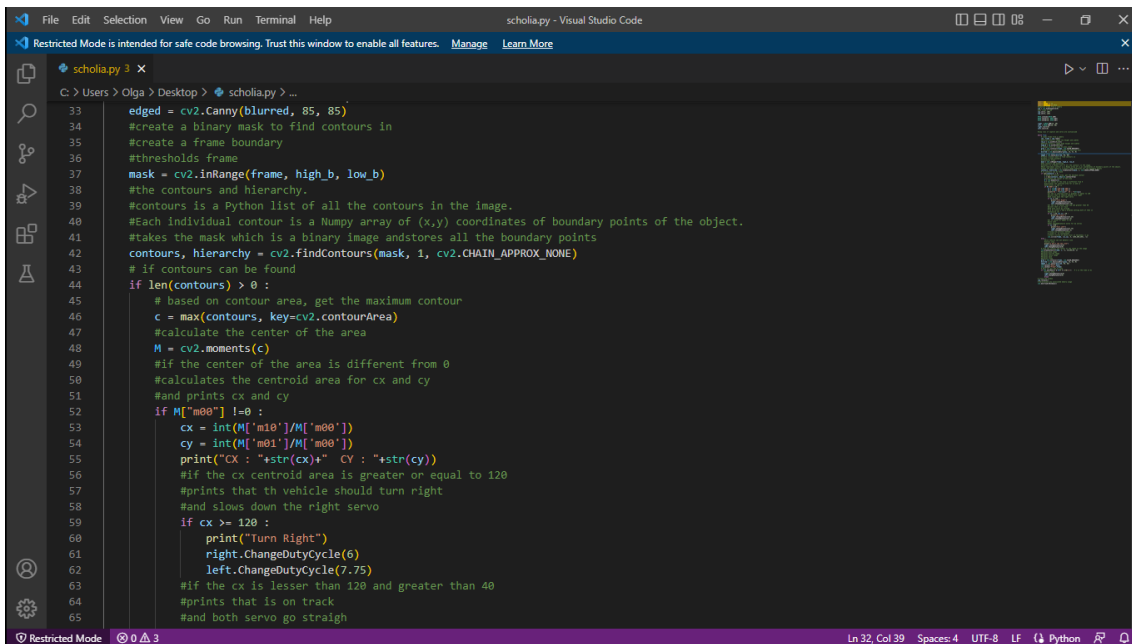
Εικόνα 40.Περιβάλλον μεταξύ ορίων. (etal, 1985)

Συνοψίζοντας, τα περιγράμματα έχουν ως προϋπόθεση την χρήση δυαδικών εικόνων, αυτό μπορεί να επιτευχθεί είτε με την εφαρμογή ανίχνευσης κατωφλιού (Sobel) είτε με την ανίχνευση ακμών. Βρίσκουν ένα λευκό αντικείμενο πάνω σε μαύρο φόντο και κάθε μεμονωμένο περίγραμμα είναι ένας πίνακας κελιών από (x,y) των οριακών σημείων του αντικειμένου. Τέλος, τα περιγράμματα είναι σαν μία καμπύλη, η οποία ενώνει όλα τα συνεχή σημεία κατά μήκος του ορίου.

5.2.1 Εναλλακτικός Αλγόριθμος σε Python.



