



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«Δημιουργία Εκπαιδευτικού Ρομποτικού Οχήματος
Έρευνας»



Της φοιτήτριας
Λιόκαυτου Παρασκευής
Αρ. Μητρώου:10/2023

Επιβλέπων
Ονοματεπώνυμο
Στάθης Κασδερίδης

Ημερομηνία 06/02/2025

Τίτλος Δ.Ε. Δημιουργία Εκπαιδευτικού Ρομποτικού Οχήματος Έρευνας

Κωδικός Δ.Ε. 24288

Όνοματεπώνυμο φοιτητή/των Παρασκευή Λιόκαντου

Όνοματεπώνυμο εισηγητή Στάθης Κασδερίδης

Ημερομηνία ανάληψης Δ.Ε. 28/10/2024

Ημερομηνία περάτωσης Δ.Ε. 06/02/2025

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

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας Λιόκαντου Παρασκευής που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

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

«Αφιέρωση»

Πρόλογος

Η επιλογή της διπλωματικής μου εργασίας έγινε μετά από διερεύνηση θεμάτων με τον επιβλέπων καθηγητή μου με σκοπό η επιλογή ενός θέματος που θα αναφερόταν σε ύλη μαθημάτων που είχαν διδαχτεί στην διάρκεια του Μεταπτυχιακού καθώς και στόχος ήταν να υπάρξει μελλοντική χρήση στην επαγγελματική μου εργασία. Το θέμα, μου παρείχε την ευκαιρία να εμβαθύνω σε επιστημονικές περιοχές της Πληροφορική όπως Μηχανική Μάθηση και Ρομποτική που θεωρώ ότι είναι ταχύτατα εξελισσόμενες με πολλαπλές εφαρμογές σε όλους τους τομείς και στην εκπαίδευση που είναι και το αντικείμενο της εργασίας μου.

Μέσα από τα εργαλεία που χρησιμοποίησα αποκόμισα γνώσεις τόσο για τον τεχνικό εξοπλισμό όσο και σε λογισμικό επίπεδο και σε αρχιτεκτονικές επικοινωνίας. Ήταν πολύ εποικοδομητικό για μένα η θεωρητική γνώση που είχα να μετατραπεί σε πρακτική εφαρμογή με τελικό αποτέλεσμα την κατασκευή του ρομποτικού μου οχήματος.

Περίληψη

Η διπλωματική θα σχεδιάσει και θα υλοποιήσει ένα μοντέλο ρομποτικού οχήματος έρευνας, σε εσωτερικό χώρο, κατάλληλο για διδασκαλία στην τάξη. Το σύστημα θα φέρει σειρά αισθητήρων για την ασφαλή πλοήγηση και την αποφυγή συγκρούσεων με εμπόδια. Μια κάμερα θα επιτρέπει την αναγνώριση αντικειμένων-στόχων και την καταμέτρησή τους. Για τον σκοπό αυτό θα αναπτυχθεί ένα μοντέλο Μηχανικής Μάθησης. Το ρομπότ θα έχει δύο τρόπους λειτουργίας: τον αυτόνομο και με καθοδήγηση από τον χρήστη. Δύο προγράμματα ελέγχου θα κατασκευαστούν. Το πρώτο (ελεγκτής) θα είναι υπεύθυνο για τον έλεγχο του υλικού του ρομπότ και την επικοινωνία με το δεύτερο πρόγραμμα. Θα εκτελεί τις εντολές κίνησης που θα λαμβάνει και θα διαβάζει τις τιμές των αισθητήρων σε τακτικά χρονικά διαστήματα. Το δεύτερο πρόγραμμα (λήπτης αποφάσεων) θα χρησιμοποιεί τις τιμές των αισθητήρων για να αντιλαμβάνεται την κατάσταση του κόσμου και θα λαμβάνει αποφάσεις σχετικά με ποιο μέρος του κόσμου θα πρέπει να εξερευνηθεί σε επόμενο χρόνο. Αντικείμενα ενδιαφέροντος θα καταμετρούνται και θα αναφέρονται στον χρήστη. Το δεύτερο πρόγραμμα θα υποστηρίζει δύο τρόπους λειτουργίας του συστήματος: είτε τελείως αυτόνομα είτε με εντολές του χρήστη. Το δεύτερο πρόγραμμα θα έχει ένα GUI για να παρουσιάζει τις πληροφορίες ενδιαφέροντος και να δέχεται εντολές χρήστη σε πραγματικό χρόνο.

Λέξεις κλειδιά

Raspberry pi 5, resnet, efficientnet, robot, client-server επικοινωνία, ρομποτικό όχημα, εκπαίδευση, τεχνητή νοημοσύνη, συλλογή δεδομένων, αναγνώριση σχημάτων, αλγόριθμοι ελέγχου, Βαθιά Μάθηση.

«Building an Educational Robotic Research Vehicle»

(στην αγγλική γλώσσα)

«Paraskevi Liokautou»

(στην αγγλική γλώσσα)

Abstract

The thesis will design and implement a model of a robotic research vehicle, operating indoors, suitable for classroom teaching. The system will carry a series of sensors for safe navigation and avoidance of collisions with obstacles. A camera will allow for the recognition of target objects and their counting. For this purpose, a Machine Learning model will be developed. The robot will have two operating modes: autonomous and user-guided. Two control programs will be built. The first (controller) will be responsible for controlling the robot's hardware and communicating with the second program. It will execute the movement commands it receives and will read the sensor values at regular intervals. The second program (decision system) will use the sensor values to understand the state of the world and make decisions about which part of the world should be explored next. Objects of interest will be counted and reported to the user. The second program will support two modes of operation of the system: either completely autonomous or with user commands. The second program will have a GUI to present the information of interest and accept user commands in real time.

Keywords

Raspberry pi 5, resnet, efficientnet, robot, client-server communication, robotic vehicle, education, artificial intelligence, data collection, pattern recognition, control algorithms, Deep Learning.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πρώτα από όλα τον σύζυγό μου που είναι πάντα ενθαρρυντικός και υπομονετικός σε όλες τις επιλογές μου . Τα παιδιά μου που κάνανε μεγάλη υπομονή με τις ώρες που αφιέρωσα στο μεταπτυχιακό μου. Την αδερφή μου Βάσω χωρίς αυτήν δεν θα μπορούσα να το υλοποιήσω.

Τέλος θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου για την καθοδήγησή του και τις συμβουλές του σε όλη την διάρκεια της διπλωματική μου εργασίας. Εκτιμώ ιδιαίτερα την υπομονή και την διαθεσιμότητά του να απαντά στις ερωτήσεις μου και να με καθοδηγεί σε ένα θέμα που δεν είχα μεγάλη εμπειρία. Είμαι ευγνώμων για την εμπιστοσύνη που μου έδειξε να συνεργαστεί μαζί μου.

Περιεχόμενα

Πρόλογος	v
Περίληψη	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xi
Κατάλογος Πινάκων	xi
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Εισαγωγή	1
1.2 Ρομποτική – Εκπαιδευτική Ρομποτική	2
1.3 Βαθιά Μάθηση –Deep Learning	3
1.3.1 Συνελκτικά νευρωνικά δίκτυα (CONVERGULATION NEURAL NETWORKS)	4
1.3.2 Resnet.....	5
1.3.3. Efficientnet	6
1.4 Αρχιτεκτονική client-server	6
1.5 Τεχνολογίες.....	7
1.5.1 Raspberry Pi	7
1.5.2 Sensor Αισθητήρες	9
1.5.3 Γλώσσα Προγραμματισμού Python.....	10
1.5.4 Google Colab / Colaboratory.....	11
1.6 Επίλογος.....	12
Κεφάλαιο 2ο: Θεωρητικό υπόβαθρο.....	13
2.1 Συλλογή δεδομένων.....	13
2.2 Προετοιμασία δεδομένων	14
2.3 Εκπαίδευση μοντέλου.....	15
2.4 Προβλέψεις από το εκπαιδευμένο μοντέλο	17
2.5 Επίλογος.....	19
Κεφάλαιο 3ο: Σχεδίαση και Ανάπτυξη Ρομποτικού Οχήματος.....	20
3.1 Εισαγωγή	20
3.2 Υλικό οχήματος.....	21
3.3 Λογισμικό Οχήματος.....	27

3.4	Απομακρυσμένη πρόσβαση	27
3.5	Χειρισμός του οχήματος με Python.....	30
3.6	Client-Server	31
3.7	Multithreading.....	32
3.8	Επίλογος.....	33
Κεφάλαιο 4ο:	Αποτελέσματα	34
Κεφάλαιο 5ο:	Συμπεράσματα ή/και προτάσεις βελτίωσης.....	41
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		42

Κατάλογος Εικόνων

Εικόνα 1: Ρομποτική Πηγή: https://www.kathimerini.gr/economy	2
Εικόνα 2: A Typical Convolutional Neural Network	5
Εικόνα 3:Raspberry Pi	8
Εικόνα 4 : Google Colab.....	11
Εικόνα 5: Κύβος Εικόνα 6: Τριγωνική σφήνα Εικόνα 7: Τραπεζοειδής Πυραμίδα .	13
Εικόνα 8: Τιμές από τα dataset.....	14
Εικόνα 9: Training and Validation Losses/Resnet18.....	16
Εικόνα 10: Training and Validation Losses/Efficientnetv2_s.....	16
Εικόνα 11: Confusion Matrix ResNet18.....	17
Εικόνα 12: Confusion Matrix EfficientNetv2_s.....	18
Εικόνα 13: Υλικό.....	22
Εικόνα 14: Robot Smart Car 4WD	22
Εικόνα 15: Στάδιο συναρμολόγησης	23
Εικόνα 16: Ενσωμάτωση και του οδηγού L298N.	23
Εικόνα 17: Raspberry Pi 5.....	24
Εικόνα 18: Pins of Raspberry Pi 5 Πηγή: https://www.hackatronic.com/raspberry-pi-5-pinout-specifications-pricing-a-complete-guide/	24
Εικόνα 19: Αισθητήρας εμποδίων	25
Εικόνα 20: Camera Module 8MP for Raspberry Pi 5 - 79.3° FOV	25
Εικόνα 21: Raspberry Pi Imager.....	27
Εικόνα 22: Επιφάνεια εργασίας Raspberry Pi Απομακρυσμένη πρόσβαση	27
Εικόνα 23:Ρυθμίσεις στο Raspberry για το VNC	28
Εικόνα 24:Η πρώτη επιφάνεια του RealVNC	28
Εικόνα 25:Γίνεται η επιβεβαίωση των κωδικών	29
Εικόνα 26:Εικόνα σύνδεσης Raspberry	29
Εικόνα 27: Πρώτη εικόνα RealVNC Viewer	34
Εικόνα 28: Authentication.....	35
Εικόνα 29: Εισαγωγή στοιχείων του Raspberry Pi.....	35
Εικόνα 30: Επιφάνεια εργασίας rpi	36
Εικόνα 31: Windows and rpi	36
Εικόνα 32: Thorny client_end.py	37
Εικόνα 33: Εντολές ελέγχου.....	38
Εικόνα 34: Σωστή πρόβλεψη ΚΥΒΟΣ	38
Εικόνα 35: Σωστή πρόβλεψη ΤΡΙΓΩΝΙΚΗ ΣΦΗΝΑ	39
Εικόνα 36: Σωστή πρόβλεψη ΑΓΝΩΣΤΟ.....	39

Κατάλογος Πινάκων

Πίνακας 1: Αισθητήρες.....	10
Πίνακας 2:Υλικά κατασκευής.....	21

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
τρι	Raspberry Pi
DP	Deep Learning
CNN	Convolutional neural network
Client	Πελάτης
Server	Εξυπηρετητής

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Η ανάπτυξη ενός εκπαιδευτικού ρομποτικού οχήματος για σκοπούς έρευνας, σε εσωτερικό χώρο, κατάλληλο για διδασκαλία στην τάξη ενσωματώνει τις αρχές της ρομποτικής και της τεχνητής νοημοσύνης, προσφέροντας μια διεπιστημονική πλατφόρμα μάθησης και εφαρμογών. Η παρούσα εργασία επικεντρώνεται στη σχεδίαση, υλοποίηση και αξιολόγηση ενός ρομποτικού οχήματος εσωτερικού χώρου που συνδυάζει την αυτονομία, την οπτική αναγνώριση μέσω μοντέλων νευρωνικών δικτύων όπως το Resnet18 και το Efficientnetv2_s.

Το σύστημα φέρει σειρά αισθητήρων για την ασφαλή πλοήγηση και την αποφυγή συγκρούσεων με εμπόδια. Επίσης μια κάμερα θα επιτρέπει την αναγνώριση αντικειμένων-στόχων. Η απομακρυσμένη επικοινωνία του οχήματος με το μοντέλο εκπαίδευσης γίνεται με την αρχιτεκτονική client-server (πελάτη-εξυπηρετητή). Το όχημα βασίζεται σε μια ευέλικτη αρχιτεκτονική με χρήση της πλατφόρμας Raspberry Pi 5. Τα αποτελέσματα δείχνουν ότι η υλοποίηση είναι οικονομικά αποδοτική και λειτουργικά αξιόπιστη, ενώ παράλληλα συμβάλλει στη διεύρυνση της γνώσης και της χρήσης της εκπαιδευτικής ρομποτικής. Οι μελλοντικές προεκτάσεις περιλαμβάνουν τη βελτίωση της ανάλυσης δεδομένων και την ενσωμάτωση προηγμένων τεχνικών τεχνητής νοημοσύνης για την πλήρη αυτονομία του συστήματος.

Η ανάπτυξη ρομπότ και η αυτονομία τους με βάση την τεχνητή νοημοσύνη μέσω της υπολογιστικής όρασης για τη διάκριση αντικειμένων και τη λήψη αποφάσεων σε πραγματικό χρόνο είναι ένα από τα σημαντικά πράγματα στον τομέα της βιομηχανίας για τη βελτίωση της αποτελεσματικότητας των βιομηχανικών ρομπότ και για να γίνουν εξυπνότερα.[1]

Βασικό συστατικό της αυτονομίας των ρομπότ είναι η ικανότητά τους να αντιλαμβάνονται και να κατανοούν το περιβάλλον τους. Αυτό επιτυγχάνεται με τη χρήση τεχνικών υπολογιστικής όρασης και αλγορίθμων βαθιάς μάθησης, οι οποίοι επιτρέπουν στα ρομπότ να αναγνωρίζουν αντικείμενα, να κατανοούν την τοποθεσία τους και να λαμβάνουν αποφάσεις σε πραγματικό χρόνο. Η ενσωμάτωση αυτών των τεχνολογιών στις βιομηχανικές εφαρμογές έχει φέρει επανάσταση, αυξάνοντας την αποδοτικότητα και μειώνοντας το ανθρώπινο σφάλμα. Η ανάπτυξη διαφόρων τύπων ρομποτικών συστημάτων, συμπεριλαμβανομένων των ρομποτικών οχημάτων έρευνας, όπως τα ρομπότ εξερεύνησης πλανητών (π.χ. NASA Mars Rover), τα αυτόνομα υποβρύχια οχήματα (AUVs) και τα ερευνητικά drones που χρησιμοποιούνται σε περιβαλλοντικές μελέτες.

Σε αυτό το πλαίσιο, η χρήση οπτικής αναγνώρισης με συστήματα βαθιάς μάθησης, όπως τα συνελκτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNNs), επιτρέπει τη γρήγορη και αξιόπιστη αναγνώριση

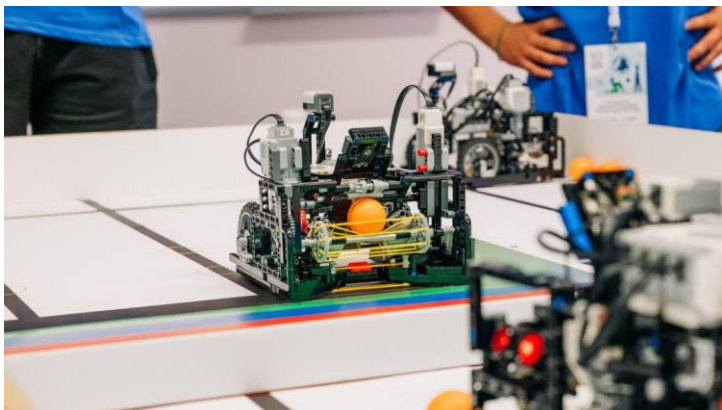
αντικειμένων και την αποφυγή εμποδίων. Τα συστήματα αυτά προσφέρουν δυνατότητες όπως η κατηγοριοποίηση αντικειμένων, η παρακολούθηση κινούμενων στόχων και η ακριβής ανάλυση δεδομένων εικόνας, συμβάλλοντας στη δημιουργία προηγμένων ρομποτικών συστημάτων.

Η εργασία αυτή εξετάζει την ανάπτυξη ενός τέτοιου συστήματος, το οποίο συνδυάζει την αυτονομία και την προηγμένη υπολογιστική όραση, εστιάζοντας σε εκπαιδευτικές εφαρμογές. Το ρομποτικό όχημα που προτείνεται ενσωματώνει σύγχρονες τεχνολογίες και προσφέρει έναν ισχυρό συνδυασμό υλικού και λογισμικού, κατάλληλο τόσο για εκπαιδευτικούς όσο και για ερευνητικούς σκοπούς.

1.2 Ρομποτική – Εκπαιδευτική Ρομποτική

Η χρήση της ρομποτικής στην εκπαίδευση ανοίγει νέες ευκαιρίες για την υλοποίηση δραστηριοτήτων που βασίζονται σε έργα, επιτρέποντας στους μαθητές να κατακτήσουν πρακτικά πολύπλοκες τεχνικές και επικοινωνιακές δεξιότητες προσφέροντας τους πολλά πλεονεκτήματα και πρακτικές δεξιότητες που είναι περιζήτητες στον σύγχρονο κόσμο.

