



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΑΝΙΧΝΕΥΣΗ ΕΛΛΗΝΙΚΗΣ ΝΟΗΜΑΤΙΚΗΣ ΓΛΩΣΣΑΣ ΜΕ ΤΕΧΝΗΤΗ
ΝΟΗΜΟΣΥΝΗ**

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ: ΙΩΑΝΝΗΣ ΠΑΝΟΠΟΥΛΟΣ

ΕΠΙΒΛΕΠΟΝΤΕΣ: Δρ. Λουκάς Χαδέλλης, Δρ. Ευάγγελος Τοπάλης

ΠΑΤΡΑ, 2024

Υπεύθυνη Δήλωση Φοιτητή

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία. Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος. Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Πανόπουλου Ιωάννη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

ΠΕΡΙΛΗΨΗ

Το θέμα της πτυχιακής εργασίας πραγματεύεται την ανίχνευση της Ελληνικής νοηματικής γλώσσας με τεχνητή νοημοσύνη. Κάθε μέρα μεγάλη μερίδα ανθρώπων δυσκολεύεται στην επικοινωνία καθώς έχουν απώλεια ακοής, βαρηκοΐα και προβλήματα λόγου εξαιτίας κάποιου συνδρόμου (π.χ. σύνδρομο Down). Επίσης, η οικογένεια και οι φίλοι αυτών των ατόμων χρειάζονται την εκπαίδευση πάνω στη νοηματική γλώσσα για να επικοινωνούν. Ακόμα είναι χρήσιμη για τους επαγγελματίες υγείας και τους διερμηνείς. Σκοπός της παρούσας πτυχιακής εργασίας είναι η υλοποίηση λογισμικού για την επικοινωνία με άτομα που πάσχουν από απώλεια ακοής ή με προβλήματα λόγου αλλά και για εκπαιδευτικό χαρακτήρα στα άτομα και τους φορείς που θέλουν να εξασκηθούν πάνω στη νοηματική γλώσσα καθώς αποτελεί ένα φυσικό και πλήρη τρόπο επικοινωνίας. Η παρούσα εργασία συνεισφέρει στην μετάφραση και την επικοινωνία της ελληνικής νοηματικής αλλά και στην εκπαίδευση αυτής. Πιο αναλυτικά θα δημιουργηθεί κώδικας με τη γλώσσα προγραμματισμού Python όπου περιέχει ανοικτό κώδικα της τεχνητής νοημοσύνης (AI) της Ultralytics yolov8 καθώς θα ανιχνεύει τα γράμματα της ελληνικής νοηματικής και κάποιες λέξεις αυτής. Η εικόνες για την ανίχνευση επέρχονται μέσω ενός Raspberry Pi 4 όπου είναι συνδεδεμένο σε μια κάμερα και μέσω δικτύου μεταφέρει την εικόνα στο πρόγραμμα που εκτελείται στη μονάδα υπολογιστή. Στη συνέχεια καθώς εκτελείται το πρόγραμμα αναδύονται δύο displays. Στο πρώτο display εμφανίζει το άτομο που πραγματοποιεί την χειρομορφή και στο δεύτερο εμφανίζεται η λέξη ή το γράμμα. Για να εμφανιστεί κάτι θα πρέπει να εκτελεστεί μια χειρομορφή που υποδηλώνει το γράμμα ή τη λέξη στην ελληνική νοηματική γλώσσα. Παρουσιάζονται οι μέθοδοι δημιουργίας των στοιχείων που χρησιμοποιούνται, η διαδικασία εκπαίδευσης του προγράμματος και η πληρέστερη ανάλυση του κώδικα της Ultralytics και του Python που γίνεται χρήση.

ΛΕΞΕΙΣ - ΚΛΕΙΔΙΑ

Ελληνική Νοηματική Γλώσσα, Τεχνητή Νοημοσύνη, Γλώσσα Προγραμματισμού Python

ABSTRACT

The topic of this thesis deals with the detection of Greek sign language with artificial intelligence. Every day a large portion of people find it difficult to communicate as they have hearing loss, hearing loss and speech problems due to a syndrome (eg Down syndrome). Also, these people's family and friends need sign language training to communicate. It is also useful for health professionals and interpreters. The purpose of this thesis is the implementation of software for communication with people suffering from hearing loss or speech problems but also for educational character to individuals and institutions who want to practice sign language as it is a natural and complete way of communication. This thesis contributes to the translation and communication of Greek sign language as well as to its education. More specifically, code will be created with the programming language Python which contains open source artificial intelligence (AI) of Ultralytics yolov8 as it will detect the letters of Greek sign language and some of its words. The images for detection occur through a Raspberry Pi 4 where it is connected to a camera and over a network transfers the image to the program running on the computer drive. Then, as the program runs, two displays pop up. The first display shows the person performing the handform and the second displays the word or letter. In order for something to appear, a chiral form denoting the letter or word in Greek sign language must be performed. The methods of creating the components used, the training process of the program and the most complete analysis of the code of Ultralytics and Python used are presented.

KEYWORDS

Greek Sign Language, Artificial Intelligence, Python Programming Language

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητές μου, κ. Λουκά Χαδέλλη και κ. Ευάγγελο Τοπάλη , που δέχθηκαν να συνεργαστούν μαζί μου για την εκπόνηση της πτυχιακής μου εργασίας. Θα ήθελα, επίσης, να ευχαριστήσω την οικογένεια μου που με στήριξε καθόλη τη διάρκεια της φοιτητικής μου σταδιοδρομίας.

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ	iii
ABSTRACT.....	iv
Ευχαριστίες	v
Πίνακας Περιεχομένων.....	vi
Κατάλογος εικόνων.....	ix
Κατάλογος πινάκων	xi
Συντομογραφίες & Ακρωνύμια	xiii
1 ΕΙΣΑΓΩΓΗ.....	1
1.1 Ελληνική Νοηματική Γλώσσα.....	1
1.2 Λογισμικά και εφαρμογές για τη νοηματική γλώσσα.....	1
1.3 Πρόγραμμα προς υλοποίηση.	2
1.4 Οργάνωση πτυχιακής.....	2
2 Η Βαρηκοΐα και η Νοηματική Γλώσσα: Από τη Διάγνωση στην Εκπαίδευση και την Ενημέρωση	4
2.1 Εισαγωγή.	4
2.2 Ορισμός και τύποι βαρηκοΐας.....	4
2.3 Συμπτώματα, αιτίες και διάγνωση.	5
2.4 Αντιμετώπιση και θεραπεία.....	6
2.5 Η δομή της νοηματικής γλώσσας.	7
2.6 Η αρχή και η αναγνώριση της ελληνικής νοηματικής γλώσσας στην Ελλάδα.	8
2.7 Η Ελληνική νοηματική γλώσσα στην εκπαίδευση	10
2.8 Η Ελληνική νοηματική γλώσσα στον τομέα της υγείας και τα ΜΜΕ.....	10
3 Η Σημασία της Προσβασιμότητας στην Ψηφιακή Εποχή για τα Άτομα με Ακουολογικές Διαταραχές.....	11
3.1 Προσβασιμότητα και ίσες ευκαιρίες στην ψηφιακή επικοινωνία.....	11

3.2	Νομοθεσία και δικαιώματα για την προσβασιμότητα των ατόμων με ακοολογικές διαταραχές.....	12
3.3	Ρόλος της τεχνολογίας στην ενσωμάτωση της νοηματικής γλώσσας στις ψηφιακές πλατφόρμες.	12
3.4	Ανάγκη για προσαρμογή των εφαρμογών και των ιστοσελίδων για βαρήκοους. ...	13
4	Τεχνητή Νοημοσύνη.....	15
4.1	Ορισμός της τεχνητής νοημοσύνης.	15
4.2	Η ιστορία και η εξέλιξη της τεχνητής νοημοσύνης.	15
4.3	Machine learning και Deep learning.....	17
4.4	Εφαρμογές της τεχνητής νοημοσύνης.	20
4.5	Επιπτώσεις της τεχνητής νοημοσύνης.	21
4.6	Ο τομέας της ηθικής στην τεχνητή νοημοσύνη.	23
5	Τεχνολογίες και Εργαλεία για Ανάλυση Εικόνας και Ανάπτυξη Εφαρμογής.....	24
5.1	Η γλώσσα προγραμματισμού Python με τις απαραίτητες βιβλιοθήκες.	24
5.2	YOLOv8 της Ultralytics.	26
5.3	MJPEG Stream.....	28
5.4	Εφαρμογή Moonlight Stream.	29
5.5	Raspberry Pi 4.....	31
5.6	Εφαρμογή LabelMe.	33
5.7	Microsoft Visual Studio.....	34
5.8	Εφαρμογή Miniconda.	36
5.9	Πλατφόρμα ZeroTier.	37
6	Ανάπτυξη Συστήματος Αναγνώρισης της Ελληνικής Νοηματικής Γλώσσας.	40
6.1	Γενική αρχιτεκτονική υλοποίηση.	40
6.2	Αποτύπωση του αλφαβήτου και των λέξεων σε φωτογραφίες.....	41
6.3	Annotation των φωτογραφιών με τη χρήση του LabelMe.....	43
6.4	Μετατροπή των labels σε άλλων τύπο αρχείου.	48

6.5	Χρήση του κώδικα της ultralytics για την εκτέλεση του training και τη δημιουργία του YOLOv8.....	50
6.6	Σύνδεση του Raspberry Pi και εγκατάσταση των απαραίτητων εφαρμογών και βιβλιοθηκών που χρειαζόμαστε.....	70
6.7	Επεξήγηση του κώδικα Python και τη χρήση του.	72
6.8	Δημιουργία του συστήματος και αποτύπωση παραδειγμάτων	78
6.9	Αποτύπωση ποσοστού επιτυχίας συστήματος.....	88
7	Επίλογος-Συμπεράσματα.....	90
7.1	Εισαγωγή.....	90
7.2	Προβλήματα και αντιμετώπιση.....	91
7.3	Συμπεράσματα και πιθανότητες εξέλιξης.....	91
8	Βιβλιογραφία	92
8.1	Διαδικτυακή Ελληνική.....	92
8.2	Διαδικτυακή Ξενόγλωσση	95
9	Παράρτημα Α: Κώδικας Python για την ανίχνευση της Ελληνικής Νοηματικής Γλώσσας	
	101	

Κατάλογος εικόνων

Εικόνα 2.1: Δακτυλικό αλφάβητο ελληνικής νοηματικής γλώσσας	8
Εικόνα 5.1: Λογότυπο Python	24
Εικόνα 5.2: Λογότυπο της ultralytics YOLOv8	26
Εικόνα 5.3: Λογότυπο Moonlight Stream	29
Εικόνα 5.4: Λογότυπο Raspberry Pi	31
Εικόνα 5.5: Raspberry-pi-4 specs	32
Εικόνα 5.6: Λογότυπο LabelMe	33
Εικόνα 5.7: Λογότυπο Microsoft Visual Studio	34
Εικόνα 5.8: Λογότυπο Anaconda	36
Εικόνα 5.9: ZeroTier λογότυπο	37
Εικόνα 6.1: Χειρομορφή για το γράμμα A	42
Εικόνα 6.2: Χειρομορφή για το γράμμα B	42
Εικόνα 6.3: Χειρομορφή για τη λέξη αυτοκίνητο	42
Εικόνα 6.4: Δεύτερη χειρομορφή για τη λέξη αυτοκίνητο	42
Εικόνα 6.5: Εγκατάσταση LabelMe	43
Εικόνα 6.6: Εφαρμογή LabelMe	44
Εικόνα 6.7: Ενεργοποίηση save with data image	44
Εικόνα 6.8: Εύρεση φακέλου φωτογραφιών	45
Εικόνα 6.9: Επιλογή create polygon	46
Εικόνα 6.10: Αρχή του annotation	47
Εικόνα 6.11: Annotation χειρομορφής με το LabelMe	47
Εικόνα 6.12: Δημιουργία label	48
Εικόνα 6.13: Εμφάνιση του label list και του file list	48
Εικόνα 6.14: Μετατροπή αρχείων JSON	49
Εικόνα 6.15: Εμφάνιση αρχείων TXT μετά τη μετατροπή	49
Εικόνα 6.16: Το display εκπαίδευσης του AI	69
Εικόνα 6.17: Εντολές για εγκατάσταση MJPEG Stream	70
Εικόνα 6.18: Συνέχεια εντολών για εγκατάσταση MJPEG Stream	71
Εικόνα 6.19: Χρήση της εφαρμογής MJPEG Stream	71
Εικόνα 6.20: Επιλογή μονάδας υπολογιστή από το Moonlight	72
Εικόνα 6.21: Οι βιβλιοθήκες που βρίσκονται στο environment	79
Εικόνα 6.22: Το περιεχόμενο του YAML file	80

Εικόνα 6.23: Δομή TXT αρχείο.....	81
Εικόνα 6.24: Ιστοσελίδα Zerotier	82
Εικόνα 6.25: Επιλογή του start free trial.....	82
Εικόνα 6.26: Δημιουργία ιδιωτικού δικτύου.	83
Εικόνα 6.27: Λήψη της εφαρμογής για υπολογιστή.....	83
Εικόνα 6.28: Λήψη του Zerotier στο Raspberry Pi 4	84
Εικόνα 6.29: Προβολή τον IP	84
Εικόνα 6.30: Παράδειγμα με χειρομορφή για το γράμμα A εντός roi.....	85
Εικόνα 6.31: Παράδειγμα με χειρομορφή για το γράμμα A εκτός roi	86
Εικόνα 6.32: Παράδειγμα με χειρομορφή για τη λέξη ελικόπτερο εκτός roi.....	86
Εικόνα 6.33: Παράδειγμα με χειρομορφή για τη λέξη ελικόπτερο εντός roi	87
Εικόνα 6.34: Παράδειγμα με χειρομορφή για τη λέξη αεροπλάνο εκτός roi	87
Εικόνα 6.35: Παράδειγμα με χειρομορφή για το γράμμα Y εντός roi.....	87
Εικόνα 6.36: Χειρομορφή με λευκό φόντο με καλό φωτισμό.....	89
Εικόνα 6.37: Χειρομορφή με μαύρο φόντο με καλό φωτισμό	89
Εικόνα 6.38: Χειρομορφή με λευκό φόντο με χαμηλό φωτισμό.....	89
Εικόνα 6.39: Χειρομορφή με μαύρο φόντο με χαμηλό φωτισμό	90

Κατάλογος πινάκων

Πίνακας 4.1: Εύρεση αριθμού	17
Πίνακας 6.1: Ανάλυση κώδικα ultalytics τμήμα 1	50
Πίνακας 6.2: Ανάλυση κώδικα ultalytics τμήμα 2	51
Πίνακας 6.3: Ανάλυση κώδικα ultalytics τμήμα 3	52
Πίνακας 6.4: Ανάλυση κώδικα ultalytics τμήμα 4	53
Πίνακας 6.5: Ανάλυση κώδικα ultalytics τμήμα 5	53
Πίνακας 6.6: Ανάλυση κώδικα ultalytics τμήμα 6	54
Πίνακας 6.7: Ανάλυση κώδικα ultalytics τμήμα 7	55
Πίνακας 6.8: Ανάλυση κώδικα ultalytics τμήμα 8	55
Πίνακας 6.9: Ανάλυση κώδικα ultalytics τμήμα 9	56
Πίνακας 6.10: Ανάλυση κώδικα ultalytics τμήμα 10	57
Πίνακας 6.11: Ανάλυση κώδικα ultalytics τμήμα 11	57
Πίνακας 6.12: Ανάλυση κώδικα ultalytics τμήμα 12	58
Πίνακας 6.13: Ανάλυση κώδικα ultalytics τμήμα 13	58
Πίνακας 6.14: Ανάλυση κώδικα ultalytics τμήμα 14	58
Πίνακας 6.15: Ανάλυση κώδικα ultalytics τμήμα 15	59
Πίνακας 6.16: Ανάλυση κώδικα ultalytics τμήμα 16	60
Πίνακας 6.17: Ανάλυση κώδικα ultalytics τμήμα 17	61
Πίνακας 6.18: Ανάλυση κώδικα ultalytics τμήμα 18	61
Πίνακας 6.19: Ανάλυση κώδικα ultalytics τμήμα 19	62
Πίνακας 6.20: Ανάλυση κώδικα ultalytics τμήμα 20	63
Πίνακας 6.21: Ανάλυση κώδικα ultalytics τμήμα 21	64
Πίνακας 6.22: Ανάλυση κώδικα ultalytics τμήμα 22	65
Πίνακας 6.23: Ανάλυση κώδικα ultalytics τμήμα 23	65
Πίνακας 6.24: Ανάλυση κώδικα ultalytics τμήμα 24	66
Πίνακας 6.25: Ανάλυση κώδικα ultalytics τμήμα 25	66
Πίνακας 6.26: Ανάλυση κώδικα ultalytics τμήμα 26	66
Πίνακας 6.27: Ανάλυση κώδικα ultalytics τμήμα 27	67
Πίνακας 6.28: Ανάλυση κώδικα ultalytics τμήμα 28	67
Πίνακας 6.29: Ανάλυση κώδικα ultalytics τμήμα 29	67
Πίνακας 6.30: Ανάλυση κώδικα ultalytics τμήμα 30	68
Πίνακας 6.31: Εντολή για την εκκίνηση του training.....	68

Πίνακας 6.32: Εγκατάσταση MJPEG Stream.....	70
Πίνακας 6.33: Εκκίνηση MJPEG Stream	71
Πίνακας 6.34: Ανάλυση κώδικα Python τμήμα 1	73
Πίνακας 6.35: Ανάλυση κώδικα Python τμήμα 2.....	73
Πίνακας 6.36: Ανάλυση κώδικα Python τμήμα 3.....	74
Πίνακας 6.37: Ανάλυση κώδικα Python τμήμα 4.....	74
Πίνακας 6.38: Ανάλυση κώδικα Python τμήμα 5.....	74
Πίνακας 6.39: Ανάλυση κώδικα Python τμήμα 6.....	75
Πίνακας 6.40: Ανάλυση κώδικα Python τμήμα 7.....	75
Πίνακας 6.41: Ανάλυση κώδικα Python τμήμα 8.....	75
Πίνακας 6.42: Ανάλυση κώδικα Python τμήμα 9.....	76
Πίνακας 6.43: Ανάλυση κώδικα Python τμήμα 10.....	76
Πίνακας 6.44: Ανάλυση κώδικα Python τμήμα 11.....	77
Πίνακας 6.45: Ανάλυση κώδικα Python τμήμα 12.....	78
Πίνακας 6.46: Ανάλυση κώδικα Python τμήμα 13.....	78
Πίνακας 6.47: Σύνδεση στο network	84
Πίνακας 6.48: Χρήση του MJPEG Stream στοRaspberry pi 4.....	85
Πίνακας 6.49:Ποσοστό επιτυχίας χειρομορφών.....	88

Συντομογραφίες & Ακρωνύμια

ABR	Auditory Brainstem Response
AI	Artificial Intelligence
AMP	Automatic Mixed Precision
ANN	Artificial Neural Network
API	Application Programming Interface
ASSR	Auditory Steady-State Response
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSPNet	Cross Stage Partial Network
dB	Decibel
IDE	Integrated Development Environment
FPN	Feature Pyramid Network
ML	Machine learning
NAT	Network Address Translation
RNN	Recurrent Neural Network
ROI	Region of Interest
STUN	Session Traversal Utilities for NAT
PANet	Path Aggregation Network
P2P	Peer-to-Peer
VPN	Virtual Private Network
WCAG	Web Content Accessibility Guidelines

YOLO	You Only Look Once
ΕΝΓ	Ελληνική Νοηματική Γλώσσα
ΜΜΕ	Μέσα μαζικής ενημέρωσης
ΚΕΝΓ	Κέντρο Ελληνικής Νοηματικής Γλωσσάς
ΚΝΕΑ	Καθολικός Νεογνικός Έλεγχος Ακοής
ΤΠΕ	Τεχνολογία Πληροφορικής και Επικοινωνιών

1 ΕΙΣΑΓΩΓΗ

1.1 Ελληνική Νοηματική Γλώσσα.

Η νοηματική γλώσσα είναι η μορφή επικοινωνίας που χρησιμοποιούν οι κωφοί και οι βαρήκοι σε όλο τον κόσμο. Στην Ελλάδα, η Ελληνική νοηματική γλώσσα (ΕΝΓ) έχει μια μακρά ιστορία και πολιτιστική σημασία. Η διαφορά της ΕΝΓ με τις προφορικές γλώσσες είναι ότι βασίζεται στην οπτικοκινητική επικοινωνία. Αυτό συμβαίνει εξαιτίας των κινήσεων των χεριών, της στάσης του σώματος και των εκφράσεων του προσώπου με τις οποίες επιτυγχάνεται η επικοινωνία. Πρέπει να επισημανθεί ότι η γλώσσα έχει αυτόνομη γραμματική και συντακτικό σύστημα το οποίο να μην αναπτύχθηκε ανεξάρτητα αλλά βασίζεται στην ελληνική προφορική γλώσσα. Αναγνωρίστηκε επίσημα από την Ελληνική βουλή τον Μάρτιο του 2000 με τον νόμο 2817 για την ειδική αγωγή. Έτσι η ΕΝΓ εισάγεται ως το βασικό εργαλείο στην εκπαίδευση των κωφών. Ωστόσο, το 2017 με τον νόμο 4488 έγινε ισότιμη με την ελληνική γλώσσα. Η ΕΝΓ πλέον είναι η γλώσσα των κωφών και των βαρήκοων στην εκπαίδευση, στα ΜΜΕ και στην Πρόσβαση αντίστοιχα. Παρόλο που η ΕΝΓ αποκαλείται ελληνική, επειδή η χρήση της γίνεται στην Ελλάδα από Έλληνες νοηματιστές, αυτό δε σημαίνει ότι απεικονίζει την ελληνική γλώσσα ή ότι προέρχεται από αυτήν. Συγκεκριμένα υπάγεται σε ένα αυτόνομο γλωσσικό σύστημα που μπορεί να μελετηθεί και να αναλυθεί όπως και κάθε άλλη φυσική γλώσσα. Η χειρομορφή είναι το χαρακτηριστικότερο συστατικό ενός νοήματος. Το σχήμα που παίρνει η παλάμη και η θέση στην οποία τοποθετούνται τα δάχτυλα σχηματίζουν ένα νόημα. Όμως, η χειρομορφή δεν είναι από μόνη της φορέας σημασίας. Για την απόκτηση σημασίας θα πρέπει να συνοδεύεται και από τα παρακάτω: α) σωστός προσανατολισμός της παλάμης θέση της χειρομορφής στο χώρο ή πάνω στο σώμα, β) η κίνηση του χεριού που είναι αναγκαία για να ολοκληρωθεί κάποιο νόημα, γ) η στάση του σώματος και δ) η έκφραση του προσώπου αποτελούν επίσης συστατικά του νοήματος. (Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος. (n.d.))

1.2 Λογισμικά και εφαρμογές για τη νοηματική γλώσσα.

Πλέον υπάρχουν λογισμικά τα οποία βοηθούν στην κατανόηση και στην εκμάθηση της ελληνικής νοηματικής γλώσσας. Το HandTalk είναι μια εφαρμογή η οποία μέσω avatar μεταφράζει κείμενο ή ομιλία σε νοηματική γλώσσα με αποτέλεσμα να διευκολύνει τα άτομα που δε γνωρίζουν νοηματική ομιλία. Το MotionSavvy είναι ένα φορητό σύστημα το οποίο

χάρη τη χρήση καμερών και αισθητήρων αναγνωρίζει τη νοηματική γλώσσα . Επιπλέον, το SignWriting είναι ένα σύστημα γραφής το οποίο σχεδιάστηκε για να καταγράφει τις κινήσεις της νοηματικής γλώσσας σε γραπτή μορφή, μέσω συμβόλων που αναπαριστούν τη χειρονομία, τη θέση που έχουν τοποθετηθεί τα χέρια και την έκφραση που πραγματοποιεί το πρόσωπο. Οι χρήστες του, έχουν τη δυνατότητα να γράφουν να διαβάζουν και να διδάσκουν νοηματική γλώσσα μέσω ενός οργανωμένου γραπτού συστήματος, του λεγόμενου *lingvano*, το οποίο είναι μια εφαρμογή η οποία παρέχει εκπαίδευση για τη νοηματική γλώσσα με διαδραστικό περιεχόμενο και παιχνίδια. (SignWriting. (n.d.), HandTalk. (n.d.), Techblog. Λορένα Σιούτη (2014, 8 Ιουνίου.), *Lingvano*. (n.d.).)

1.3 Πρόγραμμα προς υλοποίηση.

Η παρούσα εργασία έχει ως αντικείμενο την κατασκευή ενός προγράμματος, το οποίο θα αναγνωρίζει την ελληνική νοηματική γλώσσα. Το πρόγραμμα έχει γραφτεί σε γλώσσα προγραμματισμού Python, στο οποίο εμπεριέχεται και το πρόγραμμα AI, το οποίο εκπαιδεύτηκε με δεδομένα του δημιουργού της πτυχιακής. Εν συνεχεία το πρόγραμμα συνδέεται μέσω δικτύου με Raspberry Pi 4, στο οποίο είναι τοποθετημένη πάνω μια usb κάμερα. Με αυτόν τον τρόπο επιτυγχάνεται η χρήση του προγράμματος σε οποιοδήποτε σημείο, αρκεί να υπάρχει παροχή δικτύου. Επίσης, γίνεται χρήση της πλατφόρμας ZeroTier για να μην υπάρχει αλλαγή IP κατά τη σύνδεση του Raspberry Pi 4 σε οποιοδήποτε δίκτυο. Τέλος, το αποτέλεσμα προβάλλεται από την οθόνη που είναι συνδεδεμένη με το Raspberry Pi 4 καθώς φαίνονται οι χειρομορφές που επιχειρούνται με το γράμμα ή τη λέξη να αποτυπώνεται.

1.4 Οργάνωση πτυχιακής.

Η πτυχιακή εργασία στο δεύτερο κεφάλαιο αναφέρεται στα άτομα με βαρηκοΐα και στη νοηματική γλώσσα. Θα αναφερθούν οι τύποι της βαρηκοΐας , τα αίτια και η αντιμετώπιση. Επίσης, θα αναλυθεί εκτενώς η δομή της νοηματικής γλώσσας καθώς και θα επισημανθεί η αναγκαιότητα της ΕΝΓ σε διάφορες πτυχές της καθημερινότητας ενός ανθρώπου. Το τρίτο κεφάλαιο κάνει λόγο στη σημασία που υπάρχει, ώστε τα άτομα που χρησιμοποιούν την ΕΝΓ, να έχουν πρόσβαση στην ψηφιακή εποχή. Θα αναφερθούν οι νομοθεσίες αλλά και ο ρόλος της τεχνολογίας στη βοήθεια της πρόσβασης σε άτομα με βαρηκοΐα. Επακόλουθα στο τέταρτο κεφάλαιο δίνεται ο ορισμός της τεχνητής νοημοσύνης (*artificial intelligence*), αναλύεται η μηχανική μάθηση και οι αλγόριθμοι. Επιπλέον δε, τονίζονται οι τρόποι εφαρμογής και οι

επιπτώσεις της ΑΙ. Εν συνεχεία, στο πέμπτο κεφάλαιο διατυπώνονται οι τεχνολογίες και τα εργαλεία που είναι απαραίτητα για την ανάλυση της εικόνας και την ανάπτυξη της εφαρμογής. Συγκεκριμένα θα πραγματευτεί η γλώσσα προγραμματισμού Python και θα αναφερθούν οι βιβλιοθήκες που χρησιμοποιούνται. Παρουσιάζεται το πρόγραμμα της Ultralytics το YoloV8. Ακόμη θα επεξηγηθούν οι εφαρμογές MJPEG stream, Moonlight stream, LabelMe, miniconda, η πλατφόρμα ZeroTier, ενώ γίνεται αναφορά και στο Raspberry Pi 4 και τον τρόπο που λειτουργεί. Στο έκτο κεφάλαιο παρουσιάζεται η ανάπτυξη του συστήματος για την αναγνώριση της ΕΝΓ με τη γλώσσα προγραμματισμού Python αλλά και τη χρήση του yoloV8 όπως και του Raspberry Pi 4. Καταληκτικά, στο έβδομο κεφάλαιο αποτυπώνεται ο επίλογος, οι δυσκολίες και τα συμπεράσματα για τη δημιουργία της εργασίας αυτής.

2 Η Βαρηκοΐα και η Νοηματική Γλώσσα: Από τη Διάγνωση στην Εκπαίδευση και την Ενημέρωση

2.1 Εισαγωγή.

Τη σήμερον ημέρα μια από τις κοινές αισθητηριακές διαταραχές είναι η βαρηκοΐα. Η βαρηκοΐα επηρεάζει εκατομμύρια ανθρώπους ανεξαρτήτως ηλικίας. Καθημερινά εξαιτίας του εξοχρισμού αλλά και τη ραγδαία ανάπτυξη της τεχνολογίας, οι προκλήσεις που σχετίζονται με την απώλεια ακοής εντείνονται, καθώς η συνεχής χρήση των ηλεκτρονικών συσκευών, ο θόρυβος που προκαλείται από την έντονη κυκλοφορία και γενικά οι επιβλαβείς συνήθειες ακρόασης που έχουν δημιουργηθεί. Όπως αναφέρθηκε ζούμε στη σύγχρονη εποχή, ωστόσο πολλοί άνθρωποι παραμένουν χωρίς διάγνωση και χωρίς θεραπεία με αποτέλεσμα να υποβαθμίζεται η ποιότητα ζωής τους. Η βαρηκοΐα σήμερα δεν τίθεται σαν ένα απλό ιατρικό ζήτημα, αλλά σαν ένα πολυδιάστατο κοινωνικό φαινόμενο, στο οποίο επιβάλλεται να υπάρχει ενημέρωση, πρόληψη και καινοτόμες λύσεις για την αποτελεσματική αντιμετώπιση της. (Σουλαντίκας Κωνσταντίνος (2023, 27 Νοεμβρίου), Κουκουράβα Βαΐα, (2024, 24 Μαρτίου))

2.2 Ορισμός και τύποι βαρηκοΐας.

Το σύμπτωμα μειωμένης ακοής το παρατηρεί το ίδιο το άτομο ή κάποιος που βρίσκεται στον οικογενειακό του κύκλο ονομάζεται βαρηκοΐα. Άλλες φορές η διάγνωση γίνεται από γιατρό με την πραγματοποίηση ενός ακουολογικού ελέγχου. Είναι γνωστό εν τοις πάσι ότι, όσο μεγαλώνει ο άνθρωπος, η ακοή του μειώνεται σταδιακά, καθώς έρευνες έχουν δείξει ότι άτομα άνω των 65 εμφανίζουν μερικού βαθμού βαρηκοΐα. Η βαρηκοΐα που δημιουργείται με το πέρασ του χρόνου ονομάζεται πρεσβυακούσια και αφορά τις υψηλές συχνότητες ακοής. Ωστόσο, η απώλεια ακοής δεν αφορά μόνο τους ηλικιωμένους αλλά όλες τις ηλικιακές ομάδες. Συγκεκριμένα τα μικρά παιδιά μπορεί, εξαιτίας κληρονομικών παραγόντων, να πάσχουν από την επονομαζόμενη κληρονομική βαρηκοΐα. Άρθρα κάνουν αναφορά ότι περισσότερα από 300 κληρονομικά σύνδρομα έχουν αναγνωριστεί για την πρόκληση βαρηκοΐας. Ακόμη, πρέπει να επισημάνουμε ότι η συγγενής βαρηκοΐα δεν προκαλείται από τα γονίδια αλλά από λοιμώξεις της κήσης (τοξόπλασμα, ερυθρά, έρπητας, κυτταρομεγαλοϊός κ.α.), οι οποίες προκαλούν τη μηνιγγίτιδα και σαν αποτέλεσμα την οστεοποίηση του οργάνου ακοής που είναι ο κοχλίας. Η βαρηκοΐα μπορεί να προκληθεί σε οποιοδήποτε τμήμα του αυτιού. Ειδικότερα, τα τμήματα του αυτιού που μπορεί να υπάρξουν διαταραχές είναι: α) ο έξω ακουστικός πόρος, β) το μέσο ους με τα οστάρια του (σφύρα, άκμονας, αναβολέας) και γ) το έσω ους (κοχλίας, ακουστικό νεύρο,

κέντρο της ακοής στον εγκέφαλο). Τέλος, η βαρηκοΐα που προκαλείται μεταγενέστερα στον άνθρωπο, χωρίς να υπάρχει από τη γέννηση του, ονομάζεται επίκτητη. (Σύλλογος Ακοοπροθετιστών Ελλάδος. (n.d.), Σουλαντίκας Κωνσταντίνος (2023, 27 Νοεμβρίου), Κουκουράβα Βαΐα, (2024, 24 Μαρτίου))

Η βαρηκοΐα ανάλογα με τους παράγοντες που τη χαρακτηρίζουν διαχωρίζεται σε κατηγορίες, επισημαίνοντας ότι σημαντικό ρόλο παίζει η απώλεια ακοής με βάση την ένταση τα decibel(dB). Ειδικότερα:

- Φυσιολογική ακοή: 0-25 dB, (20dB στα παιδιά)
- Μικρού βαθμού βαρηκοΐα: 26-40 d
- Μετρίου βαθμού βαρηκοΐα: 41-55 dB
- Μετρίου-σοβαρού βαθμού βαρηκοΐα: 56-70 dB
- Σοβαρού βαθμού βαρηκοΐα: 71-90 dB
- Πρακτική κώφωση: πάνω από 90 dB

Εν συνεχεία αξιολογείται η έκταση της βαρηκοΐας με βάση διαφορετικές ακουστικές συχνότητες. Αναλόγως τις συχνότητες ονομάζεται βαρηκοΐα χαμηλών συχνοτήτων και βαρηκοΐα υψηλών συχνοτήτων. Η βαρηκοΐα χαμηλών συχνοτήτων είναι η μείωση ακοής που αφορά κυρίως τα μπάσα, ενώ η βαρηκοΐα υψηλών συχνοτήτων υφίσταται όταν η μεγαλύτερη απώλεια βρίσκεται στα πρίμα. Είναι πολύ σημαντικό να υπάρχει γνώση του τμήματος του αυτιού το οποίο πάσχει. Με κριτήριο το τμήμα υπάρχει η βαρηκοΐα αγωγιμότητας και η νευροαισθητήριος βαρηκοΐα. Όταν υπάρχει παθολογία του μέσου ή του έξω ώτος η βαρηκοΐα αντιμετωπίζεται άμεσα και η ακοή μπορεί να επανέλθει στα φυσιολογικά της όρια αυτή διακρίνεται σαν βαρηκοΐα αγωγιμότητας. Στην ύπαρξη βλάβης του έσω ώτος, του νεύρου της ακοής ή του τμήματος του εγκεφάλου που επεξεργάζεται την πληροφορία, ο τύπος βαρηκοΐας ονομάζεται νευροαισθητήριος βαρηκοΐα. Τέλος, ο συνδυασμός των δύο παραπάνω έχει την ονομασία μικτού τύπου βαρηκοΐας. (Σύλλογος Ακοοπροθετιστών Ελλάδος. (n.d.), Σουλαντίκας Κωνσταντίνος (2023, 27 Νοεμβρίου), Κουκουράβα Βαΐα, (2024, 24 Μαρτίου))

2.3 Συμπτώματα, αιτίες και διάγνωση.

Τα συμπτώματα της βαρηκοΐας μπορεί να εξελίσσονται με αργούς ρυθμούς και για αυτό τον λόγο να μείνει αδιάγνωστη ή ασυμπτωματική για αρκετό καιρό καθώς δεν υπάρχει η εμφάνιση μερικών εκ των κύριων ενδείξεων, ήτοι του ιλίγγου ή των εμβοών. Ένα άτομο το οποίο πάσχει από βαρηκοΐα αντιμετωπίζει μεγάλη δυσκολία στο να καταλάβει τα λεγόμενα του συνομιλητή

του, ιδιαίτερα όταν βρίσκεται σε χώρους με πολλή φασαρία. Επιπρόσθετα, μπερδεύει την ομιλία με άλλους θορύβους και δεν ακούει καθημερινούς ήχους, όπως το χτύπημα του κουδουνιού της πόρτας. Συχνά σε συζήτηση που θα πραγματοποιήσει με άλλο άτομο θα μιλάει δυνατά και θα ζητάει από τον συνομιλητή του να μιλάει δυνατά επίσης. Σε καθημερινή βάση μπορεί ασυναίσθητα κάποιος να δυναμώνει την τηλεόραση όπως και άθελα του να αγνοεί κάποιον που του απευθύνεται. Το αποτέλεσμα όλων των παραπάνω είναι πως η μείωση της ακοής θα επηρεάσει το άτομο στην επικοινωνία του, στην κοινωνικοποίηση του και την ψυχοκοινωνική ευεξία.

Οι αιτίες που προκαλούν την βαρηκοΐα ποικίλουν. Μερικές από αυτές είναι η φυσική γήρανση, η χρόνια έκθεση σε υψηλής έντασης ήχο, οι γενετικοί παράγοντες και ορισμένες καταστάσεις όπως η υπέρταση και ο διαβήτης. Επιπλέον, υπάρχουν οι ασθένειες που επηρεάζουν τον έξω ακουστικό πόρο, λόγου χάριν η εξωτερική ωτίτιδα και το βύσμα κυψελίδας. Οι ασθένειες του μέσου αυτιού, όπως η μέση ωτίτιδα, η οποία μπορεί να είναι χρόνια, η συλλογή υγρού στο μέσο αυτί και η ωτοσκλήρυνση. Όμως, τις πιο συχνές αιτίες της βαρηκοΐας τις εντοπίζουμε στις παθολογίες του έσω αυτιού, όπως είναι η πρεσβυακουσία, η αιφνίδια νευροαισθητήρια βαρηκοΐα αγνώστου αιτιολογίας και οι συγγενείς βαρηκοΐες. Η διάγνωση μπορεί να πραγματοποιηθεί σε νεογνά και ενήλικες. Στα νεογνά για την πρόωμη διάγνωση χρησιμοποιείται ο καθολικός νεογνικός έλεγχος ακοής (KNEA). Ο έλεγχος αυτός περιλαμβάνει τις Ωτοακουστικές εκπομπές (OAE's) και τα Αυτόματα Ακουστικά Προκλητά Δυναμικά Εγκεφαλικού Στελέχους (aABR), ενώ επί αρνητικών αποτελεσμάτων πραγματοποιούνται πιο εξειδικευμένες εξετάσεις υπό αναισθησία όπως τα ASSR και τα ABR. Στους ενήλικες οι ακοολογικές εξετάσεις είναι αυτές που χρησιμοποιούνται για τη διάγνωση, κάποιες εξ αυτών είναι το τονικό ακούγραμμα, το τυμπανόγραμμα και το ακουστικό αντανακλαστικό. Με τα από αυτά ο ειδικός γιατρός έχει όλα τα απαραίτητα στοιχεία για να διαγνώσει το είδος της βαρηκοΐας. Όλα τα παραπάνω μπορούν να επιφέρουν και κώφωση σε αρκετές περιπτώσεις. (Σουλαντίκας Κωνσταντίνος, (2023, 27 Νοεμβρίου), Γιωτάκης Αριστείδης, (2023, 7 Μαρτίου), MedLabNews.Κωνσταντίνου Λούβρου, (2014, 4 Μαρτίου), ΙΑΣΩ. ΚΟΡΡΕΣ ΓΕΩΡΓΙΟΣ ΦΑΙΔΩΝ (2023, 1 Μαρτίου))

2.4 Αντιμετώπιση και θεραπεία.