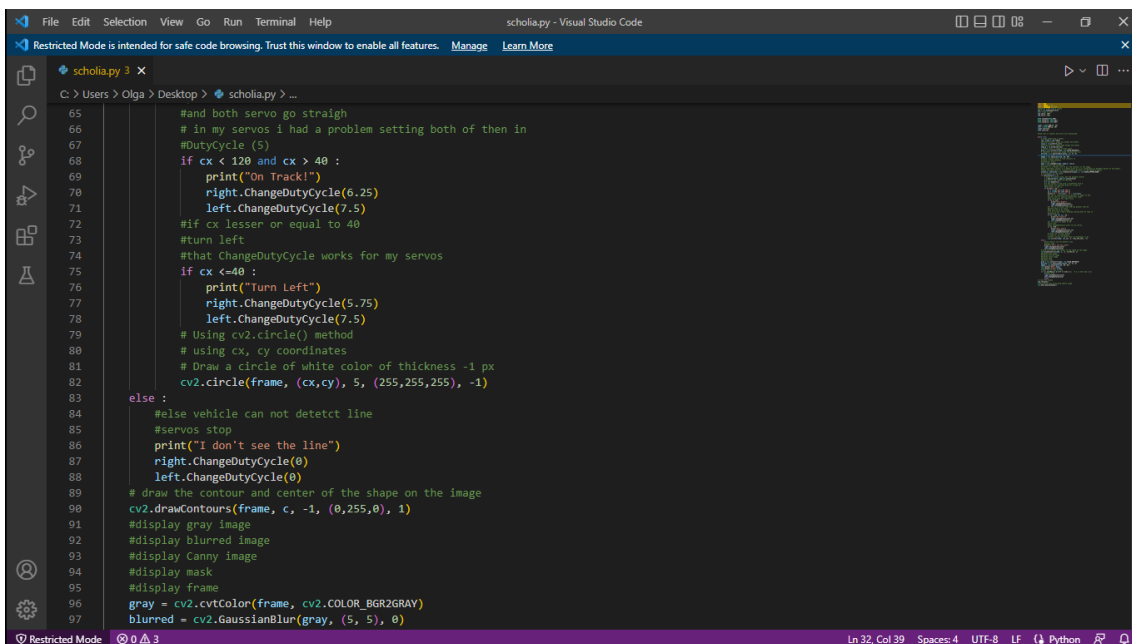
```
File Edit Selection View Go Run Terminal Help schollia.py - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
schollia.py 3 X
C:\Users\Olga\Desktop> schollia.py > ...
1 import cv2
2 import numpy as np
3 import RPi.GPIO as GPIO
4 #captures frame from a camera
5 cap = cv2.VideoCapture(0)
6 #set frame to 160
7 cap.set(3, 160)
8 cap.set(4, 120)
9
10 GPIO.setmode(GPIO.BCM)
11 GPIO.setup(14, GPIO.OUT)
12 GPIO.setup(18, GPIO.OUT)
13
14 right = GPIO.PWM(14, 50)
15 left = GPIO.PWM(18, 50)
16 right.start(0)
17 left.start(0)
18
19 #loop runs if capture and servo are initialized
20
21 while True:
22     # reads frame from a camera
23     ret, frame = cap.read()
24     #defines binary's image low heighs and widths
25     low_b = np.uint8([5,5,5])
26     #defines binary's image high heighs and widths
27     high_b = np.uint8([0,0,0])
28     # change color of image to gray scale
29     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
30     # implement to grayscale image the Gaussian blur
31     blurred = cv2.GaussianBlur(gray, (5, 5), 0)
32     #find Canny edges in blurred image
33     edged = cv2.Canny(blurred, 85, 85)
34
```



```

33 edged = cv2.Canny(blurred, 85, 85)
34 #create a binary mask to find contours in