Μία από τις κύριες θεωρητικές προϋποθέσεις είναι ο ισχυρισμός ότι η χρήση της ρομποτικής στην εκπαίδευση προωθεί την ενεργό συμμετοχή των μαθητών στη μαθησιακή διαδικασία, η οποία με τη σειρά της ενισχύει τα κίνητρά τους και βελτιώνει τα εκπαιδευτικά αποτελέσματα [2]. Επιπλέον, η εκπαιδευτική ρομποτική υποστηρίζει την εφαρμογή τεχνικών μάθησης που βασίζονται σε προβλήματα, όπου οι μαθητές αναπτύσσουν τις γνώσεις και τις δεξιότητές τους μέσω ενεργητικής αλληλεπίδρασης με εκπαιδευτικό υλικό και επίλυσης πρακτικών προβλημάτων. Κυρίως, στοχεύει στην ανάπτυξη τεχνικών δεξιοτήτων στους μαθητές, όπως προγραμματισμού, μηχανικής και εργασίας με ηλεκτρονικά. Για την επίτευξη αυτών των στόχων, πρέπει να χρησιμοποιηθούν διάφορες μεθοδολογίες που ενσωματώνουν αποτελεσματικά τη ρομποτική στην εκπαιδευτική διαδικασία.[3]



Εικόνα 1: Ρομποτική

Πηγή: <https://www.kathimerini.gr/economy>

Πρώτον, η μάθηση βάσει έργου (PBL) είναι μία από τις βασικές μεθοδολογίες. Η μάθηση βάσει έργου εστιάζει στην ενεργό και σε πραγματικό κόσμο μάθηση μέσω της ολοκλήρωσης έργων που έχουν νόημα για τους μαθητές και τις κοινότητές τους. Δεύτερον, η μάθηση με βάση το πρόβλημα (PBL) συμβάλλει επίσης αποτελεσματικά στους στόχους της εκπαιδευτικής ρομποτικής. Αυτή η μεθοδολογία περιλαμβάνει τη μάθηση μέσω της επίλυσης συγκεκριμένων, συχνά ανοιχτών προβλημάτων που απαιτούν την εφαρμογή γνώσεων και δεξιοτήτων από διάφορους τομείς. Η τρίτη σημαντική μεθοδολογία είναι η ενσωμάτωση των κλάδων STEM, όπου η εκπαιδευτική ρομποτική χρησιμοποιείται για να συνδυάσει την επιστήμη (Science), την τεχνολογία (Τεχνολογία), τη μηχανική (Engineering) και τα μαθηματικά (Mathematics) σε ένα ενιαίο εκπαιδευτικό έργο. Αυτή η προσέγγιση επιτρέπει τη συνολική ανάπτυξη των μαθητών και βελτιώνει την κατανόησή τους για τις διεπιστημονικές συνδέσεις. [3]

Η χρήση της δεν περιορίζεται μόνο στις τεχνολογικές δεξιότητες αλλά ενισχύει την συνεργασία και την επικοινωνία μεταξύ των φοιτητών/μαθητών καθιστώντας την ένα ισχυρό εργαλείο μάθησης για τα εκπαιδευτικά ιδρύματα.

1.3 Βαθιά Μάθηση –Deep Learning

Βασικό θέμα συζήτησης αποτελεί η πρόοδος των μηχανών, οι οποίες πλέον έχουν τη δυνατότητα να εκτελούν εργασίες που παλαιότερα απαιτούσαν ανθρώπινη παρέμβαση, τόσο σε εργοστάσια και αποθήκες, όσο και σε γραφεία και σπίτια. Παράλληλα, όροι όπως τεχνητή νοημοσύνη, μηχανική μάθηση και βαθιά μάθηση έχουν αναδειχθεί σε κεντρικά ζητήματα της σύγχρονης τεχνολογίας.

Η τεχνητή νοημοσύνη, εν συντομία, είναι ο επιστημονικός κλάδος που αποσκοπεί στο να καταστήσει τους υπολογιστές "έξυπνους". Μέσα από την κατανόηση της νοημοσύνης, η τεχνητή νοημοσύνη ενισχύει τη λειτουργικότητα των υπολογιστικών συστημάτων, δίνοντάς τους νέες διαστάσεις χρησιμότητας. Οι μέθοδοι βαθιάς μάθησης επιτρέπουν στις μηχανές να μαθαίνουν και να διαχειρίζονται πολύπλοκα δεδομένα με ταχύτητα, εξαλείφοντας την ανάγκη για ανθρώπινη παρέμβαση.

Η βαθιά μάθηση είναι ένας κλάδος μηχανικής μάθησης που βασίζεται σε ένα σύνολο αλγορίθμων που προσπαθούν να μοντελοποιήσουν αφαιρέσεις υψηλού επιπέδου στα δεδομένα χρησιμοποιώντας πολλαπλά επίπεδα επεξεργασίας με πολύπλοκες δομές ή με άλλους τρόπους [4].

Ο όρος βαθιά μάθηση αναφέρεται σε πολυεπίπεδα τεχνητά νευρωνικά δίκτυα. Πρόσφατα, η βαθιά μάθηση έχει γίνει ένα από τα πιο ισχυρά εργαλεία στη βιβλιογραφία, καθώς μπορεί να επεξεργαστεί μεγάλες ποσότητες δεδομένων. Τα συνελκτικά νευρωνικά δίκτυα είναι ένα από τα πιο γνωστά δίκτυα βαθιάς μάθησης και με βάση αυτά έχουν επιτευχθεί σημαντικά επιτεύγματα σε μελέτες όπως η αναγνώριση φωνής, η επεξεργασία εικόνας, η αναγνώριση προτύπων [4].

1.3.1 Συνελκτικὰ νευρωνικά δίκτυα (CONVERGULATION NEURAL NETWORKS)

Τα συνελκτικὰ δίκτυα (LeCun, 1989), επίσης γνωστά ως συνελκτικὰ νευρωνικά δίκτυα ή CNN, είναι ένα εξειδικευμένο είδος νευρωνικού δικτύου για την επεξεργασία δεδομένων που έχει μια γνωστή τοπολογία τύπου πλέγματος.[5] Ήταν εξαιρετικά επιτυχημένα σε πρακτικές εφαρμογές. Το όνομα "συνελκτικό νευρικό δίκτυο" υποδηλώνει ότι το δίκτυο χρησιμοποιεί μια μαθηματική πράξη που ονομάζεται συνέλιξη. Η συνέλιξη είναι ένα εξειδικευμένο είδος γραμμικής λειτουργίας.

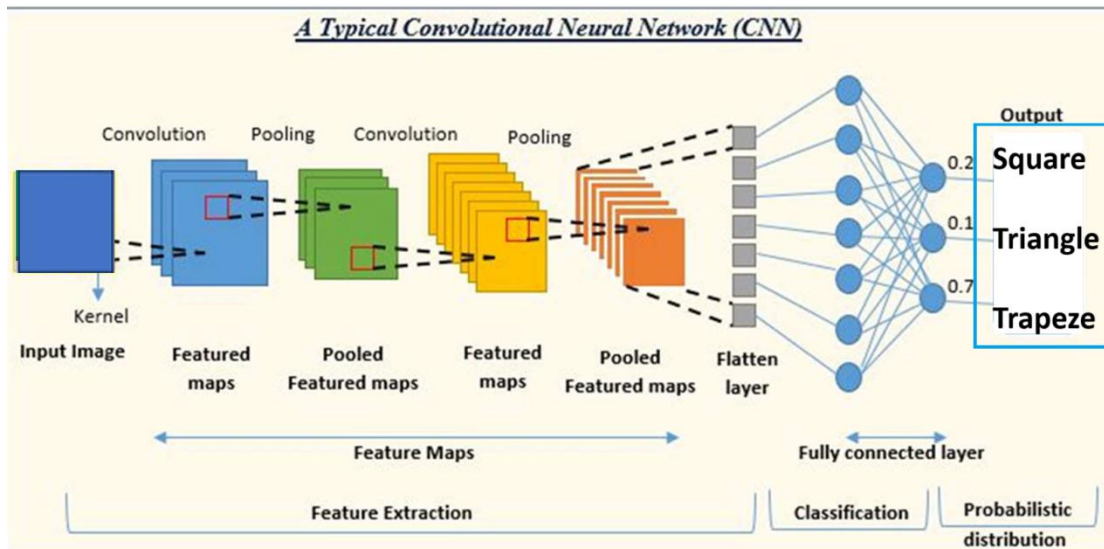
Έχουν παίξει σημαντικό ρόλο στην ιστορία της βαθιάς μάθησης. Αποτελούν βασικό παράδειγμα μιας επιτυχημένης εφαρμογής των γνώσεων που αποκτήθηκαν με τη μελέτη του εγκεφάλου σε εφαρμογές μηχανικής μάθησης. Ήταν επίσης μερικά από τα πρώτα βαθιά μοντέλα που είχαν καλή απόδοση, πολύ πριν τα αυθαίρετα βαθιά μοντέλα θεωρηθούν βιώσιμα. Επίσης είναι μερικά από τα πρώτα νευρωνικά δίκτυα που έλυσαν σημαντικές εμπορικές εφαρμογές και παραμένουν στην πρώτη γραμμή των εμπορικών εφαρμογών της βαθιάς μάθησης σήμερα.[5]

Τα συνελκτικὰ νευρωνικά δίκτυα (Convolutional Neural Networks - CNNs) έχουν αποκτήσει ιδιαίτερη δημοτικότητα λόγω της εξαιρετικής τους απόδοσης σε εφαρμογές ταξινόμησης εικόνων. Ο πυρήνας της λειτουργίας τους βρίσκεται στα συνελκτικά επίπεδα, τα οποία, σε συνδυασμό με τα φίλτρα, διευκολύνουν την εξαγωγή χωρικών και χρονικών χαρακτηριστικών από τις εικόνες. Επιπλέον, η χρήση ειδικών τεχνικών κατανομής βάρους μειώνει σημαντικά τις απαιτήσεις υπολογιστικής ισχύος.

Από την άποψη της αρχιτεκτονικής, τα CNNs είναι μια παραλλαγή των τεχνητών νευρωνικών δικτύων, με δύο βασικούς περιορισμούς: οι νευρώνες κάθε φίλτρου συνδέονται τοπικά μόνο με συγκεκριμένες περιοχές της εικόνας, διατηρώντας τη χωρική της δομή, ενώ τα βάρη είναι κοινά (shared weights), γεγονός που περιορίζει τον συνολικό αριθμό παραμέτρων του μοντέλου και βελτιώνει την αποδοτικότητα.

Ένα συνελκτικό νευρωνικό δίκτυο αποτελείται από τρία βασικά δομικά στοιχεία. Το πρώτο είναι το επίπεδο συνέλιξης, το οποίο είναι υπεύθυνο για την εκμάθηση χαρακτηριστικών από τις εισερχόμενες εικόνες, χρησιμοποιώντας φίλτρα για να αναγνωρίσει τοπικά χαρακτηριστικά όπως άκρα, υφές και μοτίβα σε διαφορετικές θέσεις της εικόνας. Στη συνέχεια, ακολουθεί το επίπεδο max-pooling, το οποίο μειώνει τη διάσταση της εικόνας και τις υπολογιστικές απαιτήσεις, διατηρώντας τα πιο σημαντικά χαρακτηριστικά μέσω της υποδειγματοληψίας. Τέλος, το πλήρως συνδεδεμένο επίπεδο συνδέει κάθε νευρώνα με κάθε νευρώνα του επόμενου επιπέδου και χρησιμοποιεί τα χαρακτηριστικά που εξάγονται από τα προηγούμενα επίπεδα για να πραγματοποιήσει την τελική ταξινόμηση. Αυτή η αρχιτεκτονική συνδυάζει τα παραπάνω στοιχεία για να βελτιώσει την απόδοση στην αναγνώριση και ταξινόμηση εικόνων, ενώ παράλληλα μειώνει τις υπολογιστικές ανάγκες μέσω

της μείωσης της διάστασης και της κοινής χρήσης των βαρών. Η αρχιτεκτονική ενός συνελκτικού νευρωνικού δικτύου απεικονίζεται στην Εικόνα 2



Εικόνα 2: A Typical Convolutional Neural Network

Ένα CNN για να αναγνωρίσει μια εικόνα, πρέπει πρώτα να εκπαιδευτεί σε ένα μεγάλο σύνολο δεδομένων εικόνων με ετικέτα που περιέχει τα αντικείμενα ενδιαφέροντος. Κατά τη διάρκεια της εκπαίδευσης, το CNN μαθαίνει να συσχετίζει τα εξαγόμενα χαρακτηριστικά με τις σωστές ετικέτες μέσω μιας διαδικασίας αντίστροφης διάδοσης και βελτιστοποίησης. Μόλις το CNN έχει εκπαιδευτεί, μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις για νέες, εικόνες περνώντας την εικόνα μέσω του δικτύου και επιλέγοντας την ετικέτα με την υψηλότερη προβλεπόμενη πιθανότητα.

1.3.2 ResNet

Το ResNet (Residual Network) είναι ένας τύπος συνελκτικού νευρωνικού δικτύου (CNN) που χρησιμοποιείται ευρέως στον τομέα της βαθιάς μάθησης για διάφορες εργασίες αναγνώρισης και ταξινόμησης εικόνων. Μια σημαντική εφαρμογή του είναι η αναγνώριση γεωμετρικών σχημάτων και προσώπων.

Το ResNet εισάγει μια καινοτόμο αρχιτεκτονική που ονομάζεται "συνδέσεις παράλειψης" ή "υπολειμματικές συνδέσεις". Αυτές οι συνδέσεις επιτρέπουν στο δίκτυο να παραλείπει ένα ή περισσότερα στρώματα και να αντιμετωπίζει αποτελεσματικά το πρόβλημα της εξαφάνισης των κλίσεων (gradients), το οποίο είναι συχνό στα βαθιά νευρωνικά δίκτυα. Με αυτές τις συνδέσεις, το ResNet μπορεί να υποστηρίξει πολύ βαθύτερες αρχιτεκτονικές (π.χ. ResNet-50, ResNet-101) χωρίς να μειώνεται η απόδοσή του.

Κατά την εφαρμογή του στην αναγνώριση σχημάτων, το ResNet εκπαιδεύεται σε μεγάλα σύνολα δεδομένων εικόνων που περιέχουν διάφορα γεωμετρικά σχήματα. Το δίκτυο μαθαίνει να αναγνωρίζει χαρακτηριστικά κάθε σχήματος, όπως οι

γωνίες και οι πλευρές, ώστε να διακρίνει μεταξύ τετραγώνων, τριγώνων και τραπεζίων. Αφού ολοκληρωθεί η εκπαίδευση, το μοντέλο μπορεί να αναλύσει και να ταξινομήσει νέες εικόνες γρήγορα και με υψηλή ακρίβεια.

Η χρήση του ResNet για την αναγνώριση σχημάτων είναι ιδιαίτερα χρήσιμη σε εφαρμογές όπως η ρομποτική όραση, τα εκπαιδευτικά λογισμικά και οι αυτοματοποιημένοι ελεγκτές ποιότητας, προσφέροντας μια σταθερή και αξιόπιστη προσέγγιση στη συγκεκριμένη πρόκληση (κυρίως το ResNet18).

1.3.3. EfficientNet

Το EfficientNetv2 είναι μια οικογένεια συνελικτικών νευρωνικών δικτύων (CNNs) που εισήχθη για να επιτύχει υψηλή απόδοση στην ταξινόμηση εικόνων με βελτιστοποίηση τόσο της ακρίβειας όσο και της αποδοτικότητας. Έχει σχεδιαστεί για να παρέχει εξαιρετικά αποτελέσματα με χαμηλότερο υπολογιστικό κόστος σε σύγκριση με άλλα μοντέλα βαθιάς μάθησης. Η αρχιτεκτονική του EfficientNet βασίζεται σε μια καινοτόμο προσέγγιση κλιμάκωσης, όπου η ανάλυση εισόδου, το πλάτος και το βάθος του δικτύου κλιμακώνονται συντονισμένα, εξασφαλίζοντας έτσι βέλτιστη ισορροπία μεταξύ απόδοσης και υπολογιστικής ισχύος.

Κατά την αναγνώριση γεωμετρικών σχημάτων, όπως τετράγωνα, τρίγωνα και τραπέζια, το EfficientNet ξεχωρίζει για την ικανότητά του να αναγνωρίζει λεπτές διαφορές στις ιδιότητες των σχημάτων. Με την εκπαίδευση σε σύνολα δεδομένων που περιέχουν εικόνες με ποικιλία σχημάτων και διαστάσεων, το EfficientNet μπορεί να μάθει να αναγνωρίζει χαρακτηριστικά, όπως ο αριθμός των πλευρών, το μήκος των πλευρών και η γωνιακή συμμετρία.

Η ευελιξία του EfficientNet το καθιστά κατάλληλο για εφαρμογές που απαιτούν την ανάλυση και ταξινόμηση γεωμετρικών σχημάτων με υψηλή ακρίβεια. Επιπλέον, το EfficientNet διατηρεί χαμηλές απαιτήσεις σε μνήμη και υπολογιστική ισχύ, καθιστώντας το ιδανικό για ενσωμάτωση σε εφαρμογές κινητών συσκευών ή συστημάτων πραγματικού χρόνου.