Υπάρχουν κάποιοι παράγοντες με καθοριστικό ρόλο στην πλήρη θεραπεία του ασθενή όπως η αιτία που προκάλεσε τη βαρηκοΐα, όταν λόγου χάριν προκαλείται από ανατομικές

δυσλειτουργίες του συστήματος αγωγής του ήχου δηλαδή βαρηκοΐα αγωγιμότητας, καθώς και και ο χρόνος που έμεινε αθεράπευτη.

Η θεραπεία της επιτυγχάνεται πλήρως με α) την αφαίρεση του βύσματος κυψελίδας, ακόμη και στο ιατρείο κάποιου ιδιώτη, β) την αποκατάσταση της τυμπανικής μεμβράνης και γ) την οσταριοπλαστική (ωτοχειρουργική) στην περίπτωση προχωρημένων βλαβών. Γενικά η θεραπεία μπορεί να είναι συντηρητική, δηλαδή με την παροχή κάποιων φαρμάκων ή χειρουργική. Επιπλέον, υπάρχουν και οι μέθοδοι διαχειρίσεις που δεν είναι άλλοι από τα ακουστικά, που αποτελεί συχνή μέθοδο για τη διόρθωση της βαρηκοΐας, την εκπαίδευση χειλικής ανάγνωσης με την οποία κάποιος κατανοεί τον λόγο μέσω των κινήσεων των χειλιών και η νοηματική γλώσσα. (Σουλαντίκας Κωνσταντίνος, (2023, 27 Νοεμβρίου), Γιωτάκης Αριστείδης, (2023, 7 Μαρτίου), MedLabNews.Κωνσταντίνου Λούβρου, (2014, 4 Μαρτίου))

2.5 Η δομή της νοηματικής γλώσσας.

Όπως έχει ειπωθεί η νοηματική γλώσσα είναι πλήρης και αυτόνομη. Έχει δική της γραμματική και δική της σύνταξη, που αποτυπώνονται με τις κινήσεις των χεριών τις εκφράσεις του προσώπου και τις κινήσεις του σώματος για να υπάρχει επικοινωνία με μηνύματα. Ανά τον κόσμο κάθε νοηματική γλώσσα έχει τη δικιά της δομή, η οποία είναι διαφορετική από τις ομιλούμενες γλώσσες, παρά το γεγονός ότι περιλαμβάνει τα βασικά στοιχεία, ήτοι υποκείμενο, ρήμα και αντικείμενο. Ωστόσο, υπάρχουν χαρακτηριστικά στοιχεία της δομής τα οποία χρησιμοποιούν όλες οι νοηματικές γλώσσες όπως τα διεθνή νοήματα.

1) Χειρομορφές (εικόνα 2.1): Είναι οι θέσεις των χεριών οι οποίες χρησιμοποιούνται από τους νομάτισες.

2) Η τοποθεσία του χεριού: Ανάλογα την τοποθέτηση του χεριού ενός νοηματιστή είτε κοντά στο σώμα είτε στον χώρο αποκτά διαφορετικό νόημα.

3) Η κίνηση του χεριού: Η κίνηση του χεριού έχει σημαντικό ρόλο στην αποτύπωση του νοήματος. Για παράδειγμα, το γράμμα ύψιλον και η λέξη αεροπλάνο είναι παρόμοια ωστόσο με την κίνηση καταλαβαίνουμε τη διαφορά.

4) Η κατεύθυνση: Το νόημα ενός σημείου μπορεί να αλλάξει με την κατεύθυνση της παλάμης ή των δακτύλων. Είναι σημαντικό για την κατεύθυνση του υποκείμενου, του αντικείμενου ή της δράσης.

5) Οι εκφράσεις προσώπου: Οι εκφράσεις που πραγματοποιεί ο νοηματιστής με το πρόσωπο είναι απαραίτητες για την έκφραση των ερωτήσεων, των αρνήσεων, του επιτονισμού.

6) Η ένταση: Ανάλογα την ταχύτητα της κίνησης εάν είναι αργή ή γρήγορη αποτυπώνεται η ένταση ή ο χρόνος.

(Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος. (2018, 9 Οκτωβρίου))



Εικόνα 2.1: Δακτυλικό αλφάβητο ελληνικής νοηματικής γλώσσας

(Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος. (2018, 26 Σεπτεμβρίου))

2.6 Η αρχή και η αναγνώριση της ελληνικής νοηματικής γλώσσας στην Ελλάδα.

Από τον 18ο αιώνα σε πολλές χώρες άρχισε ήδη η εκπαίδευση των κωφών παιδιών . Σαν πρώτη αναφορά υπάρχει η περίπτωση του ιερέα Abbe Charle Michel de l' Epee, ο οποίος ανέλαβε την εκπαίδευση των κοριτσιών μιας ενορίτισσας. Το 1755 ο ίδιος ιερέας πραγματοποίησε την ίδρυση του πρώτου σχολείου κωφών στο Παρίσι με τον ίδιο να διδάσκει εκεί.

Μεταγενέστερα αρκετοί μαθητές από αυτό το σχολείο ιδρύουν παρόμοια σχολεία σε άλλες χώρες όπως η Ολλανδία, η Πολωνία, η Ιρλανδία, η Σουηδία. Το 1817 ένας μαθητής του ιερέα ο Laurent Clerk μεταβαίνει στην Αμερική για να το πρώτο σχολείο κωφών με τον Thomas Hopkins Gallaudet και τον Mason Cogswell στις 15 Απριλίου του ίδιου έτους. Να σημειωθεί ότι ο Cogswell είχε μια κωφή εννιάχρονη κόρη. Το 1880 στο Μιλάνο πραγματοποιείται Διεθνές συνέδριο όπου συμμετείχαν εκπαιδευτικοί κωφών παιδιών. (Βικιπαίδεια. (n.d.)Ελληνική Νοηματική Γλώσσα, Εργασίες Φοιτητών ΕΤΤΑΠ (n.d.), Νοηματική Κιβωτός Πολυχώρος (2022, 20 Ιουνίου), Wikipedia. (n.d.). Thomas Hopkins Gallaudet)

Το συγκεκριμένο συνέδριο είχε μεγάλο αντίκτυπο στην ιστορία των κωφών καθώς γεννήθηκε η διαμάχη για τη χρήση ή όχι της νοηματικής γλώσσας. Το αποτέλεσμα ήταν να επικρατήσει ο προφορικός λόγος, η νοηματική γλώσσα να κριθεί ακατάλληλη και να επέλθει η απαγόρευση χρήσης της σε σχολεία. Έπαυσε να διαχωρίζεται η έννοια ομιλίας και γλώσσας, αρά ο κωφός που δε μιλάει θεωρήθηκε ασθενής και ίσως άτομο με νοητική καθυστέρηση. Η στάση αυτή σχετικά με την έννοια της νοηματικής γλώσσας αλλάζει το 1960 από τον William Stokoe. Η γλωσσική έρευνα που έκανε στη δομή της αμερικανικής νοηματικής γλώσσας καθώς και μελέτες σε θέματα κοινωνικά και γλωσσολογικά καταλήγουν στο συμπέρασμα πως οι κωφοί αποτελούν πολιτιστική και γλωσσική μειονότητα. Αρχής γενομένης από τη δεκαετία του '80 πολλές χώρες ανά το κόσμο κερδίζουν την αναγνώριση της εθνικής τους νοηματικής γλώσσας. Φτάνοντας στο πρόσφατο παρελθόν, το καλοκαίρι του 2010 στο Βανκούβερ του Καναδά, διεξάγεται ένα πολύ σημαντικό συνέδριο για το μέλλον της εκπαίδευσης των κωφών. Έγινε η αποκήρυξη όλων των αποφάσεων που πάρθηκαν στο Μιλάνο το 1880 και αναγνωρίστηκε η γλώσσα, η ιστορία και ο πολιτισμός των κωφών. (Βικιπαίδεια. (n.d.)Ελληνική Νοηματική Γλώσσα, Νοηματική Κιβωτός Πολυχώρος (2022, 20 Ιουνίου))

Στην Ελλάδα παραπομπές για τη νοηματική γλώσσα έρχονται από τα αρχαία χρόνια καθώς υπάρχουν αναφορές, όπως η ύπαρξη των κωφών στα έργα του Αριστοτέλη και του Πλάτωνα. Στους Πλατωνικούς διαλόγους φαίνεται η ύπαρξη τη νοηματικής γλώσσας με τα πλεονεκτήματα της να είναι γνωστά για εκείνη την εποχή. Η πρώτη ιστορικά τεκμηριωμένη αναφορά στη νοηματική γλώσσα εν γένει έγινε στον διάλογο Κρατύλος του Πλάτωνος το 399 π.Χ. Επίσης, αναφορά στους κωφούς και στη νοηματική γλώσσα γίνεται και σε άλλο διάλογο του Πλάτωνα τον Θεαίτητο, κάτι που δείχνει τη θετική του στάση απέναντι στους κωφούς και τη νοηματική γλώσσα. Η πρώτες προσπάθειες εκπαίδευσης κωφών στην Ελλάδα ξεκίνησαν το έτος 1907 ενώ στην Αθήνα το πρώτο σχολείο κωφών ιδρύθηκε το 1923. Μέχρι το 1984 η νοηματική γλώσσα ήταν απαγορευμένη στα σχολεία των κωφών όλης της Ελλάδας. Το 2000

με τον νόμο 2817 η ελληνική νοηματική γλώσσα αναγνωρίστηκε και εισήχθη ως το βασικό εκπαιδευτικό υλικό στην εκπαίδευση των κωφών. Τέλος, από το 2017 με τον νόμο 4488 η ελληνική νοηματική γλώσσα αναγνωρίστηκε ως επίσημη γλώσσα των κωφών και των βαρήκοων ατόμων στα ΜΜΕ, στην Εκπαίδευση και στην Πρόσβαση. (Βικιπαίδεια. (n.d.)Ελληνική Νοηματική Γλώσσα, Νοηματική Κιβωτός Πολυχώρος (2022, 20 Ιουνίου))

2.7 Η Ελληνική νοηματική γλώσσα στην εκπαίδευση .

Με την κατάλληλη εκπαίδευση των κωφών και των βαρήκοων μέσω της νοηματικής γλώσσας, μπορούν πλήρως να συμμετάσχουν στην κοινωνία διότι η γλώσσα είναι το βασικό εργαλείο επικοινωνίας.

Επιπλέον, μέσω της νοηματικής γλώσσας μπορούν τα παιδιά να διδαχθούν και να διατηρήσουν μετέπειτα τις παραδόσεις, τις αξίες, την κουλτούρα της κοινότητας των κωφών. Ωστόσο, η νοηματική γλώσσα δε θα έπρεπε να μαθαίνεται μόνο από τους κωφούς, τους βαρήκοους, τους δασκάλους, και το οικογενειακό περιβάλλον, αλλά θα έπρεπε να διδάσκεται στα σχολεία σε όλους τους μαθητές συμπεριλαμβανομένων των ατόμων με υγιή ακοή, καθώς θα υπήρχε μεγαλύτερη κατανόηση της κουλτούρας των κωφών. (Νοηματική Κιβωτός Πολυχώρος. (2023, 10 Νοεμβρίου))

2.8 Η Ελληνική νοηματική γλώσσα στον τομέα της υγείας και τα ΜΜΕ.

Η χρήση της νοηματικής γλώσσας είναι απαραίτητη σε όλους τους τομείς και ιδιαίτερα στον τομέα της υγείας. Η γνώση της νοηματικής γλώσσας από το ιατρικό προσωπικό κρίνεται αναγκαία, καθώς σε καταστάσεις έκτακτης ανάγκης η ακριβής επικοινωνία ανάμεσα στον γιατρό και τον ασθενή με προβλήματα ακοής μπορεί να αποτελέσει ζωτικής σημασίας. Επιπροσθέτως για τη σωστή ενημέρωση και διάγνωση χρειάζεται η σαφή κατανόηση των αναγκών του ασθενή. Επομένως, οι λόγοι που αναφέρθηκαν καθιστούν αναγκαία τη γνώση της νοηματικής γλώσσας στον υγειονομικό τομέα.

Για τον τομέα των μέσων μαζικής ενημερώσεως η νοηματική γλώσσα χρήζει επιτακτικής ανάγκης. Στην τηλεόραση, σε δημοσιές ανακοινώσεις η χρήση της νοηματικής γλώσσας διασφαλίζει την ισότιμη ενημέρωση των κωφών. Στην Ελλάδα υπάρχει πρόοδος στη χρήση υποτίτλων ή διερμηνέων για τη διευκόλυνση των ατόμων με βαρηκοΐα ή κώφωση. Αλλά η νομοθεσία παρά το γεγονός ότι απαιτεί τη συμπερίληψη της νοηματικής γλώσσας, δεν υπάρχει εφαρμογή της σε όλο το φάσμα των Ελληνικών καναλιών. (Νοηματική Κιβωτός Πολυχώρος. (2023, 10 Νοεμβρίου), Λαμπροπούλου Ρ. (2022, 11 Ιουλίου))

3 Η Σημασία της Προσβασιμότητας στην Ψηφιακή Εποχή για τα Άτομα με Ακουολογικές Διαταραχές.

3.1 Προσβασιμότητα και ίσες ευκαιρίες στην ψηφιακή επικοινωνία.

Η ψηφιακή προσβασιμότητα είναι θεμελιώδης για τη διασφάλιση ίσων ευκαιριών στη σύγχρονη επικοινωνία. Αδιαμφισβήτητη στη σημερινή κοινωνία, η πρόσβαση στις τεχνολογίες της πληροφορίας και της επικοινωνίας είναι απαραίτητη για την εκπαίδευση, την εργασία και την κοινωνική συμμετοχή/ενσωμάτωση. Ωστόσο, για πολλούς, η πρόσβαση στις ψηφιακές υπηρεσίες είναι περιορισμένη, έως και αδύνατη κάποιες φορές, λόγω φυσικών περιορισμών ή έλλειψης τεχνολογικών δεξιοτήτων. Αυτή η "ψηφιακή ανισότητα" μπορεί να επηρεάσει αρνητικά ευάλωτες κοινωνικές ομάδες, όπως άτομα με αναπηρίες ή μειωμένες τεχνολογικές δεξιότητες, ηλικιωμένους, άτομα που ζουν σε δύσβατες περιοχές ή απλά δεν έχουν την οικονομική δυνατότητα να εγκαταστήσουν ψηφιακές υποδομές, με αποτέλεσμα την περιθωριοποίηση του και την κοινωνική του απομόνωση. Η ψηφιακή προσβασιμότητα βάζει τα θεμέλια για την δημιουργία ενός περιβάλλοντος, μέσα στο οποίο όσοι δυσκολεύονται να αποκτήσουν πρόσβαση, να έχουν άμεση πρόσβαση, ανεξαρτήτως φυσικών ή τεχνολογικών ικανοτήτων και δυσκολιών. Η προσβασιμότητα καθορίζεται από διεθνείς κατευθυντήριες γραμμές, όπως οι WCAG (Web Content Accessibility Guidelines), που ορίζουν τα κριτήρια για την ανάπτυξη προσβάσιμων ιστοσελίδων και εφαρμογών. Οι οδηγίες αυτές προβλέπουν ότι οι ιστότοποι πρέπει να είναι αντιληπτοί, λειτουργικοί και κατανοητοί από όλους τους χρήστες, περιλαμβανομένων ατόμων με αναπηρίες. Η ψηφιακή επικοινωνία περιλαμβάνει την ανάπτυξη λύσεων που ενσωματώνουν τη δυνατότητα πλοήγησης μέσω φωνητικών εντολών ή την προσθήκη υποτίτλων σε βίντεο. Ειδικά για τα άτομα με αναπηρίες, οι τεχνολογικές λύσεις, όπως τα προγράμματα ανάγνωσης οθόνης ή η αυτόματη αναγνώριση φωνής, είναι ζωτικής σημασίας για τη διευκόλυνση της πρόσβασης τους σε υπηρεσίες και πληροφορίες. Αξιοθαύμαστο είναι δε ότι η Ελλάδα έχει κάνει σημαντικά άλματα προόδου για την ενίσχυση της ψηφιακής προσβασιμότητας μέσω του νόμου 4727/2020, που ενσωματώνει την Οδηγία 2016/2102 της ΕΕ, ο οποίος και υποχρεώνει τους δημόσιους φορείς να εξασφαλίζουν ότι οι ιστοσελίδες τους είναι άμεσα προσβάσιμες από άτομα με αναπηρίες ή άλλες δυσκολίες, ενισχύοντας έτσι την ισότητα στις ψηφιακές υπηρεσίες και ταυτόχρονα αμβλύνοντας το κοινωνικό και οικονομικό χάσμα. (GSIS. (n.d.), SecDigital. (2023, Μάιος), W3C. (2023, 21 September))

3.2 Νομοθεσία και δικαιώματα για την προσβασιμότητα των ατόμων με ακοολογικές διαταραχές.

Η νομοθεσία σχετικά με την ψηφιακή προσβασιμότητα για τα άτομα με ακοολογικές διαταραχές αποτελεί ένα από τα πιο σημαντικά βήματα για την ισότιμη συμμετοχή τους στην κοινωνία. Συγκεκριμένα, Ο νόμος 4727/2020, που ενσωματώνει την Οδηγία 2016/2102 της ΕΕ, θέτει αυστηρά πρότυπα για την προσβασιμότητα των δημόσιων ιστοσελίδων και εφαρμογών. Αυτή η νομοθεσία διασφαλίζει ότι οι ιστότοποι του δημόσιου τομέα θα είναι προσβάσιμοι και θα περιλαμβάνουν εργαλεία που διευκολύνουν την αλληλεπίδραση των ατόμων με ακοολογικές διαταραχές. Οι ιστοσελίδες, εξάλλου, οφείλουν να παρέχουν εναλλακτικές λύσεις για τους χρήστες που δεν μπορούν να ακούσουν το περιεχόμενο, όπως υπότιτλους για τα βίντεο ή γραπτές περιγραφές ηχητικών πληροφοριών. Επιπλέον, η χρήση τεχνολογιών όπως οι μετατροπείς φωνής σε κείμενο ή οι αυτοματοποιημένοι υπότιτλοι συμβάλλουν σημαντικά στη βελτίωση της εμπειρίας αυτών των ατόμων κατά την πλοήγηση στον ψηφιακό κόσμο. Παρά τις θετικές εξελίξεις ως προς την δημιουργία ενός νομοθετικού πλαισίου, στην πράξη η εφαρμογή της νομοθεσίας αυτής παρουσιάζει προκλήσεις, καθώς απαιτεί σημαντικές προσαρμογές στις ήδη υπάρχουσες πλατφόρμες και ιστοσελίδες, το σίγουρο όμως είναι ότι οι οργανισμοί του δημόσιου και ιδιωτικού τομέα οφείλουν να επενδύσουν στην ανάπτυξη προσβάσιμων εργαλείων και υπηρεσιών, διασφαλίζοντας ότι η προσβασιμότητα δεν είναι απλώς τυπική υποχρέωση, αλλά ουσιαστική δέσμευση για την ισότητα. (GSIS. (n.d.), SecDigital. (2023, Μαιος), W3C. (2023, 21 September))

3.3 Ρόλος της τεχνολογίας στην ενσωμάτωση της νοηματικής γλώσσας στις ψηφιακές πλατφόρμες.

Η τεχνολογία παίζει κεντρικό ρόλο στην ενσωμάτωση της νοηματικής γλώσσας στις ψηφιακές πλατφόρμες, καθιστώντας την επικοινωνία ευκολότερη για τα άτομα με ακοολογικές διαταραχές. Ειδικότερα, οι ψηφιακές πλατφόρμες που υποστηρίζουν τη νοηματική γλώσσα επιτρέπουν στα άτομα με ακοολογικές διαταραχές να έχουν πρόσβαση σε περιεχόμενο που διαφορετικά θα ήταν απρόσιτο. Ένα καλό παράδειγμα είναι οι πλατφόρμες κοινωνικής δικτύωσης και οι εφαρμογές βίντεο, όπως το YouTube, που προσφέρουν δυνατότητες υποτίτλων ή τη χρήση ειδικών εφαρμογών για τη νοηματική γλώσσα. Ακόμη, οι νέες τεχνολογίες αναγνώρισης φωνής και μετάφρασης νοηματικής γλώσσας επιτρέπουν τη ζωντανή μετάφραση της νοηματικής γλώσσας σε γραπτό κείμενο ή ομιλία, διευκολύνοντας τη συμμετοχή των ατόμων με αναπηρίες σε ψηφιακές συζητήσεις και τηλεδιασκέψεις. Επιπλέον,

δε, η τεχνητή νοημοσύνη, μια τεχνολογία τόσο σημαντική και ανερχόμενη τη σήμερον ημέρα, χρησιμοποιείται για την αυτόματη αναγνώριση της νοηματικής γλώσσας και τη μετατροπή της σε κατανοητό κείμενο ή φωνή, δίνοντας έτσι τη δυνατότητα σε άτομα με ακοολογικές διαταραχές, έχοντας άμεση πρόσβαση σε κάθε είδους πληροφορία, να ενσωματωθούν στην εκπαίδευση και αργότερα να αφομοιωθούν “αναίμακτα” σε κάθε εργασιακό χώρο. Εξίσου σημαντικό όμως είναι το εργαλείο αυτό, ήτοι η χρήση των τεχνολογιών αυτών, όχι μόνο για την επικοινωνία, αλλά και για την εκμάθηση της νοηματικής γλώσσας μέσω ειδικών εφαρμογών, που χρησιμοποιούν την τεχνητή νοημοσύνη για την αναγνώριση της νοηματικής γλώσσας και την εκμάθησή της, ώστε τα άτομα χωρίς αναπηρίες να μάθουν να επικοινωνούν καλύτερα με τους κωφούς ή βαρήκοους χρήστες. (Open Standards. Γιάννης Παπαβασιλείου (2020, 2 Δεκεμβρίου))

3.4 Ανάγκη για προσαρμογή των εφαρμογών και των ιστοσελίδων για βαρήκοους.

Η ανάγκη για προσαρμογή των εφαρμογών και των ιστοσελίδων για βαρήκοους είναι εξίσου σημαντική για τη διασφάλιση της ίσης πρόσβασης στις ψηφιακές υπηρεσίες και πληροφορίες. Συγκεκριμένα, πολλές πλατφόρμες δεν παρέχουν τις απαραίτητες δυνατότητες ώστε να καλύπτουν τις ανάγκες αυτής της πληθυσμιακής ομάδας, με αποτέλεσμα την αποκλειστική εξάρτησή τους από ηχητικά μέσα ή περιεχόμενο χωρίς επαρκή υποστήριξη υποτίτλων. Όμως για να είναι μια ιστοσελίδα κατ’ ουσίαν προσβάσιμη σε βαρήκοους χρήστες, πρέπει να προσφέρει εναλλακτικές λύσεις σε κάθε ακουστικό περιεχόμενο, όπως η ύπαρξη πλήρους γραπτού αντιγράφου ή υποτίτλων για τα βίντεο, αλλά και σαφή περιγραφή για οποιαδήποτε πληροφορία παρέχεται μόνο μέσω ηχητικών αρχείων. Επιπλέον, είναι σημαντικό να ενσωματωθούν εργαλεία αναγνώρισης ομιλίας που μετατρέπουν την ομιλία σε κείμενο, όπως τα live-captioning συστήματα, τα οποία προσφέρουν τη δυνατότητα σε βαρήκοους χρήστες να παρακολουθούν συζητήσεις ή βίντεο σε πραγματικό χρόνο. Η προσαρμογή των ιστοσελίδων ώστε να ακολουθούν τις οδηγίες του WCAG είναι απαραίτητη για να διασφαλιστεί η δυνατότητα εύκολης πλοήγησης και κατανόησης από χρήστες με περιορισμένη ακουστική ικανότητα. Αυτά περιλαμβάνουν και την ενσωμάτωση μεγάλων, καθαρών κειμένων και οπτικών εργαλείων που αντικαθιστούν ή υποστηρίζουν το ηχητικό περιεχόμενο. Η πραγματική πρόκληση που έρχονται να αντιμετωπίσουν πολλές επιχειρήσεις και δημόσιοι οργανισμοί είναι η δυσκολία και το κόστος που απαιτείται για την προσαρμογή υφιστάμενων ιστοσελίδων. Παρόλα αυτά, η επένδυση σε τεχνολογίες που προσφέρουν προσβάσιμες λύσεις όχι μόνο αυξάνει την προσβασιμότητα, αλλά βελτιώνει και τη συνολική εμπειρία του χρήστη, κάνοντας

τις ψηφιακές υπηρεσίες περισσότερο λειτουργικές για όλους τους πολίτες. Η προσβασιμότητα δεν είναι απλώς ένα τεχνικό ζήτημα, αλλά θεμελιώδες δικαίωμα κάθε πολίτη, και η εξασφάλισή της προάγει την ισότητα και την ένταξη. (GSIS. (n.d.), SecDigital. (2023, Μαιος), W3C. (2023, 21 September))

4 Τεχνητή Νοημοσύνη.

4.1 Ορισμός της τεχνητής νοημοσύνης.

Η τεχνητή νοημοσύνη (Artificial Intelligence ή AI) είναι η ανάπτυξη των μηχανών που έχουν πρόσβαση σε πληροφορίες και μπορούν να σκέφτονται σαν τους ανθρώπους. Δηλαδή, επιτρέπει στις μηχανές να λαμβάνουν αποφάσεις, να επιλύουν προβλήματα και να πολλά ακόμη ζητήματα, χωρίς να απαιτείται η συνεχής βοήθεια από τον άνθρωπο. Τα “συστατικά” για τη δημιουργία της AI είναι η μηχανική μάθηση, η επεξεργασία της φυσικής γλώσσας, η όραση υπολογιστών και η ρομποτική. Πλέον με τη χρήση που γίνεται είναι αποδεδειγμένο ότι η AI είναι από τα πιο καινοτόμα εργαλεία της εποχής μας. Ένα συμπέρασμα το οποίο επέρχεται καθώς πλέον είναι αναπόσπαστο κομμάτι πολλών βιομηχανιών και τομέων, από την υγεία έως τις επιχειρήσεις. Η ικανότητα της να επεξεργάζεται τεράστια ποσοστά δεδομένων γρήγορα και αποτελεσματικά, βοηθά στην εξοικονόμηση χρόνου και χρήματος, βελτιώνοντας παράλληλα την ακρίβεια και την παραγωγικότητα στις διαδικασίες που της ζητούνται. Πρέπει να σημειωθεί ότι η AI έχει βοηθήσει σε πολύπλοκα ζητήματα όπως είναι η κλιματική αλλαγή, η ανίχνευση ασθενειών και η ανακάλυψη νέων φαρμάκων. Με τη χρήση των εξελιγμένων αλγορίθμων, η AI προσφέρει λύσεις σε τομείς που απαιτούν υψηλό επίπεδο ανάλυσης. Εν κατακλείδι η τεχνητή νοημοσύνη με τις δυνατότητες που προσφέρει δε βοηθά μόνο στην επίλυση των δύσκολων προβλημάτων αλλά βελτιώνει το ποιοτικό επίπεδο ζωής του ανθρώπου καθιστώντας τις καθημερινές εργασίες πιο εύκολες και πιο αποδοτικές. (Greeco. Greeco Team (2023, 1 Δεκεμβρίου)).

4.2 Η ιστορία και η εξέλιξη της τεχνητής νοημοσύνης.

Η ιστορία της τεχνητής νοημοσύνης ξεκινάει περισσότερο από 2000 χρόνια, όπου φιλόσοφοι όπως ο Αριστοτέλης, ο Ηράκλειτος, ο Descartes και άλλοι προσπάθησαν να περιγράψουν τους μηχανισμούς μάθησης, απομνημόνευσης, όρασης, αντίληψης και συλλογισμού. Οι απαρχές της τεχνητής νοημοσύνης ανάγονται στους συλλογισμούς του Αριστοτέλη (384-322 π.χ.), οι οποίοι παρείχαν πρότυπα εκφράσεων που έδιναν πάντα σωστά συμπεράσματα από σωστές υποθέσεις βασισμένα στην Αριστοτέλεια Συλλογιστική. Η AI εξετάζει ερευνητικά πεδία τα οποία ακουμπούν πάνω σε αυτές τις δεξιότητες. Το 1956 ο John McCarthy χρησιμοποίησε για πρώτη φορά τον όρο AI για να περιγράψει ένα θερινό εργαστήριο με την ονομασία “The Dartmouth Summer Research Project on Artificial Intelligence”. Το βασικό θέμα προς συζήτηση ήταν “οι μηχανές που σκέφτονται”.

Αυτό θεωρήθηκε το πρώτο συνέδριο της ΑΙ που ήταν καθοριστικό στη γέννηση και την υλοποίηση της. Δόθηκε η εγκυκλοπαιδική ερμηνεία “ικανότητα της μηχανής να μπορεί να σκέφτεται και να μιμείται την ανθρώπινη συμπεριφορά και ευφυΐα, αλλά να μην την αντικαθιστά”. Η αρχική έρευνα της ΑΙ εστίασε στην επίλυση προβλημάτων και σε θέματα συμβολικών μεθόδων. Στη δεκαετία του 1960 υπήρξε ενδιαφέρον για αυτόν τον τύπο εργασίας από το υπουργείο άμυνας των ΗΠΑ. Ήταν το έναυσμα για να ξεκινήσει η εκπαίδευση των υπολογιστών στη μίμηση της βασικής ανθρώπινης συλλογιστικής. Η προεργασία αυτή δημιούργησε τις βάσεις για την αυτοματοποίηση και την τυπική συλλογιστική, με την οποία έρχονται σε επαφή οι άνθρωποι τη σήμερον ημέρα μέσω των υπολογιστών, έχοντας συμπεριλάβει τα συστήματα υποστήριξης λήψης αποφάσεων και των έξυπνων συστημάτων αναζήτησης που μπορούν να σχεδιαστούν για τη συμπλήρωση και τη βελτίωση των ανθρώπινων ικανοτήτων. Ειδική αναφορά πρέπει να γίνει στον μαθηματικό Alan Matheson Turing, ο οποίος θεωρείται ο Πατέρας της επιστήμης των υπολογιστών. Η συνεισφορά του στην πρόοδο της τεχνικής νοημοσύνης είναι μεγάλη, καθώς κατά τη διάρκεια του Δευτέρου Παγκοσμίου Πολέμου εργάστηκε για την αποκρυπτογράφηση των γερμανικών κρυπτογραφημένων μηνυμάτων. Η εμπειρία αυτή τον οδήγησε στην ανάπτυξη του μοντέλου μηχανής Turing, ένα θεωρητικό μοντέλο υπολογιστικής μηχανής που θα μπορούσε να προσομοιώσει οποιονδήποτε αλγόριθμο. Φτάνοντας, τέλος, στην περίοδο που αποκαλείται χειμώνας της τεχνητής νοημοσύνης, όρος που αποδόθηκε επειδή οι προσδοκίες για την τεχνητή νοημοσύνη ήταν υπερβολικά υψηλές, αλλά η αδυναμία των υπολογιστών να αναπτύξουν πραγματική νοημοσύνη οδήγησε στην απόσυρση της έρευνας σε αυτόν τον τομέα μέχρι την περίοδο που πραγματοποιήθηκε η εξέλιξη των νευρωνικών δικτύων. Το νευρωνικό δίκτυο είναι το δίκτυο που αποτελείται από υπολογιστικούς κόμβους (νευρώνες , νευρόνια) διασυνδεδεμένους μεταξύ τους. Το δίκτυο αυτό είναι εμπνευσμένο από τη λειτουργία του ανθρώπινου εγκεφάλου. Με την εξέλιξη των νευρωνικών δικτύων ήρθε και η επανεκκίνηση της τεχνητής νοημοσύνης. Εξαιτίας των νευρωνικών δικτύων οι υπολογιστές κατάφεραν να μαθαίνουν από τα δεδομένα και να βελτιώνουν την απόδοσή τους με τον χρόνο. Με αυτή την προσέγγιση κατάφερε να ανοίξει ο δρόμος για νέες δυνατότητες στην αναγνώριση μοτίβων, τη μηχανική μάθηση και την επεξεργασία της φυσικής γλώσσας. (Greek Tech. (2023, 9 Αυγούστου), Βικιπαίδεια. (n.d.). Άλαν Τούρινγκ, Σαπκάς, Dimitris Stefanidis (2020, 19 Σεπτεμβρίου)).

4.3 Machine learning και Deep learning.

Το ΑΙ βασίζεται σε πολλές τεχνολογίες. Το Machine learning (ML) είναι μια από αυτές. Σκεπτόμενοι τη λειτουργία της ανθρώπινης σκέψης και της γνώσης έχει γίνει κατανοητό, όπως ακριβώς ο τρόπος που αντλούνται πληροφορίες από το περιβάλλον δημιουργώντας συγκεκριμένα συμπεράσματα. Μια αντίστοιχη λογική είναι και αυτή που βασίζεται το ML. Το ML αποτελείται από αλγόριθμους που επεξεργάζονται ένα σύνολο πληροφοριών. Ανάλογα τις πληροφορίες που δέχεται, δηλαδή τα δεδομένα, ο αλγόριθμος θα κατασκευάσει τη δική του λογική και θα προσαρμόσει τη λειτουργία του. Ένα παράδειγμα (πίνακας 4.1) για την ύπαρξη καλύτερης κατανόησης σε αυτά που αναφέρονται :

1	6
2	12
3	18
4	24
5	?

Πίνακας 4.1: Εύρεση αριθμού

Ο άνθρωπος εύκολα θα καταλάβει ότι στη θέση του ερωτηματικού θα τοποθετηθεί ο αριθμός 30. Ένα απλό συμβατικό πρόγραμμα θα έπρεπε να είναι γραμμένο για να λύσει ειδικά αυτό το πρόβλημα. Συγκεκριμένα, το πρόβλημα που δημιουργείται είναι πως το ίδιο πρόγραμμα θα είναι ανεπαρκές για κάποιο άλλο πρόβλημα που θα χρήζει επίλυσης. Μέσω του ML γίνεται προσπάθεια ένας υπολογιστής να λειτουργεί περισσότερο σαν άνθρωπος. Το ML επιτρέπει στους αλγόριθμους να μαθαίνουν εμπειρικά και να βελτιώνουν την απόδοσή τους στην πάροδο του χρόνου. Καθώς ο αριθμός των δειγμάτων αυξάνεται , τόσο ο αλγόριθμος γίνεται αποδοτικότερος. Για τη λειτουργία του ML είναι γνωστές μόνο τις πολύ βασικές αρχές αυτής της μεθόδου, καθώς ο αλγόριθμος που χρησιμοποιεί κάθε επιχείρηση αποτελεί σημαντικό εμπορικό μυστικό. Ένα απλό παράδειγμα είναι η σύγκριση της Google με το Bing. Μια αναζήτηση στην Google θα δώσει καλύτερα αποτελέσματα. Αυτό συμβαίνει καθώς οι αλγόριθμοι που χρησιμοποιούνται από την Google είναι πολύ ανώτεροι από των ανταγωνιστών της. Το ML αποτελείται από διαφορετικούς τύπους μοντέλων μηχανικής μάθησης χρησιμοποιώντας διάφορες αλγοριθμικές τεχνικές που ανάλογα με τη φύση των δεδομένων και το επιθυμητό αποτέλεσμα επάγονται σε τέσσερα μοντέλα μάθησης: επιβλεπόμενη μάθηση, μην επιβλεπόμενη μάθηση, ημι-επιβλεπόμενη μάθηση και η ενισχυμένη μάθηση.

- Η επιβλεπόμενη μάθηση: Στους επιτηρούμενους αλγορίθμους μάθησης, η μηχανή διδάσκεται από το παράδειγμα. Δηλαδή η εκπαίδευση του γίνεται από ένα σύνολο μοντέλου με ετικέτα. Το μοντέλο μαθαίνει από τα παρεχόμενα παραδείγματα, όπου κάθε παράδειγμα αποτελείται από μια είσοδο και την αντίστοιχη αναμενόμενη έξοδο.
- Η μη επιβλεπόμενη μάθηση: Στο μοντέλο μη επιβλεπόμενης μάθησης δεν υπάρχει κλειδί απάντησης. Ο στόχος είναι η ανακάλυψη κρυφών μοτίβων ή δομών που είναι εγγενείς στα δεδομένα. Η συγκεκριμένη μάθηση είναι ιδιαίτερα χρήσιμη όταν η δομή των δεδομένων είναι άγνωστη.
- Η ημι-επιβλεπόμενη μάθηση: Στην ημι-επιβλεπόμενη μάθηση το μοντέλο χρησιμοποιείται όταν υπάρχουν τεράστιες ποσότητες ακατέργαστων και μη δομημένων δεδομένων. Το μοντέλο αποτελείται από την εισαγωγή μικρών ποσοτικών δεδομένων με ετικέτα για να αυξήσουν το σύνολο δεδομένων χωρίς ετικέτα. Με αυτό τον τρόπο τα επισημασμένα δεδομένα ενεργούν για να δώσουν μια αρχή λειτουργίας στο σύστημα και βελτιώνουν σημαντικά την ταχύτητα και την ακρίβεια της μάθησης.
- Η ενισχυμένη μάθηση: Το μοντέλο της ενισχυμένης μάθησης δεν περιλαμβάνει ένα κλειδί απάντησης αλλά εισάγει ένα σύνολο επιτρεπόμενων ενεργειών, κανόνων και πιθανών τελικών καταστάσεων. Με αυτό το μοντέλο οι μηχανές μπορούν να μάθουν με το παράδειγμα όταν ο επιθυμητός στόχος είναι σταθερός ή δυαδικός. Όταν το επιθυμητό αποτέλεσμα είναι μεταβλητό τότε το σύστημα μαθαίνει από την εμπειρία και την ανταμοιβή. Αυτό το είδος της μάθησης έχει εφαρμοστεί σε διάφορους τομείς, όπως η ρομποτική και τα παιχνίδια. Η ενισχυμένη μάθηση είναι μια πολλά υποσχόμενη προσέγγιση για τη δημιουργία ευφυών παραγόντων που είναι ικανοί να μαθαίνουν και να λαμβάνουν αποφάσεις αυτόνομα. (PC Steps. Γιάννης Φυτίλης (2018, 11 Απριλίου), SAP. (n.d.)).