35 #create a frame boundary
36 #thresholds frame
37 mask = cv2.inRange(frame, high_b, low_b)
38 #the contours and hierarchy.
39 #contours is a Python list of all the contours in the image.
40 #Each individual contour is a Numpy array of (x,y) coordinates of boundary points of the object.
41 #takes the mask which is a binary image and stores all the boundary points
42 contours, hierarchy = cv2.findContours(mask, 1, cv2.CHAIN_APPROX_NONE)
43 # if contours can be found
44 if len(contours) > 0 :
45     # based on contour area, get the maximum contour
46     c = max(contours, key=cv2.contourArea)
47     #calculate the center of the area
48     M = cv2.moments(c)
49     #if the center of the area is different from 0
50     #calculates the centroid area for cx and cy
51     #and prints cx and cy
52     if M["m00"] != 0 :
53         cx = int(M["m10"]/M["m00"])
54         cy = int(M["m01"]/M["m00"])
55         print("CX : "+str(cx)+" CY : "+str(cy))
56         #if the cx centroid area is greater or equal to 120
57         #prints that th vehicle should turn right
58         #and slows down the right servo
59         if cx >= 120 :
60             print("Turn Right")
61             right.ChangeDutyCycle(6)
62             left.ChangeDutyCycle(7.75)
63         #if the cx is lesser than 120 and greater than 40