1.4 Αρχιτεκτονική client-server

Η αρχιτεκτονική πελάτη-εξυπηρετητή περιγράφει ένα υπολογιστικό μοντέλο όπου ένας υπολογιστής που ονομάζεται πελάτης (client), ζητά έναν πόρο από έναν άλλο υπολογιστή που ονομάζεται εξυπηρετητής (server) μέσω μιας σύνδεσης δικτύου (Guynes & Windsor, 2011). Ο εξυπηρετητής λαμβάνει το αίτημα, το επεξεργάζεται και απαντά στον πελάτη.[6]

Τα αιτήματα προέρχονται από τον πελάτη και αποστέλλονται στον εξυπηρετητή μέσω ενός καναλιού επικοινωνίας όπως ένα δίκτυο. Οι πελάτες θα μπορούσαν να

είναι απλοί υπολογιστές που εκτελούν προγράμματα-πελάτες όπως προγράμματα περιήγησης ιστού, εφαρμογές για κινητά ή οποιαδήποτε άλλη εφαρμογή που μπορεί να ζητήσει μια υπηρεσία ή έναν πόρο (Borrie, 2004). Τέτοιοι υπολογιστές δεν χρειάζονται πολλές ρυθμίσεις παραμέτρων ή πολύπλοκα προγράμματα αφού δεν εξυπηρετούν κανένα αίτημα. Από την άλλη πλευρά, οι εξυπηρετητές είναι σύνθετοι υπολογιστές με υψηλή επεξεργαστική ισχύ με βάση τα προγράμματα στα οποία εκτελούνται. Ο εξυπηρετητής λαμβάνει τυπικά αιτήματα, τα επεξεργάζεται και απαντά στον πελάτη.[7]

Στην περίπτωση του εκπαιδευτικού ρομποτικού οχήματος, το Raspberry Pi λειτουργεί ως client, συλλέγοντας δεδομένα από τους αισθητήρες και αποστέλλοντάς τα στον server (ένα laptop), όπου γίνεται η ανάλυση εικόνας και η επεξεργασία δεδομένων υψηλής υπολογιστικής ισχύς. Όλη η επικοινωνία γίνεται με την Python socket library, μία βιβλιοθήκη που επιτρέπει στον χρήστη να δημιουργεί open sockets τα οποία μπορούν να τρέχουν βασισμένα σε πρωτόκολλα όπως τα TCP/IP και τα UDP.

1.5 Τεχνολογίες

1.5.1 Raspberry Pi

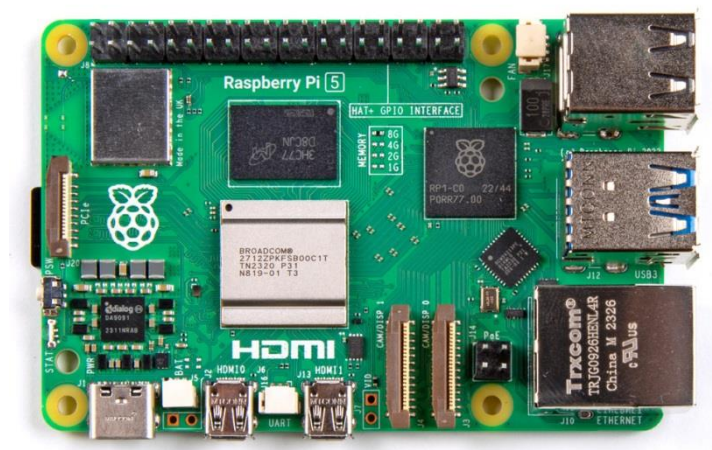
Το Raspberry Pi (/raɪ/) είναι μια σειρά μικρών υπολογιστών μονής πλακέτας (SBC) που αναπτύχθηκε στο Ηνωμένο Βασίλειο. Ο αρχικός υπολογιστής Raspberry Pi αναπτύχθηκε από το Ίδρυμα Raspberry Pi σε συνεργασία με την Broadcom. Από το 2012, όλα τα προϊόντα Raspberry Pi έχουν αναπτυχθεί από την Raspberry Pi Ltd, η οποία ξεκίνησε ως θυγατρική του Ιδρύματος που του ανήκει εξ ολοκλήρου. Το έργο Raspberry Pi αρχικά έκλινε προς την προώθηση της διδασκαλίας της βασικής επιστήμης των υπολογιστών στα σχολεία. Το αρχικό μοντέλο έγινε πιο δημοφιλές από ότι αναμενόταν, πουλώντας εκτός της αγοράς-στόχου του για ποικίλες χρήσεις όπως ρομποτική, οικιακός αυτοματισμός, βιομηχανικός αυτοματισμός και από λάτρεις υπολογιστών και ηλεκτρονικών χομπιστών, λόγω του χαμηλού κόστους, και της υιοθέτησης των προτύπων HDMI ,USB και 40 pins γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά. [8]

Η τελευταία έκδοση το Pi 5 διαθέτει έναν τετραπύρρηνο επεξεργαστή 64-bit Arm Cortex-A76 που λειτουργεί στα 2,4 GHz και προσφέρει 2-3 φορές μεγαλύτερη απόδοση στην CPU σε σχέση με το Raspberry Pi 4. Παράλληλα με τη σημαντική αύξηση των επιδόσεων γραφικών από μια GPU 800MHz VideoCore VII, διπλή έξοδο οθόνης με 4Kp60 μέσω HDMI και την υποστήριξη κάμερας από ένα ανασχεδιασμένο Raspberry Pi Image Signal Processor, παρέχει μια ομαλή εμπειρία στην επιφάνεια εργασίας. Ένα από τα βασικά πλεονεκτήματα του Raspberry Pi 5 είναι η βελτιωμένη υποστήριξη συνδεσιμότητας. Διαθέτει διπλή έξοδο HDMI με υποστήριξη 4K ανάλυσης στα 60Hz, δύο θύρες USB 3.0 για γρήγορη μεταφορά δεδομένων και μια θύρα USB-C για τροφοδοσία, η οποία υποστηρίζει μεγαλύτερη ισχύ έως 5V/5A. Επίσης, διαθέτει Gigabit Ethernet, ενώ

η ασύρματη συνδεσιμότητα Wi-Fi 5 και Bluetooth 5.0 επιτρέπουν ευέλικτη επικοινωνία σε διάφορες εφαρμογές IoT και ρομποτικής. Στον τομέα της υποστήριξης περιφερειακών, το Raspberry Pi 5 προσφέρει μια αναβαθμισμένη 40-pin GPIO θύρα, επιτρέποντας τη σύνδεση με διάφορους αισθητήρες, μοτέρ και άλλες συσκευές. Παράλληλα, το νέο RP1 I/O controller βελτιώνει τη διαχείριση των εισόδων/εξόδων, μειώνοντας την καθυστέρηση και αυξάνοντας τη σταθερότητα των δεδομένων. Αυτές οι αναβαθμίσεις καθιστούν το Raspberry Pi 5 ιδανικό για εφαρμογές ρομποτικής, όπου η αλληλεπίδραση με αισθητήρες και ενεργοποιητές είναι κρίσιμη.

Τέλος, η βελτιωμένη ενεργειακή απόδοση το 4.1 σύστημα ψύξης κάνουν το Raspberry Pi 5 πιο αξιόπιστο για συνεχή λειτουργία σε απαιτητικές εφαρμογές. Ο νέος σχεδιασμός επιτρέπει καλύτερη απαγωγή θερμότητας, ενώ η υποστήριξη ενεργής ψύξης μέσω ανεμιστήρα βοηθά στη διατήρηση χαμηλών θερμοκρασιών κατά τη λειτουργία. Συνολικά, το Raspberry Pi 5 αποτελεί ένα ισχυρό και ευέλικτο εργαλείο για εκπαιδευτικές, ερευνητικές και βιομηχανικές εφαρμογές, με σημαντικά πλεονεκτήματα στον τομέα της τεχνητής νοημοσύνης, της αυτοματοποιημένης επεξεργασίας δεδομένων και των συστημάτων αυτονομίας. [9]








Το Raspberry Pi 5 υποστηρίζει μια ποικιλία λειτουργικών συστημάτων, με το Raspberry Pi OS (πρώην Raspbian) να αποτελεί τη βασική και προτεινόμενη διανομή. Το Raspberry Pi OS είναι μια έκδοση του Debian Linux, βελτιστοποιημένη για την αρχιτεκτονική ARM του Raspberry Pi, παρέχοντας ένα ελαφρύ, σταθερό και φιλικό προς τον χρήστη περιβάλλον. Περιλαμβάνει προεγκατεστημένες εφαρμογές όπως τον Chromium browser, τον interpreter κώδικα Thonny Python IDE και τη σουίτα LibreOffice, καθιστώντας το ιδανικό για εκπαιδευτική και επαγγελματική χρήση.



Εικόνα 3:Raspberry Pi

1.5.2 Αισθητήρες (Sensors)

Αισθητήρας ονομάζεται μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος όπως η θερμοκρασία, η πίεση, η υγρασία, το φως, ο ήχος, η κίνηση, τα χρώματα. Παράγει από αυτό μία μετρήσιμη έξοδο η οποία μπορεί να αναλυθεί από συστήματα ή από ανθρώπους. Υπάρχουν Αισθητήρες Περιβάλλοντος, Χώρου/Απόστασης, Δύναμης, Κίνησης, Ήχου/Φωνής, Αναγνώρισης, Χρώματος/Φωτός, Βιομετρικοί, Ρεύματος, Υπερύθρων, Υγρών / Στάθμης, Αφής, Αερίων, Πίεσης.[10]

Αισθητήρες	Εικόνα
Περιβάλλοντος	
Χώρου/Απόστασης	
Δύναμης	
Κίνησης	
Ήχου/Φωνής	
Αναγνώρισης	
Χρώματος/Φωτός	

Υπερύθρων	
-----------	--

Πίνακας 1: Αισθητήρες

Οι αισθητήρες χρησιμοποιούνται σε καθημερινά αντικείμενα, όπως κουμπιά ανελκυστήρων ευαίσθητα στην αφή και λάμπες φωτισμού που εκπέμπουν λαμπρότερα ή απαλότερα αγγίζοντας τη βάση τους. Υπάρχουν αναρίθμητες ακόμη χρήσεις που οι περισσότεροι άνθρωποι δεν αντιλαμβάνονται. Εφαρμογές τους συναντούμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική.

1.5.3 Γλώσσα Προγραμματισμού Python

Η Python είναι αυτή τη στιγμή η πιο ευρέως χρησιμοποιούμενη αντικειμενοστραφής γλώσσα προγραμματισμού (object-oriented) υψηλού επιπέδου, με ενσωματωμένες δομές δεδομένων και δυναμικές ιδιότητες. Δημιουργήθηκε το 1991 από τον Guido van Rossum και σήμερα είναι στην έκδοση 3. Χαρακτηριστικά της είναι η φιλικότητα που έχει προς τον χρήστη, είναι απλή και κατανοητή που την καθιστά ιδανική για αρχάριους. Επιπλέον είναι συμβατή με πολλά λειτουργικά συστήματα (Windows, Linux, macOS). Χρησιμοποιείται για πολλές εφαρμογές όπως η Τεχνητή Νοημοσύνη, η Μηχανική Μάθηση (ML), για ανάπτυξη αυτοματισμών, ιστοσελίδων, παιχνιδιών καθώς και για διαχείριση δεδομένων.

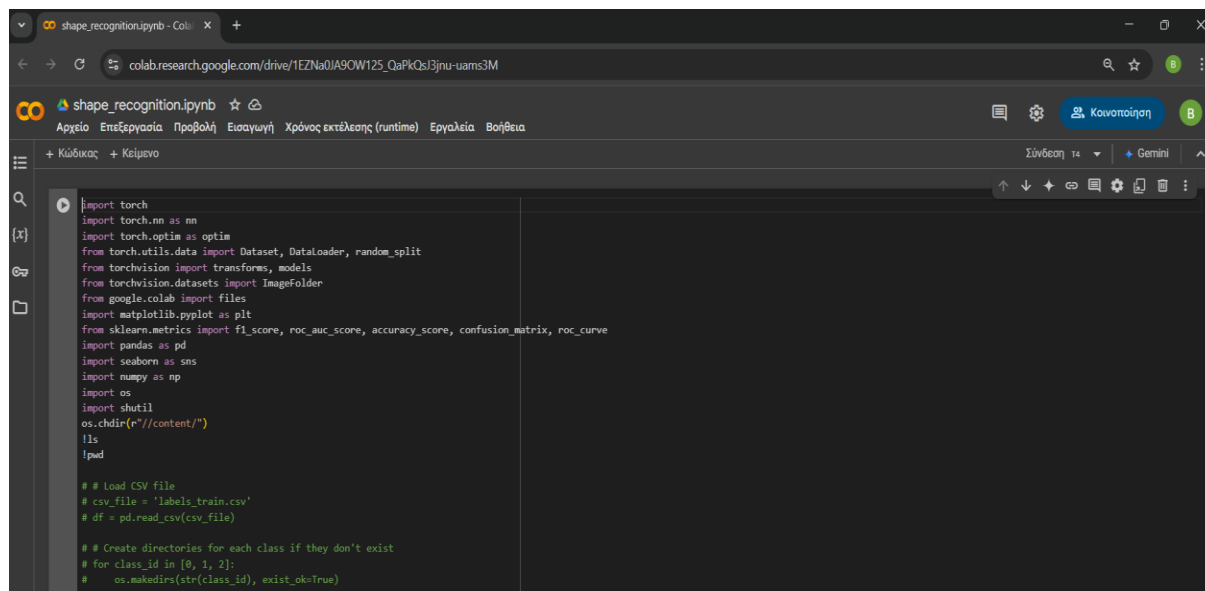
Σε μια δημοσκόπηση που δημοσιεύτηκε τον Μάιο του 2018 από την έγκυρη πύλη KD Nuggets [11], στην κατηγορία «Top Analytics, Data Science, Machine Learning Tools», διαπιστώθηκε ότι η Python χρησιμοποιείται από το 65,2% περίπου των 2000 συμμετεχόντων, έναντι 52,7% για τη RapidMiner και 48,5% για την R, τους δύο σημαντικούς ανταγωνιστές της.[12]

Η μεγάλη συλλογή βιβλιοθηκών (libraries) που διαθέτει όπως NumPy για αριθμητικούς υπολογισμούς, Pandas για επεξεργασία δεδομένων, Matplotlib για απεικόνιση δεδομένων, PyTorch /TensorFlow πλατφόρμα για deep learning και υπολογιστικά νευρωνικά δίκτυα, Pillow(PIL) για βασική επεξεργασία εικόνων, OpenCV για επεξεργασία εικόνων, βίντεο και επεξεργασία αντικειμένων, Requests για αιτήματα HTTP, os για διαχείριση λειτουργικού συστήματος, sys για διαχείριση Python runtime περιβάλλοντος, time για χρονικές λειτουργίες που επιτρέπει την επαναχρησιμοποίηση του κώδικα και μειώνουν σημαντικά το κόστος συντήρησής της.

1.5.4 Google Colab / Colaboratory

Το Google Colaboratory (γνωστό και ως Colab) είναι μια υπηρεσία cloud που βασίζεται στα Notebooks Jupyter για τη διάδοση της εκπαίδευσης και της έρευνας στη μηχανική μάθηση. Για εφαρμογές μηχανικής μάθησης και επιστήμης δεδομένων, το Google Colab είναι μια δημοφιλής πλατφόρμα με πολλά πλεονεκτήματα σε σχέση με τις ανταγωνιστικές. Η χρήση του είναι εντελώς δωρεάν και δεν χρειάζεται να γίνει εγκατάσταση κάποιου λογισμικού ή να δημιουργηθεί ένα τοπικό περιβάλλον ανάπτυξης. Όλα είναι προσβάσιμα από οποιονδήποτε υπολογιστή με σύνδεση στο διαδίκτυο και φιλοξενούνται στους εξυπηρετητές της Google. Η Colab προσφέρει δωρεάν πρόσβαση σε ισχυρές GPU που μπορούν να συντομεύσουν σημαντικά τη διάρκεια εκπαίδευσης του μοντέλου βαθιάς μάθησης. Η εισαγωγή και η εξαγωγή δεδομένων απλοποιούνται χάρη στην απρόσκοπτη ενσωμάτωση του Colab με άλλες υπηρεσίες της Google, όπως το Google Drive και τα Φύλλα Google. Χρησιμοποιείται η γλώσσα Python .[13]

Μελέτες έχουν καταλήξει ότι είναι πιο γρήγορο να εκπαιδευτεί ένα CNN στον επιταχυνόμενο υλικό (HW) εκτέλεσης του Colaboratory παρά να χρησιμοποιηθούν 20 φυσικοί πυρήνες ενός εξυπηρετητή Linux. Η απόδοση της GPU που διατίθεται από το Colaboratory μπορεί να είναι αρκετή για πολλά προφίλ ερευνητών και φοιτητών. Ωστόσο, αυτοί οι δωρεάν πόροι υλικού απέχουν πολύ από το να λύσουν απαιτητικά προβλήματα του πραγματικού κόσμου και δεν είναι επεκτάσιμοι. Ο πιο σημαντικός περιορισμός που βρέθηκε είναι η έλλειψη πυρήνων CPU.[13]



```
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader, random_split
from torchvision import transforms, models
from torchvision.datasets import ImageFolder
from google.colab import files
import matplotlib.pyplot as plt
from sklearn.metrics import f1_score, roc_auc_score, accuracy_score, confusion_matrix, roc_curve
import pandas as pd
import seaborn as sns
import numpy as np
import os
import shutil
os.chdir(r"//content/")
!ls
!pwd

## Load CSV file
# csv_file = 'labels_train.csv'
# df = pd.read_csv(csv_file)

## Create directories for each class if they don't exist
# for class_id in [0, 1, 2]:
#     os.makedirs(str(class_id), exist_ok=True)
```

Εικόνα 4 : Google Colab

1.6 Επίλογος

Στο κεφάλαιο αυτό έγινε μια προσπάθεια ανάλυσης των βασικών εννοιών που θα χρησιμοποιηθούν στην διπλωματική. Δίνονται εισαγωγικά στοιχεία για τον όρο εκπαιδευτική ρομποτική ώστε να γίνει αντιληπτό γιατί γίνεται χρήση στην παρακάτω εργασία. Η Βαθιά Μάθηση αναλύεται πιο πολύ με αναφορές στα Συνελικτικά νευρωνικά δίκτυα και κυρίως στα μοντέλα Resnet και Efficientnet στα οποία στηρίχθηκε η εκπαίδευση του μοντέλου για την αναγνώριση αντικειμένων. Κρίθηκε απαραίτητο να αναφερθεί και η πλατφόρμα Raspberry και οι αισθητήρες που χρησιμοποιήθηκαν στην κατασκευή του οχήματος. Η γλώσσα προγραμματισμού Python με τις βιβλιοθήκες που περιγράφηκαν είναι αυτή με την οποία δημιουργήθηκαν όλοι οι κώδικες της εργασίας. Τέλος το όχημα (client) συνδέονταν με τον υπολογιστή (server) μέσω της αρχιτεκτονικής client-server.