Το Deep learning είναι ένα υποπεδίο της μηχανικής μάθησης που χαρακτηρίζεται από τη χρήση αλγορίθμων και τεχνητών νευρωνικών δικτύων για την εκμάθηση ιεραρχικών αναπαραστάσεων δεδομένων. Το Deep learning επιτρέπει στις μηχανές να μαθαίνουν αυτόματα από σύνθετα μοτίβα μέσω πολλαπλών επιπέδων επεξεργασίας. Αυτή είναι η διαφορά που υπάρχει με την παραδοσιακή μάθηση η οποία βασίζεται σε συγκεκριμένους αλγορίθμους με προκαθορισμένους κανόνες. Η αποτελεσματικότητα του Deep learning επάγεται σε εργασίες που απαιτούν υψηλό επίπεδο αφαίρεσης και κατανόησης δεδομένων όπως είναι η αναγνώριση ομιλίας, η ανίχνευση αντικειμένων από εικόνες και η αυτόματη μετάφραση. Στον τομέα του Deep learning χρησιμοποιούνται διάφορες αρχιτεκτονικές και

μοντέλα για την επίλυση σύνθετων προβλημάτων στην επεξεργασία δεδομένων. Οι αρχιτεκτονικές αυτές αποτελούνται από δομές τεχνητών νευρωνικών δικτύων που έχουν σχεδιαστεί για να μαθαίνουν αυτόματα και να αναπαριστούν μοτίβα και χαρακτηριστικά δεδομένων. Η αρχιτεκτονική Convolutional Neural Network (CNN) αποτελεί μια από τις πιο χρήσιμες στο Deep learning είναι ιδιαίτερα αποτελεσματική στην επεξεργασία εικόνας και η χρήση της έχει πραγματοποιηθεί σε διάφορες εφαρμογές όπως η αναγνώριση αντικειμένων, η ανίχνευση αντικειμένων και η ταξινόμηση εικόνων. Αυτά τα δίκτυα χαρακτηρίζονται από την ικανότητα τους να μαθαίνουν ιεραρχικές αναπαραστάσεις μέσω επιπέδων συνέλιξης και υποδειγματοληψίας. Η Recurrent Neural Network (RNN) είναι άλλη μια αρχιτεκτονική η οποία είναι ιδανική για την επεξεργασία της ροής και χρησιμοποιείται σε εφαρμογές όπως η αναγνώριση ομιλίας, η αυτόματη μετάφραση και η δημιουργία κειμένου. Αυτά τα δίκτυα επάγονται σε συνδέσεις μεταξύ νευρώνων που σχηματίζουν βρόχους που τους επιτρέπει να διατηρούν μια εσωτερική μνήμη και να συλλαμβάνουν μακροπρόθεσμη εξάρτηση από τα δεδομένα.(PCLT. (2024, 3 Φεβρουαρίου.), Βικιπαίδεια.(n.d.). Νευρωνικό Δίκτυο).

Το Deep learning είναι ένα ισχυρό εργαλείο σε πολλούς και διάφορους τομείς την σήμερα. Ο τομέας της τεχνητής όρασης είναι ένας από αυτούς τους τομείς. Εξαιτίας των συνελκτικών νευρωνικών δικτύων είναι δυνατόν να γίνει ο εντοπισμός, η ταξινόμηση σε εικόνες με ακρίβεια και αποτελεσματικότητα. Αυτό έχει σαν αποτέλεσμα να επέλθει μεγάλη πρόοδος σε τομείς όπως η ανίχνευση αντικειμένων σε ιατρικές εικόνες, την αυτόνομη οδήγηση και την ασφάλεια στη βιντεοπαρακολούθηση. Επιπλέον, στον τομέα της επεξεργασίας φυσικής γλώσσας είχε σημαντικό αντίκτυπο το Deep learning καθώς χρησιμοποιούνται γλωσσικά μοντέλα που βασίζονται σε νευρωνικά δίκτυα. Τα συστήματα βελτίωσαν την ικανότητα τους στην κατανόηση και στη δημιουργία κειμένου με πιο συνθετικό και φυσικό τρόπο. Έτσι επήλθε η ανάπτυξη σε εφαρμογές με εικονικούς βοηθούς και συστήματα αυτόματης μετάφρασης η οποία διευκόλυνε την αλληλεπίδραση των ανθρώπων με τις μηχανές σε διαφορετικές γλώσσες και περιβάλλοντα. Ο τομέας της ιατρικής που δεν είναι καθόλου αμελητέος έχει προοδεύσει εξαιτίας του Deep learning. Με τα νευρωνικά δίκτυα οι ερευνητές έχουν επιτύχει ακριβέστερες και ταχύτερες διαγνώσεις σε τομείς που εξυπηρετεί η έγκαιρη ανίχνευση ασθενειών, ο εντοπισμός προτύπων σε ιατρικές εικόνες και η πρόβλεψη κλινικών αποτελεσμάτων. Η συμβολή αυτή είχε σαν αποτέλεσμα τη βελτίωση των θεραπειών και τη διάσωση ζωών. Καθώς υπάρχει συνεχής πρόοδος της τεχνολογίας και της έρευνας μαζί με τη βοήθεια του Deep learning θα επέλθει η ανάπτυξη καινοτόμων λύσεων στο μέλλον. Ανάμεσα στο ML και το Deep learning υπάρχουν διαφορές.

- **Δεδομένα:** Τα δεδομένα που χρησιμοποιεί το ML είναι structured data ενώ το deep learning χρησιμοποιεί Artificial Neural Network (ANN).
- **Ανεξαρτησία:** Όταν ο αλγόριθμος του ML μπει σε λειτουργία θα πρέπει να υπάρχει μία επίβλεψη από έναν data analyst, για να παρακολουθεί τις διάφορες μεταβλητές. Αντίθετα, στο deep learning ο αλγόριθμος έχει μεγαλύτερη ανεξαρτησία αναφορικά με την ανάλυση δεδομένων.
- **Αποτελεσματικότητα:** Οι αλγόριθμοι του ML δημιουργούνται αρκετά γρήγορα αλλά η αποτελεσματικότητά τους στερεί. Ενώ οι αλγόριθμοι του deep learning παρόλο που χρειάζονται χρόνο για να δημιουργηθούν έχουν υψηλή αποτελεσματικότητα. Επίσης, με την πάροδο του χρόνου και προσθέτοντας περισσότερα δεδομένα βελτιώνονται συνεχώς.
- **Διάρκεια εκτέλεσης:** Στο ML οι αλγόριθμοι χρειάζονται λιγότερο χρόνο για την εκτέλεση του μοντέλου αλλά αρκετό χρόνο για να δοκιμαστεί το μοντέλο. Οι αλγόριθμοι του deep learning χρειάζονται περισσότερο χρόνο εκτέλεσης αλλά έχουν πιο σύντομη δοκιμή.
- **Τρόπος επίλυσης προβλημάτων:** Η προσέγγιση επίλυσης στο ML είναι ο αλγόριθμος να παίρνει το πρόβλημα, να το σπάει σε μικρότερα κομμάτια, να επιλύει το καθένα ξεχωριστά και στο τέλος να λύσει ολόκληρο το πρόβλημα. Ενώ στο deep learning ο αλγόριθμος λαμβάνει το πρόβλημα και δίνει κατευθείαν την απάντηση.

Τόσο το ML όσο και το deep learning προσφέρουν διαφορετικά οφέλη καθώς δουλεύουν διαφορετικά. Το ποιο θα χρησιμοποιηθεί εξαρτάται από το μέγεθος και την πολυπλοκότητα του προβλήματος.(PC Steps. Γιάννης Φυτίλης (2018, 11 Απριλίου), Tecnobits. Σεμπάστιαν Βιδάλ (2023, 9 Ιουλίου), Big Blue Academy. (n.d.), SAP. (n.d.)).

4.4 Εφαρμογές της τεχνητής νοημοσύνης.

Η τεχνητή νοημοσύνη, όπως προαναφέρθηκε, μαζί με την ανάπτυξη της τεχνολογίας βρήκε εφαρμογή σε πολλούς επαγγελματικούς κλάδους. Στον κλάδο της υγείας η τεχνητή νοημοσύνη χρησιμοποιείται για την ανάλυση δεδομένων για την ταχύτερη διάγνωση των ασθενών. Την έγκαιρη ανίχνευση ασθενειών και τη γρήγορη έναρξη θεραπειών. Επιπλέον, με τη χρήση της ΑΙ γίνεται ευκολότερη η παρακολούθηση της πορείας των ασθενών, χρησιμοποιείται στη γενετική ερευνά καθώς και στη ρομποτική χειρουργική. Η ΑΙ βοηθάει στην πρόβλεψη ασθενειών, μπορεί να προβλέψει πιθανά προβλήματα υγείας με βάση την ηλικία, το γενετικό ιστορικό. Στον τομέα της οικονομίας η ΑΙ έχει βοηθήσει τις τράπεζες και τα

χρηματοπιστωτικά ιδρύματα στον εντοπισμό περιστατικών απάτης και κατάχρησης, στην ανάλυση της πιστοληπτικής ικανότητας, τη διαχείριση χαρτοφυλακίων, καθώς και την παροχή εξατομικευμένων οικονομικών συμβουλών. Στην αγορά η ΑΙ βοηθά αποτελεσματικά να μελετηθούν οι καταναλωτές και οι εν δυνάμει πελάτες βάσει των συνηθειών τους. Ανάλογα με την αναζήτηση προϊόντων, υπηρεσιών και των αντίστοιχων αγορών που κάνουν , δημιουργούνται εξατομικευμένες προσφορές με την επιχείρηση να προσφέρει στους πελάτες αυτό που ακριβώς επιθυμούν. Σε τομείς όπως ο γεωργικός τομέας βρήκε πεδίο άνθησης η ΑΙ, καθώς γίνεται χρήση μηχανημάτων, όπως οι αυτοκίνητες θεριζοαλωνιστικές μηχανές, για τη συγκομιδή σιτηρών και καλαμποκιού, ενώ υπάρχει ρομπότ για τη συγκομιδή σταφυλιών. Αυτά τα οχήματα με τη χρήση της ΑΙ μπορούν να ανιχνεύσουν το περιβάλλον, να λάβουν αποφάσεις και να πλοηγούνται με ασφάλεια. Η χρήση τέτοιων οχημάτων γίνεται ευρέως σε μέρη που είναι επικίνδυνα ή επιβλαβή για τον άνθρωπο όπως τα ορυχεία και πολλά εργοστάσια χρησιμοποιούν φορτηγά χωρίς οδηγό. Ένα σημαντικό περιβαλλοντικό πρόβλημα είναι η ενεργειακή κρίση. Η ΑΙ βοηθάει στη διαχείριση της βιομηχανικής και οικιακής κατανάλωσης. Επίσης, συμβάλει στη διαχείριση των ανανεώσιμων πηγών ενέργειας. Προσφέρει ακόμα βοήθεια στον συντονισμό του επιστημονικού έργου στον αγώνα κατά της ρύπανσης. Στο διαδίκτυο πλέον οι χρήστες με τη βοήθεια της ΑΙ αποκτούν νέα εργαλεία στις αναζητήσεις τους και στην ενημέρωσή τους. Στον εκπαιδευτικό τομέα η ΑΙ έχει σημαντικό αντίκτυπο με τη δημιουργία εξατομικευμένων συστημάτων μάθησης. Οι αλγόριθμοι που βασίζονται στην ΑΙ μπορούν να διευκολύνουν τους μαθητές να κατανοήσουν ένα θέμα καθώς αναγνωρίζουν τα μοτίβα στο διδακτικό υλικό.(Newsit. (2023, 12 Αυγούστου), Greeco. Greeco Team (2023, 1 Δεκεμβρίου))

4.5 Επιπτώσεις της τεχνητής νοημοσύνης.

Όπως είναι φανερό η κοινωνία έχει επηρεαστεί σημαντικά από τη χρήση της ΑΙ. Η επανάσταση που έχει επέλθει από τη χρήση της ΑΙ σε πολλούς τομείς δεν παρέχει μόνο θετικό πρόσημο. Με την πάροδο του χρόνου πραγματοποιείται αλλαγή εργασιακού σκηνικού. Η ανησυχία που έχει προκληθεί από τη συνεχή ανάπτυξη της τεχνητής νοημοσύνης είναι η αντικατάσταση του ανθρωπίνου δυναμικού από τις μηχανές. Επίσης, το κόστος για την εγκατάσταση των συστημάτων ΑΙ είναι αρκετά υψηλό. Όπως και το κόστος που θα πρέπει να καταβάλει η εκάστοτε επιχείρηση για να εκπαιδεύσει ορθά το προσωπικό δυναμικό της για την ασφαλή και σωστή χρήση των μηχανών ΑΙ. Τα συστήματα μπορούν να αντικαταστήσουν την ανθρώπινη εργασία εκτελώντας την πιο γρήγορα και πιο αποτελεσματικά. Αυτό που δε θα μπορέσει να

αντικαταστήσει καμία μηχανή είναι ο παράγοντας συναισθήματος. Οι μηχανές και τα συστήματα ΑΙ λειτουργούν καθαρά με τη λογική, έτσι δεν μπορεί να δημιουργηθεί μια σύνδεση με ένα χρήστη όπως θα έκανε ο άνθρωπος. Επιπλέον, αρκετοί κίνδυνοι ελλοχεύουν με τη συνεχή χρήση της τεχνητής νοημοσύνης. Τα συστήματα ΑΙ για τη λήψη των αποφάσεων χρησιμοποιούν νευρωνικά δίκτυα τα οποία αποτελούνται από πολυπληθείς κόμβους. Αρκετές φορές η ανάλυση που οδήγησε το σύστημα να πάρει μια συγκεκριμένη απόφαση είναι ιδιαίτερα δυσχερής ακόμη και για τον ίδιο του τον δημιουργό. Το πρόβλημα που δημιουργείται έγκειται πως δεν μπορεί να διαπιστωθεί η πορεία που χρησιμοποίησε το σύστημα για να προβεί σε μια απόφαση όταν αυτή αποδεικνύεται λανθασμένη. Αυτό έχει απασχολήσει ιδιαίτερα τομείς, όπως η υγεία ή η άμυνα στους οποίους η διαφάνεια στη λήψη αποφάσεων είναι απαραίτητη. Παράλληλα τα συστήματα μπορούν να λάβουν λάθος αποφάσεις όπως προαναφέρθηκε είτε προσβάλλοντας τα δικαιώματα ατόμων είτε προβαίνουν σε παραπτώματα με το ερώτημα που δημιουργείται να είναι σε ποιόν θα υπάρξει ο καταλογισμός της ευθύνης του παραπτώματος στον δημιουργό ή το μηχανικό σύστημα. Συγχρόνως δεν πρέπει να παραβλέπεται ότι η δημιουργία ενός συστήματος ΑΙ πραγματοποιείται από τον άνθρωπο. Εάν ένα σύστημα έχει τροφοδοτηθεί με δεδομένα προκατειλημμένου περιεχομένου είναι εύλογο οι αποφάσεις που θα παρθούν να εμπεριέχουν τις ίδιες προκαταλήψεις. Η κοινωνική χειραγώγηση που μπορεί να υπάρξει είναι ένα ακόμα ζήτημα. Αυτό μπορεί να συμβεί εξαιτίας των εξατομικευμένων διαφημίσεων και προτάσεων περιεχομένου στους διαδικτυακούς τόπους και τα μέσα που εφαρμόζεται η ΑΙ. (Big Blue Academy. (2023,31 Ιανουαρίου), Big Blue Academy. (2022, 3 Μαΐου)).

Μερικά από αυτά τα παραδείγματα είναι αρχικά στις αμερικάνικες εκλογές του 2016 και εν συνεχεία στο Brexit του 2020. Ομοίως, μείζον είναι το θέμα που ταλανίζει την κοινωνία, η παραβίασή της ιδιωτικότητας. Τα τελευταία δυο χρόνια έχουν παραχθεί μέχρι σήμερα το 90% των ψηφιακών δεδομένων. Τα δεδομένα αυτά μπορεί να είναι ταυτοποίησης, επικοινωνίας, υγείας, πληρωμών, αγορών, προτιμήσεων. Τα οποία με τη χρήση της ΑΙ μπορούν να τα εκμεταλλευτούν η επιχειρήσεις και να επωφεληθούν στον τομέα του marketing. Κάτι τέτοιο όμως δημιουργεί ένα μεγάλο κενό στο πλαίσιο της προστασίας των δεδομένων και της ιδιωτικότητας για τα διαφορετικά είδη που υπάρχουν στο διαδίκτυο. Τέλος, η τεχνολογία μπορεί να είναι είτε καλή είτε κακή ανάλογα με τη χρήση της. Έτσι η ΑΙ μπορεί να χρησιμοποιηθεί για αθέμιτους σκοπούς όπως είναι το hacking. Πολλαπλές κυβερνοεπιθέσεις προκαλούνται από hackers με τη βοήθεια της ΑΙ καθώς προσφέρει ταχύτερη σάρωση των συσκευών ενός δικτύου, δοκιμή εναλλακτικών κωδικών για το κρακάρισμα των συστημάτων

και εφαρμογή social engineering. Συγχρόνως η ΑΙ δημιουργεί αυτοματοποιημένα πλήθος πολλαπλών διευθύνσεων IP για τους hackers ώστε να μην μπορούν να ιχνηλατήσουν στο διαδίκτυο. Οι κίνδυνοι που προαναφέρθηκαν θέτει σημαντικά ερωτήματα στην επιστημονική κοινότητα για τη χρήση της ΑΙ.(Greeco. Greeco Team (2023, 1 Δεκεμβρίου),Big Blue Academy. (2023,31 Ιανουαρίου), Big Blue Academy. (2022, 3 Μαΐου)).

4.6 Ο τομέας της ηθικής στην τεχνητή νοημοσύνη.

Όσο προχωράει η ανάπτυξη της ΑΙ οι εφαρμογές γίνονται ολοένα και πιο προηγμένες. Αυτός είναι ο λόγος που πρέπει να εξεταστούν οι ηθικές επιπτώσεις της ανάπτυξης και της χρήσης τους. Είναι επιτακτική ανάγκη οι προγραμματιστές να διερευνήσουν τις σχετικές ηθικές πτυχές, όπως η προσβασιμότητα των δεδομένων, η προστασία της ιδιωτικής ζωής, η διασφάλιση της δικαιοσύνης στη λήψη αποφάσεων, η μείωση της προκατάληψης και η δυνατότητα των λειτουργιών της ΑΙ. Για τη χρήση της ΑΙ πρέπει οι προγραμματιστές να υιοθετήσουν και να λάβουν υπόψη τους τις ανάγκες όλων των επηρεαζόμενων ενδιαφερομένων μερών, συμπεριλαμβανομένης σε αυτό της προστασίας των ευαίσθητων δεδομένων, όσων παρέχουν και χρησιμοποιούν συστήματα ΑΙ, καθώς και την αξιολόγηση περιβαλλοντικών και ψυχικών επιπτώσεων που προκαλούνται από τις τεχνολογίες που βασίζονται στην ΑΙ. Είναι αναγκαίο να θεσπιστούν και να εφαρμοστούν κανονισμοί, πολιτικές προκειμένου να διασφαλιστεί ότι η χρήση των συστημάτων ΑΙ πραγματοποιείται με υπευθυνότητα, ηθική, διασφαλίζοντας τα συμφέροντα των σημερινών αλλά και των μελλοντικών γενεών. Κατά την ανάπτυξη της ΑΙ θα πρέπει η ηθική να είναι αναπόσπαστο κομμάτι της, θέτοντας ισχυρά θεμέλια για την ηθική εφαρμογή της ΑΙ. (Greeco. Greeco Team (2023, 1 Δεκεμβρίου), Big Blue Academy. (2022, 3 Μαΐου))

5 Τεχνολογίες και Εργαλεία για Ανάλυση Εικόνας και Ανάπτυξη Εφαρμογής.

5.1 Η γλώσσα προγραμματισμού Python με τις απαραίτητες βιβλιοθήκες.



Εικόνα 5.1: Λογότυπο Python

(Python Software Foundation. (n.d.). Python Community Logos.)

Η Python (εικόνα 5.1) είναι μια υψηλού επιπέδου, ερμηνευόμενη γλώσσα προγραμματισμού γνωστή για την απλότητα και την ευκολία χρήσης της. Είναι ευρέως χρησιμοποιούμενη σε διάφορους τομείς όπως η ανάπτυξη ιστοσελίδων, η επιστήμη δεδομένων, η μηχανική μάθηση, η αυτοματοποίηση, και πολλά άλλα.

Βασικές Βιβλιοθήκες της Python:

- NumPy (Numerical Python): Για υπολογισμούς με πίνακες και μαθηματικές λειτουργίες. Το NumPy είναι μια βιβλιοθήκη ανοιχτού κώδικα για τη γλώσσα προγραμματισμού Python, η οποία χρησιμοποιείται ευρέως για υπολογισμούς με πολυδιάστατους πίνακες (arrays) και γενικά αριθμητικούς υπολογισμούς. Παρέχει εργαλεία για γρήγορες και αποτελεσματικές λειτουργίες με πίνακες, καθώς και διάφορες συναρτήσεις για γραμμική άλγεβρα, στατιστικές και μαθηματικές πράξεις. Χρησιμοποιείται κυρίως για την αποδοτική διαχείριση μεγάλων δεδομένων και την πραγματοποίηση αριθμητικών υπολογισμών με υψηλή ακρίβεια. Επιτρέπει πράξεις με πολυδιάστατους πίνακες, γνωστούς ως ndarray, και υποστηρίζει βασικές πράξεις όπως indexing, slicing, reshaping.
- Pandas: Για ανάλυση και διαχείριση δεδομένων. Το Pandas είναι μια δημοφιλής βιβλιοθήκη για ανάλυση δεδομένων και χειρισμό πινάκων που μοιάζουν με τα φύλλα εργασίας του Excel, γνωστά ως DataFrames. Παρέχει δομές δεδομένων και εργαλεία υψηλού επιπέδου για καθαρισμό, φιλτράρισμα, συγκεντρωτικές λειτουργίες και επεξεργασία δεδομένων. Διαχειρίζεται δεδομένα σε μορφή σειρών (Series) και πινάκων (DataFrames), τα οποία μπορούν να είναι πολυδιάστατα και να περιέχουν διαφορετικούς τύπους δεδομένων. Είναι ιδανικό για εργασία με μεγάλες ποσότητες

δεδομένων, εισαγωγή ή εξαγωγή και μετασχηματισμούς δεδομένων από αρχεία CSV, Excel, SQL.

- **Matplotlib & Seaborn:** Για δημιουργία γραφημάτων και οπτικοποιήσεων δεδομένων

Matplotlib: Η Matplotlib είναι μια από τις πιο παλιές και ισχυρές βιβλιοθήκες για την οπτικοποίηση δεδομένων στην Python. Χρησιμοποιείται για τη δημιουργία στατικών, κινούμενων και διαδραστικών γραφημάτων, από απλά γραφήματα γραμμής έως σύνθετες διαγράμματα scatter και γραφήματα 3D. Υποστηρίζει έναν τεράστιο αριθμό τύπων γραφημάτων όπως histogram, bar plots, scatter plots και χρησιμοποιείται ευρέως σε συνδυασμό με το Pandas για ανάλυση δεδομένων.

Seaborn: Το Seaborn είναι μια βιβλιοθήκη οπτικοποίησης βασισμένη στη Matplotlib, η οποία προσφέρει υψηλού επιπέδου διεπαφές για τη δημιουργία κομψών και στατιστικών γραφημάτων. Δίνει έμφαση στις οπτικοποιήσεις δεδομένων που βασίζονται σε διανεμήσεις και στατιστικά συμπεράσματα. Διευκολύνει τη δημιουργία γραφημάτων όπως heatmaps, pairplots και boxplots, επιτρέποντας την εύκολη εξερεύνηση των σχέσεων μεταξύ των μεταβλητών.

- **TensorFlow & PyTorch:** Για ανάπτυξη μοντέλων μηχανικής μάθησης.

TensorFlow: Το TensorFlow είναι μια ανοιχτού κώδικα πλατφόρμα για μηχανική μάθηση (ML). Χρησιμοποιείται για τη δημιουργία και την εκπαίδευση βαθιών νευρωνικών δικτύων (Deep Learning), καθώς και για την ανάπτυξη μοντέλων μηχανικής μάθησης γενικά. Παρέχει ευελιξία και επεκτασιμότητα, με ισχυρή υποστήριξη για υπολογισμούς σε GPU και TPU. Έχει μια υποβιβλιοθήκη, το Keras, που κάνει τη χρήση της πιο απλή για γρήγορη ανάπτυξη μοντέλων.

PyTorch: Το PyTorch είναι μια ανοιχτού κώδικα βιβλιοθήκη, η οποία χρησιμοποιείται κυρίως για εφαρμογές βαθιάς μάθησης και μοντέλα μηχανικής μάθησης που απαιτούν δυναμικά γραφήματα υπολογισμών. Η ευκολία στη χρήση και η δυνατότητα για debugging σε πραγματικό χρόνο το καθιστούν δημοφιλές στους ερευνητές και τους μηχανικούς ML. Παρέχει επίσης υποστήριξη για γρήγορους υπολογισμούς σε GPU και είναι ιδιαίτερα κατάλληλο για πειραματισμούς και έρευνα.

- **OpenCV:** Για επεξεργασία εικόνας και υπολογιστική όραση.

Το OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη ανοιχτού κώδικα που παρέχει εργαλεία για επεξεργασία εικόνας και υπολογιστική όραση

(computer vision). Χρησιμοποιείται ευρέως σε εφαρμογές όπως αναγνώριση προσώπου, επεξεργασία βίντεο, ανάλυση εικόνας, ανίχνευση αντικειμένων, και άλλες παρόμοιες εφαρμογές. Διαθέτει συναρτήσεις για την επεξεργασία και ανάλυση εικόνας, συμπεριλαμβανομένων φίλτρων, μετασχηματισμών, ανίχνευσης ακμών, και πολλά άλλα. Μπορεί να χρησιμοποιηθεί τόσο για απλές εφαρμογές επεξεργασίας εικόνας όσο και για περίπλοκες εφαρμογές όπως η αναγνώριση αντικειμένων και η επεξεργασία σε πραγματικό χρόνο. (Python Software Foundation. (n.d.), TutorialsPoint. (n.d.)Real Python. (n.d.), Van der Plas, J. (n.d.), Python Software Foundation. (n.d.), Numpy Developers. (n.d.), Pandas Developers. (n.d.), McKinney, W. (2022, January 23), Waskom, M. (2020), Matplotlib. (n.d.), TensorFlow. (n.d.), PyTorch. PyTorch Contributors. (n.d.), OpenCV Team(2019. 6 March))

5.2 YOLOv8 της Ultralytics.



Εικόνα 5.2: Λογότυπο της ultralytics YOLOv8

(YOLOv8. (n.d.))

YOLO (You Only Look Once) (εικόνα 5.2) είναι μια οικογένεια αλγορίθμων για ανίχνευση αντικειμένων σε εικόνες και βίντεο σε πραγματικό χρόνο. Η έκδοση YOLOv8 της Ultralytics είναι η πιο πρόσφατη βελτιστοποίηση που προσφέρει υψηλή ακρίβεια και ταχύτητα.

Δομή και Αρχιτεκτονική του YOLOv8 Segmentation:

Το YOLOv8 βασίζεται σε μια εξελιγμένη αρχιτεκτονική, η οποία συνδυάζει την ευελιξία του YOLO με τη δυνατότητα ακριβούς πρόβλεψης masks (μάσκες) για κάθε αντικείμενο που ανιχνεύεται σε μια εικόνα. Παρακάτω περιγράφεται η δομή του YOLOv8 με segmentation.

Neck (Δίκτυο Διασύνδεσης):

Το neck του YOLOv8 χρησιμοποιεί επίπεδα για την ενίσχυση και τη συνένωση των χαρακτηριστικών από διαφορετικές κλίμακες. Χρησιμοποιεί συνήθως αρχιτεκτονικές όπως:

- FPN (Feature Pyramid Network): Συνδυάζει χαρακτηριστικά από πολλαπλά επίπεδα ανάλυσης.
- PANet (Path Aggregation Network): Ενισχύει την πληροφορία από διάφορα επίπεδα του δικτύου για καλύτερη ανίχνευση μικρών και μεγάλων αντικειμένων.

Head (Δίκτυο Κεφαλής):

Το head είναι το τμήμα του δικτύου που προβλέπει τις τελικές εξόδους, δηλαδή τα bounding boxes, τις κατηγορίες, και τις μάσκες segmentation για τα ανιχνευόμενα αντικείμενα. Στο YOLOv8 για segmentation, η διαδικασία είναι διπλή:

- Bounding Box Prediction: Παράγει ένα bounding box για κάθε αντικείμενο στην εικόνα.
- Mask Prediction: Παράγει μια δυαδική μάσκα για κάθε αντικείμενο, η οποία δείχνει ποια pixels ανήκουν στο αντικείμενο.

Στην έξοδο, το μοντέλο επιστρέφει τόσο τα bounding boxes όσο και τις μάσκες segmentation. Αυτή η συνδυασμένη πρόβλεψη καθιστά το YOLOv8 κατάλληλο για εφαρμογές όπου απαιτείται ακριβής οριοθέτηση των αντικειμένων σε εικόνες.

Loss Function (Συνάρτηση Απώλειας):

Η συνάρτηση απώλειας στο YOLOv8 είναι ένας συνδυασμός διαφόρων παραγόντων που επιτρέπουν στο μοντέλο να εκπαιδευτεί σωστά τόσο για την ανίχνευση όσο και για την τμηματοποίηση.

Περιλαμβάνει:

- Box Loss: Απώλεια για την ακρίβεια του bounding box.
- Class Loss: Απώλεια για την σωστή ταξινόμηση του αντικειμένου.
- Mask Loss: Απώλεια για την ακρίβεια της μάσκας τμηματοποίησης.

Η συνάρτηση απώλειας είναι βελτιστοποιημένη ώστε να ισορροπεί την ανίχνευση και την τμηματοποίηση, επιτρέποντας στο μοντέλο να μάθει και τα δύο με ακρίβεια.

Χαρακτηριστικά YOLOv8:

- Ταχύτητα: Ικανή για χρήση σε εφαρμογές που απαιτούν real-time επεξεργασία.

- Ακρίβεια: Βελτιωμένη απόδοση στην ανίχνευση μικρών και πυκνών αντικειμένων.
- Ευελιξία: Υποστηρίζει διάφορα πρότυπα δεδομένων και μπορεί να προσαρμοστεί εύκολα σε διάφορες εφαρμογές.

Χρήσεις:

- Ασφάλεια και παρακολούθηση.
- Αυτοκινούμενα οχήματα.
- Βιομηχανική αυτοματοποίηση.
- Ιατρική απεικόνιση.

(Ultralytics. (2015),YOLOv8. (n.d.),Dev Kit (2024, January 28),Juan R. Terven Instituto Politecnico Nacional CICATA-Qro &Diana M. Cordova-Esparza Universidad Autónoma de Querétaro Facultad de Informática (2024, 29 February), MMYOLO. (n.d.), Muhammad Yaseen Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan (2024, September 5), Ultralytics. (n.d.), YOLOv8. (n.d.), YOLOv8. Jane Torres (2024, March 19), YOLOv8. Jane Torres (2024, March 22))

5.3 MJPEG Stream.

Το MJPEG (Motion JPEG) είναι ένας κωδικοποιητής βίντεο στον οποίο κάθε καρέ ενός βίντεο συμπιέζεται ως μια ξεχωριστή εικόνα JPEG. Αυτό σημαίνει ότι σε αντίθεση με άλλες τεχνολογίες συμπίεσης όπως το H.264, το MJPEG δεν χρησιμοποιεί διαφορική κωδικοποίηση μεταξύ καρέ (interframe compression), γεγονός που εξασφαλίζει σταθερή ποιότητα για κάθε καρέ. Η τεχνολογία αυτή είναι απλή και ευρέως υποστηριζόμενη από πολλές κάμερες και συστήματα streaming, αν και καταναλώνει περισσότερη αποθήκευση και εύρος ζώνης σε σχέση με νεότερα πρότυπα συμπίεσης.

Χαρακτηριστικά του MJPEG

- Κάθε καρέ ως JPEG εικόνα: Κάθε βίντεο αποτελείται από διαδοχικές JPEG εικόνες, κάτι που σημαίνει σταθερή ποιότητα εικόνας για κάθε καρέ.
- Ευκολία στην υλοποίηση: Το MJPEG βασίζεται σε ένα ώριμο πρότυπο συμπίεσης εικόνων (JPEG), το οποίο είναι καλά υποστηριζόμενο από βιβλιοθήκες λογισμικού και εύκολο στη διαχείριση.

- Χαμηλή πολυπλοκότητα: Σε σύγκριση με άλλες μορφές συμπίεσης, το MJPEG απαιτεί λιγότερη επεξεργαστική ισχύ και είναι πιο ανεκτικό σε απώλειες καρτέ, ειδικά σε συνθήκες χαμηλού εύρους ζώνης.
- Υψηλή κατανάλωση εύρους ζώνης: Επειδή κάθε καρτέ είναι μια πλήρης εικόνα JPEG, απαιτεί περισσότερο χώρο αποθήκευσης και εύρος ζώνης, ειδικά σε υψηλότερες αναλύσεις.
- Συμβατότητα με παλαιότερα συστήματα: Το MJPEG μπορεί να χρησιμοποιηθεί σε παλαιότερα συστήματα, και πολλές κάμερες ασφαλείας ή streaming συστήματα το υποστηρίζουν ακόμα.

Χρήσεις του MJPEG.

- Κάμερες ασφαλείας: Το MJPEG χρησιμοποιείται ευρέως σε κάμερες παρακολούθησης, καθώς προσφέρει σταθερή ποιότητα εικόνας χωρίς συμπίεση που θα επηρέαζε την αξιοπιστία των δεδομένων.
- Ζωντανή ροή βίντεο (Live Streaming): Χρησιμοποιείται για streaming μέσω HTTP με χαμηλή καθυστέρηση, ιδανικό για εφαρμογές όπως κάμερες παρακολούθησης ή κάμερες σε πραγματικό χρόνο (π.χ., Raspberry Pi).
- Παλαιότερες συσκευές: Λόγω της απλότητας και της χαμηλής επεξεργαστικής ισχύος που απαιτεί, το MJPEG εξακολουθεί να είναι χρήσιμο σε συστήματα και συσκευές που δεν υποστηρίζουν πιο προηγμένα πρότυπα συμπίεσης.
- Ψηφιακές κάμερες: Πολλές παλαιότερες ψηφιακές κάμερες χρησιμοποιούσαν MJPEG για την εγγραφή βίντεο, καθώς ήταν εύκολο να υλοποιηθεί με τα υπάρχοντα JPEG εξαρτήματα τους. (Movani. Gayane Sargsyan (2019, 5 June), Wikipedia. (n.d.). Motion JPEG).

5.4 Εφαρμογή Moonlight Stream.



Εικόνα 5.3: Λογότυπο Moonlight Stream

Το Moonlight Stream (εικόνα 5.3) είναι μια ανοιχτού κώδικα εφαρμογή που επιτρέπει στους χρήστες να κάνουν streaming των PC παιχνιδιών τους από έναν υπολογιστή με συμβατότητα NVIDIA GameStream σε διάφορες συσκευές. Πρόκειται για έναν πελάτη (client) που αξιοποιεί το GameStream πρωτόκολλο της NVIDIA, προσφέροντας τη δυνατότητα να παίζετε τα παιχνίδια σας σε smartphones, tablets, έξυπνες τηλεοράσεις, ή άλλες πλατφόρμες ανεξαρτήτως τοποθεσίας, είτε βρίσκεστε σε άλλο δωμάτιο είτε μίλια μακριά.

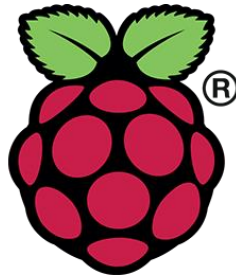
Χαρακτηριστικά του Moonlight Stream:

- Streaming σε 4K με HDR: Υποστηρίζει ροή έως 4K ανάλυση με HDR, αν ο πελάτης σας έχει τέτοια δυνατότητα.
- 120 FPS: Επιτρέπει streaming έως 120 καρέ ανά δευτερόλεπτο, εξασφαλίζοντας ομαλή εμπειρία.
- Υποστήριξη Πολλαπλών Συσκευών: Διαθέσιμο για Windows, macOS, Linux, Android, iOS, Apple TV, PS Vita, Nintendo Switch, και πολλές άλλες συσκευές.
- Ανοιχτός Κώδικας: Το Moonlight είναι πλήρως ανοιχτού κώδικα και διατίθεται δωρεάν χωρίς διαφημίσεις ή συνδρομές.
- Δημιουργία Προσωπικού Cloud Gaming Server: Μπορεί να φιλοξενήσει ένα gaming server εγκαθιστώντας το εργαλείο Moonlight Internet Hosting Tool στον gaming υπολογιστή σας.