64         #prints that is on track
65         #and both servo go straigh

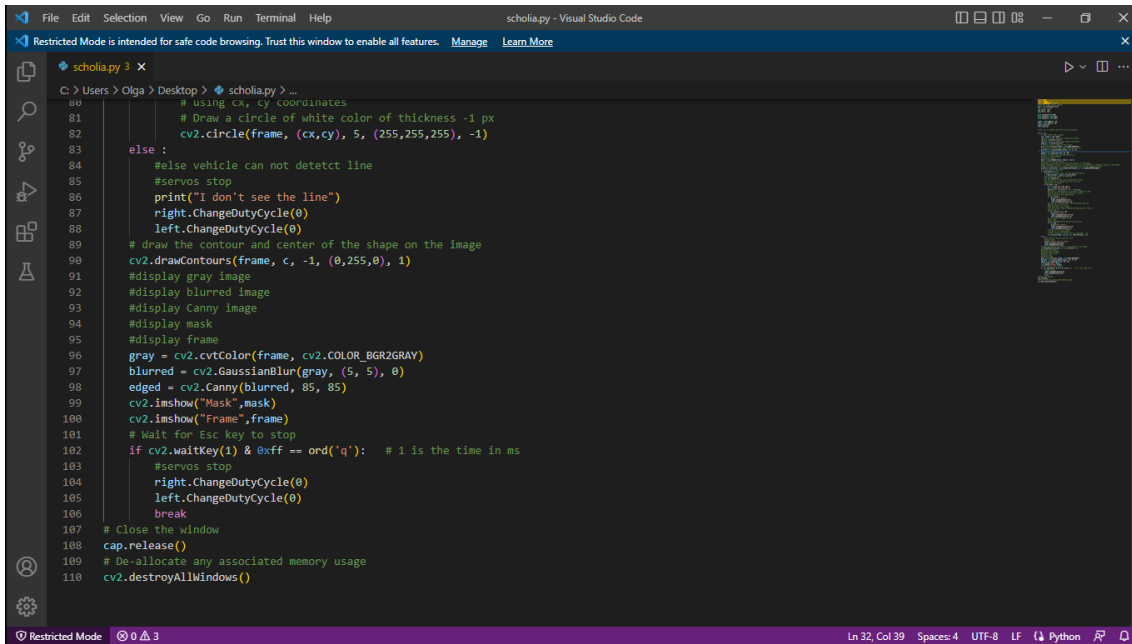
```



```

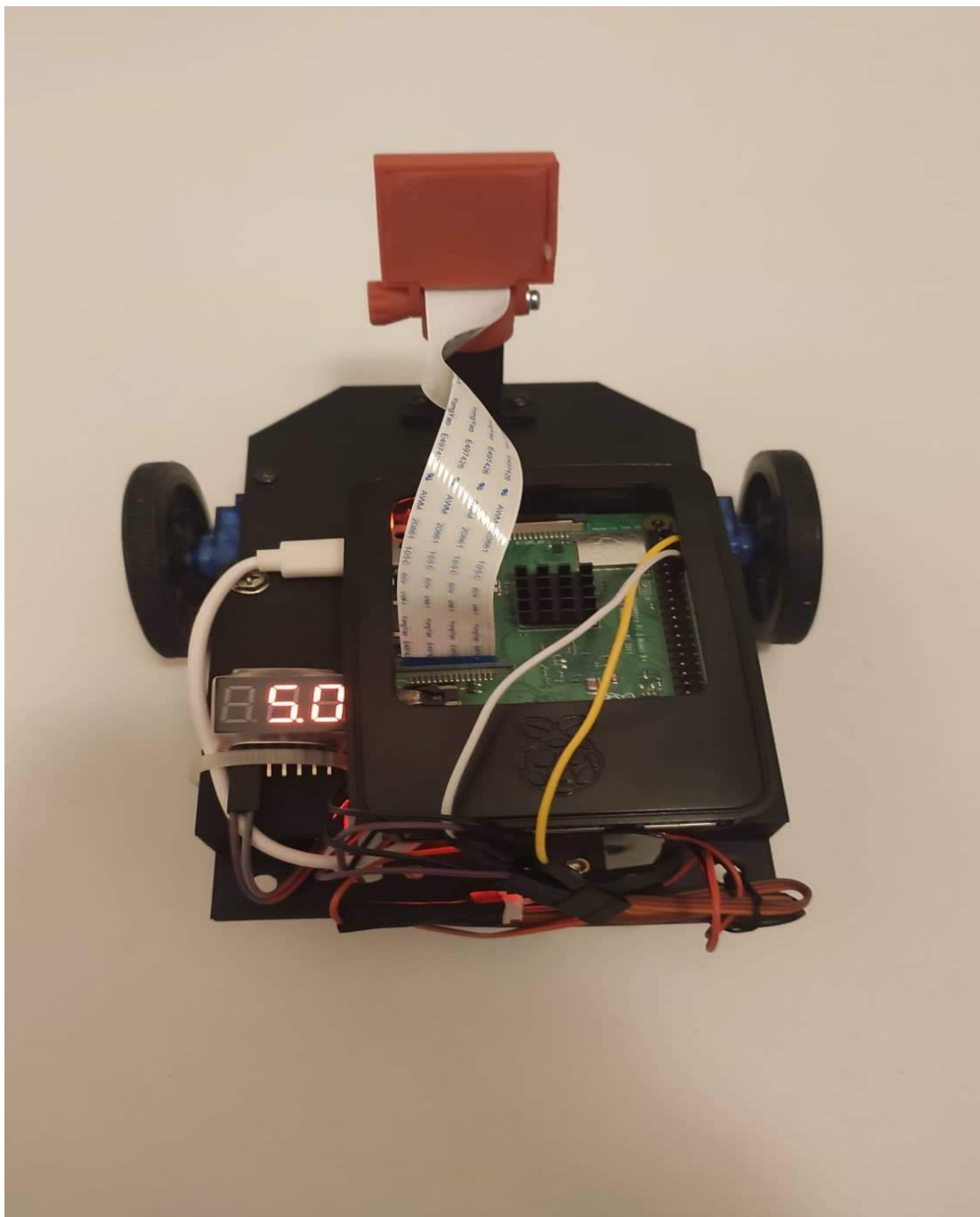
65 #and both servo go straigh
66 # in my servos i had a problem setting both of then in
67 #DutyCycle (5)
68 if cx < 120 and cx > 40 :
69     print("On Track!")
70     right.ChangeDutyCycle(6.25)
71     left.ChangeDutyCycle(7.5)
72 #if cx lesser or equal to 40
73 #turn left
74 #that ChangeDutyCycle works for my servos
75 if cx <=40 :
76     print("Turn Left")
77     right.ChangeDutyCycle(5.75)
78     left.ChangeDutyCycle(7.5)
79 # Using cv2.circle() method
80 # using cx, cy coordinates
81 # Draw a circle of white color of thickness -1 px
82 cv2.circle(frame, (cx,cy), 5, (255,255,255), -1)
83 else :
84     #else vehicle can not detctct line
85     #servos stop
86     print("I don't see the line")
87     right.ChangeDutyCycle(0)
88     left.ChangeDutyCycle(0)
89 # draw the contour and center of the shape on the image
90 cv2.drawContours(frame, c, -1, (0,255,0), 1)
91 #display gray image
92 #display blurred image
93 #display Canny image
94 #display mask
95 #display frame
96 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
97 blurred = cv2.GaussianBlur(gray, (5, 5), 0)