Κεφάλαιο 2ο: Θεωρητικό υπόβαθρο

2.1 Συλλογή δεδομένων

Η προετοιμασία των δεδομένων ξεκινά με την ταξινόμηση εικόνων που απεικονίζουν 3D σχήματα. Την κλάση 0 που είναι Κύβος, την κλάση 1 που είναι η Τριγωνική σφήνα και την κλάση 2 που είναι η Τραπεζοειδής Πυραμίδα. Συλλέχθηκαν 175 δείγματα.



Εικόνα 5: Κύβος

Εικόνα 6: Τριγωνική σφήνα

Εικόνα 7: Τραπεζοειδής Πυραμίδα

Η συλλογή των δεδομένων έγινε από την κάμερα του οχήματος. Πρώτα επιδιώχθηκε η απομακρυσμένη επικοινωνία οχήματος με τον υπολογιστή μέσω αρχιτεκτονικής client-server. Client είναι το όχημα, server ο υπολογιστής και με χρήση κώδικα rpython το όχημα μπορούσε να κινηθεί με τις επιλογές του χρήστη. Ο χρήστης παρακολουθώντας, μέσω της κάμερας την διαδρομή, μπορούσε πατώντας το πλήκτρο C, να συλλεγεί φωτογραφία και να την στέλνει στον server. Εκεί γίνονταν και η αποθήκευση της σε φάκελο. Όλη η συλλογή βρίσκεται στον server. Για την ταυτόχρονη εκτέλεση των παραπάνω διαδικασιών έγινε χρήση multithreading μέσω της βιβλιοθήκης threading.

Μετά γίνονταν ο διαχωρισμός σε τρεις φακέλους, στον φάκελο μηδέν (0) αποθηκεύονταν τα σχήματα κύβου, στον φάκελο ένα (1) οι εικόνες τριγωνικής πυραμίδας και στον φάκελο δύο (2) οι τραπεζοειδή πυραμίδες. Αυτή η ολοκληρωμένη συλλογή αποτέλεσε τη βάση του συνόλου δεδομένων (train dataset) η οποία τοποθετήθηκε στο google drive με όνομα train_d. Επιπλέον δημιουργήθηκε ένας φάκελος με όνομα test_image όπου τοποθετήθηκαν εικόνες και από τις παραπάνω κλάσεις αλλά και από άλλα γεωμετρικά σχήματα για να χρησιμοποιηθούν στο τέλος στην αξιολόγηση του εκπαιδευμένου μοντέλου.

2.2 Προετοιμασία δεδομένων

Η διαδικασία της εκπαίδευσης ξεκινά στο Google Colab με την ενσωμάτωση προηγμένων βιβλιοθηκών της Python, όπως το PyTorch και το TorchVision, που διευκολύνουν τη διαχείριση και επεξεργασία των δεδομένων εικόνας, καθώς και την ανάπτυξη και εκπαίδευση των νευρωνικών δικτύων.[14] Επιπλέον ζητείται η άδεια για πρόσβαση σε λογαριασμό Google που ανήκει το σημειωματάριο.

Φορτώνεται στη μνήμη το σύνολο των δεδομένων και ξεκινάει η εφαρμογή μιας σειράς μετασχηματισμών για τη βελτίωση της ποιότητας και της ενιαίας αναπαράστασης των δεδομένων. Αυτό περιλαμβάνει τη διαδικασία του padding με την συνάρτηση SquarePad, δηλαδή να κάνει την εικόνα τετράγωνη προσθέτοντας μαύρο γέμισμα στις πλευρές που χρειάζεται. Στην συνέχεια γίνονται μια σειρά από μετασχηματισμούς καλώντας την βιβλιοθήκη transforms.Compose για να προετοιμάσει τις προηγούμενες εικόνες για εκπαίδευση σε ένα μοντέλο μάθησης. Οι εικόνες που είναι ήδη τετράγωνες γίνεται αλλαγή μεγέθους σε 224x224 με την εντολή transform.Resize ((224,224)) που είναι το πρότυπο μέγεθος για πολλά Συνελικτικά δίκτυα όπως το ResNet. Για να γίνει αύξηση της γενίκευσης του μοντέλου προστέθηκε μια οπτική παραλλαγή με την χρήση της εντολής transform.RandomHorizontalFlip(). Τυχαία οριζόντια αναστροφή, γίνεται με την RandomRotation(20) όπου περιστρέφεται η εικόνα κατά μία τυχαία γωνία μέσα σε εύρος [-20ο,20ο] και αυτό βελτιώνει την αντοχή του μοντέλου. Η εφαρμογή τυχαίων χρωματικών αλλαγών για να αυξηθεί η γενικευσιμότητα του μοντέλου γίνεται με την

transform.ColorJitter(brightness=0.1, contrast=0.1).

Η εντολή **transforms.ToTensor()** μετατρέπει την εικόνα από μορφή PIL ή NumPy array σε μορφή tensor.

Από την συλλογή αυτή έγινε χωρισμός (split) σε 80% για το train set και το 20% για το test set. Στην συνέχεια το train set χωρίζεται σε 90% train set και το 10% σε validation set. Σύμφωνα με τα δεδομένα της συλλογής από τα 175 σχήματα στον πρώτο διαχωρισμό τα 140 ήταν το train set και τα 35 ήταν το test set. Ο δεύτερος διαχωρισμός έχει από τα 140 σχήματα τα 126 για το train set και τα 14 για το validation.

Για τον έλεγχο των τιμών από τα set υπάρχουν εκτυπώσιμα μηνύματα.

```

↔ Total dataset size: 175
   Train set size: 126
   Validation set size: 14
   Test set size: 35
   tensor([0.3476, 0.3565, 0.2958])

```

Εικόνα 8: Τιμές από τα dataset

Ο υπολογισμός των βαρών (weights) είναι πολύ σημαντικό στοιχείο στην εκπαίδευση καθώς υπολογίζοντας τα βάρη κάθε κλάσης μπορεί να διασφαλιστεί

ότι είναι κλιμακωμένα σωστά και αν τυχόν κάποια κλάση έχει λιγότερα δείγματα να πάρει μεγαλύτερο βάρος. Είναι σαφές ότι κλάσεις με τα περισσότερα δείγματα θα έχουν και το μικρότερο βάρος στην εκπαίδευση. Επιπλέον οι μαθηματικοί υπολογισμοί χρησιμοποιούνται σε συναρτήσεις απώλειας (loss function) .

2.3 Εκπαίδευση μοντέλου

Μετά την προετοιμασία των δεδομένων, επιλέχθηκε ένα προ-εκπαιδευμένο μοντέλο από μια σειρά διαθέσιμων αρχιτεκτονικών δικτύων όπως ResNet, DenseNet, EfficientNet και Swin. Όλα τα μοντέλα δοκιμάστηκαν και σε 25 και σε 40 εποχές. Το Swin κατά την δοκιμή τόσο σε 40 όσο και σε 25 εποχές δημιουργούσε μεγάλη βεβαιότητα για τις κλάσεις αλλά στην αξιολόγηση δεν μπορούσε να κάνει καλές προβλέψεις. Το Desnet121 και το Desnet161 δεν ήταν καλό στις προβλέψεις. Το Efficientnet_v2_s ήταν πολύ καλό στις προβλέψεις το ίδιο και το Resnet18.

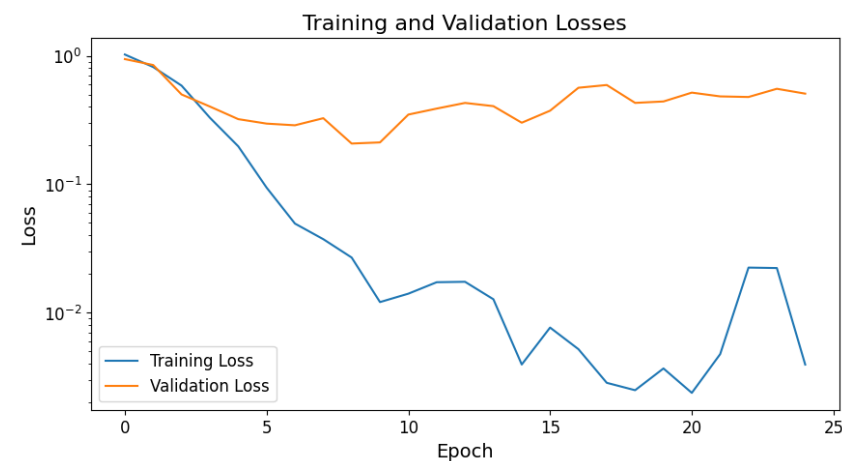
Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο εκπαιδεύεται σε προκαθορισμένο αριθμό εποχών, χρησιμοποιώντας τη συνάρτηση απώλειας CrossEntropyLoss. Ο Adam optimizer χρησιμοποιείται για τη βελτιστοποίηση των παραμέτρων του μοντέλου. Οι φάσεις εκπαίδευσης (training) και η επικύρωση (validation) εναλλάσσονται για την καλύτερη παρακολούθηση της απόδοσης του μοντέλου. Υπολογίστηκαν οι δύο πιο κοινές μετρικές αξιολόγησης μοντέλων το F-score που δίνει έμφαση στη σωστή ανίχνευση των κλάσεων με λίγα δείγματα και η Ακρίβεια (accuracy) ή οποία υπολογίζει την αναλογία των σωστών προβλέψεων προς το συνολικό αριθμό.

Στην συνέχεια εμφανίζονται τα διαγράμματα σφάλματος που παρατηρούνται στην εκπαίδευση και την επικύρωση με την βοήθεια της βιβλιοθήκης matplotlib που απεικονίζει την εξέλιξη της απώλειας εκπαίδευσης (training loss) και την απώλεια επικύρωσης (validation loss) σε κάθε εποχή εκπαίδευσης. Η ιδανική κατάσταση θα πρέπει να δείχνει την καμπύλη training loss και validation loss να μειώνεται και να συγκλίνει. Αυτό σημαίνει ότι το μοντέλο γενικεύει καλά στα δεδομένα. Αν η training loss μειώνεται αλλά η validation loss παραμένει υψηλή ή αρχίζει να αυξάνεται τότε το μοντέλο έχει υπερ-εκπαιδευτεί και έχουμε το φαινόμενο Overfitting.



Εικόνα 9: Training and Validation Losses/Resnet18

Η μπλε γραμμή αντιπροσωπεύει το training loss δηλαδή το σφάλμα αναγνώρισης κατά την εκπαίδευση και η πορτοκαλί γραμμή το validation loss το σφάλμα αναγνώρισης κατά τη δοκιμή σε δεδομένα επικύρωσης. Το διάγραμμα του μοντέλου Resnet18 δείχνει ότι και οι δυο γραμμές μειώνονται με την αύξηση των epochs κάτι που δηλώνει ότι το μοντέλο μαθαίνει αποτελεσματικά. Η διαφορά μεταξύ τους είναι μικρή όποτε το μοντέλο γενικεύει καλά και δεν δημιουργεί overfitting.



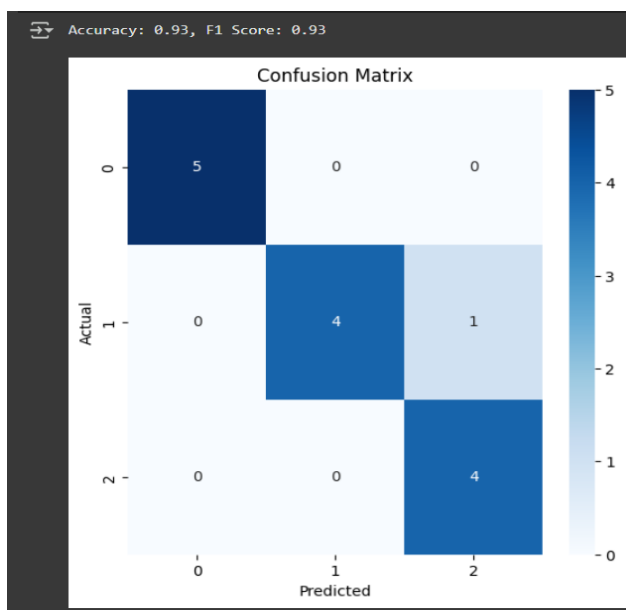
Εικόνα 10: Training and Validation Losses/Effcientnetv2_s

Στο διάγραμμα το training loss μειώνεται σταθερά με την πάροδο των epochs δείχνοντας ότι μαθαίνει αποτελεσματικά. Το validation loss παρουσιάζει αστάθεια ειδικά μετά από τις 10 epochs . Αυτό μπορεί να είναι overfitting

δηλαδή το μοντέλο μαθαίνει να ταιριάζει υπερβολικά με τα δεδομένα και έτσι χάνει τη γενίκευση στα δεδομένα επικύρωσης.

2.4 Προβλέψεις από το εκπαιδευμένο μοντέλο

Τώρα το μοντέλο έχει ολοκληρώσει την εκπαίδευση και με την εντολή `model.eval()` γίνεται η χρήση του. Κάθε εικόνα υποβάλλεται σε επεξεργασία ξεχωριστά και στη συνέχεια περνάει από το μοντέλο για να παραχθεί μια πρόβλεψη. Κατά τη διάρκεια της αξιολόγησης, οι παράγωγοι των βαρών του μοντέλου δεν υπολογίζονται (`torch.no_grad()`), καθώς δεν πραγματοποιούνται ενημερώσεις των βαρών. Αυτό μειώνει τους περιττούς υπολογισμούς και βελτιστοποιεί την ταχύτητα. Η προβλεπόμενη κλάση για κάθε εικόνα καθορίζεται από τη μέγιστη πιθανότητα, η οποία εξάγεται μετά την εφαρμογή της συνάρτησης ενεργοποίησης SoftMax. Υπολογίζονται οι μετρικές `F1_score` και `Accuracy` για να δημιουργηθεί στο τέλος ένας πίνακας σύγχυσης (Confusion Matrix) που ουσιαστικά είναι ένας πίνακας ταξινόμησης των σωστών και λανθασμένων προβλέψεων για κάθε κλάση.

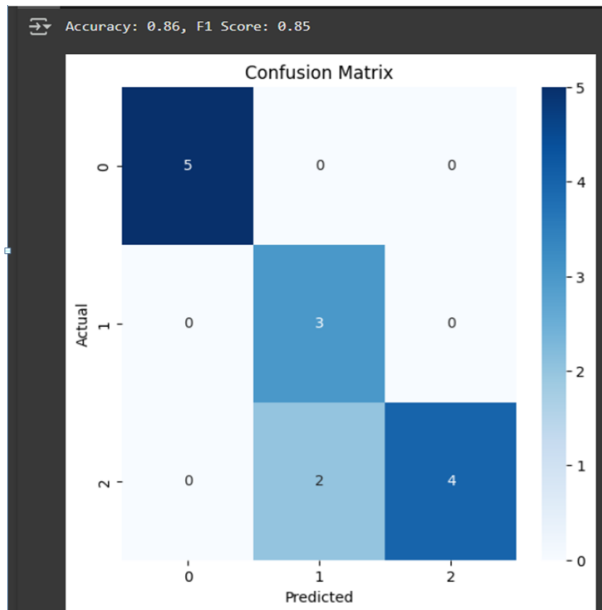


Εικόνα 11: Confusion Matrix ResNet18

Στην Εικόνα 11, παρουσιάζεται ο πίνακας σύγχυσης των προβλέψεων του μοντέλου ResNet18 για τρεις κατηγορίες (0, 1, 2). Από τα στοιχεία του πίνακα βλέπουμε ότι η κλάση 0 έχει υψηλό ποσοστό σωστής ταξινόμησης με 5 εικόνες να ταξινομούνται σωστά, ενώ καμία δεν ταξινομήθηκε λάθος. Η κλάση 1 με 4 σωστές ταξινομήσεις και 1 εικόνα που ταξινομήθηκε λανθασμένα, ενώ η κλάση 2 έχει ίδια αποτελέσματα με την κλάση 0. Η γενική ακρίβεια του μοντέλου είναι

Κεφάλαιο 2

0.93 και το F1 Score είναι 0.93, κάτι που δείχνει ότι το μοντέλο έχει μια πολύ καλή συνολική απόδοση στο σύνολο δεδομένων που δοκιμάστηκε. Ίσως χρειάζεται επιπλέον δεδομένα για την κλάση 1.



Εικόνα 12: Confusion Matrix EfficientNet2_s

Στην Εικόνα 12, παρουσιάζεται ο πίνακας σύγχυσης των προβλέψεων του μοντέλου EfficientNet2_s για τρεις κατηγορίες (0, 1, 2). Από τα στοιχεία του πίνακα βλέπουμε ότι η κλάση 0 έχει υψηλό ποσοστό σωστής ταξινόμησης με 5 εικόνες να ταξινομούνται σωστά, ενώ καμία δεν ταξινομήθηκε λάθος. Η κλάση 1 έχει ίδια αποτελέσματα με την κλάση 0, ενώ η κλάση 2 4 σωστές ταξινομήσεις και 2 εικόνες που ταξινομήθηκαν λανθασμένα. Η γενική ακρίβεια του μοντέλου είναι 0.85 και το F1 Score είναι 0,86 κάτι που δείχνει ότι το μοντέλο έχει μια πολύ καλή συνολική απόδοση στο σύνολο δεδομένων που δοκιμάστηκε. Ίσως χρειάζεται επιπλέον δεδομένα για την κλάση 2.

2.5 Επίλογος

Στο κεφάλαιο αυτό αναλύθηκε η διαδικασία εκπαίδευσης ενός μοντέλου. Από τα πρώτα στάδια που είναι η συλλογή των δεδομένων, η προετοιμασία τους με συγκεκριμένες συναρτήσεις (transform, padding) στην εκπαίδευση προ-εκπαιδευμένων μοντέλων και τέλος στην αξιολόγησή τους. Σε όλες τις φάσεις έγινε προσπάθεια να αναλυθούν όλες οι παράμετροι και να γίνει κατανοητό και για τους μη έμπειρους αναγνώστες. Επίσης η χρήση γραφικών αναπαραστάσεων όπως ο Confusion Matrix και το Training and Validation Losses έκανε σαφή το λόγο που επιλέχθηκε τελικά το μοντέλο ResNet18.