Χρήσεις του Moonlight Stream:

- Απομακρυσμένο Gaming: Τα PC παιχνίδια μπορεί να παιχτούν από οποιοδήποτε σημείο του κόσμου σε υποστηριζόμενες συσκευές.
 - Προσωπικός Cloud Server: Δημιουργία cloud gaming περιβάλλοντος χωρίς συνδρομές σε τρίτους παρόχους.
 - Ευρεία Πλατφορμική Υποστήριξη: Κατάλληλο για μια τεράστια γκάμα συσκευών, από smartphones μέχρι κονσόλες και smart TVs, προσφέροντας ευελιξία στη χρήση.
- (AirGPU. (n.d.), Moonlight. (n.d.), Moonlight. (n.d.), Parker, J. (2023, July 21)

5.5 Raspberry Pi 4.



Εικόνα 5.4: Λογότυπο Raspberry Pi

(GitHub. (n.d.). iiiiyrpk/rpi-icon)

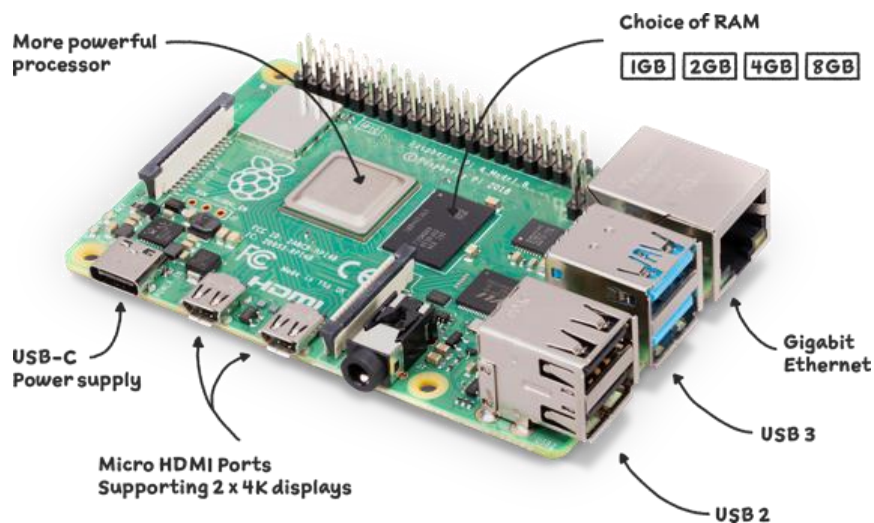
Το Raspberry Pi 4 (εικόνα 5.4) είναι ένας μικρού μεγέθους και χαμηλού κόστους υπολογιστής που προσφέρει εξαιρετικές δυνατότητες επεξεργασίας και συνδεσιμότητας. Σχεδιασμένο για εκπαιδευτικούς, προγραμματιστές και χομπίστες, το Pi 4 μπορεί να χρησιμοποιηθεί για διάφορα έργα, από απλές εφαρμογές μέχρι πιο απαιτητικές, όπως η κατασκευή servers και οικιακών υπολογιστών. Με βελτιωμένη απόδοση σε σχέση με τους προκατόχους του, παρέχει ισχυρή CPU και GPU, υποστηρίζει διπλές οθόνες 4K και προσφέρει γρήγορη συνδεσιμότητα μέσω USB 3.0 και Gigabit Ethernet.

Χαρακτηριστικά του Raspberry Pi 4 (εικόνα 5.5):

- Επεξεργαστής: Quad-core ARM Cortex-A72 στα 1.5 GHz, ο οποίος προσφέρει σημαντική βελτίωση στην επεξεργαστική ισχύ σε σύγκριση με προηγούμενα μοντέλα.
- Μνήμη RAM: Διαθέσιμες επιλογές για 1GB, 2GB, 4GB και 8GB LPDDR4 RAM, κατάλληλες για multitasking και απαιτητικές εφαρμογές.
- Συνδεσιμότητα: Gigabit Ethernet, δύο θύρες USB 3.0 για υψηλές ταχύτητες μεταφοράς δεδομένων, Bluetooth 5.0, και Wi-Fi 802.11ac.
- Υποστήριξη Διπλών Οθονών 4K: Δύο θύρες micro HDMI που μπορούν να οδηγήσουν δύο οθόνες 4K ταυτόχρονα.
- Επεκτασιμότητα: Υποστήριξη για GPIO, USB 3.0, και PCIe, καθιστώντας το Pi 4 ιδανικό για ενσωματωμένες εφαρμογές και αυτοματισμούς.

Χρήσεις του Raspberry Pi 4:

- Οικιακός Υπολογιστής: Με τη δυνατότητα να λειτουργήσει ως desktop PC, το Raspberry Pi 4 μπορεί να εκτελέσει εργασίες όπως περιήγηση στο web, επεξεργασία κειμένων και ακόμα και αναπαραγωγή βίντεο.
- HTPC (Home Theater PC): Με ισχυρή απόδοση και χαμηλή κατανάλωση ενέργειας, μπορεί να χρησιμοποιηθεί για αναπαραγωγή πολυμέσων, streaming και εφαρμογές όπως το Kodi για οικιακή ψυχαγωγία.
- Smart Home Hub: Χρησιμοποιείται για τη δημιουργία ενός οικονομικού και ισχυρού κέντρου έξυπνου σπιτιού, με υποστήριξη για πλατφόρμες όπως το Home Assistant και το OpenHAB.
- Retro Gaming: Ιδανικό για αναπαραγωγή παλαιών παιχνιδιών μέσω εξειδικευμένων λειτουργικών συστημάτων όπως το RetroPie και το Lakka.
- NAS (Network Attached Storage): Λόγω του μικρού μεγέθους και της χαμηλής κατανάλωσης, είναι κατάλληλο για δημιουργία δικτύων αποθήκευσης στο σπίτι ή στο γραφείο. (Electromaker. Moe Long (2019, 10 September), PiCockpit. Adam (2023, 8 September),Raspberry Pi. (n.d.), Raspberry Pi Foundation. (n.d.))



Εικόνα 5.5: Raspberry-pi-4 specs

(Raspberry Pi. (n.d.). Raspberry Pi 4 Model B)

5.6 Εφαρμογή LabelMe.



Εικόνα 5.6: Λογότυπο LabelMe

(LabelMe. (n.d.))

Το LabelMe (εικόνα 5.6) είναι ένα ανοιχτού κώδικα εργαλείο σχολιασμού εικόνων (image annotation) που χρησιμοποιείται κυρίως για την εκπαίδευση μοντέλων μηχανικής μάθησης σε προβλήματα υπολογιστικής όρασης. Δημιουργήθηκε από το MIT και επιτρέπει στους χρήστες να προσθέτουν ετικέτες και να οριοθετούν αντικείμενα μέσα σε εικόνες, προσφέροντας ένα σύνολο εργαλείων όπως πολύγωνα, ορθογώνια, κύκλους και γραμμές.

Χαρακτηριστικά του LabelMe:

- Διάφορα εργαλεία σχολιασμού: Το LabelMe προσφέρει πολλαπλές επιλογές annotation όπως πολύγωνα, ορθογώνια, γραμμές και κύκλους, καθιστώντας το ιδανικό για πολλές χρήσεις, όπως ανίχνευση αντικειμένων και τμηματοποίηση εικόνων.
- Αποθήκευση σε JSON: Τα annotation αποθηκεύονται σε μορφή JSON, κάτι που διευκολύνει τη χρήση των δεδομένων σε μοντέλα μηχανικής μάθησης.
- Εγκατάσταση: Το LabelMe μπορεί να εγκατασταθεί σε Windows, macOS και Linux, με απλή εγκατάσταση μέσω Python ή Anaconda.
- Εξαγωγή δεδομένων: Υποστηρίζει εξαγωγή σχολιασμένων δεδομένων σε μορφές όπως Pascal VOC και COCO, που είναι δημοφιλείς στη μηχανική μάθηση.
- Ανοιχτού κώδικα: Είναι δωρεάν και ανοιχτού κώδικα, επιτρέποντας την κοινότητα να συνεισφέρει στη βελτίωση του εργαλείου.

Χρήσεις του LabelMe:

- Εκπαίδευση μοντέλων μηχανικής μάθησης: Χρησιμοποιείται ευρέως για την εκπαίδευση μοντέλων υπολογιστικής όρασης, επιτρέποντας το annotation εικόνων για αναγνώριση αντικειμένων και τμηματοποίηση.
- Δημιουργία data base εικόνων: Χρησιμοποιείται από ερευνητές και προγραμματιστές για τη δημιουργία μεγάλων βάσεων δεδομένων σχολιασμένων εικόνων.
- Ανίχνευση αντικειμένων και τμηματοποίηση: Χρησιμοποιείται για τον εντοπισμό και την οριοθέτηση αντικειμένων σε εικόνες μέσω εργαλείων όπως τα bounding boxes και οι πολύγωνοι σχολιασμοί. (Innovatiana. Nanobaly (2024, 3 September), LabelMe. (n.d.), V7 Labs. Hmrishav Bandyopadhyay (2022, 14 October)

5.7 Microsoft Visual Studio.



Εικόνα 5.7: Λογότυπο Microsoft Visual Studio

(Wikipedia. (n.d.). File: Visual Studio Icon)

Το Microsoft Visual Studio (εικόνα 5.7) είναι ένα πλήρες και ισχυρό περιβάλλον ανάπτυξης εφαρμογών Integrated Development Environment (IDE), το οποίο υποστηρίζει περισσότερες από 30 γλώσσες προγραμματισμού, όπως C#, C++, Python, JavaScript. Προσφέρει εργαλεία για όλες τις φάσεις του κύκλου ανάπτυξης λογισμικού, από τον προγραμματισμό, το debugging μέχρι τη δοκιμή και την ανάπτυξη. Το Visual Studio είναι εξαιρετικά επεκτάσιμο, επιτρέποντας στους χρήστες να χρησιμοποιούν πρόσθετα (plugins) και επεκτάσεις για να καλύψουν συγκεκριμένες ανάγκες.

Χαρακτηριστικά του Visual Studio:

- Επεξεργασία Κώδικα με IntelliSense: Παρέχει εργαλεία αυτόματης συμπλήρωσης κώδικα και κωδικοποίηση (IntelliSense) για να επιταχύνει τη συγγραφή κώδικα και να βοηθήσει στη διόρθωση σφαλμάτων κατά τη σύνταξη.

- Εντοπισμός Σφαλμάτων: Διαθέτει ολοκληρωμένα εργαλεία για (debugging) που επιτρέπουν στους χρήστες να εντοπίζουν προβλήματα σε επίπεδο κώδικα και μηχανής. Μπορεί βήμα-βήμα μέσω του κώδικα και να εξεταστούν οι μεταβλητές και οι τιμές.
- Ενσωματωμένα Εργαλεία Διαχείρισης Εκδόσεων: Περιλαμβάνει δυνατότητες για την ενσωμάτωση με Git και άλλα συστήματα διαχείρισης εκδόσεων, επιτρέποντας εύκολη παρακολούθηση αλλαγών στον κώδικα.
- Συνεργασία σε πραγματικό χρόνο: Το εργαλείο Live Share επιτρέπει σε πολλούς προγραμματιστές να συνεργάζονται σε πραγματικό χρόνο πάνω στον ίδιο κώδικα.
- AI-Υποβοηθούμενη Ανάπτυξη: Χρησιμοποιεί εργαλεία όπως το GitHub Copilot και το IntelliCode για να βοηθήσει τους προγραμματιστές να γράφουν κώδικα πιο γρήγορα και με μεγαλύτερη ακρίβεια.
- Δημιουργία και Διαχείριση Βάσεων Δεδομένων: Παρέχει εργαλεία για τη διαχείριση βάσεων δεδομένων, την ανάπτυξη σε SQL Server και Azure SQL, καθώς και την ανάπτυξη σύνθετων εφαρμογών data base.

Χρήσεις του Visual Studio:

- Ανάπτυξη Εφαρμογών Ιστού και Κινητών Συσκευών: Το Visual Studio υποστηρίζει τη δημιουργία σύγχρονων εφαρμογών ιστού και εφαρμογών για κινητές συσκευές που λειτουργούν σε Windows, Android, και iOS.
- Ανάπτυξη Παιχνιδιών: Παρέχει εργαλεία για την ανάπτυξη παιχνιδιών με υποστήριξη για scripting και σχεδιασμό γραφικών, καθώς και για μηχανές παιχνιδιών όπως το Unity.
- Ανάπτυξη Cloud Εφαρμογών: Το Visual Studio διευκολύνει την ανάπτυξη και διαχείριση cloud εφαρμογών, κυρίως μέσω της ενσωμάτωσης του με το Azure.
- Ανάπτυξη Γλώσσας Προγραμματισμού και Επεκτάσεων: Επιτρέπει τη δημιουργία επεκτάσεων για την υποστήριξη διαφορετικής γλώσσας προγραμματισμού και την προσθήκη εργαλείων στο IDE. (GeeksforGeeks. anshul_aggarwal(2023, 22 September),HubSpot. Madhu Murali (2023, 30 January), Microsoft. (2024, 19 June), Microsoft. (n.d.), Wikipedia. (n.d.). Visual Studio)

5.8 Εφαρμογή Miniconda.



Εικόνα 5.8: Λογότυπο Anaconda

(Anaconda. (n.d.). Miniconda Documentation.)

Το Miniconda (εικόνα 5.8) είναι μια ελαφριά έκδοση της Anaconda που περιλαμβάνει μόνο το απαραίτητο περιβάλλον για να ξεκινήσει κανείς τη χρήση της γλώσσας προγραμματισμού Python και των σχετικών βιβλιοθηκών. Αποτελείται από τον διαχειριστή πακέτων Conda και τον διερμηνέα της Python. Το Miniconda επιτρέπει την εγκατάσταση μόνο των πακέτων που χρειάζεστε, σε αντίθεση με την πλήρη έκδοση της Anaconda, η οποία περιλαμβάνει πολλές προ εγκατεστημένες βιβλιοθήκες.

Χαρακτηριστικά του Miniconda:

- Ελαφριά Εγκατάσταση: Περιλαμβάνει μόνο τον διαχειριστή πακέτων Conda και τον Python interpreter, προσφέροντας μια ελάχιστη εγκατάσταση.
- Διαχείριση Πακέτων και Εικονικών Περιβαλλόντων: Γίνετε χρήση του Conda για να εγκαθιστά, αναβαθμίζει, να διαχειρίζεται πακέτα, και για την δημιουργία εικονικού περιβάλλοντος.
- Υποστήριξη για Πολλαπλές Πλατφόρμες: Υποστηρίζει Windows, macOS και Linux, δίνοντας τη δυνατότητα στους χρήστες να δουλεύουν σε διάφορα λειτουργικά συστήματα.
- Εύκολη Επέκταση: Μπορείτε να εγκαταστήσετε μόνο τις βιβλιοθήκες που χρειάζεστε, όπως το NumPy, Pandas, SciPy, και άλλες, με ελάχιστο αποθηκευτικό χώρο.
- Συμβατότητα με την Anaconda: Αν και πιο ελαφριά, είναι συμβατό με το οικοσύστημα της Anaconda και μπορεί να εγκαταστήσει τα ίδια πακέτα.

Χρήσεις του Miniconda:

- Δημιουργία Εικονικών Περιβαλλόντων: Οι χρήστες μπορούν να δημιουργούν διαφορετικά περιβάλλοντα για διάφορα projects, με διαφορετικές εκδόσεις Python και πακέτα.
- Προγραμματισμός με Python: Παρέχει έναν απλό και ευέλικτο τρόπο για να γίνει η διαχείριση του διερχόμενου της Python και των πακέτων που απαιτούνται για ανάπτυξη λογισμικού.
- Ανάπτυξη Λογισμικού: Είναι ιδανικό για προγραμματιστές που θέλουν ένα πιο ευέλικτο εργαλείο για διαχείριση εξαρτήσεων των έργων τους.
- Εκπαίδευση και Εκμάθηση: Το Miniconda είναι επίσης χρήσιμο για σπουδαστές και καθηγητές, καθώς παρέχει έναν ελαφρύ τρόπο να εγκαταστήσουν μόνο τα εργαλεία που χρειάζονται για να μάθουν Python ή άλλες γλώσσες που υποστηρίζονται από Conda. (Anaconda. (n.d.), Conda. (n.d.), Data School. (2024, 7 January), DNMTEchs. Auden Lily (2022, 9 March))

5.9 Πλατφόρμα ZeroTier.



Εικόνα 5.9: ZeroTier λογότυπο

(ZeroTier. (2024, 15 April.))

Το ZeroTier (εικόνα 5.9) είναι μια πλατφόρμα εικονικών ιδιωτικών δικτύων (VPN) που επιτρέπει σε συσκευές να συνδέονται μέσω του διαδικτύου με ασφάλεια, σαν να βρίσκονται στο ίδιο τοπικό δίκτυο (LAN). Προσφέρει peer-to-peer επικοινωνία, κάτι που μειώνει την καθυστέρηση, και χρησιμοποιεί 256-bit end-to-end κρυπτογράφηση για την ασφάλεια των δεδομένων. Μπορεί να χρησιμοποιηθεί σε διάφορα λειτουργικά συστήματα, όπως Windows, macOS, Linux, Android και iOS, και επιτρέπει τη σύνδεση από διαφορετικές γεωγραφικές περιοχές. Η ρύθμισή του είναι εύκολη, καθιστώντας το κατάλληλο για επιχειρήσεις, απομακρυσμένους χρήστες, ακόμα και για προσωπική χρήση.

Είναι μια ευέλικτη λύση που συνδυάζει χαρακτηριστικά τόσο από τα VPN όσο και από τα SD-WAN, προσφέροντας δυνατότητες όπως απομακρυσμένη πρόσβαση σε υπολογιστές, διαχείριση εφαρμογών μέσω του intranet, καθώς και υποστήριξη IoT. Παρά την εξάρτησή του από κεντρικούς διακομιστές για τον συντονισμό, η πλατφόρμα είναι αρκετά ασφαλής και δημοφιλής για τις εφαρμογές της.

Χαρακτηριστικά του ZeroTier:

- Κρυπτογράφηση 256-bit end-to-end: Όλες οι συνδέσεις και δεδομένα κρυπτογραφούνται χρησιμοποιώντας το πρωτόκολλο AES-256, παρέχοντας υψηλό επίπεδο ασφάλειας.
- Peer-to-Peer (P2P) Επικοινωνία: Συνδέει απευθείας συσκευές μέσω του διαδικτύου, βελτιώνοντας την ταχύτητα και μειώνοντας την καθυστέρηση, χωρίς τη χρήση κεντρικού διακομιστή για τα δεδομένα.
- Υποστήριξη Πολλαπλών Πλατφορμών: Υποστηρίζει λειτουργικά συστήματα όπως Windows, macOS, Linux, Android, iOS, καθώς και συσκευές IoT.
- Ευκολία Χρήσης: Εύκολη εγκατάσταση και διαχείριση, με δυνατότητα δημιουργίας δικτύων μέσα σε λίγα λεπτά από έναν κεντρικό πίνακα ελέγχου.
- Απεριόριστα Δίκτυα: Προσφέρει δυνατότητα δημιουργίας απεριόριστων εικονικών δικτύων, που μπορούν να περιλαμβάνουν πολλές συσκευές, από μικρά οικιακά δίκτυα μέχρι εταιρικά.
- Αυτόματη Διαχείριση Δικτύων: Το ZeroTier διαχειρίζεται αυτόματα τις συνδέσεις των συσκευών, κάνοντας χρήση τεχνικών όπως STUN και hole punching για σύνδεση συσκευών πίσω από NAT.
- Ενσωματωμένο API: Παρέχει πρόσβαση σε API για αυτοματοποίηση και ενσωμάτωση με άλλες εφαρμογές.

Χρήσεις του ZeroTier:

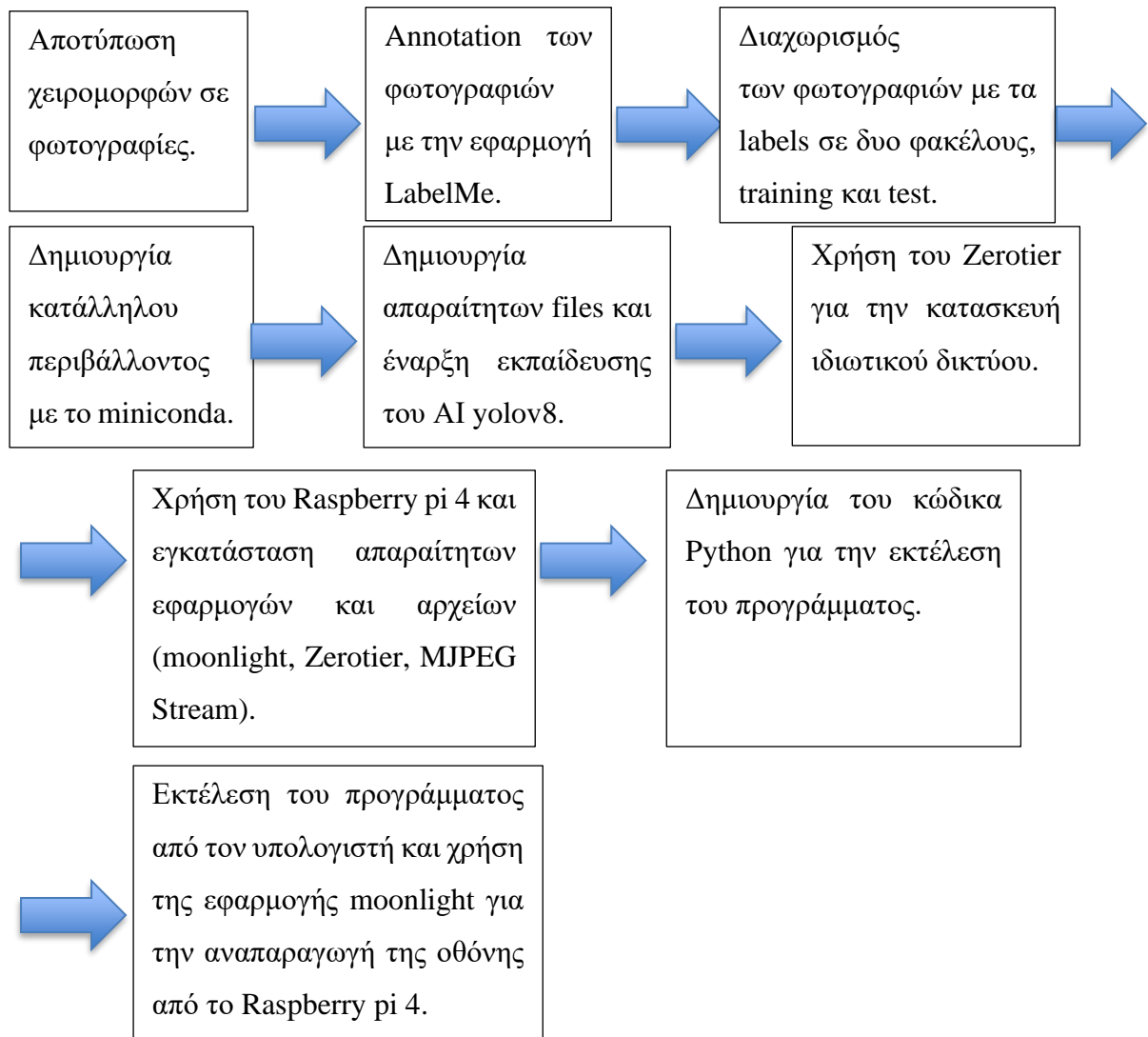
- Απομακρυσμένη Πρόσβαση: Παρέχει ασφαλή πρόσβαση σε υπολογιστές και διακομιστές από οποιαδήποτε τοποθεσία, χωρίς την ανάγκη περίπλοκων ρυθμίσεων VPN.
- Ιδιωτικά Δίκτυα: Δημιουργία ιδιωτικών δικτύων για επιχειρήσεις ή προσωπική χρήση, που επιτρέπουν την ασφαλή επικοινωνία συσκευών και τη διαχείριση εφαρμογών.
- Υποστήριξη IoT: Ιδανικό για δίκτυα συσκευών IoT, καθώς προσφέρει ασφαλή σύνδεση και έλεγχο των συσκευών αυτών απομακρυσμένα.

- Gaming: Δημιουργία ιδιωτικών δικτύων για multiplayer gaming, επιτρέποντας τη σύνδεση με φίλους σαν να βρίσκονται στο ίδιο τοπικό δίκτυο (LAN).
- Απομακρυσμένη Διοίκηση: Χρησιμοποιείται για απομακρυσμένη διαχείριση υποδομών IT, όπως server farms, cloud instances, και βάσεις δεδομένων. (DZone. Tomasz Waraksa (2021, 9 September), TechCult. Henry Quill (2023, 20 October), Wlink Technologies. (n.d.), ZeroTier. Adam Ierymenko (2023, 10 June), ZeroTier. (2024, 15 April).

6 Ανάπτυξη Συστήματος Αναγνώρισης της Ελληνικής Νοηματικής Γλώσσας.

6.1 Γενική αρχιτεκτονική υλοποίηση.

Παρακάτω παρατίθεται περιληπτική αναφορά στην υλοποίηση του συστήματος.



Αρχικά για την δημιουργία του συστήματος πρέπει να πραγματοποιηθεί η αποτύπωση όλων των χειρομορφών σε φωτογραφίες. Δηλαδή θα πρέπει οι χειρομορφές που υποδεικνύουν τα γράμματα και οι χειρομορφές που χαρακτηρίζουν τις λέξεις να απεικονιστούν σε φωτογραφίες. Η διαδικασία επαναλαμβάνετε αρκετές φορές καθώς χρειάζονται πολλές φωτογραφίες για το κάθε ένα γράμμα όπως και για την κάθε μια λέξη ξεχωριστά. Εν συνεχεία εκτελείτε η

επεξεργασία όλων των φωτογραφιών. Τοποθετούνται στην εφαρμογή LabelMe και με τον τρόπο που αναλύεται σε παρακάτω κεφάλαιο τοποθετούνται οι σωστές ετικέτες (labels). Ακόμη, αφού τελειώσει η διαδικασία με την προσθήκη ετικέτας σε κάθε φωτογραφία ξεχωριστά, στον φάκελο περιέχονται τα αρχεία των φωτογραφιών και τα αρχεία των ετικετών που αντιστοιχούν στην κάθε μια φωτογραφία. Γίνεται ο διαχωρισμός τους σε ποσοστό 60% με 65% στον φάκελο train και το υπόλοιπο 40% με 35% αντίστοιχα στον φάκελο test για να υποστούν περαιτέρω επεξεργασία. Επιπλέον για να μπορέσει το AI πρόγραμμα να εκτελέσει την εκπαίδευση του, δημιουργείται κατάλληλο περιβάλλον με το miniconda. Στο περιβάλλον εμπεριέχονται όλες οι βιβλιοθήκες που είναι απαραίτητες για την λειτουργία του AI. Επιπροσθέτως γίνεται η δημιουργία αρχείων όπου είναι απαραίτητα για το πρόγραμμα AI να εκτελέσει σωστά την εκπαίδευση. Η ανάλυση των αρχείων αυτών αντικατοπτρίζεται σε μεταγενέστερο κεφάλαιο. Στην συνέχεια με το Zerotier επιτυγχάνεται η χρήση ιδιωτικού δικτύου όπου είναι απαραίτητη για την μόνιμη σύνδεση με το Raspberry pi 4. Για την χρήση του Raspberry pi 4 είναι απαραίτητη η εγκατάσταση συγκεκριμένων εφαρμογών και αρχείων όπως moonlight, Zerotier, MJPEG Stream, για να χρησιμοποιηθεί ορθά με το τελικό σύστημα. Με την ολοκλήρωση των παραπάνω πραγματοποιείται η σύνταξη του κώδικα rython όπου ενορχηστρώνει τα αποτελέσματα όλων των παραπάνω. Τέλος πραγματοποιείται εκτέλεση του τελικού προγράμματος από τον υπολογιστή και μέσω της εφαρμογής moonlight γίνεται αναπαραγωγή στην οθόνη του Raspberry pi 4.

Με αυτόν τον τρόπο έγινε η δημιουργία του συστήματος. Στα παρακάτω κεφάλαια θα γίνει αναλυτικότερη αναφορά στην υλοποίηση του συστήματος, στις εφαρμογές και τους κώδικες που χρησιμοποιούνται.

6.2 Αποτύπωση του αλφαβήτου και των λέξεων σε φωτογραφίες.

Τα δεδομένα που χρειάστηκαν για να πραγματοποιηθεί η εκπαίδευση του AI είναι 1419 φωτογραφίες που απαθανάτιστηκαν με γράμματα και λέξεις της ελληνικής νοηματικής. Τα γράμματα χρησιμοποιούνται όλα γι' αυτό υπάρχουν φωτογραφίες για το καθένα ξεχωριστά. Οι λέξεις που χρησιμοποιούνται και μπορούν να ανιχνευτούν είναι: αυτοκίνητο, ελικόπτερο, αεροπλάνο, αερόστατο. Παράδειγμα είναι μερικές φωτογραφίες που αποτυπώνουν το γράμμα ή τη λέξη. Πρέπει να επισημανθεί ότι για τις λέξεις είναι απαραίτητο να παρθούν frames φωτογραφιών καθώς δεν είναι μια στατική κίνηση αλλά μια κίνηση στον χώρο. Ωστόσο, αυτό δεν αναιρεί το γεγονός ότι πρέπει να χρησιμοποιηθεί πλήθος φωτογραφιών για το κάθε γράμμα

και την κάθε λέξη για την ύπαρξη καλύτερου training του συστήματος με αποτέλεσμα μεγαλύτερη ευστοχία.

Για το γράμμα Α (εικόνα 6.1)

Για το γράμμα Β (εικόνα 6.2)



Εικόνα 6.1: Χειρομορφή για το γράμμα Α

Εικόνα 6.2: Χειρομορφή για το γράμμα Β

Για τη λέξη αυτοκίνητο (εικόνα 6.3 και εικόνα 6.4)



Εικόνα 6.3: Χειρομορφή για τη λέξη αυτοκίνητο

Εικόνα 6.4: Δεύτερη χειρομορφή για τη λέξη αυτοκίνητο

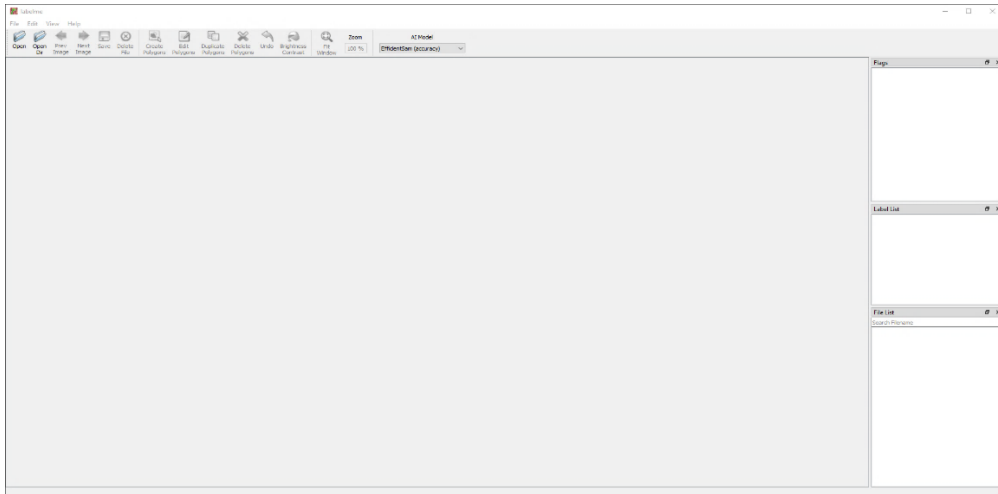
6.3 Annotation των φωτογραφιών με τη χρήση του LabelMe.

Στις φωτογραφίες που χρησιμοποιούνται για το training πρέπει να κατανοήσει το ΑΙ ποιο σημείο πρέπει να αναγνωρίζει και να βγάζει τη σωστή λέξη ή το σωστό γράμμα όταν εκτελείτε το πρόγραμμα. Αυτό είναι εφικτό εξαιτίας του LabelMe. Με την εφαρμογή LabelMe πραγματοποιείται το annotation και δημιουργούνται τα labels για την κάθε φωτογραφία ξεχωριστά. Αρχικά για τη χρήση της εφαρμογής πρέπει στο client του miniconda να πληκτρολογηθεί η εντολή `pip install label me` (εικόνα 6.5) και να γίνει η εγκατάσταση των απαραίτητων πακέτων για τη λειτουργία της εφαρμογής.

```
Anaconda Prompt (miniconda)
(takis) C:\Users\giann\miniconda3>pip install labelme
Collecting labelme
  Using cached labelme-5.5.0.tar.gz (1.4 MB)
  Preparing metadata (setup.py) ... done
Collecting gdcm (from labelme)
  Using cached gdcm-2.0-py3-none-any.whl.metadata (5.8 kB)
Collecting imgviz>=1.7.5 (from labelme)
  Using cached imgviz-1.7.5.tar.gz (7.7 MB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from labelme) (3.8.4)
Collecting natsort>=7.1.0 (from labelme)
  Using cached natsort-8.4.0-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: numpy in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from labelme) (1.26.4)
Requirement already satisfied: Pillow>=2.8 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from labelme) (10.2.0)
Requirement already satisfied: PyYAML in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from labelme) (6.0.1)
Collecting stpyqt>=1.11.2 (from labelme)
  Using cached stpyqt-2.4.1-py3-none-any.whl.metadata (12 kB)
Collecting scikit-image (from labelme)
  Downloading scikit_image-0.24.0-cp312-win_amd64.whl.metadata (14 kB)
Requirement already satisfied: termcolor in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from labelme) (2.4.0)
Collecting PyQT5=5.15.3,!=5.15.4 (from labelme)
  Using cached PyQT5-5.15.11-cp38-abi3-win_amd64.whl.metadata (2.1 kB)
Requirement already satisfied: colorama in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from labelme) (0.4.0)
Requirement already satisfied: coloredlogs in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from onnxruntime==1.16.0, >=1.14.1->labelme) (15.0.1)
Requirement already satisfied: flatbuffers in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from onnxruntime==1.16.0, >=1.14.1->labelme) (24.3.25)
Requirement already satisfied: packaging in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from onnxruntime==1.16.0, >=1.14.1->labelme) (24.1)
Requirement already satisfied: protobuf in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from onnxruntime==1.16.0, >=1.14.1->labelme) (4.25.3)
Requirement already satisfied: sympy in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from onnxruntime==1.16.0, >=1.14.1->labelme) (1.12)
Collecting PyQ5-sip<13, >=12.15 (from PyQT5=5.15.3,!=5.15.4->labelme)
  Downloading PyQ5-sip-12.15.0-cp312-win_amd64.whl.metadata (439 bytes)
Collecting PyQ5-QT5=10.0, >=5.15.2 (from PyQT5=5.15.3,!=5.15.4->labelme)
  Using cached PyQ5-QT5-5.15.2-py3-none-win_amd64.whl.metadata (552 bytes)
Collecting beautifulsoup4 (from gdcm->labelme)
  Downloading beautifulsoup4-4.12.3-py3-none-any.whl.metadata (3.8 kB)
Requirement already satisfied: filelock in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from gdcm->labelme) (3.13.4)
Requirement already satisfied: requests[socks] in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from gdcm->labelme) (2.32.3)
Requirement already satisfied: tqdm in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from gdcm->labelme) (4.66.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from matplotlib->labelme) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from matplotlib->labelme) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from matplotlib->labelme) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from matplotlib->labelme) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from matplotlib->labelme) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from matplotlib->labelme) (2.9.0.post0)
Requirement already satisfied: scipy>=1.9 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from scikit-image->labelme) (1.13.0)
Requirement already satisfied: networkx>=2.8 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from scikit-image->labelme) (3.3)
Collecting imageio>=2.33 (from scikit-image->labelme)
  Downloading imageio-2.36.0-py3-none-any.whl.metadata (5.2 kB)
Collecting tifffile>=2022.8.12 (from scikit-image->labelme)
  Downloading tifffile-2024.9.20-py3-none-any.whl.metadata (32 kB)
Collecting lazy_loader>=0.4 (from scikit-image->labelme)
  Using cached lazy_loader-0.4-py3-none-any.whl.metadata (7.6 kB)
Requirement already satisfied: six>=1.5 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from python-dateutil==2.7->matplotlib->labelme) (1.16.0)
Collecting soupsieve>=1.2 (from beautifulsoup4->gdcm->labelme)
  Downloading soupsieve-2.6-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: humanfriendly>=9.1 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from coloredlogs->onnxruntime==1.16.0, >=1.14.1->labelme) (10.0)
Requirement already satisfied: charset-normalizer<4, >=2 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from requests[socks]->gdcm->labelme) (3.3.2)
Requirement already satisfied: idna<=2, >=5 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from requests[socks]->gdcm->labelme) (3.7)
Requirement already satisfied: urllib3<3, >=1.21.1 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from requests[socks]->gdcm->labelme) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from requests[socks]->gdcm->labelme) (2024.7.4)
Collecting PySocks=1.5.7,!=1.5.6 (from requests[socks]->gdcm->labelme)
  Using cached PySocks-1.7.1-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: mpmath>=0.19 in c:\users\giann\miniconda3\envs\takis\lib\site-packages (from sympy->onnxruntime==1.16.0, >=1.14.1->labelme) (1.3.0)
```

Εικόνα 6.5: Εγκατάσταση LabelMe

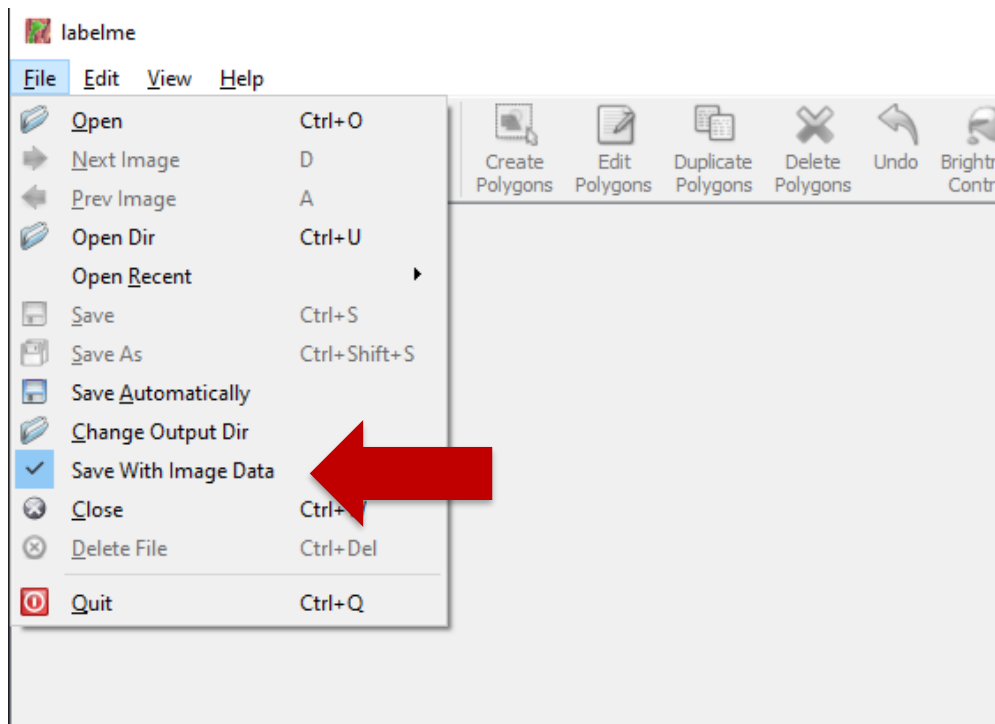
Στη συνέχεια με την εντολή `labelme` ανοίγει το πρόγραμμα και εμφανίζεται το display της εικόνας (εικόνα 6.6)



Εικόνα 6.6: Εφαρμογή LabelMe

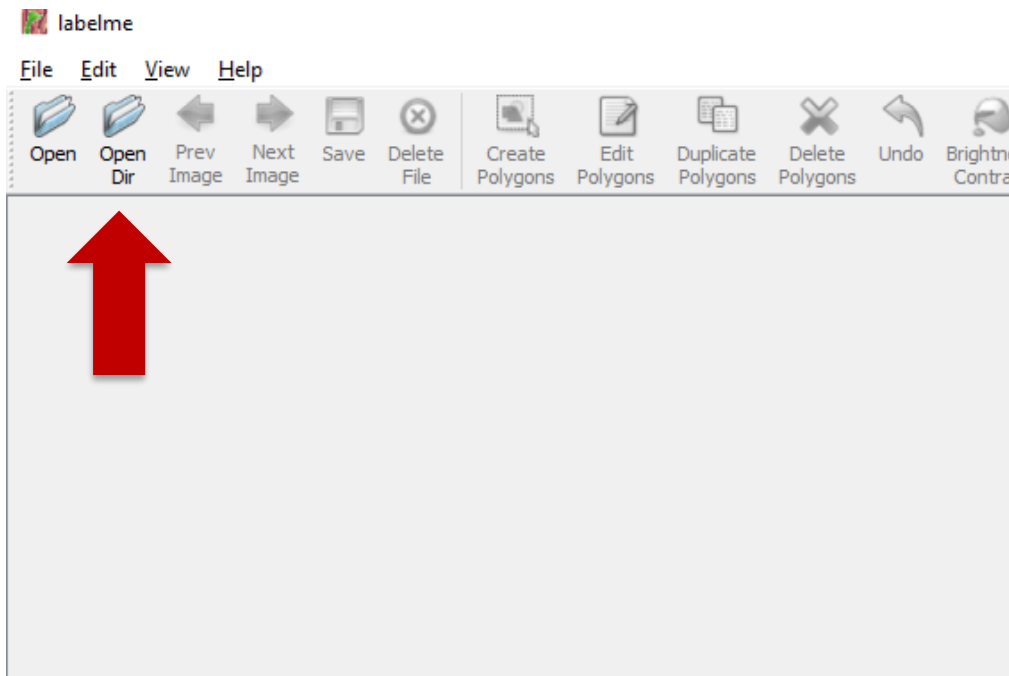
Με την εφαρμογή έτοιμη είναι αναγκαίο να ενεργοποιηθούν οι παρακάτω λειτουργίες.