```

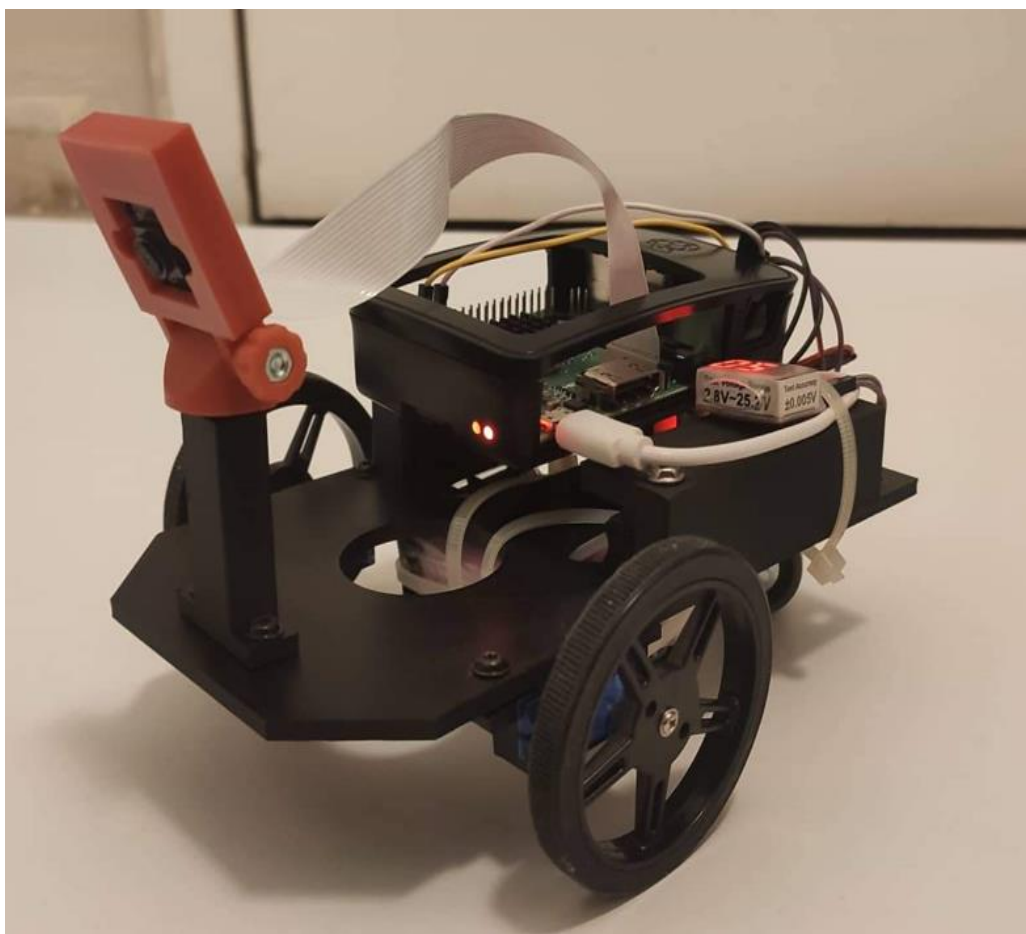


```
80 # using cx, cy coordinates
81 # Draw a circle of white color of thickness -1 px
82 cv2.circle(frame, (cx,cy), 5, (255,255,255), -1)
83
84 else :
85     #else vehicle can not detctct line
86     #servos stop
87     print("I don't see the line")
88     right.ChangeDutyCycle(0)
89     left.ChangeDutyCycle(0)
90 # draw the contours and center of the Shape on the image
91 cv2.drawContours(frame, c, -1, (0,255,0), 1)
92 #display gray image
93 #display blurred image
94 #display Canny image
95 #display mask
96 #display frame
97 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
98 blurred = cv2.GaussianBlur(gray, (5, 5), 0)
99 edged = cv2.Canny(blurred, 85, 85)
100 cv2.imshow("Mask",mask)
101 cv2.imshow("Frame",Frame)
102 # Wait for Esc key to stop
103 if cv2.waitKey(1) & 0xff == ord('q'): # 1 is the time in ms
104     #servos stop
105     right.ChangeDutyCycle(0)
106     left.ChangeDutyCycle(0)
107     break
108 # Close the window
109 cap.release()
110 # De-allocate any associated memory usage
111 cv2.destroyAllWindows()
```

Για τους σερβοκινητήρες χρησιμοποιήθηκε ο ίδιος κώδικας με αυτόν του πρώτου αλγόριθμου. (engineerkid1, n.d.)



Εικόνα 41. Η πάνω πλευρά του αυτοκατευθυνόμενου οχήματος.



Εικόνα 42. Το αυτοκατευθυνόμενο όχημα.