Κεφάλαιο 3ο: Σχεδίαση και Ανάπτυξη Ρομποτικού Οχήματος








3.1 Εισαγωγή

Το κεφάλαιο αυτό θα χωριστεί σε δύο μέρη , το πρώτο θα αναλύσει το υλικό (Hardware) που χρησιμοποιήθηκε και το δεύτερο μέρος θα αναλύσει το λογισμικό κομμάτι (Software).

Η ανάπτυξη ενός εκπαιδευτικού ρομποτικού οχήματος απαιτεί μια ολοκληρωμένη προσέγγιση που περιλαμβάνει τη μηχανική σχεδίαση, την επιλογή υλικών και εξαρτημάτων, τον προγραμματισμό και την ενσωμάτωση αλγορίθμων τεχνητής νοημοσύνης. Σε αυτό το κεφάλαιο, παρουσιάζεται η αρχιτεκτονική του ρομποτικού συστήματος, η κατανομή των λειτουργικών μονάδων, καθώς και οι μέθοδοι επικοινωνίας και ελέγχου. Ειδικότερα, γίνεται αναφορά στην επιλογή του Raspberry Pi 5 ως κεντρικής μονάδας επεξεργασίας, στη χρήση αισθητήρων για πλοήγηση και αποφυγή εμποδίων, καθώς και στην ενσωμάτωση συστημάτων οπτικής αναγνώρισης με deep learning. Παράλληλα, αναλύονται οι τεχνικές απομακρυσμένου ελέγχου του οχήματος μέσω αρχιτεκτονικής client-server, διασφαλίζοντας την αποδοτική και αξιόπιστη λειτουργία του συστήματος.

3.2 Υλικό οχήματος

Μετά από έρευνα για την προμήθεια του υλικού τόσο σε φυσικά όσο και σε ηλεκτρονικά καταστήματα αποφασίστηκε η αγορά των προϊόντων του παρακάτω πίνακα.

Κατάστημα : https://grobotronics.com/		
Υλικό	Τιμή	
Raspberry pi 5 -4GB	72.9 €	
Raspberry Pi 5 - Active Cooler	5.90€	
Κάρτα μνήμης microSDHC 32GB Class 10 - SanDisk Ultra SDSQUA4-032G-GN6MA	5.6€	
Waveshave Υπέρυθρος Αισθητήρας Εμποδίων	4x3.9€	
Camera Module 8MP for Raspberry Pi 5 - 79.3° FOV	11.9€	
Robot Smart Car 4WD-Chassis 26 cm	16.9€	
Dual Motor Driver Module L298N	2x4.20€	

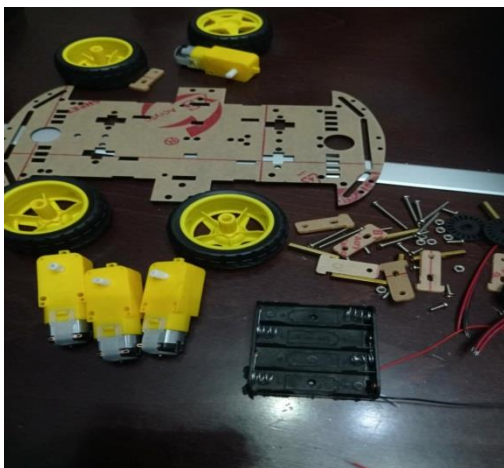
Πίνακας 2:Υλικά κατασκευής

Επιπλέον χρησιμοποιήθηκε ένα power bank 10000 mAh για ενέργεια του raspberry και τέσσερις μπαταρίες AA για ενέργεια των τροχών.



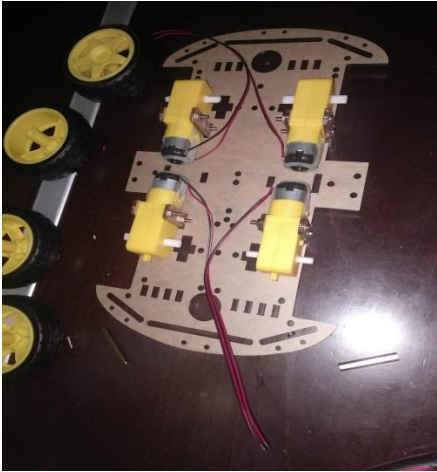
Εικόνα 13: Υλικό

Ξεκίνησε η συναρμολόγηση με το Robot Smart Car 4WD που είναι μια πλατφόρμα ρομπότ που διαθέτει τέσσερα μοτέρ με encoder και τροχούς 65 mm. Οι πλάκες πλαισίου και τα στηρίγματα είναι από μαύρο ακρυλικό (plexiglass) 3mm και έχουν οπές για την προσάρτηση πλακετών, αισθητήρων, ελεγκτών και άλλων πρόσθετων μονάδων. Στην κατασκευή της εργασίας οι κινητήρες τοποθετήθηκαν μεταξύ των δύο πλακών γιατί αν τοποθετούνταν από κάτω υπήρχε πρόβλημα με την επαφή στο έδαφος. Χρειάστηκε και κολλητήρι για να κολλήσουν τα καλώδια στα μοτέρ και να παραμένουν σταθερά. Δεν περιλαμβάνονταν μέσα στο kit ούτε οι μπαταρίες 4xAA. Στην παρακάτω εικόνα φαίνονται τα περιεχόμενα του kit.



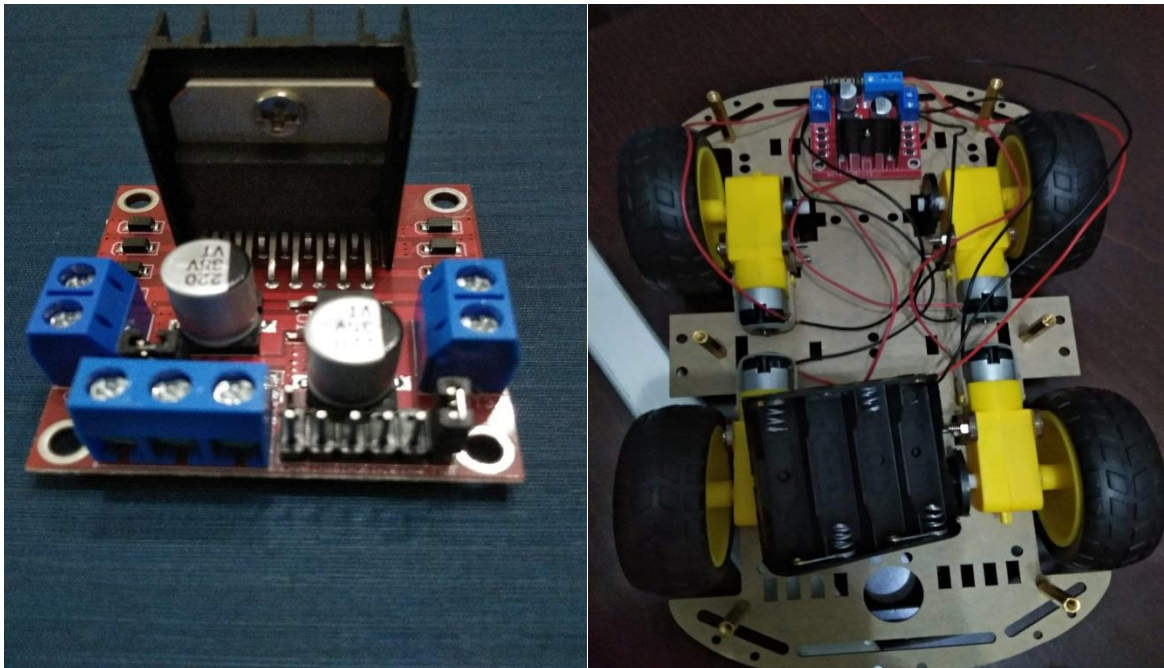
Εικόνα 14: Robot Smart Car 4WD

Δεν ήταν εύκολη η διαδικασία της συναρμολόγησης χρειάστηκαν αρκετές ώρες για να ολοκληρωθεί σωστά.



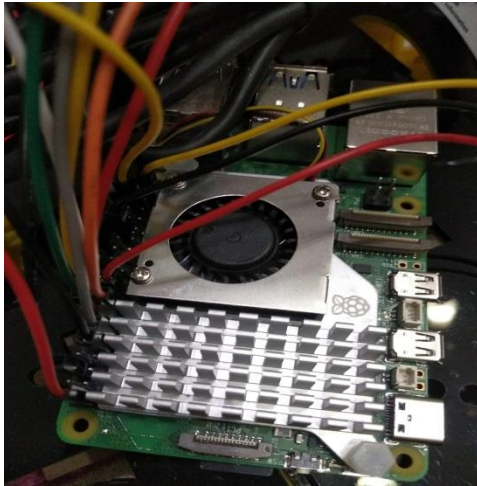
Εικόνα 15: Στάδιο συναρμολόγησης

Το L298N είναι ένα ολοκληρωμένο κύκλωμα για κινητήρες συνεχούς ρεύματος απαραίτητο για τον έλεγχο του οχήματος. Μπορεί να δώσει κίνηση στους κινητήρες, υποστηρίζει έως δύο κινητήρες με ισχύ 2A ανά κινητήρα. Η μονάδα L298N H-bridge μπορεί να χρησιμοποιηθεί με κινητήρες που έχουν τάση μεταξύ 5 και 35 V DC. Υπάρχει επίσης ενσωματωμένος ρυθμιστής 5V, οπότε αν η τάση τροφοδοσίας είναι μέχρι 12V, μπορεί επίσης να τροφοδοτηθεί με 5V από την πλακέτα. Η επικοινωνία με τον οδηγό γίνεται μέσω ψηφιακών σημάτων PWM (Pulse Width Modulation), που επιτρέπουν τον ακριβή έλεγχο της ταχύτητας των κινητήρων, καθιστώντας τον L298N μια αξιόπιστη και ευέλικτη λύση για τον έλεγχο κίνησης σε ρομποτικά οχήματα. Στην παρακάτω εικόνα φαίνεται το επόμενο στάδιο συναρμολόγησης με τον οδηγό L298N ενσωματωμένο στο όχημα.



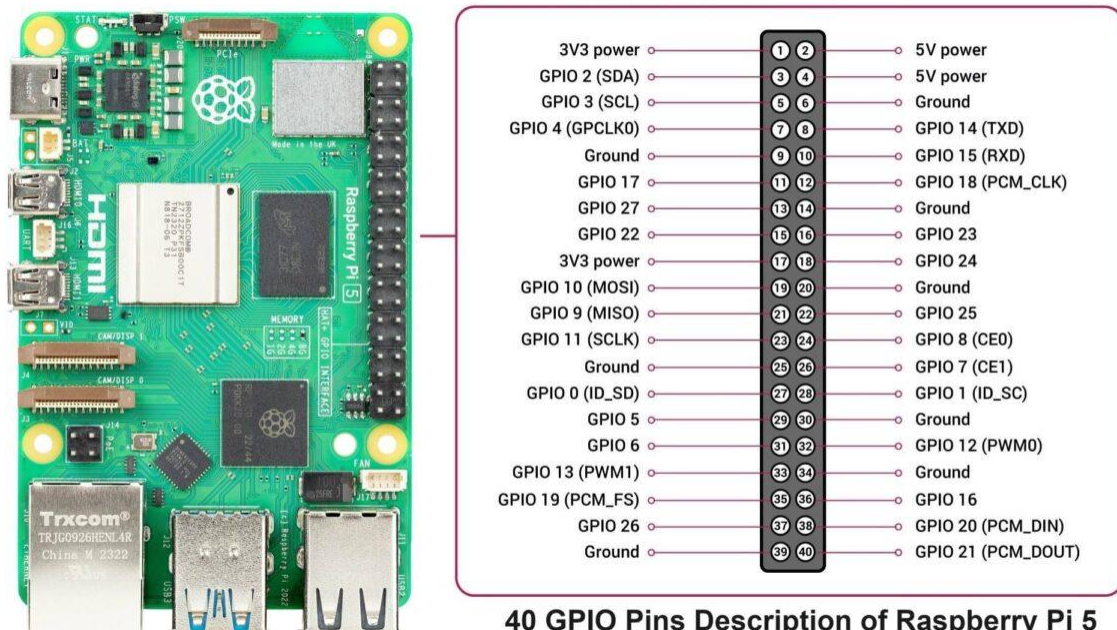
Εικόνα 16: Ενσωμάτωση και του οδηγού L298N.

Έτοιμο για το επόμενο βήμα που είναι η ενσωμάτωση της πλατφόρμας Raspberry Pi. Για ψύξη θα χρησιμοποιηθεί το Raspberry Pi Active Cooler που συνδυάζει μια μεγάλη μεταλλική ψήκτρα με έναν υψηλής ποιότητας ανεμιστήρα μεταβλητής ταχύτητας, που τροφοδοτείται και ελέγχεται μέσω του βύσματος ανεμιστήρα στο Raspberry Pi 5, που είναι συνδεδεμένος στην πλακέτα μέσω ελατηρίων των ακίδων στο ειδικό ζεύγος οπών στερέωσης.



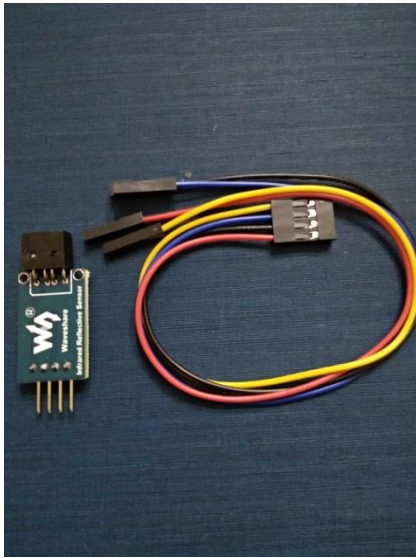
Εικόνα 17: Raspberry Pi 5

Με μεγάλη προσοχή έγινε η σύνδεση του L298N και συγκεκριμένα των ακίδων (pin) IN1, IN2, IN3 και IN4 με τα σωστά pin του Raspberry. Η κατεύθυνση του κινητήρα ελέγχεται στέλνοντας ένα σήμα HIGH ή LOW στον ηλεκτροκινητήρα για κάθε κινητήρα (ή κανάλι). Για παράδειγμα, για έναν κινητήρα, ένα HIGH στο IN1 και ένα LOW στο IN2 θα τον κάνει να στρίψει προς τη μία κατεύθυνση και ένα LOW και ένα HIGH θα κάνει να στρίψει προς την άλλη κατεύθυνση.



Εικόνα 18: Pins of Raspberry Pi 5 Πηγή: <https://www.hackatronic.com/raspberry-pi-5-pinout-specifications-pricing-a-complete-guide/>

Ο υπέρυθρος αισθητήρας εμποδίων Waveshare μπορεί να συνδεθεί στο Raspberry Pi μέσω των GPIO (General Purpose Input/Output) pins, επιτρέποντας την ανίχνευση εμποδίων σε μικρή απόσταση. Ο αισθητήρας διαθέτει τρεις ακροδέκτες: VCC (τροφοδοσία), GND (γείωση) και OUT (ψηφιακή έξοδος). Το VCC συνδέεται σε μια τάση 3.3V ή 5V από το Raspberry Pi, το GND συνδέεται στη γείωση, ενώ το OUT συνδέεται σε έναν από τους διαθέσιμους GPIO ακροδέκτες για την ανάγνωση της εξόδου του αισθητήρα. Όταν ανιχνεύσει ένα αντικείμενο, η έξοδος του αλλάζει κατάσταση (HIGH ή LOW), δίνοντας στο Raspberry Pi τη δυνατότητα να λαμβάνει τις απαραίτητες αποφάσεις, όπως η αλλαγή κατεύθυνσης ή η επιβράδυνση του ρομποτικού οχήματος.[10] Η ανάγνωση των δεδομένων γίνεται μέσω ενός απλού Python script, το οποίο χρησιμοποιεί τη βιβλιοθήκη RPi.GPIO. Αυτό έγινε και για τους δύο αισθητήρες που χρησιμοποιούνται.



Εικόνα 19: Αισθητήρας εμποδίων

Μεγάλο πρόβλημα υπήρξε με την κάμερα. Πρώτα επιλέχθηκε η Camera Module 8MP for Raspberry Pi 5 - 79.3° FOV.



Εικόνα 20: Camera Module 8MP for Raspberry Pi 5 - 79.3° FOV

Έγινε προσπάθεια να συνδεθεί στο raspberry και στα δύο module που διαθέτει αλλά δεν έγινε ποτέ αναγνώριση της κάμερας με αποτέλεσμα να μην μπορεί να υλοποιηθεί η λήψη φωτογραφιών. Έγινε προσπάθεια και με το λογισμικό και με τις ρυθμίσεις του συστήματος χωρίς όμως αποτέλεσμα. Μπορεί το πρόβλημα να είναι ότι βασίζεται στον αισθητήρα OmniVision OV5647. Αυτός ο αισθητήρας δεν υποστηρίζεται από το libcamera στο Raspberry Pi OS Bookworm, καθώς το Raspberry Pi 5 χρησιμοποιεί νέο σύστημα κάμερας. Ίσως χρειάζονταν να χρησιμοποιηθεί μια νεότερη κάμερα (IMX219 ή IMX477).

Για να αντιμετωπιστεί το παραπάνω πρόβλημα χρησιμοποιήθηκε μια κάμερα σύνδεσης USB που δεν παρουσίασε κανένα πρόβλημα.