- 1) Στο file πρέπει να ενεργοποιηθεί το save with data image (εικόνα 6.7). Η επιλογή save with data image στο LabelMe αποθηκεύει τα δεδομένα ετικετών (labels) μαζί με την εικόνα σε ένα ενιαίο αρχείο JSON. Είναι απαραίτητο να είναι ενεργοποιημένο καθώς μεταγενέστερα που θα πραγματοποιηθεί η επεξεργασία των δεδομένων θα υπάρξουν τα σωστά αποτελέσματα για την υλοποίηση της εργασίας.



Εικόνα 6.7: Ενεργοποίηση save with data image

2) Για να χρησιμοποιηθούν οι φωτογραφίες ώστε να επεξεργαστούν και να γίνουν τα annotations και τα labels των γραμμάτων ή των λέξεων, πρέπει να επιλεγεί ο φάκελος που τις περιέχει. Στη γραμμή εργαλείων του LabelMe θα επιλεγεί το open dir (εικόνα 6.8) και θα ανοίξει ο φάκελος με τις φωτογραφίες.



Εικόνα 6.8: Εύρεση φακέλου φωτογραφιών

3) Το annotation που γίνεται στις εικόνες ονομάζεται segmentation. Υπάρχουν δυο είδη annotation το segmentation και το detection. Θα αναλυθεί το καθένα ξεχωριστά για να γίνει κατανοητή η διάφορα τους. Detection (Ανίχνευση): Η ανίχνευση επικεντρώνεται στον εντοπισμό αντικειμένων σε μια εικόνα και την αναγνώριση της θέσης τους μέσω ορθογώνιων πλαισίων (*bounding boxes*). Ο στόχος του detection είναι να εντοπιστεί η παρουσία ενός αντικειμένου σε μια σκηνή και να προσδιοριστεί το σημείο του. Τα βασικά χαρακτηριστικά της ανίχνευσης είναι:

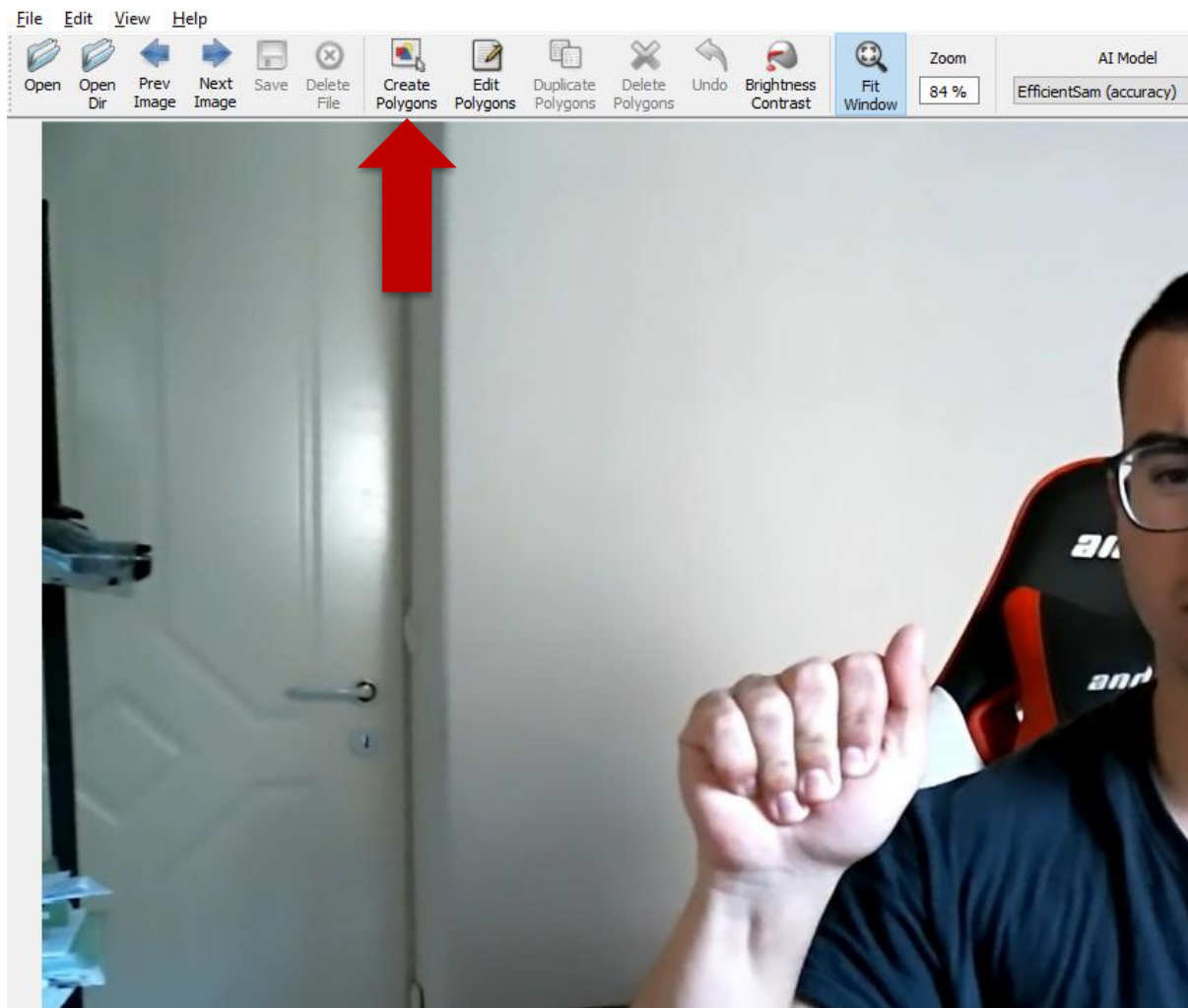
- Αναγνώριση του τύπου των αντικειμένων, π.χ., "αυτοκίνητο," "άνθρωπος," "ζώο."
- Σχεδιασμός περιγράμματος γύρω από το αντικείμενο χωρίς ακριβή καθορισμό του σχήματός του.
- Περιορισμένη ακρίβεια στον προσδιορισμό των ακριβών ορίων του αντικειμένου, καθώς χρησιμοποιεί απλά πλαίσια γύρω από αυτά.

Segmentation (Τμηματοποίηση): Η τμηματοποίηση, σε αντίθεση, επικεντρώνεται στον διαχωρισμό των αντικειμένων από το φόντο και τον εντοπισμό των ακριβών τους ορίων σε

επίπεδο pixel. Έτσι, το segmentation παρέχει μια πολύ λεπτομερή αναπαράσταση των αντικειμένων. Διακρίνεται σε:

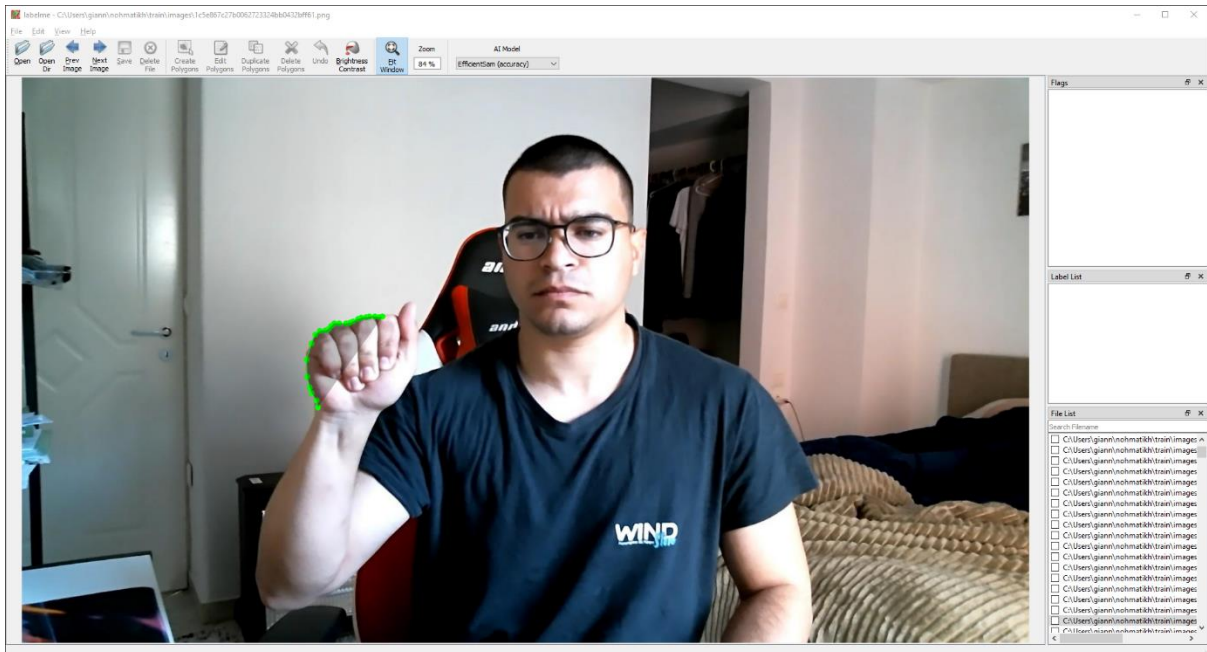
- **Semantic Segmentation:** Κατηγοριοποιεί κάθε pixel της εικόνας σε κλάσεις (π.χ., το γρασίδι, ο ουρανός, τα αυτοκίνητα), χωρίς να κάνει διάκριση ανάμεσα στα ξεχωριστά αντικείμενα της ίδιας κατηγορίας.
- **Instance Segmentation:** Συνδυάζει την ανίχνευση και την τμηματοποίηση, προσδιορίζοντας κάθε ξεχωριστό αντικείμενο στην εικόνα και σχεδιάζοντας το ακριβές του περίγραμμα. Χρησιμοποιείται για πιο εξειδικευμένες εφαρμογές, όπως η αυτόνομη οδήγηση και η ιατρική διάγνωση.

Το εργαλείο για να πραγματοποιηθεί το annotation είναι το create polygon από τη γραμμή εντολών (εικόνα 6.9).



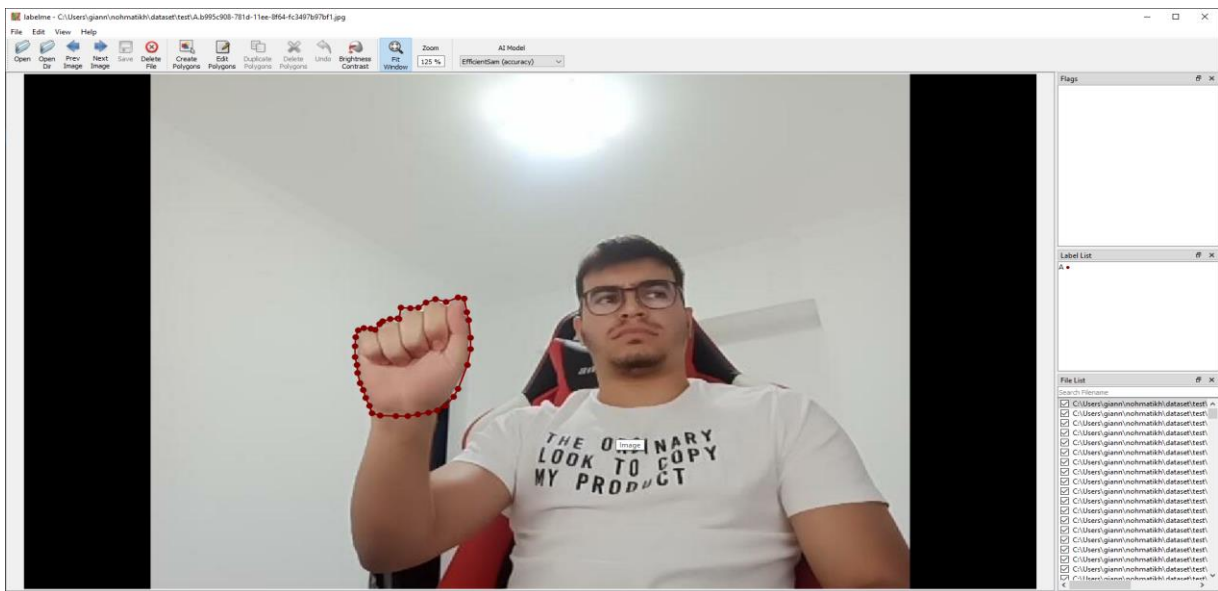
Εικόνα 6.9: Επιλογή create polygon

4) Για να γίνει το segmentation πρέπει περιμετρικά από τη χειρομορφή να δημιουργηθεί ένα πολύγωνο. Όπως φαίνεται και στην εικόνα 6.10



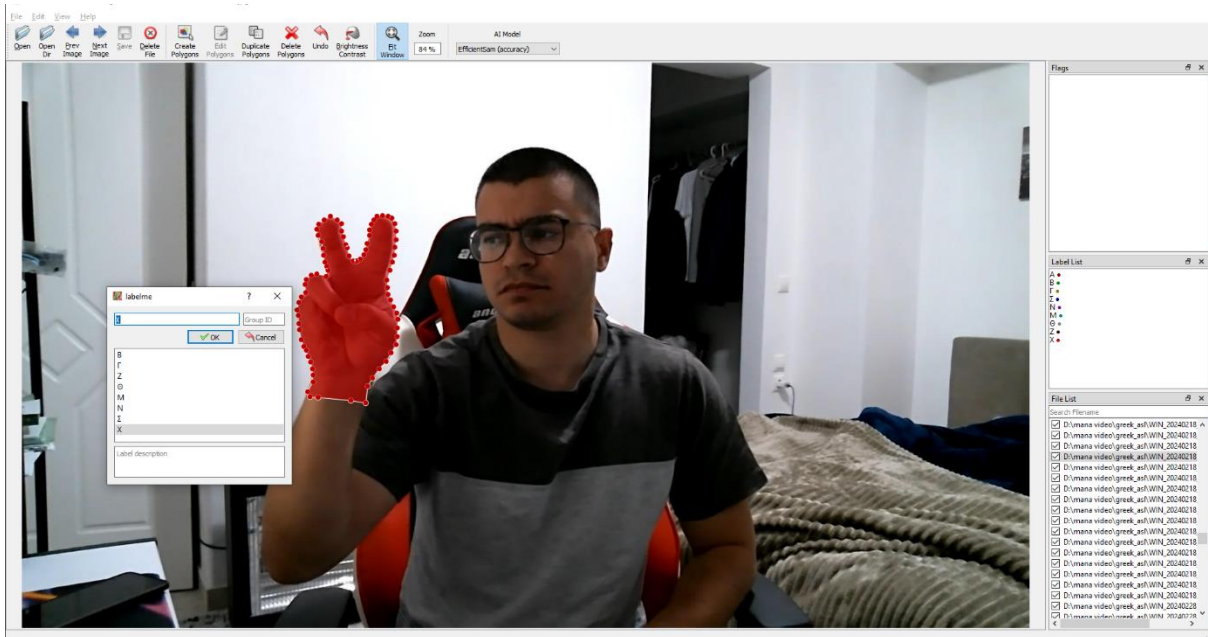
Εικόνα 6.10: Αρχή του annotation

5) Το ολοκληρωμένο αποτέλεσμα του annotation αποτυπώνεται στην εικόνα 6.11 όπου η χειρομορφή βρίσκεται μέσα στο πολύγωνο.



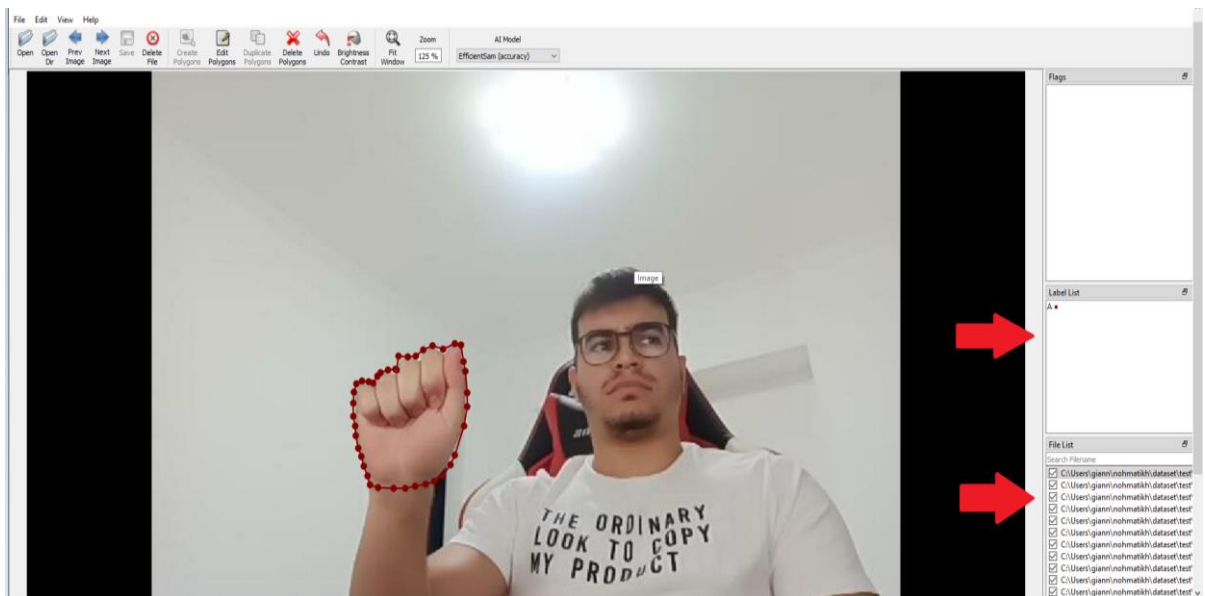
Εικόνα 6.11: Annotation χειρομορφής με το LabelMe

6) Κατά την ολοκλήρωση του annotation εμφανίζεται ένα παράθυρο. Αυτό το παράθυρο καθορίζει το label με το οποίο θα ονοματιστεί. Ανάλογα τη χειρομορφή καθορίζεται και το label που του δίνετε. Το label φαίνεται στην εικόνα 6.12



Εικόνα 6.12: Δημιουργία label

Η διαδικασία επαναλαμβάνετε για τις 1419 φωτογραφίες αλλάζοντας το label όταν είναι απαραίτητο. Τα labels όλων των annotatate φωτογραφιών φαίνονται στο παράθυρο εικόνα 6.13 όπως και σε ποια φωτογραφία έχει δημιουργηθεί label εικόνα 6.13.



Εικόνα 6.13: Εμφάνιση του label list και του file list

6.4 Μετατροπή των labels σε άλλων τύπο αρχείου.

Εφόσον τελειώσει η διαδικασία του annotation μέσα στον φάκελο υπάρχουν οι φωτογραφίες και τα labels για την κάθε μια φωτογραφία σε τύπο αρχείου JSON file (.json). Τα αρχεία αυτού

του τύπου πρέπει να προβούν σε μετατροπή για να μπορεί να τα αναγνωρίζει το yolov8. Εξαιτίας αυτού χρησιμοποιούνται οι εντολές στο client του miniconda labelme2yolo --json_dir dataset/train και labelme2yolo --json_dir dataset/test (εικόνα 6.14).

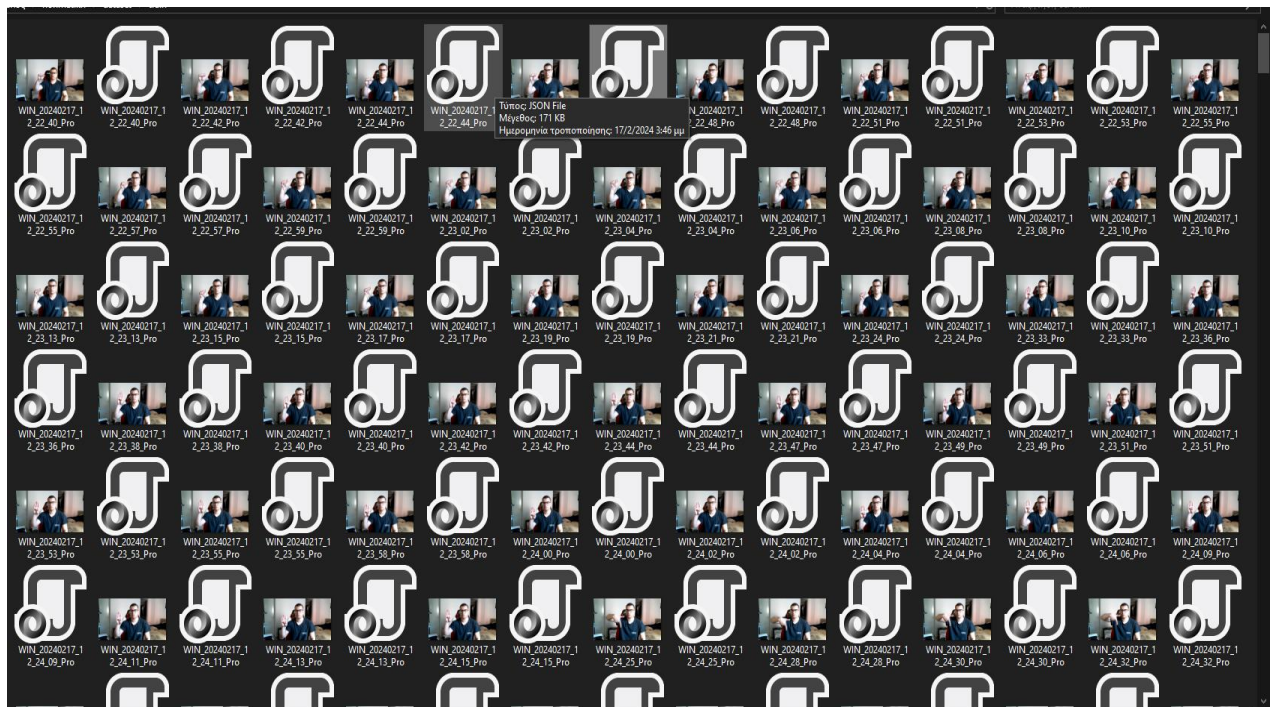
```
(takis) C:\Users\giann\nohmatikh>labelme2yolo --json_dir dataset/train
INFO:labelme2yolo:Searching label list from json files ...
INFO:labelme2yolo:Converting train set ...
Converting...-----100% 0:00:00
INFO:labelme2yolo:Converting val set ...
Converting...-----100% 0:00:00
INFO:labelme2yolo:Converting test set ...
Converting...-----0% -:--:--

(takis) C:\Users\giann\nohmatikh>labelme2yolo --json_dir dataset/test
INFO:labelme2yolo:Searching label list from json files ...
INFO:labelme2yolo:Converting train set ...
Converting...-----100% 0:00:00
INFO:labelme2yolo:Converting val set ...
Converting...-----100% 0:00:00
INFO:labelme2yolo:Converting test set ...
Converting...-----0% -:--:--

(takis) C:\Users\giann\nohmatikh>
```

Εικόνα 6.14: Μετατροπή αρχείων JSON

Με τη συγκεκριμένη εντολή τα αρχεία γίνονται TXT αρχεία (.txt) (εικόνα 6.15). Εν συνεχεία χωρίζονται σε διαφορετικούς φακέλους φωτογραφίες και labels.



Εικόνα 6.15: Εμφάνιση αρχείων TXT μετά τη μετατροπή

6.5 Χρήση του κώδικα της ultralytics για την εκτέλεση του training και τη δημιουργία του YOLOv8.

Το πρόγραμμα AI που χρησιμοποιείται είναι της ultralytics το model YOLOv8-seg. Η γενική λειτουργία του είναι: Το μοντέλο λαμβάνει μια είσοδο δηλαδή της annotate φωτογραφίες. Το backbone εξάγει τα σημαντικά χαρακτηριστικά από τις φωτογραφίες μέσω διαδοχικών επιπέδων convolution. Στη συνέχεια το neck ενσωματώνει χαρακτηριστικά από διάφορα επίπεδα για να χειριστεί αντικείμενα σε διαφορετικές κλίμακες. Στη συνέχεια το head προβλέπει τα bounding boxes και τις κατηγορίες των αντικείμενων. Επιπλέον, προβλέπει τις μάσκες segmentation που αντιστοιχούν στα ανιχνευόμενα αντικείμενα. Τέλος, το τελικό αποτέλεσμα περιλαμβάνει τα αντικείμενα που ανιχνεύτηκαν, τις κατηγορίες τους, και τις μάσκες που αντιπροσωπεύουν τα περιγράμματα των αντικειμένων στην εικόνα. Παρακάτω αναλύεται ο βασικός κώδικας που είναι για όλα τα μοντέλα του YOLO.

Από Ultralytics github. (n.d.)

```
1 import inspect
2 from pathlib import Path
3 from typing import List, Union
4 import numpy as np
5 import torch
6 from PIL import Image
7 from ultralytics.cfg import TASK2DATA, get_cfg, get_save_dir
8 from ultralytics.engine.results import Results
9 from ultralytics.hub import HUB_WEB_ROOT, HUBTrainingSession
10 from ultralytics.nn.tasks import attempt_load_one_weight,
    guess_model_task, nn, yaml_model_load
11 from ultralytics.utils import (
12     ARGV,
13     ASSETS,
14     DEFAULT_CFG_DICT,
15     LOGGER,
16     RANK,
17     SETTINGS,
18     callbacks,
19     checks,
20     emojis,
21     yaml_load,
22 )
```

Πίνακας 6.1: Ανάλυση κώδικα ultralytics τμήμα 1

Ο παραπάνω κώδικας (πίνακας 6.1) από 1 γραμμή έως 22 γραμμή εισάγει όλες τις απαραίτητες βιβλιοθήκες και modules για τη λειτουργία του YOLO μοντέλου. Περιλαμβάνει βασικές

βιβλιοθήκες της Python, όπως το inspect για τη συλλογή πληροφοριών σχετικά με τις λειτουργίες και κλάσεις και το Path για διαχείριση διαδρομών αρχείων. Επίσης, περιλαμβάνει το typing για τον ορισμό τύπων δεδομένων. Προστίθενται εξωτερικές βιβλιοθήκες, όπως το numpy και το torch, που υποστηρίζουν μαθηματικούς υπολογισμούς και εκπαίδευση νευρωνικών δικτύων αντίστοιχα, καθώς και το PIL.Image, το οποίο διευκολύνει την επεξεργασία εικόνων. Τέλος, εισάγονται modules από την ultralytics, τα οποία περιλαμβάνουν εργαλεία για διαχείριση διαμορφώσεων και δεδομένων του YOLO, συνδέει το Ultralytics HUB για λήψη προεκπαιδευμένων μοντέλων και βοηθητικά εργαλεία για τη διαχείριση του μοντέλου, όπως καταγραφές, έλεγχοι παραμέτρων, και ενσωμάτωση διαμορφώσεων YAML. Με αυτές τις εισαγωγές, ο κώδικας είναι πλέον έτοιμος να υποστηρίξει τη δημιουργία, εκπαίδευση, και χρήση του YOLO μοντέλου με όλες τις απαιτούμενες βιβλιοθήκες και εργαλεία.

```
23 class Model(nn.Module):
24     def __init__(
25         self,
26         model: Union[str, Path] = "yolo8n.pt",
27         task: str = None,
28         verbose: bool = False,
29     ) -> None:
30         super().__init__()
31         self.callbacks = callbacks.get_default_callbacks()
32         self.predictor = None
33         self.model = None
34         self.trainer = None
35         self.ckpt = None
36         self.cfg = None
37         self.ckpt_path = None
38         self.overrides = {}
39         self.metrics = None
40         self.session = None
41         self.task = task
42         model = str(model).strip()
```

Πίνακας 6.2: Ανάλυση κώδικα ultralytics τμήμα 2

Ο παραπάνω κώδικας (πίνακας 6.2) από γραμμή 23 μέχρι γραμμή 42 ορίζει την κλάση Model, η οποία αποτελεί τον βασικό πυρήνα για τη διαχείριση του YOLO μοντέλου, και αρχικοποιεί τη μέθοδο `__init__`. Η Model κληρονομεί από την `nn.Module` του PyTorch, επιτρέποντας στο μοντέλο να λειτουργεί στο πλαίσιο της βιβλιοθήκης αυτής και να ενσωματώνει όλες τις απαραίτητες δυνατότητες για τη δημιουργία και χρήση νευρωνικών δικτύων.

Παράμετροι: Η `__init__` δέχεται τρεις παραμέτρους:

model: Δέχεται είτε μια διαδρομή σε αρχείο με τα weights του μοντέλου είτε μια διαδρομή για τη διαμόρφωση του μοντέλου.

task: Προσδιορίζει τον τύπο της εργασίας που θα εκτελέσει το μοντέλο, όπως ανίχνευση αντικειμένων ή ταξινόμηση.

verbose: Επιτρέπει λεπτομερή καταγραφή της εκτέλεσης, αν είναι ενεργοποιημένο.

Αρχικοποίηση μεταβλητών: Οι διάφορες μεταβλητές, όπως callbacks, predictor, trainer, ckpt, cfg και overrides, ρυθμίζονται για να διαχειρίζονται διαφορετικές πτυχές του YOLO μοντέλου. Ειδικότερα:

callbacks: Αποθηκεύει προεπιλεγμένες ενέργειες που θα εκτελεστούν κατά την εκπαίδευση ή πρόβλεψη.

predictor και trainer: Θα χρησιμοποιηθούν για την εκτέλεση προβλέψεων και εκπαίδευσης, αντίστοιχα.

cfg, ckpt και overrides: Κρατούν τις ρυθμίσεις του μοντέλου, το αρχείο weights (checkpoint) και τυχόν προσαρμογές στις παραμέτρους εκτέλεσης.

Με την αρχικοποίηση αυτής της κλάσης, ο κώδικας διασφαλίζει ότι η κλάση Model έχει τις απαραίτητες δομές και παραμέτρους για τη σωστή διαχείριση και εκτέλεση των λειτουργιών του YOLO μοντέλου.

```
43 # Check if Ultralytics HUB model from https://hub.ultralytics.com
44 if self.is_hub_model(model):
45     # Fetch model from HUB
46     checks.check_requirements("hub-sdk>=0.0.12")
47     session = HUBTrainingSession.create_session(model)
48     model = session.model_file
49     if session.train_args: # training sent from HUB
50         self.session = session
51
52 # Check if Triton Server model
53 elif self.is_triton_model(model):
54     self.model_name = self.model = model
55     return
```

Πίνακας 6.3: Ανάλυση κώδικα *ulalytics* τμήμα 3

Αυτό το κομμάτι του κώδικα (πίνακας 6.3) από γραμμή 43 μέχρι γραμμή 55 ελέγχει αν το μοντέλο προέρχεται από το Ultralytics HUB ή τον Triton Server και εκτελεί τις κατάλληλες ενέργειες για να το φορτώσει ανάλογα με την πηγή του. Εάν το model αντιστοιχεί σε ένα μοντέλο από το Ultralytics HUB, η μέθοδος `is_hub_model` επιστρέφει True. Στην περίπτωση αυτή, το `HUBTrainingSession.create_session(model)` δημιουργεί μια νέα συνεδρία εκπαίδευσης και ανακτά το αρχείο του μοντέλου από το HUB. Αν η συνεδρία έχει παραμέτρους εκπαίδευσης (`train_args`), αυτές αποθηκεύονται στη μεταβλητή `session` για

μελλοντική χρήση. Αν το μοντέλο προέρχεται από τον Triton Server, το `is_triton_model` επιστρέφει `True`, και το μοντέλο αποθηκεύεται στις μεταβλητές `model_name` και `model`, ενώ η εκτέλεση διακόπτεται με `return` για να μην συνεχίσει με τις επόμενες ενέργειες.

Με αυτό τον τρόπο, ο κώδικας μπορεί να εντοπίσει και να ενσωματώσει μοντέλα από εξωτερικές πηγές, είτε από το Ultralytics HUB είτε από τον Triton Server, διασφαλίζοντας τη σωστή διαχείριση και τη δυνατότητα να χρησιμοποιηθεί το μοντέλο για περαιτέρω επεξεργασία ή εκπαίδευση.

```
56 # Load or create new YOLO model
57 if Path(model).suffix in {".yaml", ".yml"}:
58     self._new(model, task=task, verbose=verbose)
59 else:
60     self._load(model, task=task)
```

Πίνακας 6.4: Ανάλυση κώδικα `ultralytics` τμήμα 4

Αυτό το τμήμα του κώδικα (πίνακας 6.4) ξεκινώντας από τη γραμμή 56 μέχρι τη γραμμή 60 αποφασίζει αν θα δημιουργηθεί ένα νέο YOLO μοντέλο ή αν θα φορτωθεί ένα ήδη εκπαιδευμένο μοντέλο από ένα αρχείο `weights`, ανάλογα με τον τύπο του αρχείου που δόθηκε ως `model`. Εάν το αρχείο `model` έχει κατάληξη `.yaml` ή `.yml`, αυτό υποδηλώνει ένα αρχείο διαμόρφωσης YAML για το μοντέλο. Σε αυτή την περίπτωση, καλείται η μέθοδος `new`, η οποία δημιουργεί ένα νέο YOLO μοντέλο με βάση τις ρυθμίσεις που καθορίζονται στο YAML.

Εάν το αρχείο δεν είναι YAML, τότε ο κώδικας υποθέτει ότι πρόκειται για προεκπαιδευμένα `weights` και καλεί τη μέθοδο `_load` για να φορτώσει το μοντέλο από αυτό το αρχείο.

Με αυτό τον έλεγχο, ο κώδικας παρέχει ευελιξία ώστε το YOLO μοντέλο να μπορεί να δημιουργηθεί από την αρχή με βάση συγκεκριμένες ρυθμίσεις ή να φορτωθεί ως προεκπαιδευμένο μοντέλο, διευκολύνοντας την προσαρμογή του YOLO σε διαφορετικά σενάρια χρήσης.

```
61 def __call__(
62     self,
63     source: Union[str, Path, int, Image.Image, list, tuple, np.ndarray,
64     torch.Tensor] = None,
65     stream: bool = False,
66     **kwargs,
67 ) -> list:
68     return self.predict(source, stream, **kwargs)
```

Πίνακας 6.5: Ανάλυση κώδικα `ultralytics` τμήμα 5

Αυτό το τμήμα του κώδικα (πίνακας 6.5) αρχίζοντας από τη γραμμή 61 και τελειώνοντας στη γραμμή 67 ορίζει τη μέθοδο `__call__`, που επιτρέπει στο αντικείμενο `Model` να χρησιμοποιηθεί σαν να ήταν συνάρτηση, διευκολύνοντας την άμεση κλήση για πρόβλεψη. Η μέθοδος αυτή

ανακατευθύνει τις κλήσεις στη μέθοδο predict, η οποία θα εκτελέσει την πρόβλεψη πάνω στην καθορισμένη πηγή (source).