6 Συμπεράσματα

Ο σκοπός της παρούσας εργασίας ήταν η κατασκευή ενός ρομποτικού αυτοκατευθυνόμενου οχήματος όπου θα εντοπίζει μία λωρίδα και θα κινείται πάνω σε αυτή ακολουθώντας την. Για να πραγματοποιηθεί αυτή η εργασία χρησιμοποιήθηκε ένα βασικό εργαλείο, δηλαδή η υπολογιστική όραση που εξελίσσεται ραγδαία και εφαρμόζεται σε ένα ευρύ φάσμα τομέων της καθημερινής ζωής. Να σημειωθεί εδώ, ότι μετά από την μελέτη της διαφοράς της υπολογιστικής με την μηχανική όραση, καταλήγουμε στο συμπέρασμα ότι η μηχανική όραση είναι παρακλάδι της υπολογιστικής όρασης και χρησιμοποιείται κυρίως στον κλάδο της μηχανικής. Στην παρούσα εργασία οι δυο αυτές έννοιες συγχέονται και θεωρούνται ίδιες. Επίσης, παρατηρήσαμε την εξέλιξη των αυτοματισμών στην καθημερινότητά μας και κυρίως όσον αφορά τα οχήματα και τα επίπεδα αυτοματισμών που έχουν αναπτυχθεί με τέτοιο τρόπο ώστε να μιλάμε για αυτοκατευθυνόμενα οχήματα που δεν χρειάζονται οδηγό, αλλά μέσω αλγορίθμων βαθιάς μάθησης και κατάλληλων υλικών, όπως κάμερες και αισθητήρες, είναι σε θέση να αναγνωρίζουν το περιβάλλον στο οποίο κινούνται και να προβλέψουν είτε κινήσεις άλλων οχημάτων, είτε κινήσεις πεζών.

Για την υλοποίηση του οχήματος χρησιμοποιήθηκε ο υπολογιστής μονής πλακέτας Raspberry Pi 3B+, ο οποίος είναι ιδιαίτερα εύχρηστος και χρησιμοποιείται για την κατασκευή εφαρμογών σε ρομποτική και πληροφορική. Επίσης, οι διαστάσεις του είναι τέτοιες που δίνεται η δυνατότητα να γίνει φορητός και το κόστος του είναι χαμηλό.

Για την κίνηση στους τροχούς χρησιμοποιήθηκαν δύο σερβοκινητήρες που χρησιμοποιούνται ειδικά για συνεχή περιστροφή και είναι κατάλληλοι για να ελεγχθούν από το Raspberry και οποιονδήποτε μικροελεγκτή με απευθείας σύνδεση, με μία γραμμή εισόδου/εξόδου, το γεγονός αυτό αποτελεί και τον λόγο που επιλέχθηκαν για την κατασκευή του οχήματος, καθώς βέβαια και το μικρό τους μέγεθος.

Για τον εντοπισμό της λωρίδας χρησιμοποιήθηκε η κάμερα Raspberry Pi Camera Module V2 και συνδυάστηκε με αλγόριθμο υπολογιστικής όρασης, ούτως ώστε το όχημα να εντοπίζει τη λωρίδα. Οι βιβλιοθήκες που χρησιμοποιήθηκαν ήταν η OpenCv, NumPy και Matplotlib.

Οι απαραίτητες εντολές και αλγόριθμοι γράφτηκαν στην γλώσσα προγραμματισμού Python.

Τέλος το όχημα ικανοποιεί επαρκώς τον στόχο για τον οποίο κατασκευάστηκε, βέβαια μπορούν να γίνουν μελλοντικές βελτιώσεις όπως :

Την εγκατάσταση ενός τρίτου σερβοκινητήρα και πάνω σε αυτόν την τοποθέτηση της κάμερας, ούτως ώστε να δίνεται η δυνατότητα περιστροφής της κάμερα και με αυτό τον τρόπο να επιτυγχάνεται ο καλύτερος εντοπισμός λωρίδας στο χώρο.