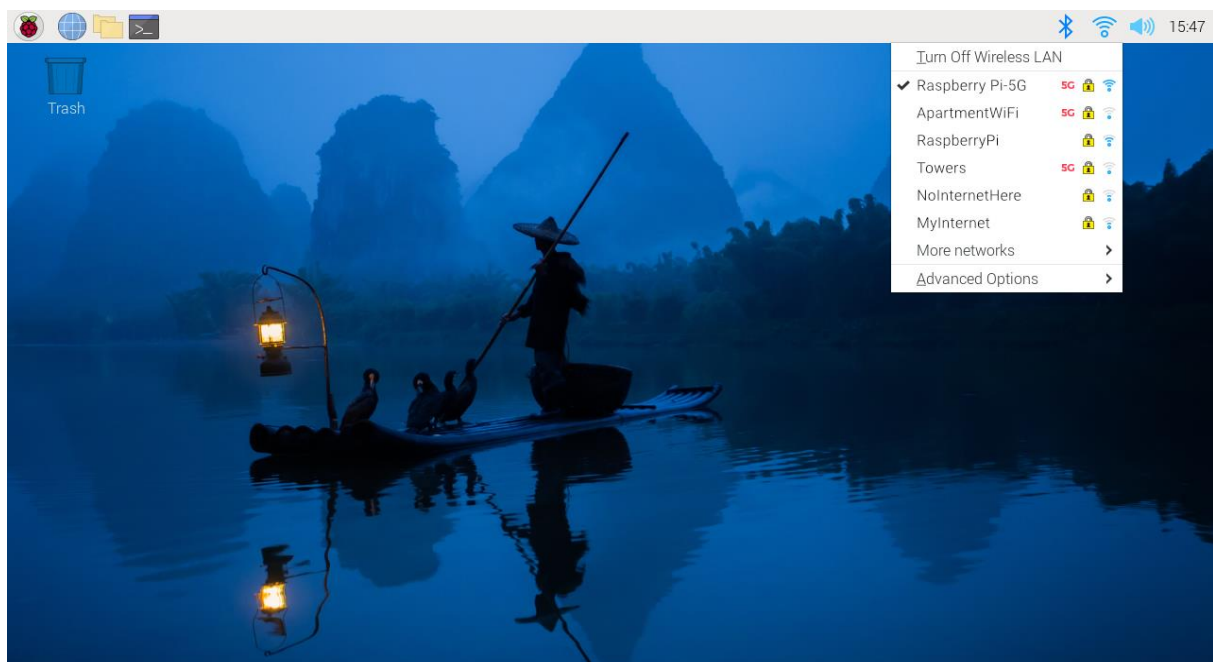
3.3 Λογισμικό Οχήματος

Η πλατφόρμα Raspberry όπως έχει ήδη αναφερθεί είναι ένας υπολογιστής και χρειάζεται λειτουργικό σύστημα . Για το λόγο αυτό αγοράστηκε μια κάρτα microSDHC 32GB .Το επίσημο λειτουργικό σύστημα είναι το Raspbian και για να γίνει η εγκατάσταση, η επίσημη σελίδα διαθέτει ένα εργαλείο το Raspberry Pi Imager όπου με πολύ αναλυτικά βήματα γίνεται η εγκατάσταση στην κάρτα SD. [15]



Εικόνα 21: Raspberry Pi Imager

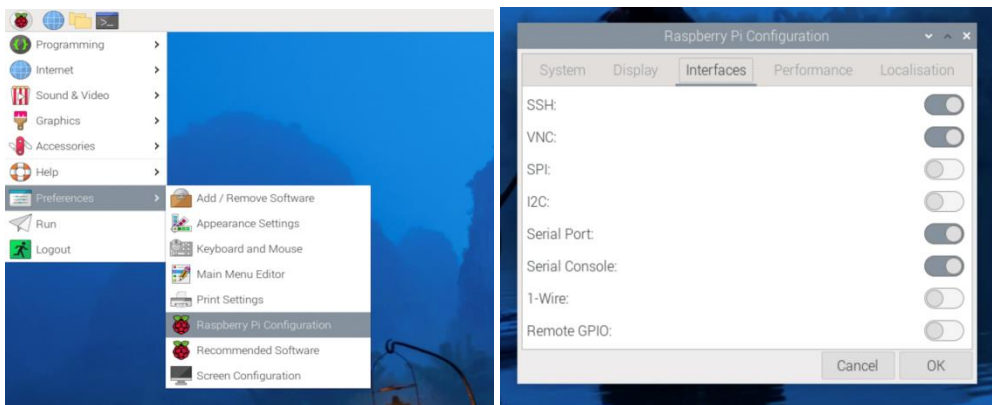
Για να μπορέσουμε να χειριστούμε το Raspberry Pi χρειάζεται μια οθόνη που συνδέεται στη microHDMI, ένα πληκτρολόγιο και ένα ποντίκι που συνδέονται σε USB θύρες και είναι έτοιμος ο υπολογιστής για να προγραμματιστεί.



Εικόνα 22: Επιφάνεια εργασίας Raspberry Pi Απομακρυσμένη πρόσβαση

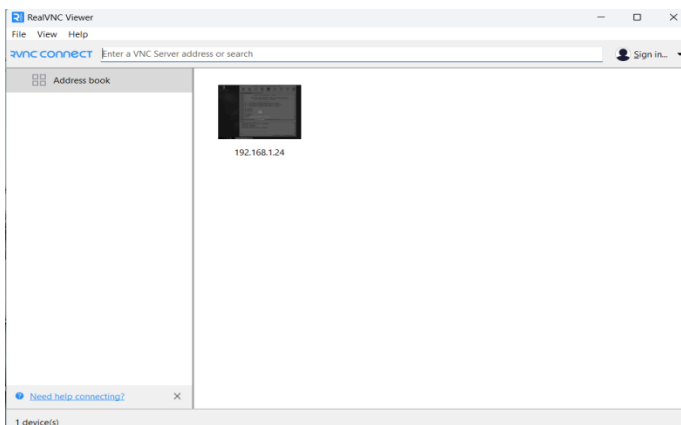
Το έργο της διπλωματικής είναι ο χειρισμός ενός οχήματος που θα κινείται στο χώρο άρα δεν μπορεί να έχουμε την παραπάνω σύνδεση. Το Virtual Network Computing (VNC) επιτρέπει τον έλεγχο της επιφάνεια εργασίας μιας συσκευής από μια άλλη. Το VNC βασίζεται στην αρχιτεκτονική πελάτη –εξυπηρετητή (client-server) . Ο πελάτης εκτελείται σε μια συσκευή με την οποία ή αλληλεπίδραση γίνεται φυσικά, όπως προσωπικό φορητό υπολογιστή, επιτραπέζιο υπολογιστή, tablet ή τηλέφωνο. Ο εξυπηρετητής τρέχει στο Raspberry Pi. Όταν χρησιμοποιείτε το VNC, ο πελάτης μεταδίδει συμβάντα πληκτρολογίου και ποντικιού στον εξυπηρετητή. Ο εξυπηρετητής εκτελεί αυτά τα συμβάντα στο Raspberry Pi και επιστρέφει ενημερώσεις οθόνης στον πελάτη. Ο πελάτης VNC εμφανίζει την επιφάνεια εργασίας του Raspberry Pi σε ένα παράθυρο.

Για να γίνει η χρήση του VNC στο Raspberry Pi, πρέπει να ενεργοποιηθεί ο εξυπηρετητής VNC. Παρακάτω φαίνονται οι ρυθμίσεις:

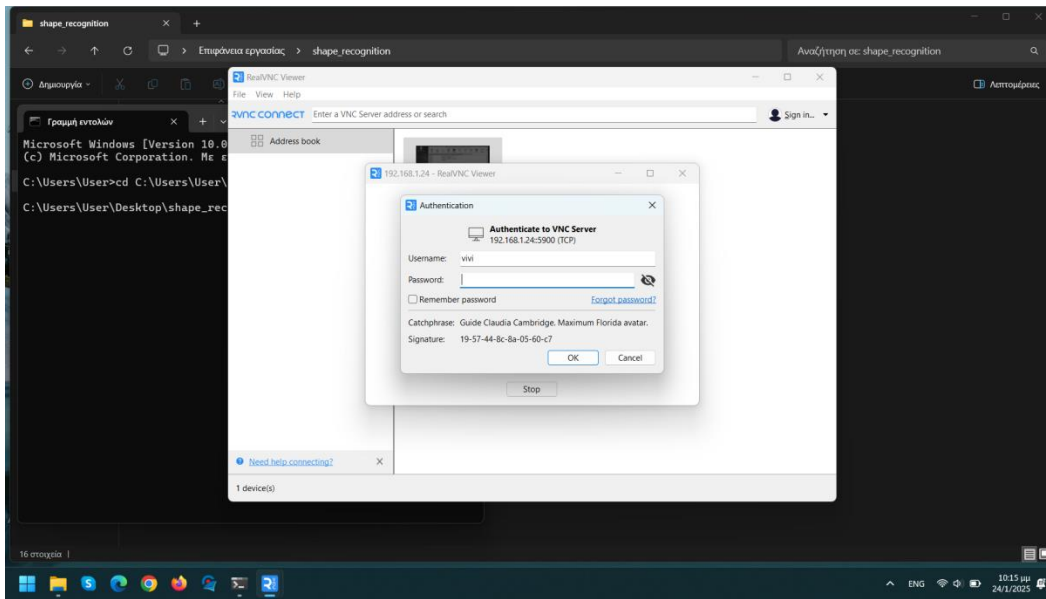


Εικόνα 23:Ρυθμίσεις στο Raspberry για το VNC

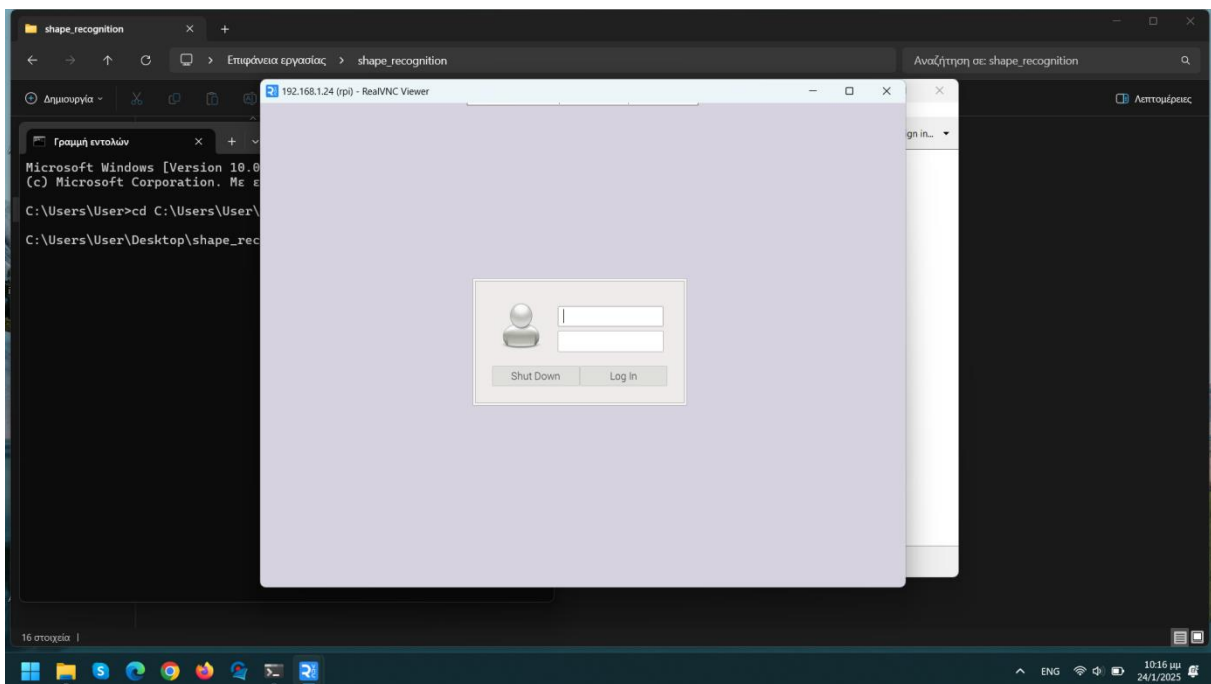
Στον υπολογιστή στον επίσημο ιστότοπο του RealVNC γίνεται η λήψη και η εγκατάσταση του RealVNC viewer. Κατά την εγκατάσταση θα πρέπει να δημιουργηθεί λογαριασμός και να εισαχθεί κωδικός που θα χρησιμοποιηθεί κάθε φορά που θα επιχειρούμε σύνδεση.



Εικόνα 24:Η πρώτη επιφάνεια του RealVNC



Εικόνα 25:Γίνεται η επιβεβαίωση των κωδικών



Εικόνα 26:Εικόνα σύνδεσης Raspberry

3.4 Χειρισμός του οχήματος με Python

Το λειτουργικό σύστημα Rasbian έχει ενσωματωμένο το εργαλείο Thorny για να γραφτεί κώδικας. Η γλώσσα Python είναι αυτή που χρησιμοποιείται. Ένα απόσπασμα του κώδικά παρουσιάζεται παρακάτω.

```
import RPi.GPIO as GPIO
```

Εισάγει τη βιβλιοθήκη RPi.GPIO για τον έλεγχο των GPIO pins του Raspberry Pi. Το as GPIO επιτρέπει να χρησιμοποιείται η αναφορά GPIO αντί για το πλήρες όνομα της βιβλιοθήκης

```
from time import sleep
```

Εισάγει τη συνάρτηση sleep , η οποία σταματά την εκτέλεση κώδικα για καθορισμένο χρόνο, συνήθως χρησιμοποιείται για καθυστερήσεις.

```
In1 = 23
```

```
In2 = 24
```

```
Ena = 25
```

```
In3 = 17
```

```
In4 = 27
```

```
Enb = 4
```

Ορίζονται τα GPIO pins του Raspberry Pi για να ελεγχθεί το μοτέρ L298N. Τα In1, In2, In3, In4 ελέγχουν την κατεύθυνση περιστροφής , τα Ena και Enb ελέγχουν την ισχύ μέσω PWM(έλεγχος πλάτους παλμών).

```
GPIO.setmode(GPIO.BCM)
```

Ορίζει τη λειτουργία αριθμοδότησης των pins σε BCM (Broadcom SOC Channel).

```
GPIO.setup(In1,GPIO.OUT)
```

```
GPIO.setup(In2,GPIO.OUT)
```

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

```
GPIO.setup(In3,GPIO.OUT)
```

```
GPIO.setup(In4,GPIO.OUT)
```

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

Κάθε pin (In1, In2, Ena, In3, In4, Enb) ορίζεται ως έξοδος (OUTPUT). Αυτό σημαίνει ότι θα χρησιμοποιούνται για να στέλνουν σήμα στο μοτέρ.

```
power_a = GPIO.PWM(Ena,100)
```

```
power_a.start(100)
```

```
power_b = GPIO.PWM(Enb,100)
```

```
power_b.start(100)
```

Δημιουργεί αντικείμενο PWM (έλεγχου πλάτους παλμών) για το pin Ena και το Enb με συχνότητα 100 Hz. Το PWM χρησιμοποιείται για τη ρύθμιση της ισχύος που παρέχεται στα μοτέρ. Ξεκινά το PWM με duty cycle 100% (δηλαδή πλήρης ισχύς).

Στον κώδικά γίνεται έλεγχος κίνησης με τα πλήκτρα f (forward) το όχημα κινείται μπροστά, με το πλήκτρο b(back) το όχημα κινείται πίσω, με το πλήκτρο r(right) το όχημα κάνει στροφή προς τα δεξιά ,με το πλήκτρο l(left) το όχημα κάνει στροφή αριστερά . Επιπλέον χρησιμοποιούνται τα πλήκτρα s(stop) για να σταματήσει η κίνηση και το e(exit) για να τερματίσει το πρόγραμμα. Όλες οι παραπάνω κινήσεις γίνονται με τον χειρισμό των GPIO pins .

3.5 Client-Server

Με την βιβλιοθήκη socket η Python ενώνει τον υπολογιστή με το Raspberry Pi για την αποστολή –παραλαβή bytes μέσω του δικτύου.

Πρώτα για τον client ορίζονται η IP διεύθυνση καθώς και η port δηλαδή η θύρα που θα χρησιμοποιηθεί για την σύνδεση. Στο πρόγραμμα ορίζεται η θύρα 5000.

```
client_socket.connect((host, port))
```

Συνδέει τον πελάτη στον εξυπηρετητή με την διεύθυνση host και τη θύρα port. Αν ο server δεν λειτουργεί ή είναι κλειστός , ο κώδικας θα προκαλέσει σφάλμα.

Για τον server δηλώνεται η IP που θα δέχεται συνδέσεις , αν είναι τοπικά μπορεί να εισαχθεί η 127.0.0.1 . Αν δηλωθεί η 0.0.0.0 τότε δέχεται συνδέσεις από οποιαδήποτε διεύθυνση IP . Ορίζεται και η θύρα επικοινωνίας του Server.

```
server_socket.bind((host, port))
```

```
server_socket.listen(1)
```

Συνδέει το socket με τη διεύθυνση host και τη θύρα port. Αυτό επιτρέπει στον server να "ακούει" για συνδέσεις στη συγκεκριμένη διεύθυνση και θύρα. Θέτει

τον server σε λειτουργία ακρόασης, περιμένοντας συνδέσεις. Ο αριθμός 1 δηλώνει ότι μπορεί να διαχειριστεί έως και μία αναμονή σύνδεσης τη φορά.