Παράμετροι:

source: Δέχεται διάφορους τύπους δεδομένων, όπως διαδρομές αρχείων, εικόνες, λίστες, πίνακες numpy ή torch.Tensor, καλύπτοντας έτσι ένα μεγάλο εύρος μορφών δεδομένων για την πηγή της πρόβλεψης.

stream: Αν το stream είναι True, υποδηλώνει ότι η πρόβλεψη θα γίνεται με συνεχή ροή δεδομένων, κάτι χρήσιμο για βίντεο ή ζωντανή ανάλυση εικόνων.

predict: Καλώντας την predict, η μέθοδος αυτή επιστρέφει τη λίστα με τα αποτελέσματα της πρόβλεψης, δίνοντας τη δυνατότητα εύκολης πρόβλεψης με μια γραμμή κώδικα, π.χ., model(source).

Η μέθοδος __call__ βελτιώνει την ευχρηστία της κλάσης Model, επιτρέποντας την άμεση κλήση της για πρόβλεψη χωρίς την ανάγκη να καλείται ξεχωριστά η predict.

```
68 @staticmethod
69 def is_triton_model(model: str) -> bool:
70     from urllib.parse import urlsplit
71     url = urlsplit(model)
72     return url.netloc and url.path and url.scheme in {"http", "grpc"}
73
74 @staticmethod
75 def is_hub_model(model: str) -> bool:
76     return model.startswith(f"{HUB_WEB_ROOT}/models/")
```

Πίνακας 6.6: Ανάλυση κώδικα ultalytics τμήμα 6

Οι μέθοδοι is_triton_model και is_hub_model (πίνακας 6.6) από γραμμή 68 μέχρι γραμμή 76 είναι στατικές μέθοδοι που χρησιμοποιούνται για να προσδιορίσουν αν ένα μοντέλο προέρχεται από τον Triton Server ή το Ultralytics HUB αντίστοιχα.

Η εντολή is_triton_model αναλύει το URL του μοντέλου χρησιμοποιώντας το urlsplit και επιστρέφει True αν το URL έχει έγκυρο scheme (http ή grpc) και περιέχει netloc και path. Αυτό υποδηλώνει ότι το μοντέλο βρίσκεται στον Triton Server.

Η εντολή is_hub_model ελέγχει αν το όνομα του μοντέλου ξεκινά με τη βασική διεύθυνση του Ultralytics HUB (HUB_WEB_ROOT), υποδεικνύοντας έτσι ότι το μοντέλο προέρχεται από το Ultralytics HUB. Αυτές οι μέθοδοι βοηθούν στον εντοπισμό της πηγής του μοντέλου, επιτρέποντας στον κώδικα να χειριστεί διαφορετικά μοντέλα ανάλογα με την πηγή τους.

```
77 def _new(self, cfg: str, task=None, model=None, verbose=False) ->
None:
78     cfg_dict = yaml_model_load(cfg)
79     self.cfg = cfg
```

```

80     self.task = task or guess_model_task(cfg_dict)
81     self.model = (model or self._smart_load("model"))(cfg_dict,
verbose=verbose and RANK == -1)
82     self.overrides["model"] = self.cfg
83     self.overrides["task"] = self.task
84     self.model.args = {**DEFAULT_CFG_DICT, **self.overrides}
85     self.model.task = self.task
86     self.model_name = cfg

```

Πίνακας 6.7: Ανάλυση κώδικα *ulalytics* τμήμα 7

Αυτό το τμήμα κώδικα (πίνακας 6.7) ξεκινώντας από τη γραμμή 77 μέχρι τη γραμμή 86 τη μέθοδο περιέχει `new`, η οποία δημιουργεί ένα νέο YOLO μοντέλο με βάση ένα αρχείο διαμόρφωσης YAML. Χρησιμοποιείται όταν το αρχείο `model` έχει κατάληξη `.yaml` ή `.yml`. επίσης η `yaml_model_load(cfg)` χρησιμοποιείται για να διαβάσει και να φορτώσει το αρχείο YAML, αποθηκεύοντας τις ρυθμίσεις στη `cfg_dict`. Αν δεν έχει ήδη οριστεί εργασία (`task`), η `guess_model_task` επιλέγει την κατάλληλη εργασία με βάση τις ρυθμίσεις του YAML και χρησιμοποιεί τις ρυθμίσεις από το YAML και δημιουργεί το μοντέλο, αποθηκεύοντας το όνομα διαμόρφωσης (`self.model_name`) και όλες τις παραμέτρους (`overrides`) για χρήση αργότερα.

Με αυτή τη μέθοδο, ο κώδικας μπορεί να δημιουργήσει ένα πλήρως διαμορφωμένο YOLO μοντέλο από την αρχή, χρησιμοποιώντας ρυθμίσεις που καθορίζονται εξωτερικά σε αρχείο YAML.

```

87 def _load(self, weights: str, task=None) -> None:
88     if weights.lower().startswith(("https://", "http://", "rtsp://",
"rtmp://", "tcp://")):
89         weights = checks.check_file(weights,
download_dir=SETTINGS["weights_dir"])
90         weights = checks.check_model_file_from_stem(weights)
91         if Path(weights).suffix == ".pt":
92             self.model, self.ckpt = attempt_load_one_weight(weights)
93             self.task = self.model.args["task"]
94             self.overrides = self.model.args =
self._reset_ckpt_args(self.model.args)
95             self.ckpt_path = self.model.pt_path
96         else:
97             weights = checks.check_file(weights)
98             self.model, self.ckpt = weights, None
99             self.task = task or guess_model_task(weights)
100            self.ckpt_path = weights
101            self.overrides["model"] = weights
102            self.overrides["task"] = self.task
103            self.model_name = weights

```

Πίνακας 6.8: Ανάλυση κώδικα *ulalytics* τμήμα 8

Αυτό το τμήμα (πίνακας 6.8) αρχίζοντας από τη γραμμή 87 και τελειώνοντας στη γραμμή 103 περιέχει τη μέθοδο `_load`, η οποία φορτώνει ένα προεκπαιδευμένο μοντέλο από αρχείο βαρών, συνήθως με κατάληξη `.pt`. Αν το `weights` είναι URL, το αρχείο κατεβαίνει τοπικά στον υπολογιστή. Αν το αρχείο έχει κατάληξη `.pt`, υποδηλώνει αρχείο `weights` PyTorch και φορτώνεται με τη `attempt_load_one_weight`. επίσης γίνεται αποθήκευση των `weights` και άλλων παραμέτρων, όπως η εργασία (`task`) και οι διαμορφώσεις `overrides`.

Με αυτή τη μέθοδο, ο κώδικας μπορεί να φορτώσει υπάρχοντα `weights` για το μοντέλο, επιτρέποντας την εύκολη πρόσβαση και χρήση των προεκπαιδευμένων παραμέτρων για πρόβλεψη ή περαιτέρω εκπαίδευση.

```
104 def _check_is_pytorch_model(self) -> None:
105     pt_str = isinstance(self.model, (str, Path)) and
Path(self.model).suffix == ".pt"
106     pt_module = isinstance(self.model, nn.Module)
107     if not (pt_module or pt_str):
108         raise TypeError(
109             f"model='{self.model}' should be a *.pt PyTorch model to
run this method, but is a different format. "
110             f"PyTorch models can train, val, predict and export, i.e.
'model.train(data=...)', but exported "
111             f"formats like ONNX, TensorRT etc. only support 'predict'
and 'val' modes, "
112             f"i.e. 'yolo predict model=yolov8n.onnx'.\nTo run CUDA or
MPS inference please pass the device "
113             f"argument directly in your inference command, i.e.
'model.predict(source=..., device=0)'"
114         )
```

Πίνακας 6.9: Ανάλυση κώδικα *ulalytics* τμήμα 9

Αυτό το κομμάτι του κώδικα (πίνακας 6.9) από γραμμή 104 μέχρι γραμμή 114 ορίζει τη μέθοδο `_check_is_pytorch_model`, η οποία επαληθεύει αν το μοντέλο είναι έγκυρο PyTorch μοντέλο (`*.pt` αρχείο). Αυτός ο έλεγχος διασφαλίζει ότι το μοντέλο έχει τη σωστή μορφή ώστε να μπορεί να εκπαιδευτεί ή να χρησιμοποιηθεί για πρόβλεψη. Ο κώδικας ελέγχει αν το `self.model` είναι είτε αρχείο `.pt` είτε αντικείμενο `nn.Module`. Αυτό σημαίνει ότι το μοντέλο πρέπει να είναι PyTorch μοντέλο. Αν το μοντέλο δεν πληροί τις παραπάνω προϋποθέσεις, η μέθοδος προκαλεί `TypeError`, μαζί με ένα επεξηγηματικό μήνυμα που περιγράφει τις μορφές μοντέλου που υποστηρίζονται και πώς να χρησιμοποιηθεί σωστά το PyTorch για εκπαίδευση ή πρόβλεψη. Με τη μέθοδο αυτή, ο κώδικας εξασφαλίζει ότι θα εντοπίσει ασυμβατότητες στο φορμάτ του μοντέλου πριν από την εκτέλεση εκπαιδευτικών ή προβλεπτικών λειτουργιών, αποτρέποντας σφάλματα σε μεταγενέστερα βήματα.

```
115 def reset_weights(self) -> "Model":
```

```

116 self._check_is_pytorch_model()
117 for m in self.model.modules():
118     if hasattr(m, "reset_parameters"):
119         m.reset_parameters()
120 for p in self.model.parameters():
121     p.requires_grad = True
122 return self

```

Πίνακας 6.10: Ανάλυση κώδικα ultalytics τμήμα 10

Η μέθοδος `reset_weights` (πίνακας 6.10) αρχίζοντας από τη γραμμή 115 και τελειώνοντας στη γραμμή 122 επαναφέρει τα βάρη του μοντέλου στις αρχικές τους τιμές, επιτρέποντας επανεκπαίδευση από την αρχή. Καλεί τη `_check_is_pytorch_model` για να βεβαιωθεί ότι το μοντέλο είναι έγκυρο PyTorch μοντέλο, όπως απαιτείται. Ακόμη η μέθοδος διατρέχει όλες τις στρώσεις του μοντέλου, και όπου υπάρχει η δυνατότητα (`reset_parameters`), επαναφέρει τις παραμέτρους στην αρχική τους κατάσταση. Επιπλέον, θέτει τις παραμέτρους σε `requires_grad=True`, επιτρέποντας την εκπαίδευση όλων των στρώσεων του μοντέλου από την αρχή.

Με την `reset_weights`, ο κώδικας καθιστά δυνατή την εκ νέου εκπαίδευση του μοντέλου, χωρίς να χρειάζεται νέα αρχικοποίηση.

```

123 def load(self, weights: Union[str, Path] = "yolo11n.pt") -> "Model":
124     self._check_is_pytorch_model()
125     if isinstance(weights, (str, Path)):
126         self.overrides["pretrained"] = weights # remember the weights
for DDP training
127         weights, self.ckpt = attempt_load_one_weight(weights)
128     self.model.load(weights)
129     return self

```

Πίνακας 6.11: Ανάλυση κώδικα ultalytics τμήμα 11

Η μέθοδος `load` (πίνακας 6.11) ξεκινώντας από τη γραμμή 123 μέχρι τη γραμμή 129 φορτώνει προεκπαιδευμένα `weights` στο μοντέλο από το αρχείο `weights`, διευκολύνοντας τη χρήση αποθηκευμένων παραμέτρων για επαναχρησιμοποίηση ή αξιολόγηση. Επαληθεύει ότι το μοντέλο είναι PyTorch μοντέλο μέσω της `_check_is_pytorch_model`. επιπλέον αν το `weights` είναι μια διαδρομή η `attempt_load_one_weight` φορτώνει τα `weights` και τα αποθηκεύει στο μοντέλο. Η μέθοδος επιστρέφει το μοντέλο, επιτρέποντας την αλληλουχία των εντολών.

Η `load` διευκολύνει την εύκολη εφαρμογή προεκπαιδευμένων `weights` στο μοντέλο, προσφέροντας τη δυνατότητα άμεσης χρήσης του για πρόβλεψη ή περαιτέρω εκπαίδευση.

```

130 def save(self, filename: Union[str, Path] = "saved_model.pt") ->
None:
131     self._check_is_pytorch_model()
132     from copy import deepcopy

```

```

133     from datetime import datetime
134
135     from ultralytics import __version__
136
137     updates = {
138         "model": deepcopy(self.model).half() if isinstance(self.model,
nn.Module) else self.model,
139         "date": datetime.now().isoformat(),
140         "version": __version__,
141         "license": "AGPL-3.0 License
(https://ultralytics.com/license)",
142         "docs": "https://docs.ultralytics.com",
143     }
144     torch.save(**self.ckpt, **updates), filename)

```

Πίνακας 6.12: Ανάλυση κώδικα ultralytics τμήμα 12

Η μέθοδος save (πίνακας 6.12) από γραμμή 130 μέχρι γραμμή 144 αποθηκεύει το μοντέλο σε αρχείο .pt, συμπεριλαμβάνοντας πληροφορίες σχετικά με την έκδοση, την άδεια χρήσης, και την ημερομηνία αποθήκευσης. Επαληθεύει ότι το μοντέλο είναι PyTorch μοντέλο. Επίσης, χρησιμοποιεί deepcopy για να αντιγράψει το μοντέλο, μαζί με πληροφορίες για την ημερομηνία, την έκδοση και την άδεια χρήσης, και τις προσθέτει στο αρχείο. Ακόμη χρησιμοποιεί τη torch.save για να αποθηκεύσει το αρχείο με το όνομα filename.

Αυτή η μέθοδος διασφαλίζει ότι το μοντέλο μπορεί να αποθηκευτεί με όλες τις πληροφορίες του, έτοιμο για επαναφόρτωση και διανομή.

```

145 def info(self, detailed: bool = False, verbose: bool = True):
146     self._check_is_pytorch_model()
147     return self.model.info(detailed=detailed, verbose=verbose)

```

Πίνακας 6.13: Ανάλυση κώδικα ultralytics τμήμα 13

Η μέθοδος info (πίνακας 6.13) αρχίζοντας από τη γραμμή 145 και τελειώνοντας στη γραμμή 147 επιστρέφει πληροφορίες για το μοντέλο, παρέχοντας λεπτομέρειες σχετικά με τις στρώσεις και τις παραμέτρους του. Βεβαιώνεται ότι το μοντέλο είναι PyTorch και προσφέρει την επιλογή να επιστραφούν βασικές (detailed=False) ή αναλυτικές (detailed=True) πληροφορίες. Με την info, ο χρήστης μπορεί να λάβει μια λεπτομερή αναφορά για το μοντέλο και τη δομή του, χρήσιμο για διαγνωστικούς σκοπούς ή έλεγχο πριν την εκπαίδευση.

```

148 def fuse(self):
149     self._check_is_pytorch_model()
150     self.model.fuse()

```

Πίνακας 6.14: Ανάλυση κώδικα ultralytics τμήμα 14

Η μέθοδος fuse (πίνακας 6.14) από γραμμή 148 μέχρι γραμμή 150 εκτελεί μια διαδικασία συνένωσης στρώσεων στο μοντέλο, βελτιώνοντας την ταχύτητα και την απόδοσή του. Εξασφαλίζει ότι το μοντέλο είναι PyTorch. Καλεί τη fuse του PyTorch για να συγχωνεύσει τις

στρώσεις του μοντέλου, μειώνοντας τους απαιτούμενους πόρους και αυξάνοντας την αποδοτικότητα.

Η fuse είναι χρήσιμη για την βελτιστοποίηση του μοντέλου, ειδικά κατά την εκτέλεση σε περιβάλλον παραγωγής όπου η ταχύτητα είναι σημαντική.

```
151 def embed(  
152     self,  
153     source: Union[str, Path, int, list, tuple, np.ndarray,  
torch.Tensor] = None,  
154     stream: bool = False,  
155     **kwargs,  
156 ) -> list:  
157     if not kwargs.get("embed"):  
158         kwargs["embed"] = [len(self.model.model) - 2]  
159     return self.predict(source, stream, **kwargs)
```

Πίνακας 6.15: Ανάλυση κώδικα *ultalytics* τμήμα 15

Η μέθοδος `embed` (πίνακας 6.15) ξεκινώντας από τη γραμμή 151 μέχρι τη γραμμή 159 επιστρέφει τα `embeddings` από τις τελευταίες στρώσεις του μοντέλου. Τα `embeddings` αποτελούν αναπαραστάσεις υψηλού επιπέδου των δεδομένων εισόδου και είναι χρήσιμα για διάφορες εργασίες ανάλυσης.

Παράμετροι:

source: Δέχεται την πηγή των δεδομένων εισόδου, η οποία μπορεί να είναι εικόνα, πίνακας numpy ή `torch.Tensor`.

stream: Καθορίζει αν τα `embeddings` θα παράγονται σε ροή.

Αν δεν έχει οριστεί συγκεκριμένο `layer`, η μέθοδος χρησιμοποιεί το δεύτερο προς τελευταίο `layer` του μοντέλου για παραγωγή `embeddings`.

Με την `embed`, ο χρήστης μπορεί να ανακτήσει τα `embeddings` των δεδομένων εισόδου, τα οποία είναι χρήσιμα για την κατανόηση της πληροφορίας που εξάγει το μοντέλο.

```
160 def predict(  
161     self,  
162     source: Union[str, Path, int, Image.Image, list, tuple,  
np.ndarray, torch.Tensor] = None,  
163     stream: bool = False,  
164     predictor=None,  
165     **kwargs,  
166 ) -> List[Results]:  
167     if source is None:  
168         source = ASSETS
```

```

169     LOGGER.warning(f"WARNING ⚠️ 'source' is missing. Using
'source={source}'.")
170     is_cli = (ARGV[0].endswith("yolo") or
ARGV[0].endswith("ultralytics")) and any(
171         x in ARGV for x in ("predict", "track", "mode=predict",
"mode=track")
172     )
173     custom = {"conf": 0.25, "batch": 1, "save": is_cli, "mode":
"predict"}
174     args = {**self.overrides, **custom, **kwargs}
175     prompts = args.pop("prompts", None)
176     if not self.predictor:
177         self.predictor = (predictor or
self._smart_load("predictor"))(overrides=args, _callbacks=self.callbacks)
178         self.predictor.setup_model(model=self.model, verbose=is_cli)
179     else:
180         self.predictor.args = get_cfg(self.predictor.args, args)
181         if "project" in args or "name" in args:
182             self.predictor.save_dir =
get_save_dir(self.predictor.args)
183     if prompts and hasattr(self.predictor, "set_prompts"):
184         self.predictor.set_prompts(prompts)
185     return self.predictor.predict_cli(source=source) if is_cli else
self.predictor(source=source, stream=stream)

```

Πίνακας 6.16: Ανάλυση κώδικα *ultralytics* τμήμα 16

Η μέθοδος `predict` (πίνακας 6.16) αρχίζοντας από τη γραμμή 160 και τελειώνοντας στη γραμμή 185 εκτελεί πρόβλεψη στο YOLO μοντέλο, χρησιμοποιώντας δεδομένα από διάφορες μορφές εισόδου. Μπορεί να διαχειριστεί πρόβλεψη τόσο μέσω εντολών CLI όσο και από εσωτερικές κλήσεις του προγράμματος.

Παράμετροι:

source: Δέχεται την πηγή εισόδου, όπως εικόνα ή βίντεο, υποστηρίζοντας πολλές μορφές δεδομένων.

stream: Αν είναι `True`, η πρόβλεψη εκτελείται σε ροή, κάτι χρήσιμο για ανάλυση βίντεο.

Προσαρμογές για CLI: Αν η `predict` εκτελείται από το CLI, ρυθμίζονται οι κατάλληλες παράμετροι για αποθήκευση και λειτουργία σε λειτουργία πρόβλεψης.

Εκτέλεση πρόβλεψης: Η `predictor` χρησιμοποιεί τις ρυθμίσεις και τα δεδομένα εισόδου για να εκτελέσει πρόβλεψη και επιστρέφει τη λίστα αποτελεσμάτων.

Με τη μέθοδο `predict`, ο χρήστης μπορεί να λάβει προβλέψεις από το μοντέλο για εικόνες ή βίντεο με ένα ενιαίο API.

```

186 def track(
187     self,

```

```

188     source: Union[str, Path, int, list, tuple, np.ndarray,
torch.Tensor] = None,
189     stream: bool = False,
190     persist: bool = False,
191     **kwargs,
192 ) -> List[Results]:
193     if not hasattr(self.predictor, "trackers"):
194         from ultralytics.trackers import register_tracker
195         register_tracker(self, persist)
196     kwargs["conf"] = kwargs.get("conf") or 0.1
197     kwargs["batch"] = kwargs.get("batch") or 1
198     kwargs["mode"] = "track"
199     return self.predict(source=source, stream=stream, **kwargs)

```

Πίνακας 6.17: Ανάλυση κώδικα *ultralytics* τμήμα 17

Η μέθοδος `track` (πίνακας 6.17) από γραμμή 186 μέχρι γραμμή 199 χρησιμοποιείται για την παρακολούθηση αντικειμένων σε βίντεο ή συνεχείς ροές εικόνων. Ο στόχος της είναι η αναγνώριση και παρακολούθηση αντικειμένων σε κάθε καρέ μιας ροής βίντεο.

Παράμετροι:

source: Δέχεται την πηγή δεδομένων, η οποία μπορεί να είναι βίντεο, λίστα εικόνων ή άλλες μορφές δεδομένων.

stream και persist: Αν είναι ενεργά, το `stream` επεξεργάζεται συνεχή ροή και το `persist` κρατά ενεργό τον εντοπιστή κατά τη ροή.

register_tracker: Αν δεν έχει καταχωρηθεί εντοπιστής αντικειμένων, γίνεται καταχώρηση με το `register_tracker`.

Με την `track`, ο χρήστης μπορεί να εφαρμόσει παρακολούθηση αντικειμένων σε βίντεο, προσφέροντας ένα ισχυρό εργαλείο για ανάλυση κινούμενων στόχων.

```

200 def val(
201     self,
202     validator=None,
203     **kwargs,
204 ):
205     custom = {"rect": True}
206     args = {**self.overrides, **custom, **kwargs, "mode": "val"}
207     validator = (validator or
self._smart_load("validator"))(args=args, _callbacks=self.callbacks)
208     validator(model=self.model)
209     self.metrics = validator.metrics
210     return validator.metrics

```

Πίνακας 6.18: Ανάλυση κώδικα *ultralytics* τμήμα 18

Η μέθοδος `val` (πίνακας 6.18) αρχίζει από τη γραμμή 200 και τελειώνει στη γραμμή 210 εκτελεί αξιολόγηση του μοντέλου με χρήση δεδομένων αξιολόγησης και επιστρέφει

μετρικές απόδοσης, όπως ακρίβεια και απώλεια. Συγχωνεύει τις ρυθμίσεις για λειτουργία σε mode="val", ενεργοποιώντας συγκεκριμένες διαμορφώσεις για την αξιολόγηση. Επιπρόσθετα καλεί τον validator να εκτελέσει την αξιολόγηση στο τρέχον μοντέλο, αποθηκεύοντας τις μετρικές (metrics) που προκύπτουν.

Με τη val, ο χρήστης μπορεί να μετρήσει την ακρίβεια και την απόδοση του μοντέλου σε δεδομένα που δεν έχουν χρησιμοποιηθεί για εκπαίδευση.

```
211 def benchmark(  
212     self,  
213     **kwargs,  
214 ):  
215     self._check_is_pytorch_model()  
216     from ultralytics.utils.benchmarks import benchmark  
217     custom = {"verbose": False}  
218     args = {**DEFAULT_CFG_DICT, **self.model.args, **custom, **kwargs,  
"mode": "benchmark"}  
219     return benchmark(  
220         model=self,  
221         data=kwargs.get("data"),  
222         imgsz=args["imgsz"],  
223         half=args["half"],  
224         int8=args["int8"],  
225         device=args["device"],  
226         verbose=kwargs.get("verbose"),  
227     )
```

Πίνακας 6.19: Ανάλυση κώδικα ultralytics τμήμα 19

Η μέθοδος benchmark (πίνακας 6.19) ξεκινώντας από τη γραμμή 211 μέχρι τη γραμμή 227 εκτελεί έναν έλεγχο απόδοσης στο μοντέλο, προσφέροντας πληροφορίες σχετικά με την ταχύτητα εκτέλεσης και τις απαιτήσεις σε υπολογιστικούς πόρους. Επαληθεύει ότι το μοντέλο είναι PyTorch. Ακόμα προετοιμάζει τις παραμέτρους για το benchmark, όπως το μέγεθος εικόνας, το αν θα χρησιμοποιηθεί half και int8 ακρίβεια, και τη συσκευή.

Με την benchmark, ο χρήστης μπορεί να αξιολογήσει την απόδοση του μοντέλου σε διαφορετικές παραμέτρους εκτέλεσης και να λάβει πληροφορίες σχετικά με τη βελτιστοποίηση ταχύτητας και πόρων.

```

228 def export(
229     self,
230     **kwargs,
231 ) -> str:
232     self._check_is_pytorch_model()
233     from .exporter import Exporter
234
235     custom = {
236         "imgsz": self.model.args["imgsz"],
237         "batch": 1,
238         "data": None,
239         "device": None,
240         "verbose": False,
241     }
242     args = {**self.overrides, **custom, **kwargs, "mode": "export"}
243     return Exporter(overrides=args,
_ callbacks=self.callbacks)(model=self.model)

```

Πίνακας 6.20: Ανάλυση κώδικα *ulalytics* τμήμα 20

Η μέθοδος `export` (πίνακας 6.20) από γραμμή 228 μέχρι γραμμή 243 χρησιμοποιείται για την εξαγωγή του μοντέλου σε διάφορες μορφές, όπως ONNX και TensorRT, επιτρέποντας την ενσωμάτωση του μοντέλου σε άλλες πλατφόρμες και περιβάλλοντα. Επιβεβαιώνει ότι το μοντέλο είναι PyTorch μοντέλο. Επίσης, ορίζει βασικές ρυθμίσεις, όπως το μέγεθος της εικόνας (`imgsz`), το μέγεθος παρτίδας (`batch`), και τη συσκευή (`device`). Επιπλέον, χρησιμοποιεί την κλάση `Exporter` για να εξαγάγει το μοντέλο στην επιθυμητή μορφή και επιστρέφει τη διαδρομή του εξαγόμενου αρχείου.

Η `export` διευκολύνει την εξαγωγή του μοντέλου για χρήση εκτός PyTorch, επιτρέποντας τη μεταφορά σε περιβάλλοντα παραγωγής.

```

244 def train(
245     self,
246     trainer=None,
247     **kwargs,
248 ):
249     self._check_is_pytorch_model()
250     if hasattr(self.session, "model") and self.session.model.id:
251         if any(kwargs):
252             LOGGER.warning("WARNING ⚠ using HUB training arguments,
ignoring local training arguments.")
253             kwargs = self.session.train_args
254
255     checks.check_pip_update_available()
256
257     overrides = yaml_load(checks.check_yaml(kwargs["cfg"])) if
kwargs.get("cfg") else self.overrides
258     custom = {

```

```

259     "data": overrides.get("data") or DEFAULT_CFG_DICT["data"] or
TASK2DATA[self.task],
260     "model": self.overrides["model"],
261     "task": self.task,
262 }
263 args = {**overrides, **custom, **kwargs, "mode": "train"}
264 if args.get("resume"):
265     args["resume"] = self.ckpt_path
266
267 self.trainer = (trainer or
self._smart_load("trainer"))(overrides=args, _callbacks=self.callbacks)
268 if not args.get("resume"):
269     self.trainer.model = self.trainer.get_model(weights=self.model
if self.ckpt else None, cfg=self.model.yaml)
270     self.model = self.trainer.model
271
272 self.trainer.hub_session = self.session
273 self.trainer.train()
274 if RANK in {-1, 0}:
275     ckpt = self.trainer.best if self.trainer.best.exists() else
self.trainer.last
276     self.model, _ = attempt_load_one_weight(ckpt)
277     self.overrides = self.model.args
278     self.metrics = getattr(self.trainer.validator, "metrics",
None)
279     return self.metrics

```

Πίνακας 6.21: Ανάλυση κώδικα *ultralitics* τμήμα 21

Η μέθοδος `train` εκτελεί (πίνακας 6.21) αρχίζοντας από τη γραμμή 243 και τελειώνοντας στη γραμμή 279 εκπαίδευση του YOLO μοντέλου με τα δεδομένα εκπαίδευσης και αποθηκεύει τις βελτιστοποιημένες παραμέτρους. Εξασφαλίζει ότι το μοντέλο είναι PyTorch.

Χρησιμοποιεί παραμέτρους είτε από το Ultralytics HUB (`session.train_args`) είτε από τοπικές ρυθμίσεις. Επιπρόσθετα η `trainer` εκτελεί την εκπαίδευση και αποθηκεύει τις καλύτερες παραμέτρους, καθώς και τις μετρικές (`metrics`). Η `train` διευκολύνει την εκπαίδευση του YOLO μοντέλου, καθιστώντας το έτοιμο για πρόβλεψη με τις βέλτιστες δυνατές παραμέτρους.

```

280 def tune(
281     self,
282     use_ray=False,
283     iterations=10,
284     *args,
285     **kwargs,
286 ):
287     self._check_is_pytorch_model()
288     if use_ray:
289         from ultralytics.utils.tuner import run_ray_tune
290         return run_ray_tune(self, max_samples=iterations, *args,
**kwargs)
291     else:

```

```

292     from .tuner import Tuner
293     custom = {}
294     args = {**self.overrides, **custom, **kwargs, "mode": "train"}
295     return Tuner(args=args, _callbacks=self.callbacks) (model=self,
iterations=iterations)

```

Πίνακας 6.22: Ανάλυση κώδικα *ultralitics* τμήμα 22

Η μέθοδος `tune` (πίνακας 6.22) από γραμμή 280 μέχρι γραμμή 295 βελτιστοποιεί τις παραμέτρους του μοντέλου μέσω διαδικασίας συντονισμού (tuning), που περιλαμβάνει πολλαπλές επαναλήψεις και δοκιμές. Επίσης, διασφαλίζει ότι το μοντέλο είναι PyTorch. Επιπλέον, οι δυο επιλογές για συντονισμό είναι:

Αν είναι ενεργό το `use_ray`, καλεί τη `run_ray_tune` για βελτιστοποίηση με τη βιβλιοθήκη Ray. Αλλιώς, χρησιμοποιεί την `Tuner` για εσωτερικό συντονισμό παραμέτρων.

Με τη `tune`, ο κώδικας διευκολύνει τη βελτιστοποίηση των παραμέτρων του μοντέλου, εξασφαλίζοντας καλύτερη απόδοση και ακρίβεια στην πρόβλεψη.

```

296 def _apply(self, fn) -> "Model":
297     self._check_is_pytorch_model()
298     self = super()._apply(fn)
299     self.predictor = None
300     self.overrides["device"] = self.device
301     return self

```

Πίνακας 6.23: Ανάλυση κώδικα *ultralitics* τμήμα 23

Η `_apply` εκτελεί (πίνακας 6.23) αρχίζοντας από τη γραμμή 296 και τελειώνοντας στη γραμμή 301 μια συνάρτηση σε όλα τα επίπεδα του μοντέλου και επιτρέπει την αλλαγή της συσκευής εκτέλεσης (π.χ., από CPU σε GPU). Βεβαιώνει ότι το μοντέλο είναι PyTorch και εφαρμόζει τη συνάρτηση `fn` σε κάθε επίπεδο του μοντέλου, επιτρέποντας αλλαγές στη συσκευή εκτέλεσης (device).

Η `_apply` επιτρέπει την ευέλικτη αλλαγή της συσκευής για το μοντέλο, βελτιώνοντας την απόδοση σε εξειδικευμένο hardware.

```

302 @property
303 def names(self) -> list:
304     from ultralytics.nn.autobackend import check_class_names
305     if hasattr(self.model, "names"):
306         return check_class_names(self.model.names)
307     if not self.predictor:
308         self.predictor =
self._smart_load("predictor") (overrides=self.overrides,
_callbacks=self.callbacks)
309         self.predictor.setup_model(model=self.model, verbose=False)
310     return self.predictor.model.names
311 @property
312 def device(self) -> torch.device:

```

```

313     return next(self.model.parameters()).device if
isinstance(self.model, nn.Module) else None
314 @property
315 def transforms(self):
316     return self.model.transforms if hasattr(self.model, "transforms")
else

```

Πίνακας 6.24: Ανάλυση κώδικα *ulalytics* τμήμα 24

Αυτές οι ιδιότητες (πίνακας 6.24) ξεκινώντας από τη γραμμή 302 μέχρι τη γραμμή 316 παρέχουν εύκολη πρόσβαση σε βασικές πληροφορίες του μοντέλου, όπως τα ονόματα των κατηγοριών που το μοντέλο μπορεί να αναγνωρίσει χρησιμοποιώντας έλεγχο συμβατότητας ονομάτων, τη συσκευή (device) στην οποία βρίσκεται το μοντέλο, π.χ., CPU ή GPU και τους προκαθορισμένους μετασχηματισμούς που εφαρμόζει το μοντέλο στις εισόδους.

Με αυτές τις ιδιότητες, ο χρήστης μπορεί να αντλήσει βασικές πληροφορίες για το μοντέλο, εξασφαλίζοντας τη σωστή διαμόρφωση του για την εκτέλεση προβλέψεων.

```

317 def add_callback(self, event: str, func) -> None:
318     self.callbacks[event].append(func)

```

Πίνακας 6.25: Ανάλυση κώδικα *ulalytics* τμήμα 25

Η μέθοδος `add_callback` (πίνακας 6.25) από γραμμή 317 μέχρι γραμμή 318 προσθέτει μια συνάρτηση `callback` σε ένα συγκεκριμένο γεγονός (`event`) της εκπαίδευσης ή της πρόβλεψης, επιτρέποντας την εκτέλεση επιπλέον ενεργειών κατά τη διάρκεια αυτών των διαδικασιών.

Παράμετροι:

event: Το όνομα του γεγονότος στο οποίο θα προστεθεί το `callback`, π.χ., ένα στάδιο της εκπαίδευσης.

func: Η συνάρτηση `callback` που θα εκτελείται όταν συμβαίνει το συγκεκριμένο `event` της εκπαίδευσης.

Με την `add_callback`, ο χρήστης μπορεί να προσαρμόσει τις ενέργειες που εκτελούνται σε συγκεκριμένα σημεία της διαδικασίας εκπαίδευσης ή πρόβλεψης, καθιστώντας τον κώδικα πιο ευέλικτο.

```

319 def clear_callback(self, event: str) -> None:
320     self.callbacks[event] = []

```

Πίνακας 6.26: Ανάλυση κώδικα *ulalytics* τμήμα 26

Η μέθοδος `clear_callback` (πίνακας 6.26) από γραμμή 319 μέχρι γραμμή 320 αφαιρεί όλες τις συναρτήσεις `callback` από ένα συγκεκριμένο `event`, επαναφέροντάς το στην αρχική του κατάσταση χωρίς καμία προγραμματισμένη ενέργεια.

Παράμετροι:

event: Το όνομα του γεγονότος από το οποίο θα αφαιρεθούν τα `callbacks`.

Η `clear_callback` επιτρέπει την εύκολη διαχείριση των `callbacks`, καθαρίζοντας όλα τα προγραμματισμένα `callbacks` για ένα συγκεκριμένο γεγονός.

```
321 def reset_callbacks(self) -> None:
322     for event in callbacks.default_callbacks.keys():
323         self.callbacks[event] =
[callbacks.default_callbacks[event][0]]
```

Πίνακας 6.27: Ανάλυση κώδικα *ulalytics* τμήμα 27

Η μέθοδος `reset_callbacks` (πίνακας 6.27) ξεκινώντας από τη γραμμή 321 μέχρι τη γραμμή 323 επαναφέρει όλα τα `callbacks` στα προεπιλεγμένα τους, αποκαθιστώντας τις βασικές ενέργειες που έχουν οριστεί από προεπιλογή για κάθε γεγονός. Διατρέχει όλα τα διαθέσιμα `events` και αντικαθιστά τα `callbacks` με τα αρχικά προεπιλεγμένα.

Η `reset_callbacks` εξασφαλίζει ότι τα γεγονότα επανέρχονται στις προκαθορισμένες τους λειτουργίες, διευκολύνοντας τη διαχείριση και την αποσφαλμάτωση του κώδικα.

```
324 @staticmethod
325 def _reset_ckpt_args(args: dict) -> dict:
326     include = {"imgsz", "data", "task", "single_cls"}
327     return {k: v for k, v in args.items() if k in include}
```

Πίνακας 6.28: Ανάλυση κώδικα *ulalytics* τμήμα 28

Η μέθοδος `_reset_ckpt_args` (πίνακας 6.28) αρχίζοντας από τη γραμμή 324 και τελειώνοντας στη γραμμή 327 είναι στατική και επιλέγει συγκεκριμένες παραμέτρους από ένα σύνολο ρυθμίσεων (`args`) που χρησιμοποιούνται για την ανανέωση των `weights` κατά τη φόρτωση αρχείου `checkpoint`. Από το λεξικό `args`, κρατά μόνο τις παραμέτρους `imgsz`, `data`, `task`, και `single_cls`, αγνοώντας όλες τις υπόλοιπες.

Αυτή η μέθοδος διασφαλίζει ότι μόνο οι απαραίτητες παράμετροι περνούν κατά τη διαδικασία ανανέωσης των βαρών από ένα `checkpoint`.

```
328 def _smart_load(self, key: str):
329     try:
330         return self.task_map[self.task][key]
331     except Exception as e:
332         name = self.__class__.__name__
333         mode = inspect.stack()[1][3]
334         raise NotImplementedError(
335             emojis(f"WARNING ⚠️ '{name}' model does not support
' {mode}' mode for '{self.task}' task yet.")
336         ) from e
```

Πίνακας 6.29: Ανάλυση κώδικα *ulalytics* τμήμα 29

Η μέθοδος `_smart_load` (πίνακας 6.29) από γραμμή 328 μέχρι γραμμή 336 προσπαθεί να φορτώσει μια συγκεκριμένη ρύθμιση για το μοντέλο με βάση το `task` που εκτελεί το μοντέλο. Αν δεν βρει την ρύθμιση για το συγκεκριμένο `task`, προκαλεί εξαίρεση `NotImplementedError`.