Την χρήση συνελκτικών Νευρωνικών δικτύων που αποτελεί τον κυριότερο αλγόριθμο για την αναγνώριση διαφορετικών τμημάτων του δρόμου, καθώς και την ταξινόμηση αυτών των τμημάτων και τέλος την λήψη κατάλληλων αποφάσεων. (Barla, 2022)




Κάτι ακόμα που θα βοηθούσε στην βελτίωση του οχήματος είναι η χρήση PID controller για τον καλύτερο έλεγχο των σερβομηχανισμών.

Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές

- Industrial Quick Search Directory , n.d. *www.iqsdirectory.com*. [Online]
Available at: <https://www.iqsdirectory.com/articles/power-supply/dc-power-supply.html>
[Accessed 2022].
- Wikipedia, n.d. *en.wikipedia.org*. [Online]
Available at: https://en.wikipedia.org/wiki/Raspberry_Pi_OS
(1968-2012), J. D. H., 2008. *matplotlib.org*. [Online].
- 5 March 2021, 5 March 2021. *components101.com*. [Online]
[Accessed 5 ΜΑΡΤΙΟΣ 2021].
- abhinav, n.d. *www.hackster.io*. [Online]
Available at: <https://www.hackster.io/>
[Accessed 20 4 2018].
- Alarcon, N., 2018 . *developer.nvidia.com*. [Online].
- Anon., n.d. *en.wikibooks.org*. [Online]
Available at: <https://en.wikibooks.org/wiki/File:Python-logo-notext.svg>
- Anon., n.d. *gr.rsdelivers.com*. [Online]
Available at: <https://gr.rsdelivers.com/product/raspberry-pi/raspberry-pi-camera-module-v21/raspberry-pi-camera-module-csi-2-with-3280-x-2464/9132664>
- Anon., n.d. *grobotronics.com*. [Online]
Available at: <https://grobotronics.com/dc-gear-motor-metal-dual-shaft-yellow.html>
- Anon., n.d. *www.raspberrypi.com*. [Online]
Available at: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>
- Antoniadis, P., 2022. *www.baeldung.com*. [Online].
- avayan, n.d. *ebldc.com*. [Online]
Available at: <https://ebldc.com/?p=48>
[Accessed 29 8 2009].
- Barla, N., 2022. *neptune.ai/blog*. [Online]
Available at: <https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn>
[Accessed 22 7 2022].
- Barr, M., n.d. *barrgroup.com*. [Online]
Available at: <https://barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation>
[Accessed 1 9 2001].
- Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

BenQ Color Technology Lab, 2019. [Online]

Available at: www.benq.com/en-us/knowledge-center/knowledge/color-gamut-monitor.html

Charles R. Harris¹, K. J. M. , S. J. v. d. W. , R. G. , 2020. Array programming with NumPy. *Nature*, 16 9, Issue 585 (7825), p. 357–362.

DigitalSreeni, n.d. *DigitalSreeni*. [Online]

Available at: <https://www.youtube.com/c/DigitalSreeni>

[Accessed 21 2 2020].

en.wikipedia.org, 13 October 2022. [Online]

Available at: https://en.wikipedia.org/wiki/HSL_and_HSV#cite_note-FR841335-6

[Accessed 13 10 2022].

en.wikipedia.org, 2009. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Pulse-width_modulation

[Accessed 27 9 2022].

en.wikipedia.org, n.d. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Servo_control

[Accessed 14 April 2021].

en.wikipedia.org, n.d. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Hough_transform

engineerkid1, n.d. *www.instructables.com*. [Online]

Available at: <https://www.instructables.com/OpenCV-Based-Line-Following-Robot/>

etal, S. S., 1985. Topological Structural Analysis of Digitized Binary. *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING*.

eTechnophiles , n.d. *www.etechnophiles.com*. [Online].

Geeks, Geeks for, 2022. *Geeks for Geeks*. [Online]

Available at: <https://www.geeksforgeeks.org/opencv-real-time-road-lane-detection/>

Heath, J., n.d. *www.analogictips.com*. [Online]

Available at: <https://www.analogictips.com/pulse-width-modulation-pwm/>

[Accessed 4 4 2017].

James Allen on September 4, 2., 2019. *traffictoday.com*. [Online].

Khillar, S., 2022. *differencebetween.net*. [Online].