Όταν τελειώσει το πρόγραμμα τερματίζει και η σύνδεση που έχει δημιουργηθεί.

```
conn.close()
```

```
server_socket.close()
```

3.6 Multithreading

Το επόμενο βήμα μετά τον έλεγχο κινήσεων του οχήματος είναι ο ταυτόχρονος χειρισμός της κάμερας. Μία κάμερα που είναι ενσωματωμένη στο Raspberry Pi ανοίγει και μπορεί ο χρήστης να παρακολουθεί ζωντανά την κίνηση του οχήματος καθώς και να συλλέγονται φωτογραφίες με το πάτημα του πλήκτρου C. Για να πραγματοποιηθούν ταυτόχρονα αυτές οι διαδικασίες είναι απαραίτητο να γίνει χρήση Multithreading όπου στην Python γίνεται με την βιβλιοθήκη threading. Στην ουσία είναι δύο νήματα, ο έλεγχος του οχήματος και η διαχείριση της κάμερας.

```
t1 = threading.Thread(target=camera_stream)
```

```
t2 = threading.Thread(target=control_robot)
```

3.7 Επίλογος

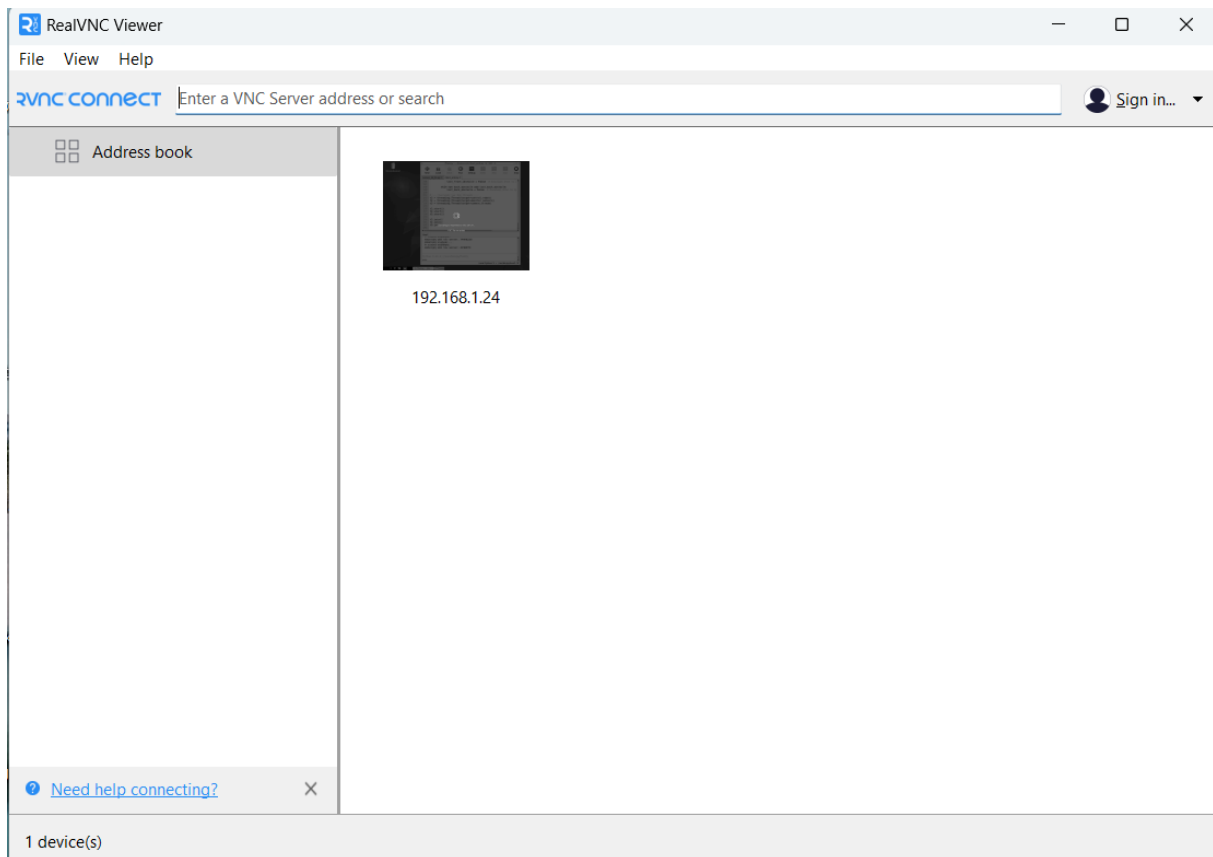
Συνοψίζοντας, η έννοια της επικοινωνίας client-server, σε συνδυασμό με την ισχύ του multithreading και τη χρήση πλατφορμών όπως το Raspberry Pi, ανοίγει απεριόριστες δυνατότητες για την ανάπτυξη σύγχρονων έργων όπως είναι το ρομποτικό όχημα. Η αρχιτεκτονική client-server παρέχει μια σταθερή βάση για την επικοινωνία και την ανταλλαγή δεδομένων μεταξύ των συσκευών. Παράλληλα, το multithreading επιτρέπει την αποτελεσματική εκμετάλλευση των διαθέσιμων πόρων, εξασφαλίζοντας παράλληλη εκτέλεση εργασιών και βελτιστοποιώντας τη συνολική απόδοση. Το Raspberry Pi, ως ένας ευέλικτος και οικονομικός υπολογιστής, γίνεται το ιδανικό εργαλείο για την εκμάθηση και την πρακτική εφαρμογή αυτών των τεχνολογιών.

Κεφάλαιο 4ο: Αποτελέσματα

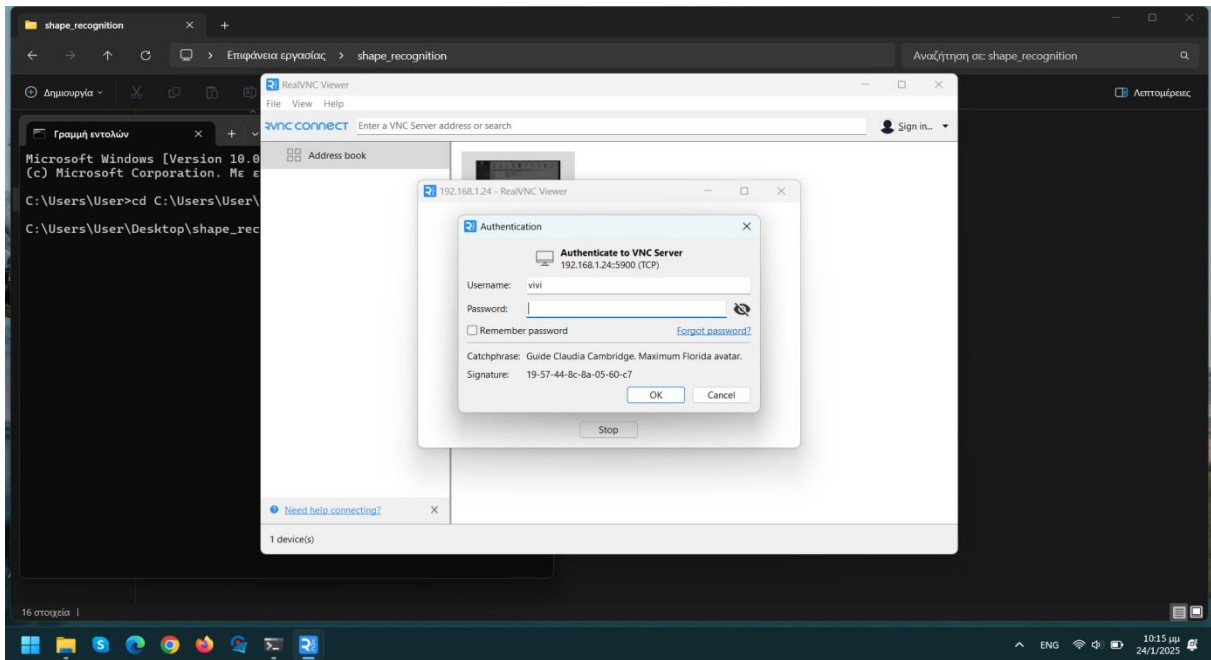
Το έργο έχει τα εξής βήματα:

Βήμα 1

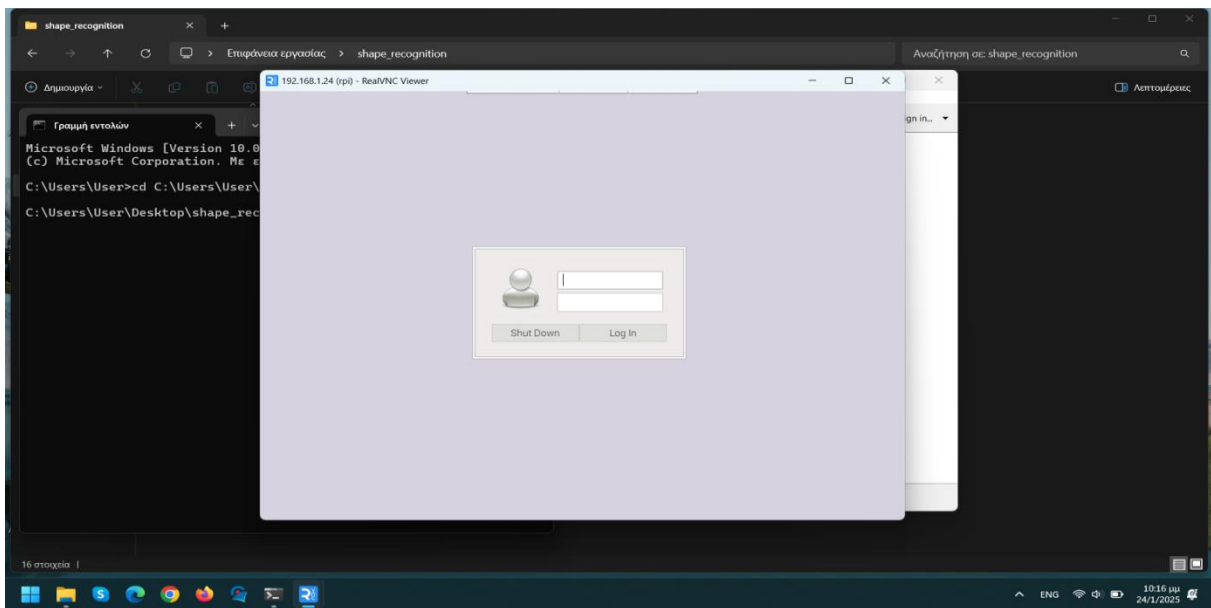
Στον υπολογιστή υπάρχει ήδη εγκατεστημένη η εφαρμογή RealVNC Viewer με την οποία συνδέεται απομακρυσμένα το Raspberry Pi με τον υπολογιστή και εμφανίζεται η επιφάνεια εργασίας του. Για να επιτευχθεί θα πρέπει να εισαχθούν το username και το password ώστε να αναγνωριστεί ο χρήστης και να επιτραπεί η σύνδεση.



Εικόνα 27: Πρώτη εικόνα RealVNC Viewer

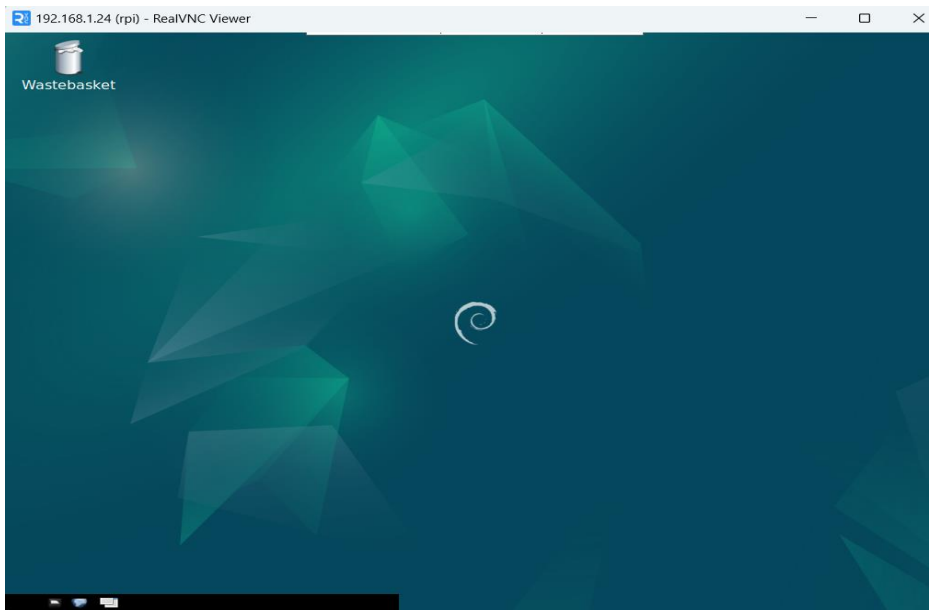


Εικόνα 28: Authentication



Εικόνα 29: Εισαγωγή στοιχείων του Raspberry Pi

Κεφάλαιο 4ο



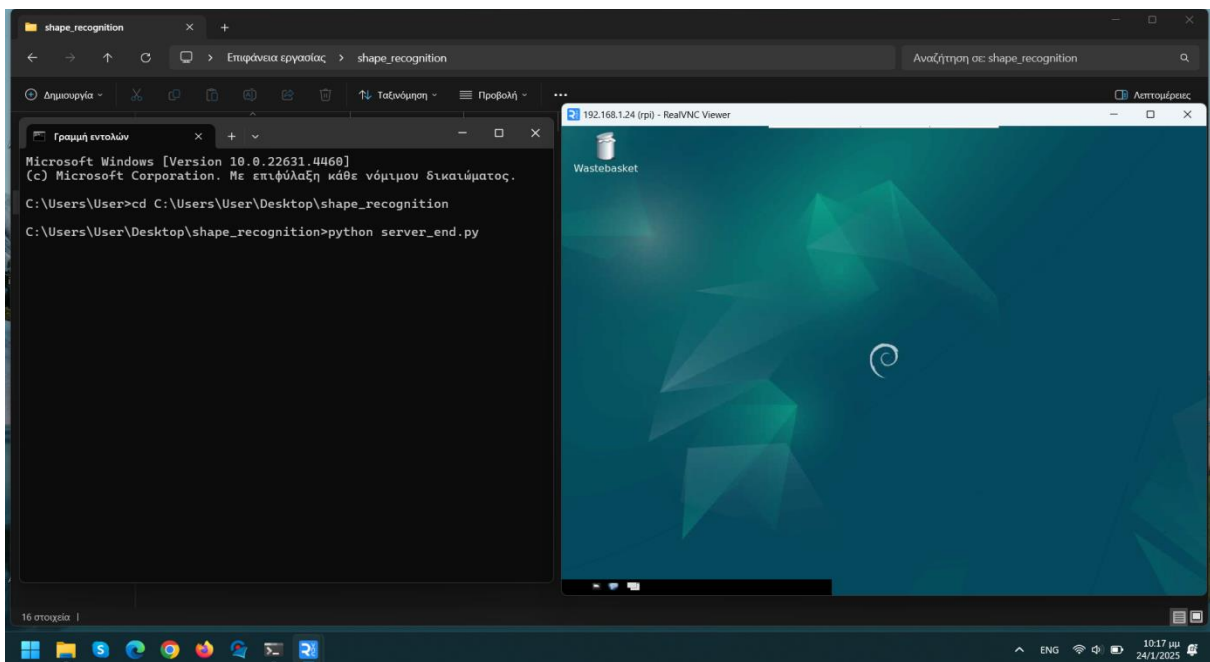
Εικόνα 30: Επιφάνεια εργασίας rpi

Βήμα 2

Τώρα που η σύνδεση πραγματοποιήθηκε στον υπολογιστή κάνοντας χρήση του command line (cmd) πληκτρολογείται η εντολή σύνδεσης του server.

```
python server_end.py
```

Η εντολή αυτή μέσω της Python εκτελεί το αρχείο `server_end.py`. Το αρχείο αυτό πρώτα ανοίγει μια θύρα επικοινωνίας την 5000 στο έργο και περιμένει να δεχτεί αίτημα σύνδεσης από κάποιον client.

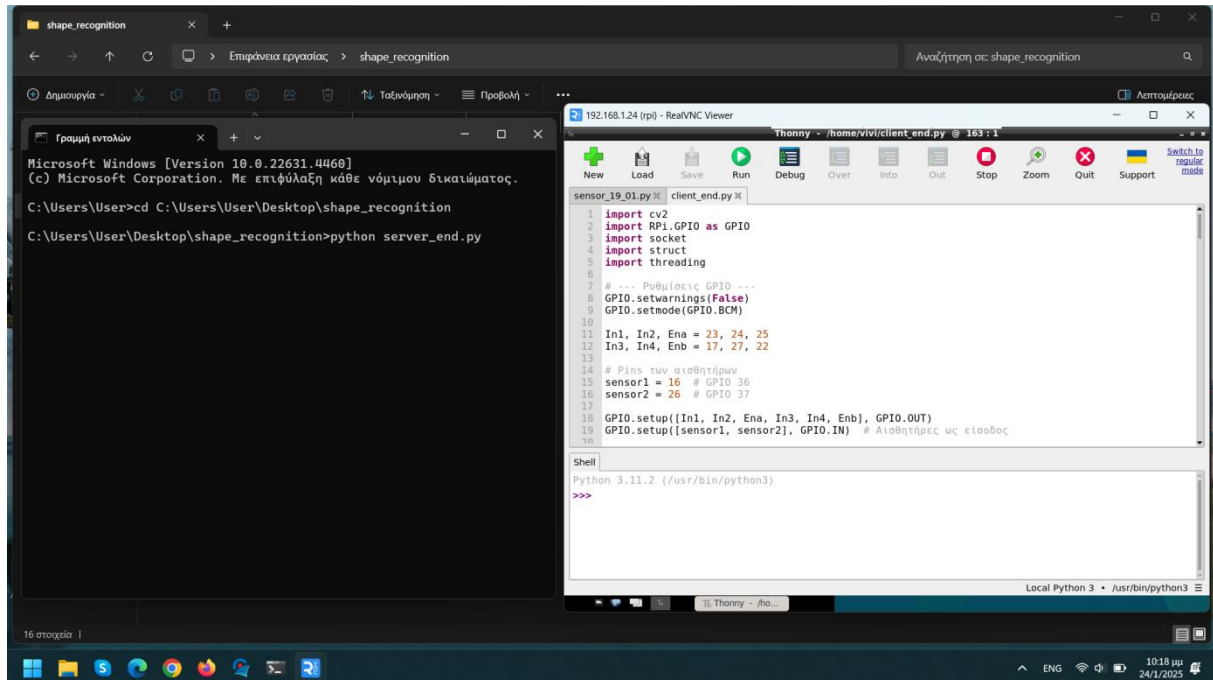


Εικόνα 31: Windows and rpi

Εμφανίζονται και οι δύο οθόνες και το cmd και η επιφάνεια εργασίας rpi .