Η μέθοδος αναζητά το key στο task_map για το τρέχον task. Αν δεν το βρει, εμφανίζει ένα προειδοποιητικό μήνυμα που εξηγεί την έλλειψη υποστήριξης για το συγκεκριμένο task. Με αυτή τη μέθοδο, ο κώδικας διασφαλίζει ότι φορτώνει τις κατάλληλες ρυθμίσεις για κάθε task, εμφανίζοντας κατάλληλα μηνύματα για μη υποστηριζόμενες λειτουργίες.

```
337 @property
338 def task_map(self) -> dict:
339     raise NotImplementedError("Please provide task map for your
model!")
```

Πίνακας 6.30: Ανάλυση κώδικα *ultralitics* τμήμα 30

Η ιδιότητα task_map (πίνακας 6.30) ξεκινώντας από τη γραμμή 337 μέχρι τη γραμμή 339 ορίζει έναν βασικό χάρτη των διαθέσιμων tasks για το μοντέλο. Η ιδιότητα προκαλεί εξαίρεση NotImplementedError, υποδεικνύοντας ότι θα πρέπει να οριστεί από τον προγραμματιστή. Το μήνυμα καλεί τον χρήστη να παρέχει έναν χάρτη με τα διαθέσιμα tasks για το μοντέλο.

Αυτή η ιδιότητα αποτελεί υπενθύμιση για τον προγραμματιστή να ορίσει συγκεκριμένα tasks που θα υποστηρίζονται από το μοντέλο, προσδίδοντας ευελιξία στη χρήση του.

Στη συνέχεια υπάρχει το training. Για να γίνει training (εικόνα 6.16) με τα δεδομένα που δημιουργήθηκαν από τον συγγραφέα αυτής της εργασίας, θα πρέπει να ακολουθηθούν μερικά βήματα. Αρχικά θα πρέπει στο client να έχει εγκατασταθεί η Python, οι βιβλιοθήκες της *ultralitics* και η PyTorch CUDA για να μπορέσει να χρησιμοποιηθεί η κάρτα γραφικών ώστε να εκπαιδευτεί το σύστημα. Έχοντας δημιουργήσει τα απαραίτητα δεδομένα πληκτρολογείται στο client (πίνακας 31) γραμμή 1:

```
1 yolo task=segment mode=train data=datasetdata.yaml model=yolov8m-seg.pt
epochs=100 batch=6 imgsz=640 amp=False
```

Πίνακας 6.31: Εντολή για την εκκίνηση του training

Automatic Mixed Precision (AMP) κατά την εκπαίδευση. Όλες οι πράξεις θα γίνουν με ακρίβεια FP32 (32-bit floating point). Η συγκεκριμένη παράμετρος χρησιμοποιείται συγκεκριμένα στο σύστημα που πραγματοποιήθηκε η εκπαίδευση καθώς εάν `amp=True` το σύστημα δε θα κάνει την εκπαίδευση σωστά. Όταν θα λάβει τέλος η εκπαίδευση θα δημιουργηθεί ένας φάκελος με δυο αρχεία. Χρησιμοποιείται το αρχείο που ονομάζεται `best.pt`. Αυτό είναι το αρχείο με την μεγαλύτερη ακρίβεια.

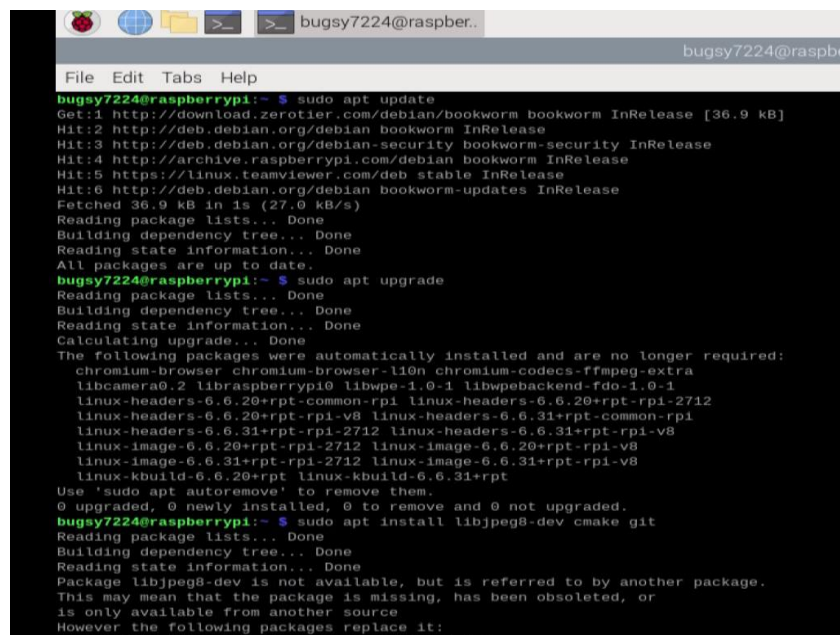
6.6 Σύνδεση του Raspberry Pi και εγκατάσταση των απαραίτητων εφαρμογών και βιβλιοθηκών που χρειαζόμαστε.

Αφού έχει ολοκληρωθεί η εκπαίδευση του AI με όσο τον δυνατόν πιο ακριβή αποτελέσματα, χρειάζεται ένα εργαλείο για να γίνεται η χρήση του προγράμματος σε οποιοδήποτε σημείο χωρίς να είναι αναγκαίο να υπάρχει μια ολόκληρη μονάδα υπολογιστή. Η επίτευξη αυτή επιτυγχάνεται με το Raspberry Pi. Αρχικά πρέπει να εγκατασταθεί το MJPEG Stream, αυτό πραγματοποιείται με την είσοδο στο client του Raspberry Pi 4 χρησιμοποιώντας τις εντολές (πίνακας 6.32) από γραμμή 1 μέχρι γραμμή 4:

```
1 sudo apt update
2 sudo apt upgrade
3 sudo apt install libjpeg8-dev cmake git
4 git clone https://github.com/jacksonliam/mjpg-streamer.git
```

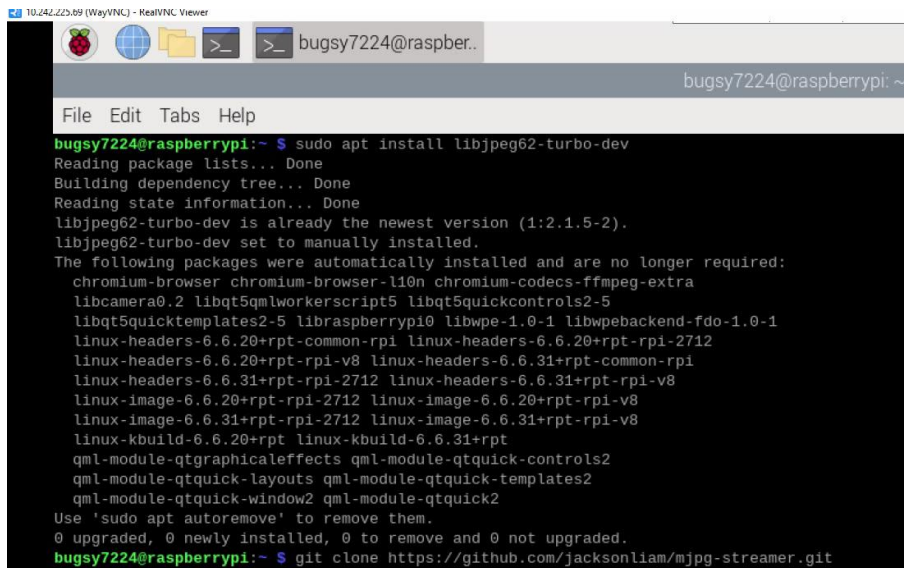
Πίνακας 6.32: Εγκατάσταση MJPEG Stream

που φαίνονται στις εικόνες 6.17 και 6.18.



```
bugsy7224@raspberrypi:~$ sudo apt update
Get:1 http://download.zerotier.com/debian/bookworm bookworm InRelease [36.9 kB]
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:4 http://archive.raspberrypi.com/debian bookworm InRelease
Hit:5 https://linux.teamviewer.com/deb stable InRelease
Hit:6 http://deb.debian.org/debian bookworm-updates InRelease
Fetched 36.9 kB in 1s (27.0 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
bugsy7224@raspberrypi:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra
libcamera0.2 libraspberrypi0 libwpe-1.0-1 libwpebackend-fdo-1.0-1
linux-headers-6.6.20+rpt-common-rpi linux-headers-6.6.20+rpt-rpi-2712
linux-headers-6.6.20+rpt-rpi-v8 linux-headers-6.6.31+rpt-common-rpi
linux-headers-6.6.31+rpt-rpi-2712 linux-headers-6.6.31+rpt-rpi-v8
linux-image-6.6.20+rpt-rpi-2712 linux-image-6.6.20+rpt-rpi-v8
linux-image-6.6.31+rpt-rpi-2712 linux-image-6.6.31+rpt-rpi-v8
linux-kbuild-6.6.20+rpt linux-kbuild-6.6.31+rpt
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
bugsy7224@raspberrypi:~$ sudo apt install libjpeg8-dev cmake git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package libjpeg8-dev is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
libjpeg8-dev-raspbian libjpeg8-dev
```

Εικόνα 6.17: Εντολές για εγκατάσταση MJPEG Stream



```
bugsy7224@raspberrypi: ~  
File Edit Tabs Help  
bugsy7224@raspberrypi:~$ sudo apt install libjpeg62-turbo-dev  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
libjpeg62-turbo-dev is already the newest version (1:2.1.5-2).  
libjpeg62-turbo-dev set to manually installed.  
The following packages were automatically installed and are no longer required:  
chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra  
libcamera0.2 libqt5qmlworkerscript5 libqt5quickcontrols2-5  
libqt5quicktemplates2-5 libraspberrypi0 libwpe-1.0-1 libwpebackend-fdo-1.0-1  
linux-headers-6.6.20+rpt-common-rpi linux-headers-6.6.20+rpt-rpi-2712  
linux-headers-6.6.20+rpt-rpi-v8 linux-headers-6.6.31+rpt-common-rpi  
linux-headers-6.6.31+rpt-rpi-2712 linux-headers-6.6.31+rpt-rpi-v8  
linux-image-6.6.20+rpt-rpi-2712 linux-image-6.6.20+rpt-rpi-v8  
linux-image-6.6.31+rpt-rpi-2712 linux-image-6.6.31+rpt-rpi-v8  
linux-kbuild-6.6.20+rpt linux-kbuild-6.6.31+rpt  
qml-module-qtgraphicaleffects qml-module-qtquick-controls2  
qml-module-qtquick-layouts qml-module-qtquick-templates2  
qml-module-qtquick-window2 qml-module-qtquick2  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
bugsy7224@raspberrypi:~$ git clone https://github.com/jacksonliam/mjpg-streamer.git
```

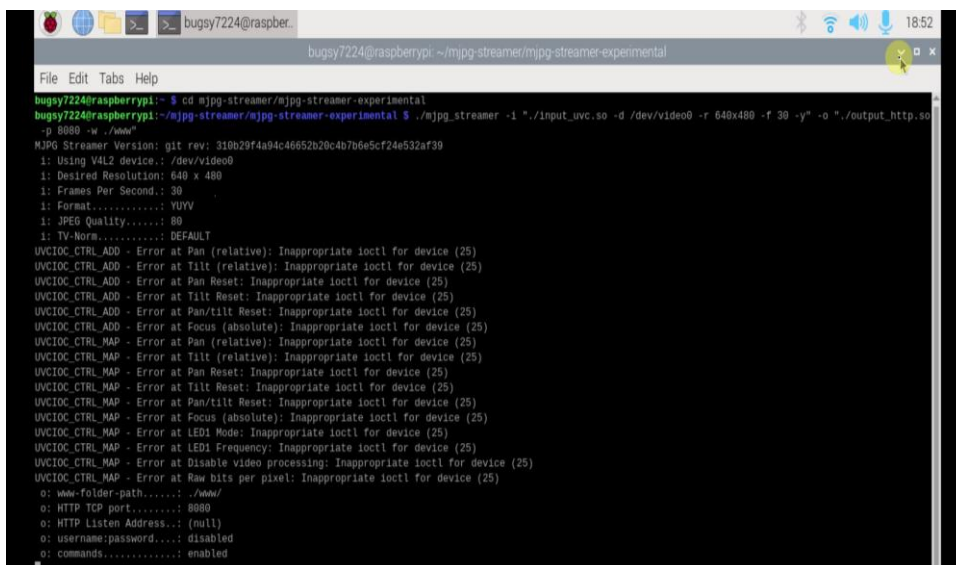
Εικόνα 6.18: Συνέχεια εντολών για εγκατάσταση MJPEG Stream

Όταν πραγματοποιηθεί η εγκατάσταση, πληκτρολογούνται οι εντολές στο client για να ενεργοποιηθεί το MJPEG Stream. Οι εντολές είναι (πίνακας 6.33) από γραμμή 5 μέχρι γραμμή 6:

```
5 cd mjpg-streamer/mjpg-streamer-experimental  
6 ./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -r 640x480 -f 30 -y" -  
o "./output_http.so -p 8080 -w ./www"
```

Πίνακας 6.33: Εκκίνηση MJPEG Stream

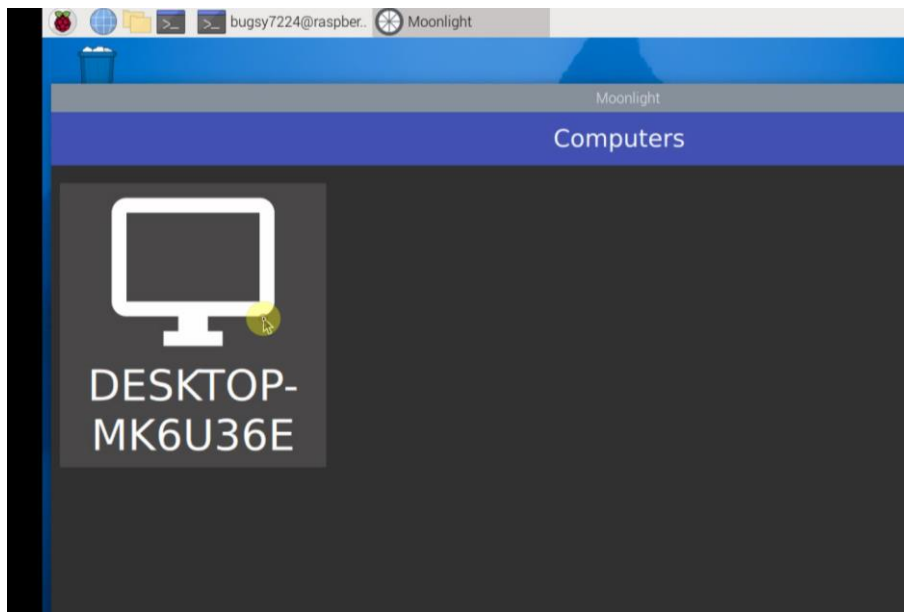
Η διαδικασία αποτυπώνεται στην εικόνα 6.19



```
bugsy7224@raspberrypi: ~/mjpg-streamer/mjpg-streamer-experimental  
File Edit Tabs Help  
bugsy7224@raspberrypi:~$ cd mjpg-streamer/mjpg-streamer-experimental  
bugsy7224@raspberrypi:~/mjpg-streamer/mjpg-streamer-experimental$ ./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -r 640x480 -f 30 -y" -  
-p 8080 -w ./www"  
MJPEG Streamer Version: git rev: 316b29f4a94c46852b20c4b7b6e5cf24e532af39  
i: Using V4L2 device.: /dev/video0  
i: Desired Resolution: 640 x 480  
i: Frames Per Second.: 30  
i: Format.....: YUYV  
i: JPEG Quality.....: 80  
i: TV Norm.....: DEFAULT  
UVCIOC_CTRL_ADD - Error at Pan (relative): Inappropriate ioctl for device (25)  
UVCIOC_CTRL_ADD - Error at Tilt (relative): Inappropriate ioctl for device (25)  
UVCIOC_CTRL_ADD - Error at Pan Reset: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_ADD - Error at Tilt Reset: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_ADD - Error at Pan/tilt Reset: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_ADD - Error at Focus (absolute): Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Pan (relative): Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Tilt (relative): Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Pan Reset: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Tilt Reset: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Pan/tilt Reset: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Focus (absolute): Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at LED1 Mode: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at LED1 Frequency: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Disable video processing: Inappropriate ioctl for device (25)  
UVCIOC_CTRL_MAP - Error at Raw bits per pixel: Inappropriate ioctl for device (25)  
o: www-folder-path.....: ./www/  
o: HTTP TCP port.....: 8080  
o: HTTP Listen Address.: (null)  
o: username:password....: disabled  
o: commands.....: enabled
```

Εικόνα 6.19: Χρήση της εφαρμογής MJPEG Stream

Καθώς είναι ενεργοποιημένο το MJPEG Stream, το IP έχει παρατεθεί στον κώδικα της Python με αποτέλεσμα να εκπέμπει από την εκάστοτε τοποθεσία που βρίσκεται το Raspberry Pi 4. Στη συνέχεια πρέπει να προβάλλεται από το Raspberry Pi 4 η διαδικασία κατά την εκτέλεση του βασικού προγράμματος Python από την μονάδα. Επειδή χρειάζεται το Raspberry Pi 4 να είναι όσο το δυνατόν λιγότερο επιβαρυνμένο από προγράμματα, αλλά και να υπάρχει η μικρότερη δυνατή καθυστέρηση χρησιμοποιείται η εφαρμογή Moonlight (εικόνα 6.20).



Εικόνα 6.20: Επιλογή μονάδας υπολογιστή από το Moonlight

6.7 Επεξήγηση του κώδικα Python και τη χρήση του.

Ο κώδικας συντάχθηκε ώστε να εξυπηρετεί τους παρακάτω σκοπούς. Αρχικά η `opencv` δεν αναγνωρίζει ελληνικούς χαρακτήρες. Γι' αυτό τον λόγο τοποθετήθηκε ελληνικό font στο πρόγραμμα. Επίσης, πολλά γράμματα είναι παρόμοια με τις λέξεις. Εξαιτίας αυτού χρησιμοποιούνται οι περιοχές `roi` για να είναι δυνατόν ο διαχωρισμός και να μην υπάρχουν λάθη. Επιπλέον, δημιουργείται ένα δεύτερο `display` που εκεί εμφανίζονται τα γράμματα και οι λέξεις των χειρομορφών. Τέλος, υπάρχει η παράμετρος να αναγνωρίζει ένα γράμμα ή μια λέξη μια φορά μέσα σε δέκα δευτερόλεπτα καθώς διαφορετικά όσο γινόταν η χειρομορφή θα συνέχιζε να αποτυπώνεται το συγκεκριμένο γράμμα. Με το κουμπί `ESC` πραγματοποιείται ο τερματισμός του προγράμματος. Παρακάτω δίνεται η εξήγησή του κώδικα:

```
1 import cv2
2 import pandas as pad
3 import numpy as num
4 import time
5 from PIL import Image, ImageDraw, ImageFont
```

```
6 from ultralytics import YOLOdetected_characters = {}
```

Πίνακας 6.34: Ανάλυση κώδικα Python τμήμα 1

Ο παραπάνω κώδικας (πίνακας 6.34) ξεκινώντας από τη γραμμή 1 μέχρι τη γραμμή 6 εισάγει τις απαραίτητες βιβλιοθήκες και εργαλεία για την ανίχνευση αντικειμένων, την ανάλυση εικόνων και δεδομένων, και τη διαχείριση της εκτέλεσης του προγράμματος.

cv2: Χρησιμοποιείται για επεξεργασία εικόνας και βίντεο με το OpenCV, ειδικά για πρόσβαση στην κάμερα και σχεδίαση πάνω σε εικόνες.

pandas (ως pad): Βοηθά στη διαχείριση και ανάλυση δεδομένων σε δομές τύπου DataFrame, διευκολύνοντας την επεξεργασία αποτελεσμάτων ανίχνευσης.

numpy (ως num): Παρέχει δυνατότητες αριθμητικών υπολογισμών και επεξεργασίας πολυδιάστατων πινάκων, που είναι χρήσιμοι στην ανάλυση εικόνας.

time: Χρησιμοποιείται για τη διαχείριση του χρόνου (timestamp) των ανιχνεύσεων, αποθηκεύοντας τη χρονική στιγμή κάθε ανίχνευσης.

PII: Προσφέρει εργαλεία για τη δημιουργία και επεξεργασία εικόνων, ιδιαίτερα χρήσιμη για την προσθήκη κειμένου πάνω σε εικόνες.

YOLO από ultralytics: Εισάγει το μοντέλο YOLO, το οποίο χρησιμοποιείται για ανίχνευση αντικειμένων σε εικόνες και βίντεο.

Με αυτές τις εισαγωγές, ο κώδικας είναι έτοιμος να εκτελέσει την ανίχνευση αντικειμένων, επεξεργασία και ανάλυση των αποτελεσμάτων, καθώς και την προβολή πληροφοριών πάνω στις εικόνες.

```
7 model = YOLO('best1.pt')
8 with open("asl_classes.txt", "r", encoding="utf-8") as my_file:
9     data = my_file.read()
10 class_list = data.split("\n")
```

Πίνακας 6.35: Ανάλυση κώδικα Python τμήμα 2

Αυτό το κομμάτι κώδικα (πίνακας 6.35) από γραμμή 7 μέχρι γραμμή 10 φορτώνει το μοντέλο YOLO χρησιμοποιώντας το προεκπαιδευμένο αρχείο best1.pt. Το μοντέλο είναι έτοιμο να χρησιμοποιηθεί για ανίχνευση χειρομορφών. Επίσης, ανοίγει το αρχείο asl_classes.txt, το οποίο περιέχει τις κατηγορίες (labels) που μπορεί να ανιχνεύσει το μοντέλο, και αποθηκεύει κάθε κατηγορία στη λίστα class_list.

Ακόμη παρέχει εύκολη πρόσβαση στις κατηγορίες ανίχνευσης, διευκολύνοντας την αναγνώριση αντικειμένων κατά την πρόβλεψη.

```
11 characters_in_roi = ['A', 'B', 'Γ', 'Δ', 'E', 'Z', 'H', 'Θ', 'I', 'K',
                       'Λ', 'M', 'N', 'Ξ', 'O', 'Π', 'P', 'Σ', 'T', 'Y', 'Φ', 'Ψ', 'Ω', 'X']
```

```
12 characters_out_roi = ['Ελικόπτερο', 'Αεροπλανο', 'Αεροστατο',  
    'Αυτοκίνητο']
```

Πίνακας 6.36: Ανάλυση κώδικα Python τμήμα 3

Αυτό το τμήμα κώδικα (πίνακας 6.36) αρχίζοντας από τη γραμμή 11 μέχρι τη γραμμή 12 ορίζει δύο λίστες που περιέχουν τις κατηγορίες χαρακτήρων για ανίχνευση:

characters_in_roi: Χαρακτήρες που ανιχνεύονται εντός της περιοχής ενδιαφέροντος roi.

characters_out_roi: Κατηγορίες αντικειμένων που ανιχνεύονται εκτός του roi.

Οι λίστες αυτές βοηθούν στην οργάνωση των αντικειμένων που το σύστημα μπορεί να ανιχνεύσει και στην ταξινόμησή τους ανάλογα με τη θέση τους στο frame.

```
13 cap = cv2.VideoCapture('http://10.242.225.69:8080/?action=stream')  
14 count = 0  
15 area = [(0, 150), (500, 150), (500, 780), (0, 780)]
```

Πίνακας 6.37: Ανάλυση κώδικα Python τμήμα 4

Αυτό το τμήμα (πίνακας 6.37) ξεκινώντας από τη γραμμή 13 μέχρι τη γραμμή 15 συνδέεται με την κάμερα μέσω URL, χρησιμοποιώντας το OpenCV, και ορίζει μια περιοχή ενδιαφέροντος roi.

cap: Χρησιμοποιείται για να λάβει τα frame από την απομακρυσμένη κάμερα που λειτουργεί σε ροή (streaming).

count: Μετρητής για τον αριθμό των frame που επεξεργάζεται.

area: Ορίζει τις συντεταγμένες του πολυγώνου που χρησιμοποιείται ως roi για την ανίχνευση των αντικειμένων.

Με αυτό το τμήμα, ο κώδικας αποκτά πρόσβαση σε συνεχή ροή από την κάμερα και ορίζει το μέρος του frame όπου θέλει να ανιχνεύει αντικείμενα.

```
16 font_path = "C:\\Users\\giann\\nohmatikh\\GentiumBookPlus-Bold.ttf"  
17 font = ImageFont.truetype(font_path, 24)
```

Πίνακας 6.38: Ανάλυση κώδικα Python τμήμα 5

Αυτό το κομμάτι κώδικα (πίνακας 6.38) από γραμμή 1 μέχρι γραμμή 4 φορτώνει μια γραμματοσειρά από το σύστημα, η οποία θα χρησιμοποιηθεί για την εμφάνιση ελληνικών χαρακτήρων πάνω στην εικόνα.

font_path: Η διαδρομή προς το αρχείο γραμματοσειράς GentiumBookPlus-Bold.ttf.

font: Η γραμματοσειρά που φορτώνεται για χρήση στην εμφάνιση κειμένου.

Με τη φόρτωση της γραμματοσειράς, ο κώδικας μπορεί να εμφανίσει ελληνικούς χαρακτήρες πάνω στις εικόνες ανίχνευσης.

```
18 def draw_text(image, text, position, font, color):  
19     pil_image = Image.fromarray(image)
```



```
20 draw = ImageDraw.Draw(pil_image)
21 draw.text(position, text, fill=color, font=font)
22 return num.array(pil_image)
```

Πίνακας 6.39: Ανάλυση κώδικα Python τμήμα 6

Η συνάρτηση `draw_text` (πίνακας 6.39) αρχίζοντας από τη γραμμή 18 μέχρι τη γραμμή 22 προσθέτει κείμενο σε μια εικόνα χρησιμοποιώντας τη βιβλιοθήκη PIL. Μετατρέπει την εικόνα από `numpy array` σε αντικείμενο PIL. Σχεδιάζει το κείμενο στη θέση `position`, με το χρώμα `color` και την επιλεγμένη `font`. Ακόμη επιστρέφει την ενημερωμένη εικόνα ως `numpy array`.

Με αυτή τη συνάρτηση, ο κώδικας μπορεί να προσθέσει προσαρμοσμένο κείμενο στις εικόνες ανίχνευσης με τη γραμματοσειρά που έχει φορτωθεί.

```
23 detected_characters = {}
```

Πίνακας 6.40: Ανάλυση κώδικα Python τμήμα 7

Αυτή η γραμμή του κώδικα (πίνακας 6.40) γραμμή 23 αρχικοποιεί ένα κενό λεξικό, το οποίο θα χρησιμοποιηθεί για την αποθήκευση των χαρακτήρων που έχουν ήδη ανιχνευθεί. Στη συνέχεια αποθηκεύει κάθε ανιχνευμένο χαρακτήρα ως κλειδί και τον χρόνο ανίχνευσής του ως τιμή, για την αποφυγή επαναλαμβανόμενων ανιχνεύσεων.

Με το `detected_characters`, ο κώδικας μπορεί να ελέγχει ποιοι χαρακτήρες έχουν ήδη ανιχνευτεί, αποτρέποντας την πολλαπλή εμφάνιση του ίδιου χαρακτήρα σε σύντομο χρονικό διάστημα.

```
24 while True:
25     ret, frame = cap.read()
26     if not ret:
27         break
28     count += 1
29     if count % 3 != 0:
30         continue
31     frame = cv2.resize(frame, (1920, 1080))
```

Πίνακας 6.41: Ανάλυση κώδικα Python τμήμα 8

Αυτό το τμήμα κώδικα (πίνακας 6.41) ξεκινώντας από τη γραμμή 24 μέχρι τη γραμμή 31 ξεκινά το loop για την επεξεργασία κάθε `frame` που λαμβάνεται από την κάμερα. Το `ret, frame = cap.read()` διαβάζει το επόμενο `frame` από την κάμερα. Αν δεν υπάρχει άλλο `frame`, το loop σταματά.

Χρησιμοποιεί μόνο κάθε τρίτο `frame`, παραλείποντας τα υπόλοιπα για να μειώσει την επεξεργασία και αλλάζει το μέγεθος του `frame` σε 1920x1080.

Με αυτόν τον τρόπο το loop επεξεργάζεται μόνο επιλεγμένα `frames`, εξασφαλίζοντας αποδοτική χρήση των πόρων.

```

32 results = model.predict(frame, conf=0.80)
33 a = results[0].boxes.data.cpu().numpy()
34 px = pad.DataFrame(a).astype("float")
35 current_text = ""

```

Πίνακας 6.42: Ανάλυση κώδικα Python τμήμα 9

Αυτό το τμήμα του κώδικα (πίνακας 6.42) αρχίζοντας από τη γραμμή 32 μέχρι γραμμή 35 εκτελεί ανίχνευση αντικειμένων στο τρέχον καρέ χρησιμοποιώντας το YOLO.

results = model.predict(frame, conf=0.80): Εκτελεί την ανίχνευση αντικειμένων στο καρέ με όριο εμπιστοσύνης 0.80.

Μετατροπή αποτελεσμάτων: Μετατρέπει τα αποτελέσματα ανίχνευσης σε πίνακα numpy και μετά σε DataFrame της pandas, για ευκολότερη ανάλυση.

Αρχικοποίηση current_text: Κρατά το τρέχον κείμενο ανίχνευσης για χρήση στην εμφάνιση.

Με αυτό το τμήμα, ο κώδικας εξάγει και οργανώνει τα δεδομένα ανίχνευσης για περαιτέρω επεξεργασία.

```

36 for result in results:
37     for box in result.boxes.data:
38         x1, y1, x2, y2 = box[0], box[1], box[2], box[3]
39         c = class_list[int(box[5])]
40         result = cv2.pointPolygonTest(num.array(area, num.int32),
(int(x1), int(y2)), False)
41         if result >= 0 and any(char in c for char in
characters_in_roi):
42             for char in characters_in_roi:
43                 if char in c and (char not in detected_characters or
time.time() - detected_characters[char] >= 10):
44                     cv2.rectangle(frame, (int(x1), int(y1)), (int(x2),
int(y2)), (0, 0, 255), 2)
45                     cv2.circle(frame, (int(x1), int(y2)), 4, (255, 0,
0), -1)
46                     frame = draw_text(frame, char, (int(x1), int(y1) -
10), font, (0, 0, 255))
47                     current_text += char
48                     detected_characters[char] = time.time()

```

Πίνακας 6.43: Ανάλυση κώδικα Python τμήμα 10

Αυτό το τμήμα (πίνακας 6.43) αρχίζοντας από τη γραμμή 36 μέχρι γραμμή 48 επεξεργάζεται τα ανιχνευμένα αντικείμενα που βρίσκονται εντός της περιοχής ενδιαφέροντος roi. Επίσης, εξάγει τις συντεταγμένες του ανιχνευμένου αντικειμένου και την κατηγορία του. επιπλέον χρησιμοποιεί την pointPolygonTest για να ελέγξει αν το αντικείμενο βρίσκεται εντός του roi. Ακόμη προσθέτει τη χειρομορφή μόνο αν δεν έχει ήδη ανιχνευθεί πρόσφατα. Έτσι σχεδιάζει

το πλαίσιο και το κείμενο γύρω από την ανιχνευμένη χειρομορφή και ενημερώνει το `current_text`.

Με αυτό τον τρόπο, ο κώδικας επιλέγει χειρομορφές εντός του `roi` και τα εμφανίζει στην εικόνα με οπτικά χαρακτηριστικά.

```
49 for result in results:
50     for box in result.boxes.data:
51         x1, y1, x2, y2 = box[0], box[1], box[2], box[3]
52         c = class_list[int(box[5])]
53         result = cv2.pointPolygonTest(num.array(area, num.int32),
(int(x1), int(y2)), False)
54         if result < 0 and any(char in c for char in
characters_out_roi):
55             for char in characters_out_roi:
56                 if char in c and (char not in detected_characters or
time.time() - detected_characters[char] >= 10):
57                     cv2.rectangle(frame, (int(x1), int(y1)), (int(x2),
int(y2)), (0, 0, 255), 2)
58                     cv2.circle(frame, (int(x1), int(y2)), 4, (255, 0,
0), -1)
59                     frame = draw_text(frame, char, (int(x1), int(y1) -
10), font, (0, 0, 255))
60                     current_text += char
61                     detected_characters[char] = time.time()
```

Πίνακας 6.44: Ανάλυση κώδικα Python τμήμα 11

Αυτό το τμήμα (πίνακας 6.44) ξεκινώντας από τη γραμμή 49 μέχρι τη γραμμή 61 επεξεργάζεται τα ανιχνευμένα αντικείμενα που βρίσκονται εκτός του `roi`.

Ελέγχει αν η ανιχνεύσιμη χειρομορφή βρίσκεται εκτός του `roi` και αν ανήκει στην `characters_out_roi`. Επιπροσθέτως σχεδιάζει πλαίσιο και εμφανίζει το κείμενο της ανιχνευμένης χειρομορφής. Επιπλέον, ελέγχει αν η χειρομορφή έχει ήδη ανιχνευθεί πρόσφατα για να αποφεύγονται διπλές εγγραφές.

Με αυτόν τον τρόπο, ο κώδικας διαχειρίζεται αντικείμενα που βρίσκονται εκτός του `roi` και τα εμφανίζει κατάλληλα στην εικόνα.

```
62 frame = draw_text(frame, current_text, (10, 30), font, (255, 255,
255))
63 display_window = num.ones((300, 500, 3), dtype=num.uint8) * 255
64 y_offset = 30
65 for char, det_time in detected_characters.items():
66     if time.time() - det_time <= 10:
67         display_window = draw_text(display_window, char, (10,
y_offset), font, (0, 0, 0))
68         y_offset += 30
69 cv2.imshow("labels", display_window)
70 cv2.polylines(frame, [num.array(area, num.int32)], True, (255, 0,
255), 2)
```

```
71 cv2.imshow("greek_asl", frame)
```

Πίνακας 6.45: Ανάλυση κώδικα Python τμήμα 12

Αυτό το τμήμα (πίνακας 6.45) από γραμμή 62 μέχρι γραμμή 71 εμφανίζει το τρέχον κείμενο ανιχνεύσεων και τη σχεδιασμένη περιοχή ενδιαφέροντος roi πάνω στην εικόνα. Το `current_text` προστίθεται στην πάνω αριστερή γωνία του `frame`, ακόμη αρχικοποιεί ένα παράθυρο για να εμφανίσει τις ανιχνευμένες χειρομορφές που έχουν ανιχνευθεί πρόσφατα. Επίσης, εμφανίζει το κυριο `display (greek_asl)` και τα ανιχνευμένα `labels`.

Με αυτό το τμήμα, ο κώδικας παρέχει μια οπτική αναπαράσταση της περιοχής ενδιαφέροντος `roi` και των πρόσφατων ανιχνεύσεων των χειρομορφών.

```
72 if cv2.waitKey(1) & 0xFF == 27:  
73     break  
74 cap.release()  
75 cv2.destroyAllWindows()
```

Πίνακας 6.46: Ανάλυση κώδικα Python τμήμα 13

Αυτό το τελευταίο τμήμα (πίνακας 6.46) από γραμμή 72 μέχρι γραμμή 75 διαχειρίζεται την έξοδο και καθαρίζει τους πόρους του προγράμματος. Αν ο χρήστης πατήσει το πλήκτρο ESC, το `loop` σταματά. `akomh` απελευθερώνει την κάμερα (`cap.release()`) και κλείνει όλα τα παράθυρα (`cv2.destroyAllWindows()`).

Αυτό το τμήμα εξασφαλίζει ότι το πρόγραμμα τερματίζεται σωστά και όλοι οι πόροι καθαρίζονται, αποτρέποντας προβλήματα στην επόμενη εκτέλεση.