Kunanbaeva, A., 2020. *towardsdatascience.com*. [Online].

madmortigan, n.d. *planetmath.org*. [Online]

Available at: <https://planetmath.org/houghtransform>

[Accessed 22 3 2013].

Mark S. Nixon and Alberto S. Aguado, 2002. *Feature Extraction & Image Processing for Computer Vision*. [Online]

Available at: https://www.southampton.ac.uk/~msn/book/new_demo/gaussian/

Nayar, S., n.d. *First Principles of Computer Vision*. [Online]

Available at: https://www.youtube.com/watch?v=XRbc_xkZREg

[Accessed 3 4 2021].

Neuromation, 2018. *How Computer Vision Can Change the Automotive Industry*. [Online]

Available at: <https://medium.com/neuromation-blog/how-computer-vision-can-change-the-automotive-industry-b8ba0f1c08d1>

[Accessed 8 8 2018].

NumPy developers, n.d. *numpy.org*. [Online].

OpenCV team , 2022. *opencv.org*. [Online]

Available at: <https://opencv.org/about/>

Piltch, A., n.d. *www.tomshardware.com*. [Online]

Available at: <https://www.tomshardware.com/news/raspberry-pi-os-no-longer-raspbian>

[Accessed 30 5 2020].

Pololu.com, n.d. *Pololu*. [Online]

Available at: <https://www.pololu.com/product/1420>

Pramoditha, R., 2021. *towardsdatascience.com*. [Online].

Rafiaue, Z., 2020. Improving Efficiency of Computer Vision for Autonomous Vehicles. *ResearchGate*, 12.

Rafique, Z., 2020. *Improving Efficiency of Computer Vision for Autonomous Vehicles*. s.l.:Bournemouth University.

Rambus Press, 2022. [Online]

Available at: <https://www.rambus.com/blogs/driving-automation-levels/>

Rein van den Boomgaard, L. D., 2021. *Gaussian Convolutions and Derivatives*. [Online]

Available at:

<https://staff.fnwi.uva.nl/r.vandenboomgaard/ComputerVision/LectureNotes/IP/LocalStructure/GaussianDerivatives.html>

Rizzoli, A., 2022. *www.v7labs.com*. [Online].

Σχεδιασμός και Υλοποίηση Αυτοκατευθυνόμενου Οχήματος με Ανίχνευση Λωρίδας

Sharda, A., 2021. *medium.com*. [Online].

Stanislav Smith, Technical Transmission General Articles , 2022. *go4trans.com*. [Online] [Accessed 3 1 2022].

The Pi Hut, n.d. *thepihut.com*. [Online]

Available at: <https://thepihut.com/blogs/raspberry-pi-roundup/whats-the-difference-between-dc-servo-amp-stepper-motors>

[Accessed 20 4 2019].

Torrens, R., n.d. *www.4qdtec.com*. [Online]

Available at: <https://www.4qdtec.com/pwpm.html>

[Accessed 1996-2011].

Wikipedia, n.d. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Canny_edge_detector

Wikipedia, n.d. *en.wikipedia.org*. [Online]

Available at: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

Wikipedia, n.d. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Computer_vision#cite_note-Sebe2005-14

Wikipedia, n.d. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Gaussian_filter

Wikipedia, n.d. *en.wikipedia.org*. [Online]

Available at: https://en.wikipedia.org/wiki/Machine_vision

www.heason.com, n.d. *www.heason.com*. [Online]

[Accessed 27 10 2015].

ΒΑΣΙΛΟΠΟΥΛΟΣ ΝΙΚΟΛΑΟΣ, Δ. Α. Κ. Ε., n.d. *csdweb.aegean.gr*. [Online]

Available at: <https://icsdweb.aegean.gr/aegeanrbtcs/wp-content/uploads/2021/09/9.-%CE%A3%CF%85%CE%BD%CE%B4%CF%85%CE%B1%CF%83%CE%BC%CF%8C%CF%82-%CE%BC%CE%B5-%CE%BA%CE%AC%CE%BC%CE%B5%CF%81%CE%B1-Raspberry.pdf>

[Accessed 9 9 2021].