Βήμα3

Χρησιμοποιείται το Thorny για να εκτελεστεί το πρόγραμμα `client_end.py` σε γλώσσα Python



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. Με επιφύλαξη κάθε νόμιμου δικαιώματος.

C:\Users\User>cd C:\Users\User\Desktop\shape_recognition
C:\Users\User\Desktop\shape_recognition>python server_end.py
```

```
1 import cv2
2 import RPi.GPIO as GPIO
3 import socket
4 import struct
5 import threading
6
7 # -- Ρυθμίσεις GPIO --
8 GPIO.setwarnings(False)
9 GPIO.setmode(GPIO.BCM)
10
11 In1, In2, Ena = 23, 24, 25
12 In3, In4, Enb = 17, 27, 22
13
14 # Pins των αισθητήρων
15 sensor1 = 16 # GPIO 36
16 sensor2 = 26 # GPIO 37
17
18 GPIO.setup([In1, In2, Ena, In3, In4, Enb], GPIO.OUT)
19 GPIO.setup([sensor1, sensor2], GPIO.IN) # Αισθητήρες ως είσοδος
20
```

```
Python 3.11.2 (/usr/bin/python3)
>>>
```

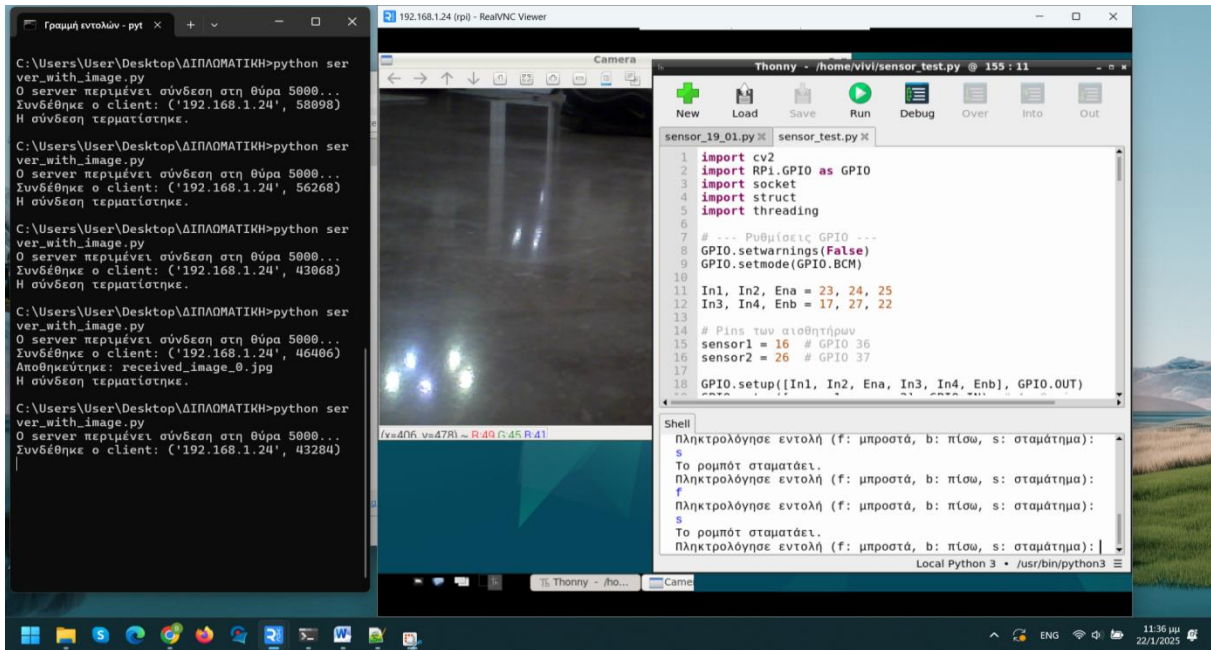
Εικόνα 32: Thorny client_end.py

Το αρχείο αυτό έχει 3 λειτουργίες την σύνδεση με τον sever, τον έλεγχο του οχήματος , την λήψη φωτογραφιών και αποστολή τους στον server. Τέλος αναμένει την πρόβλεψη για την απεικόνιση της φωτογραφίας.

Κεφάλαιο 4ο

Βήμα 4

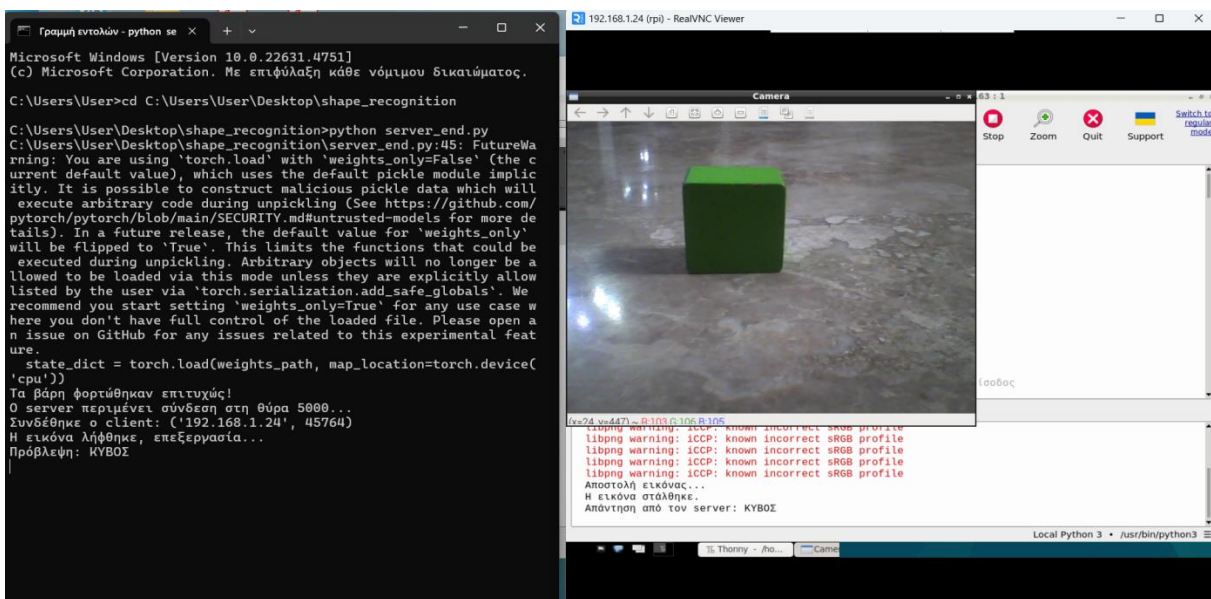
Το όχημα ξεκινάει παίρνοντας εντολές.



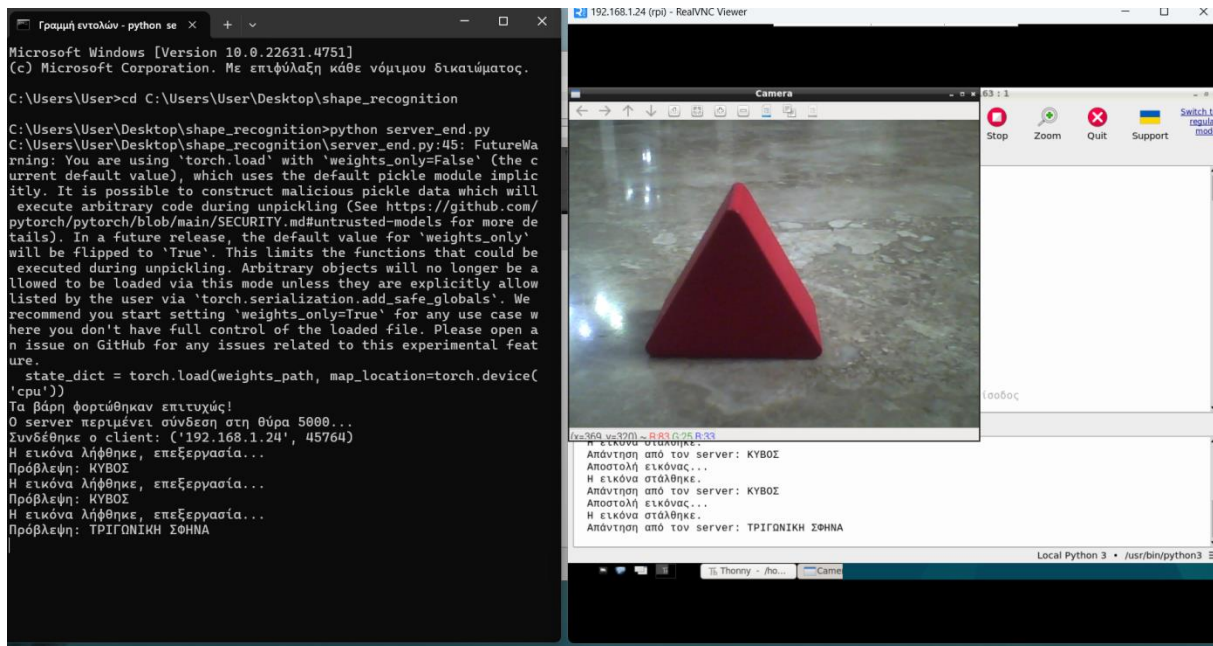
Εικόνα 33: Εντολές ελέγχου

Βήμα 5

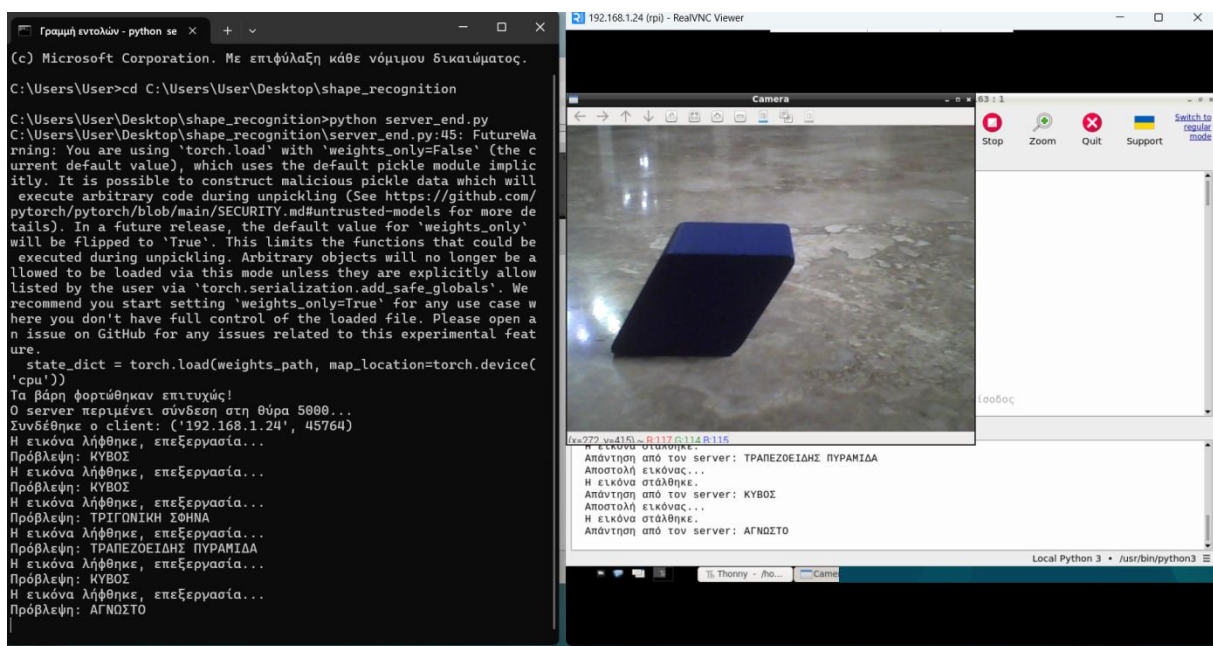
Όταν γίνει λήψη μιας εικόνας με το πλήκτρο C την λαμβάνει ο υπολογιστής, μέσω του προγράμματος server_end.py γίνεται η πρόβλεψη. Το μοντέλο που χρησιμοποιείται έχει ληφθεί από το google colab.



Εικόνα 34: Σωστή πρόβλεψη ΚΥΒΟΣ



Εικόνα 35: Σωστή πρόβλεψη ΤΡΙΓΩΝΙΚΗ ΣΦΗΝΑ



Εικόνα 36: Σωστή πρόβλεψη ΑΓΝΩΣΤΟ

Κεφάλαιο 5ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

Ο κύριος στόχος της παρούσας διπλωματικής ήταν η υλοποίηση ενός ρομποτικού οχήματος που θα μπορούσε να αναγνωρίζει αντικείμενα. Η υλοποίηση του ρομποτικού οχήματος κατέδειξε την αποτελεσματικότητα της χρήσης τεχνολογιών ρομποτικής και βαθιάς μάθησης στην ανάπτυξη ενός συστήματος πλοήγησης και αναγνώρισης. Το ρομποτικό όχημα κατάφερε να εκτελεί βασικές ενέργειες όπως αποφυγή εμποδίων με την χρήση αισθητήρων, παρακολούθηση της κίνησης μέσω κάμερας.

Η αναγνώριση έγινε μέσω της ανάπτυξης και εφαρμογής ενός μοντέλου βαθιάς μάθησης . Τα αποτελέσματα που προέκυψαν από το συγκεκριμένο μοντέλο πέτυχε ικανοποιητικά αποτελέσματα με ποσοστό ακρίβειας 83% . Ωστόσο , παρατηρήθηκε ότι υπήρξαν λανθασμένες προβλέψεις και ειδικά όταν χρησιμοποιούνταν το σχήμα σταυρός λόγω της γραμμικής εξάρτησης σταυρού – τετραγώνου. Μία από τις κύριες προκλήσεις που αντιμετωπίστηκε κατά τη διάρκεια της έρευνας ήταν η μικρή ποικιλία δεδομένων κάτι που περιορίζει αρκετά την γενίκευση του μοντέλου.

Προτάσεις Βελτίωσης

Μία κύρια πρόταση είναι η επέκταση του συνόλου δεδομένων , προσθέτοντας πιο πολλές εικόνες στα δείγματα ειδικά στις κατηγορίες που είχαν μικρά βάρη. Αυτό θα βοηθήσει στην καλύτερη πρόβλεψη των αντικειμένων , μειώνοντας τις πιθανότητες υπερπροσαρμογής (overfitting) και αυξάνοντας την αξιοπιστία του.

Η ανάπτυξη ενός φιλικού γραφικού περιβάλλοντος για τον χρήστη θα βοηθούσε πολύ στην αλληλεπίδραση με το μοντέλο και τον χειρισμό του οχήματος. Οι απαντήσεις των προβλέψεων θα ήταν σε γραφικό περιβάλλον. Ακόμα δεν θα χρειαζόνταν ο χρήστης να έχει προγραμματιστικές γνώσεις.

Για την βελτίωση του ρομποτικού οχήματος θα μπορούσε να χρησιμοποιηθούν καλύτεροι αισθητήρες όπως αισθητήρες βάθους ή lidar για την καλύτερη ανίχνευση εμποδίων όταν ο φωτισμός δεν είναι επαρκής. Επιπλέον πολύ σημαντική βελτίωση θα ήταν η αυτονομία της πλοήγησης του οχήματος όποτε ο χρήστης θα μπορούσε να έχει επιλογές στον χειρισμό του. Η ενεργειακή αυτονομία του ρομπότ θα το έκανε πιο αυτόνομο ενεργειακά χωρίς την συνεχή αλλαγή μπαταριών ή την συνεχή φόρτιση του power bank και θα μπορούσε να διανύει μεγάλες αποστάσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] N. Guedri, B. Guedri and R. Gharbi, "Detection and Recognition of Objects in Real Time Using an Intelligent Camera by a Delta Robot," 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia, 2023, pp. 1-5.
- [2] D. M. Plotnikov, "Trends in the development of end-to-end technologies in education in the context of the implementation of the digital economy in Russia", *Modern pedagogical education*, no. 3, pp. 13-17, 2021.
- [3] E. A. Ignatyeva, "Possibilities of Applying Educational Robotics In students' Project Activities," 2024 4th International Conference on Technology Enhanced Learning in Higher Education (TELE), Lipetsk, Russian Federation, 2024, pp. 35-38.
- [4] M. Darici, Z. Dokur, T. Olmez, "Pneumonia Detection and Classification Using Deep Learning on Chest X-Ray Images", *International Journal of Intelligent Systems And Applications In Engineering*, 2020.
- [5] Deep Learning, 2025. [Online]. Available: <https://www.deeplearningbook.org/> .
- [6] Natesan, Abirami N. (2019). A Detailed Study of Client-Server and its Architecture.
- [7] Mbuguah, Samuel & Mony, Victor & Nyabuto, Geoffrey. (2024). Architectural Review of Client-Server Models. *International Journal of Scientific Research and Engineering Trends*. 10. 139-143. 10.47679/ijasca.v3i1.48.
- [8] Raspberry Pi, 2024. [Online]. Available: <https://www.nettop.gr/index.php/raspberry-pi/kit-plaketes/raspberry-pi-5-4gb.html>.
- [9] Raspberry Pi, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi
- [10] Αισθητήρες, 2024. [Online]. Available: <https://grobotronics.com/robotics/aisthitires/> .
- [11] Piatetsky-Shapiro, G., & Mayo, M. (2013). KDnuggets. Data Science and AI Consulting (<http://www.kdnuggets.com/2013/06/kdnuggets-annual-software-poll-rapidminer-r-viefor-first-place.html>).
- [12] I. Stančin and A. Jović, "An overview and comparison of free Python libraries for data mining and big data analysis," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2019, pp. 977-982.
- [13] T. Carneiro, R. V. Medeiros Da Nóbrega, T. Nepomuceno, G. -B. Bian, V. H. C. De Albuquerque and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," in *IEEE Access*, vol. 6, pp. 61677-61685, 2018.
- [14] PyTorch, 2025. [Online]. Available: <https://pytorch.org/> [Accessed: January- 2025].
- [15] Raspberry Pi OS , 2024. [Online]. Available: <https://www.raspberrypi.com/software/>.