6.8 Δημιουργία του συστήματος και αποτύπωση παραδειγμάτων

Για να κατασκευαστεί το σύστημα χρησιμοποιείται η εφαρμογή της `miniconda` για την δημιουργία κατάλληλου περιβάλλοντος για την εκτέλεσή των απαραίτητων εντολών. Με την εντολή `'cd nohmatikh'` δημιουργείται το μονοπάτι μέχρι τον φάκελο `nohmatikh` που εμπεριέχονται οι φωτογραφίες με τα `labels`, μοιρασμένα σε δυο φακέλους με ονόματα `test` και `train` αντίστοιχα. Εν συνεχεία γίνεται χρήση της εντολής `'conda create --name ptyxiakh python=3.12'` για την δημιουργία του περιβάλλοντος που προαναφέρθηκε. Ακολούθως με την εντολή `'conda activate ptyxiakh'` ενεργοποιείται το περιβάλλον. Έπειτα με τις εντολές `'pip'` και `'conda'` εγκαθιστώνται οι βιβλιοθήκες που χρειαζόμαστε οι οποίες είναι κατά σειρά `'pip install ultralytics'`, `'conda install -c conda-forge opencv'`, `'conda install pandas'`, `'conda install numpy'`, `'conda install pillow'`. Επίσης, για να προστεθεί η βιβλιοθήκη της `PyTorch CUDA` για να γίνει το `training` του κώδικα AI με την χρήση της κάρτας γραφικών `Nvidia 1660 super 6gb` για να έχουμε γρηγορότερη εκπαίδευση χωρίς να καταρρεύσει το σύστημά

Εάν δεν γινόταν χρήση της PyTorch τότε η εκπαίδευση θα γινόταν με την CPU. Το αποτέλεσμα θα ήταν να διακοπεί η λειτουργία του συστήματος καθώς ο επεξεργαστής θα δούλευε στο 100% και θα ενεργοποιούταν το θερμικό. Στη συνέχεια γίνεται η δημιουργία ενός αρχείου YAML (εικόνα 6.22) με το όνομα datasetdata. Το περιεχόμενο του είναι τα paths προς τους φακέλους test και train. Το nc σημαίνει number class και είναι ο αριθμός των κλάσεων που επέρχονται στη συνέχεια. Δηλαδή τοποθετείται ο αριθμός των λέξεων και των γραμμάτων, στην περίπτωση αυτή 24 γράμματα + 4 λέξεις = 28 nc.

```
train: C:\Users\giann\nohmatikh\train
val: C:\Users\giann\nohmatikh\test

nc: 28

names: ["A", "B", "Γ", "Δ", "Ε", "Ζ", "Η", "Θ", "Ι", "Κ", "Λ", "Μ", "Ν", "Ξ", "Ο", "Π",
"Ρ", "Σ", "Τ", "Υ", "Φ", "Ψ", "Ω", "Χ", "Αεροπλανο", "Αεροστατο", "Αυτοκινητο", "Ελικοπτερο"]
```

Εικόνα 6.22: Το περιεχόμενο του YAML file

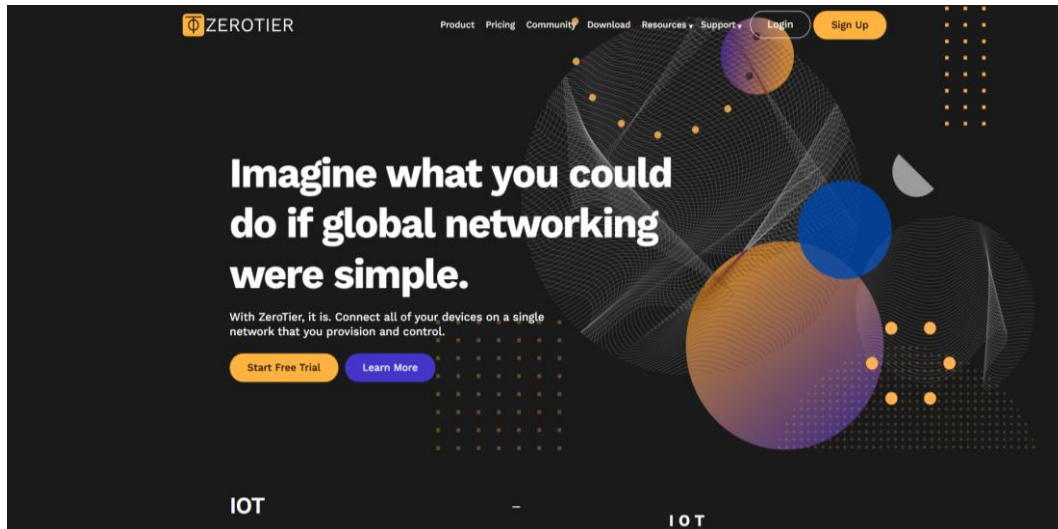
Όταν έχουν δημιουργηθεί τα απαραίτητα αρχεία τότε μπορεί να πραγματοποιηθεί η εκπαίδευση του AI με την εντολή στο client του miniconda 'yolo task=segment mode=train data=datasetdata.yaml model=yolon8m-seg.pt epochs=100 batch=6 imgsz=640 amp=False'. Η διαδικασία της εκπαίδευσης εμφανίζεται λεπτομερώς στην εικόνα 6.16 και επέρχεται ο τερματισμός του όταν φτάσει στο εκατοστό epoch. Πρέπει να σημειωθεί ότι η διαδικασία της εκπαίδευσης μπορεί να διαρκέσει για μεγάλο χρονικό διάστημα. Ανάλογα με το πλήθος των δεδομένων που χρησιμοποιούν. Για την συγκεκριμένη εργασία που έγινε χρήση 1419 φωτογραφιών η διάρκεια της εκπαίδευσης ήταν δυο μέρες. Όταν τελειώσει η εκπαίδευση στον φάκελο της nohmatikh δημιουργείται ένας άλλος φάκελος με ονομασία runs. Στον φάκελο runs υπάρχει ο υποφάκελος segment. Μέσα στον segment είναι ο φάκελος train οπύ περιέχονται δυο αρχεία τα οποία είναι best.pt και last.pt. Συγκεκριμένα το best.pt θα χρησιμοποιηθεί στον κωδικά Python καθώς έχει την μεγαλύτερη ακρίβεια από όλες τις άλλες εκπαιδεύσεις που πραγματοποιήθηκαν. Πλέον με το AI έτοιμο γίνεται η σύνταξη του κώδικα για να εκτελεστεί το αρχείο. Η σύνταξη του κώδικα χρειάστηκε αρκετές παραμέτρους. Αρχικά για να πραγματοποιείται η φόρτωση των ετικετών με τα γράμματα και τις λέξεις δημιουργήθηκε ένας φάκελος TXT με το όνομα asl_classes.txt όπου αποτυπώνονται μέσα τα γράμματα και οι λέξεις με την δομή του να φαίνεται στην εικόνα 6.23.

1	Α
2	Β
3	Γ
4	Δ
5	Ε
6	Ζ
7	Η
8	Θ
9	Ι
10	Κ
11	Λ
12	Μ
13	Ν
14	Ξ
15	Ο
16	Π
17	Ρ
18	Σ
19	Τ
20	Υ
21	Φ
22	Ψ
23	Ω
24	Χ
25	Αεροπλano
26	Αεροστατο
27	Αυτοκινητο
28	Ελικοπτερο

Εικόνα 6.23: Δομή TXT αρχείο

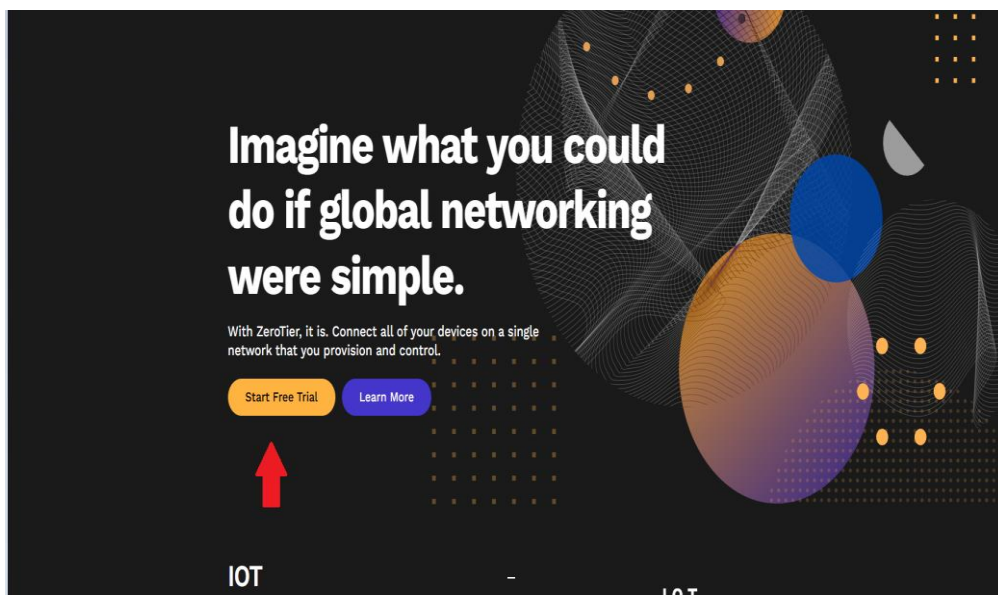
Κατόπιν γνωρίζοντας ότι το ΑΙ που εκπαιδεύτηκε παρερμηνεύει μια χειρομορφή όταν είναι στατική με την χειρομορφή που είναι στην κίνηση, για τον διαχωρισμό τους δημιουργήθηκε η περιοχή roi. Μέσα στη συγκεκριμένη περιοχή αναγνωρίζονται μόνο τα γράμματα. Αντίθετα, έξω από αυτή αναγνωρίζει μόνο τις λέξεις το ΑΙ. Για να γίνει καλύτερη επεξήγηση για την παρερμηνεία που αναφέρθηκε παραπάνω, η συγκεκριμένη παράμετρος δημιουργήθηκε για να μη συγχέει το ΑΙ κάποιες χειρομορφές γραμμάτων που είναι ίδιες με την χειρομορφή της λέξεις που πραγματοποιείται με την κίνηση του χεριού. Ένα παράδειγμα είναι η χειρομορφή με το γράμμα Υ. Όταν είναι στατική η χειρομορφή η ερμηνεία είναι το Υ. Ενώ όταν η χειρομορφή γίνεται με κίνηση είναι η λέξη αεροπλano. Στη συνέχεια για να γίνεται αναγνώριση των ελληνικών γραμμάτων έπρεπε να χρησιμοποιηθεί ελληνικό font καθώς η βιβλιοθήκη της opencv δεν μπορεί να αναγνωρίσει ελληνικούς χαρακτήρες. Εφόσον το σύστημα εκτελείται πρέπει να αναδύεται ένα δεύτερο display το οποίο απεικονίζει τα γράμματα και τις λέξεις που ανιχνεύονται. Ταυτόχρονα δημιουργείται η παράμετρος όπου: όταν ένα γράμμα εντοπίζεται εμφανίζεται στο δεύτερο display για δέκα δευτερόλεπτα και δεν μπορεί να ξανά εντοπιστεί και να γραφτεί. Η συγκεκριμένη παράμετρος αποσκοπεί να μην ανιχνεύεται και γράφεται ένα γράμμα ή μια λέξη στο display πολλές φορές. Εάν δεν υπήρχε αυτή η παράμετρος με τον εντοπισμό της χειρομορφής θα γραφόταν το ίδιο πράγμα όσες φορές μπορεί να εντοπιστεί μέχρι να γίνει άλλη χειρομορφή. Τέλος, με το κουμπί ESC πραγματοποιείται ο τερματισμός του προγράμματος. Με την ολοκλήρωση του προγράμματος χρησιμοποιείται ένα Raspberry Pi 4 για να μπορεί να εφαρμοστεί το πρόγραμμα σε οποίο σημείο υπάρχει παροχή ίντερνετ χωρίς να χρειάζεται ένας φορητός υπολογιστής. Οπότε το Raspberry Pi 4 αξιοποιείται σαν αναμεταδότης. Στον κώδικα έχει τοποθετηθεί η IP του

Raspberry Pi 4 αλλά για να είναι μια συγκεκριμένη και να μην αλλάζει κάθε φορά που το Raspberry Pi 4 συνδέεται σε διαφορετικό δίκτυο χρησιμοποιείται το Zerotier. Το Zerotier επιτρέπει δωρεάν την δημιουργία ιδιωτικού δικτύου μεταξύ την μονάδας και του Raspberry Pi με συγκεκριμένο IP. Με αυτόν τον τρόπο όποιο δίκτυο και να χρησιμοποιηθεί θα λειτουργήσει το πρόγραμμα. Για να δημιουργηθεί το ιδιωτικό δίκτυο πρέπει να πληκτρολογηθεί στη μηχανή αναζήτησης στο διαδίκτυο η λέξη Zerotier για να γίνει μετάβαση στην ιστοσελίδα του (εικόνα 6.24).



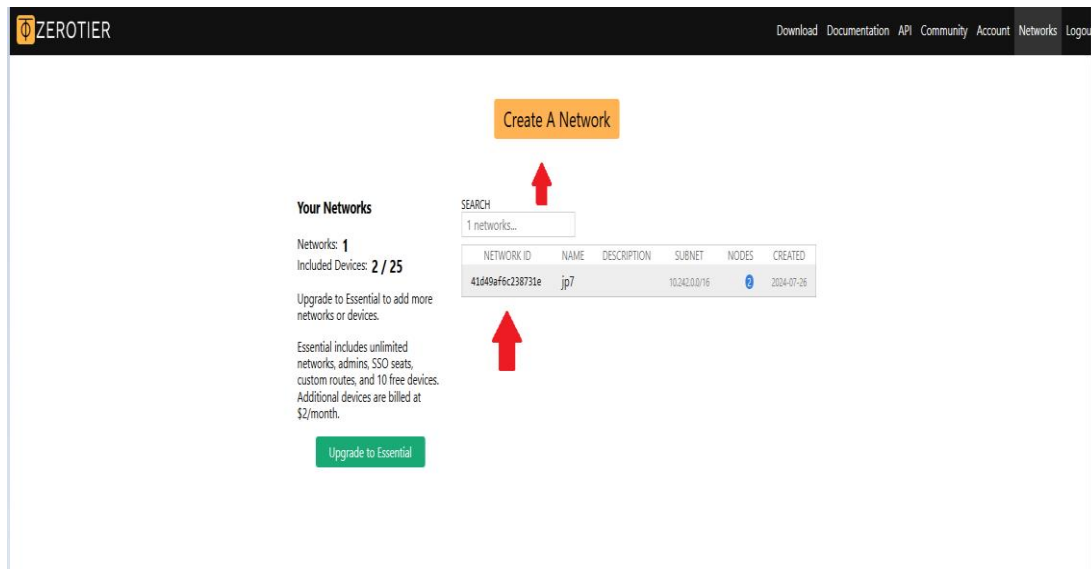
Εικόνα 6.24: Ιστοσελίδα Zerotier

Στη συνέχεια γίνεται επιλογή το start free trial (εικόνα 6.25).



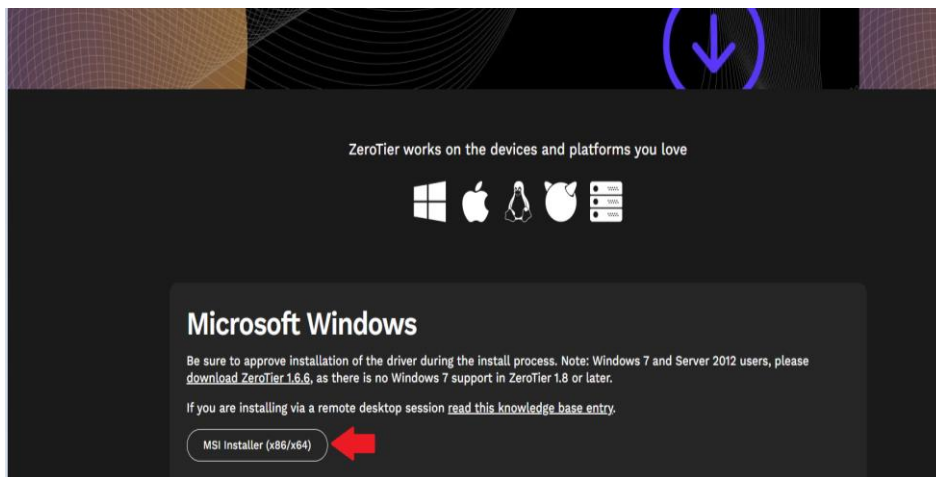
Εικόνα 6.25: Επιλογή του start free trial

Με την επιλογή αυτή γίνεται ανακατεύθυνση για sign up με το email. Με την εισαγωγή του email γίνεται μεταφορά στη σελίδα για την δημιουργία του network (εικόνα 6. 26).



Εικόνα 6.26: Δημιουργία ιδιωτικού δικτύου.

Πατώντας το create a network δημιουργείται το δίκτυο το οποίο σημαδεύεται από το δεύτερο βέλος. Εν συνέχεια με την επιλογή download γίνεται η λήψη της εφαρμογής του Zerotier για την μονάδα του υπολογιστή (εικόνα 6.27).



Εικόνα 6.27: Λήψη της εφαρμογής για υπολογιστή

Λήψη του Zerotier πραγματοποιείται και στο Raspberry Pi 4 μέσω των παρακάτω εντολών εικόνα 6.28.

```

bugsy7224@raspberrypi:~$ sudo apt update
Hit:1 http://deb.debian.org/debian bookworm InRelease
Hit:2 http://deb.debian.org/debian-security bookworm-security InRelease
Hit:3 http://archive.raspberrypi.com/debian bookworm InRelease
Hit:4 http://deb.debian.org/debian bookworm-updates InRelease
Get:5 http://download.zerotier.com/debian/bookworm bookworm InRelease [36.9 kB]
Hit:6 https://linux.teamviewer.com/deb stable InRelease
Fetched 36.9 kB in 1s (27.2 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
bugsy7224@raspberrypi:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
chromium-browser chromium-browser-l10n chromium-codecs-ffmpeg-extra libcamera0.2 libqt5qmlworkerscript5 libqt5quickcontrols2-5 libqt5quicktemplates2-5
libraspberrypi0 libwpe-1.0-1 libwpebackend-fdo-1.0-1 linux-headers-6.6.20+rpt-common-rpi linux-headers-6.6.20+rpt-rpi-2712 linux-headers-6.6.20+rpt-rpi-v8
linux-headers-6.6.31+rpt-common-rpi linux-headers-6.6.31+rpt-rpi-2712 linux-headers-6.6.31+rpt-rpi-v8 linux-image-6.6.20+rpt-rpi-2712
linux-image-6.6.20+rpt-rpi-v8 linux-image-6.6.31+rpt-rpi-2712 linux-image-6.6.31+rpt-rpi-v8 linux-kbuild-6.6.20+rpt linux-kbuild-6.6.31+rpt
qml-module-qtgraphicaleffects qml-module-qtquick-controls2 qml-module-qtquick-layouts qml-module-qtquick-templates2 qml-module-qtquick-window2
qml-module-qtquick2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
bugsy7224@raspberrypi:~$ curl -s https://install.zerotier.com | sudo bash

```

Εικόνα 6.28: Λήψη του Zerotier στο Raspberry Pi 4

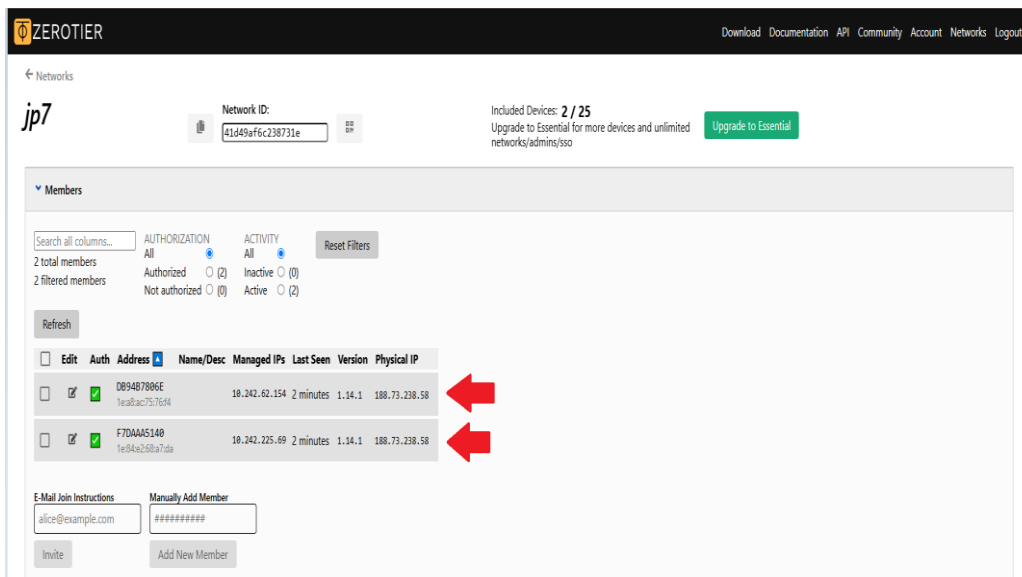
Για να συνδεθεί το Raspberry Pi 4 στο network πληκτρολογείται η εντολή (πίνακας 6.47):

```
1 sudo zerotier-cli join 41d49af6c238731e
```

Πίνακας 6.47: Σύνδεση στο network

Το 41d49af6c238731e είναι το network id.

Οι λήψεις αυτές γίνονται καθώς στο ιδιωτικό δίκτυο θα έχουν συγκεκριμένες IP και ο υπολογιστής και το Raspberry Pi 4. Στην εικόνα 6.29 τα δυο φαίνονται τα IP.



Εικόνα 6.29: Προβολή των IP

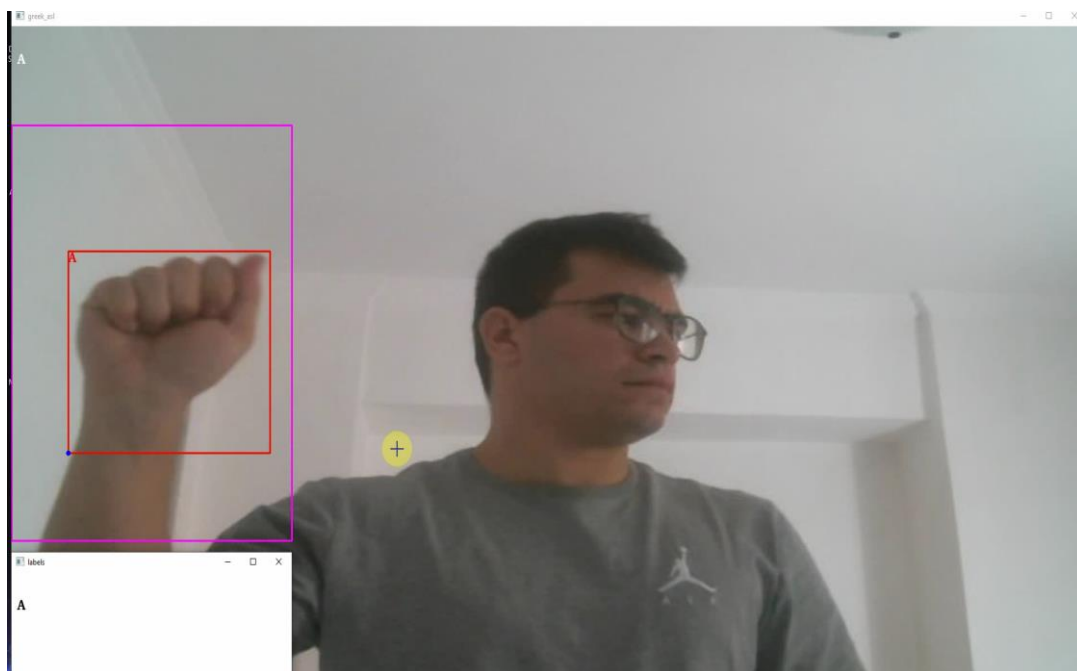
Για να μπορέσει να γίνει μετάδοση από το Raspberry Pi γίνεται χρήση των εντολών του MJPEG Stream στο client οι οποίες είναι :

```
1 sudo zerotier-cli join 41d49af6c238731ecd mjpg-streamer/mjpg-streamer-experimental
```

```
2 ./mjpg_streamer -i "./input_uvc.so -d /dev/video0 -r 640x480 -f 30 -y"  
-o "./output_http.so -p 8080 -w ./www"
```

Πίνακας 6.48: Χρήση του MJPEG Stream στο Raspberry pi 4

Με την σύνδεση της κάμερας με το πρόγραμμα είναι αναγκαία η χρήση μιας οθόνης για να προβάλλεται ότι εμφανίζεται στον υπολογιστή. Η χρήση της εφαρμογής moonlight μέσω του Raspberry Pi προβάλλει την οθόνη. Πρέπει να ειπωθεί ότι δεν γίνεται η εκτέλεση του προγράμματος στο Raspberry Pi καθώς είναι πολύ βαρύ και κολλάει βλέποντας το πρόγραμμα σε frames με πολλά ms. Στις παρακάτω εικόνες (εικόνα 6.30, εικόνα 6.31, εικόνα 6.32, εικόνα 6.33, εικόνα 6.34, εικόνα 6.35) εμφανίζονται παραδείγματα με επεξηγήσεις για να κατανοηθεί πλήρως η λειτουργία του συστήματος.



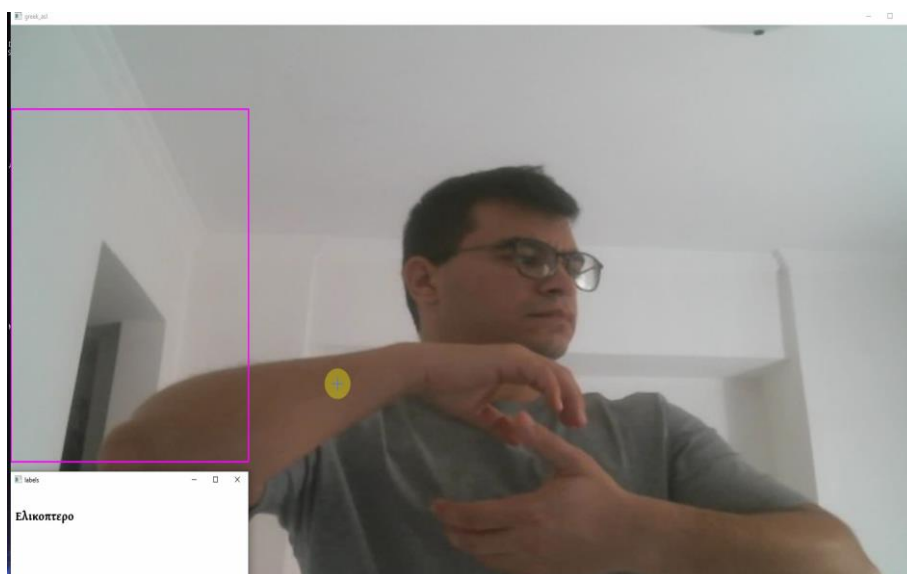
Εικόνα 6.30: Παράδειγμα με χειρομορφή για το γράμμα A εντός roi

Στην την εικόνα 6. 30 παρατηρείται η χειρομορφή του γράμματος A. Αντίθετα, στην εικόνα 6. 31 προβάλλεται η ίδια χειρομορφή αλλά το σύστημα δεν εμφανίζει το νόημα της που είναι το γράμμα A. Η διάφορα αυτή έγκειται στις περιοχές roi που έχει γίνει αναφορά. Όταν η χειρομορφή που το νόημα της είναι γράμμα βρίσκεται εντός roi τότε πραγματοποιείται ανίχνευση. Ενώ όταν είναι εκτός περιοχής roi τότε δεν γίνεται καμία ανίχνευση.

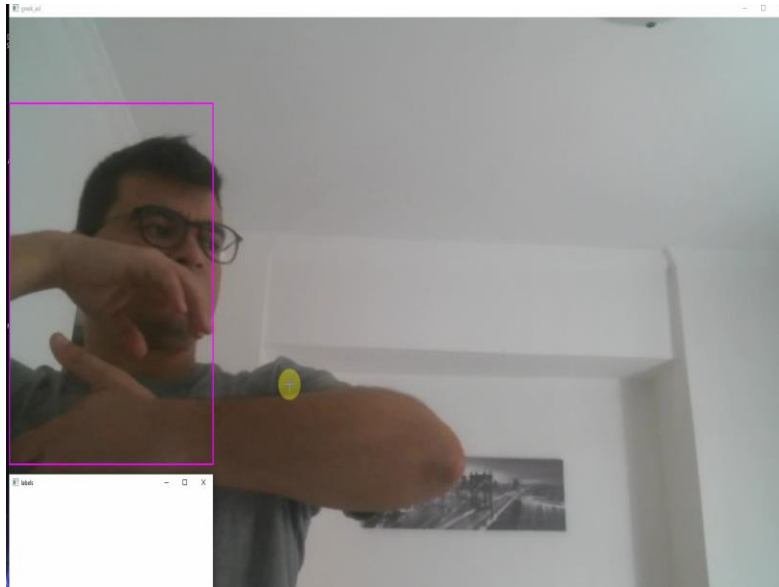


Εικόνα 6.31: Παράδειγμα με χειρομορφή για το γράμμα Α εκτός roi

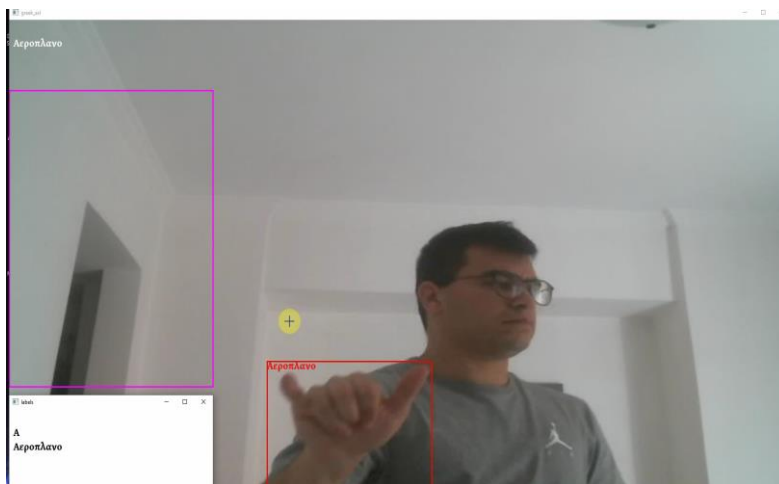
Το ίδιο συμβαίνει και με τις λέξεις. Εάν η χειρομορφή της λέξης εντοπιστεί εκτός roi τότε γίνεται η ανίχνευση της, εικόνα 6.32. Διαφορετικά εάν η χειρομορφή εμφανιστεί εντός roi δεν υπάρχει η ανίχνευση της από το σύστημα εικόνα 6.33. Ο λόγος της δημιουργίας των περιοχών roi πραγματοποιήθηκε για την παράμετρο που απεικονίζεται στην εικόνα 6.34 και εικόνα 6.35. Η χειρομορφή του γράμματος Υ και της λέξης αεροπλάνο είναι ίδιες με την διαφορά σε πραγματικό χρόνο να είναι ότι η χειρομορφή της λέξης κάνει μια κίνηση στον χώρο. Αυτό μπορεί να το καταλάβει το ανθρώπινο μάτι. Ωστόσο, το σύστημα και ειδικότερα το ΑΙ πρόγραμμα δεν μπορεί να κατανοήσει την διάφορα. Το αποτέλεσμα στην εικόνα 6.34 που ζητείται να αποτυπώνεται είναι η λέξη αεροπλάνο. Ενώ στην εικόνα 6.35 επιδιώκεται το γράμμα Υ να αποτυπώνεται και αυτό χωρίς να υπάρχει η παραμικρή παρερμηνεία.



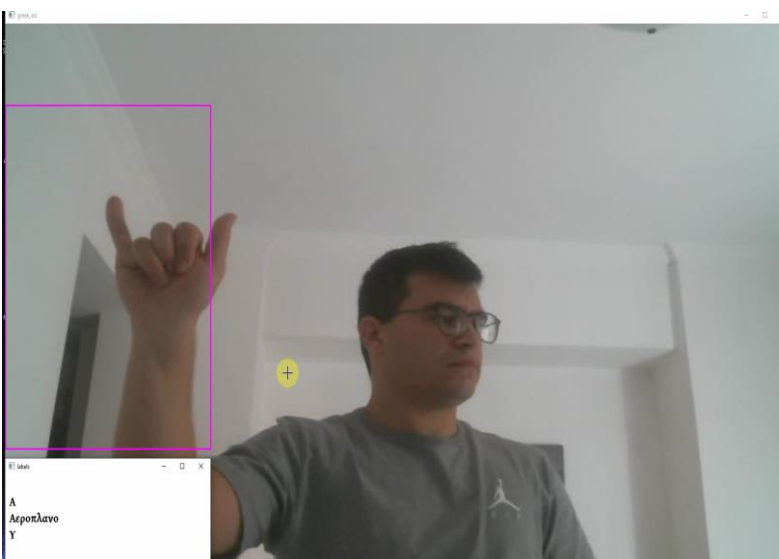
Εικόνα 6.32: Παράδειγμα με χειρομορφή για τη λέξη ελικοπτερο εκτός roi



Εικόνα 6.33: Παράδειγμα με χειρομορφή για τη λέξη ελικόπτερο εντός roi



Εικόνα 6.34: Παράδειγμα με χειρομορφή για τη λέξη αεροπλάνο εκτός roi



Εικόνα 6.35: Παράδειγμα με χειρομορφή για το γράμμα Υ εντός roi

6.9 Αποτύπωση ποσοστού επιτυχίας συστήματος

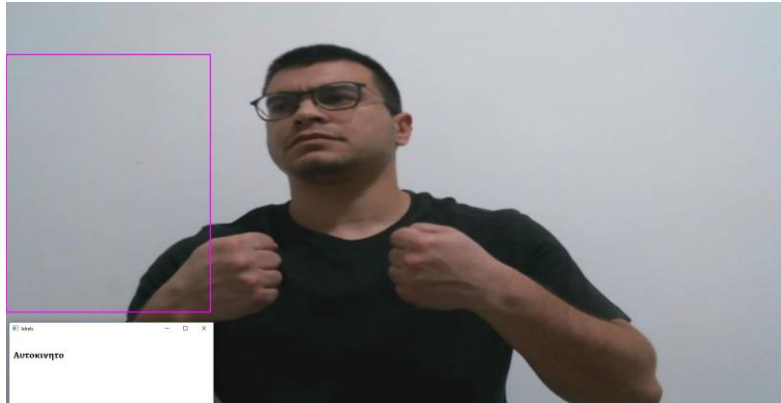
Στον πίνακα που ακολουθεί, αποτυπώνεται το ποσοστό επιτυχίας κάθε χειρομορφής. Δηλαδή κάθε ενός γράμματος του ελληνικού αλφαβήτου και των πέντε λέξεων που έχουν εισαχθεί. Τα πειραματικά αποτελέσματα ελήφθησαν σε ξεχωριστό φόντο και ξεχωριστό φωτισμό με δέκα φορές η κάθε μια χειρομορφή. Ήτοι δέκα φορές η κάθε μια χειρομορφή σε λευκό φόντο με καλό φωτισμό (εικόνα 6.36), δέκα φορές σε μαύρο φόντο με καλό φωτισμό (εικόνα 6.37), δέκα φορές σε με λευκό φόντο με χαμηλό φωτισμό (εικόνα 6.38) και δέκα φορές σε με μαύρο φόντο με χαμηλό φωτισμό (εικόνα 6.39) . Τα αποτελέσματα παρατίθενται στον πίνακα 6.49

Γράμματα/ Λέξεις	Ποσοστό επιτυχίας με λευκό φόντο με καλό φωτισμό	Ποσοστό επιτυχίας με μαύρο φόντο με καλό φωτισμό	Ποσοστό επιτυχίας με λευκό φόντο και χαμηλό φωτισμό	Ποσοστό επιτυχίας με μαύρο φόντο και χαμηλό φωτισμό
A	9/10	9/10	9/10	7/10
B	10/10	10/10	10/10	10/10
Γ	10/10	10/10	10/10	10/10
Δ	9/10	9/10	9/10	9/10
E	9/10	9/10	9/10	9/10
Z	10/10	10/10	10/10	10/10
H	10/10	10/10	10/10	10/10
Θ	7/10	8/10	7/10	7/10
I	10/10	10/10	10/10	10/10
K	10/10	10/10	10/10	10/10
Λ	8/10	9/10	8/10	6/10
M	8/10	7/10	7/10	5/10
N	10/10	10/10	8/10	7/10
Ξ	10/10	10/10	10/10	8/10
O	10/10	10/10	10/10	10/10
Π	9/10	10/10	9/10	8/10
P	8/10	10/10	8/10	8/10
Σ	10/10	10/10	10/10	8/10
T	9/10	8/10	9/10	8/10
Υ	7/10	4/10	5/10	3/10
Φ	10/10	10/10	10/10	9/10
X	10/10	10/10	10/10	9/10
Ψ	10/10	10/10	10/10	9/10
Ω	10/10	10/10	10/10	10/10
Αεροπλάνο	10/10	10/10	10/10	10/10
Αερόστατο	10/10	10/10	10/10	10/10
Αυτοκίνητο	10/10	10/10	10/10	10/10
Ελικόπτερο	10/10	10/10	10/10	10/10

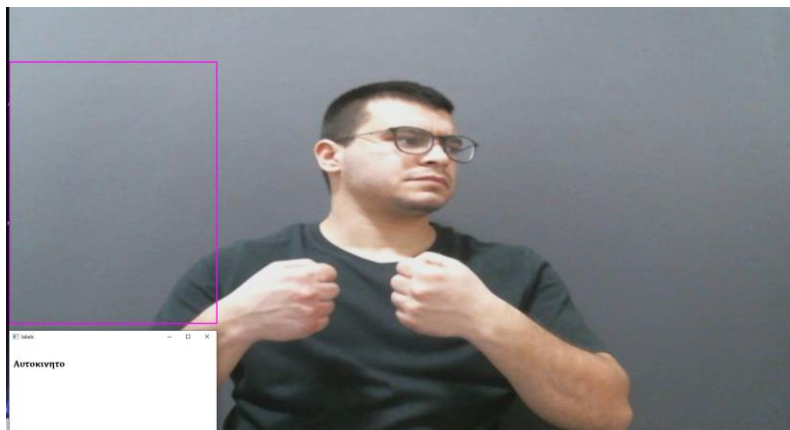
Πίνακας 6.49: Ποσοστό επιτυχίας χειρομορφών

*οι επιτυχημένες φορές της χειρομορφής/στις δέκα φορές που πραγματοποιήθηκαν

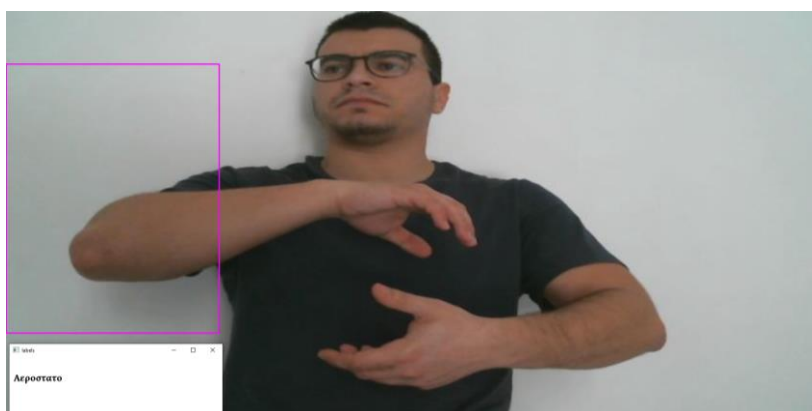
Οι αποτυχημένες προσπάθειες των χειρομορφών μπορεί να προκαλούνται από: τις φωτογραφίες που έχουν εισαχθεί στο σύστημα, την λάθος εκπαίδευση του συστήματος, την αδυναμία σταθερής σύνδεσης δικτύου, την αδυναμία συστημικού προβλήματος που εκτελεί το πρόγραμμα και η ύπαρξη ανθρωπίνου λάθους δηλαδή η λανθασμένη κίνηση για να αποτυπωθεί η χειρομορφή.



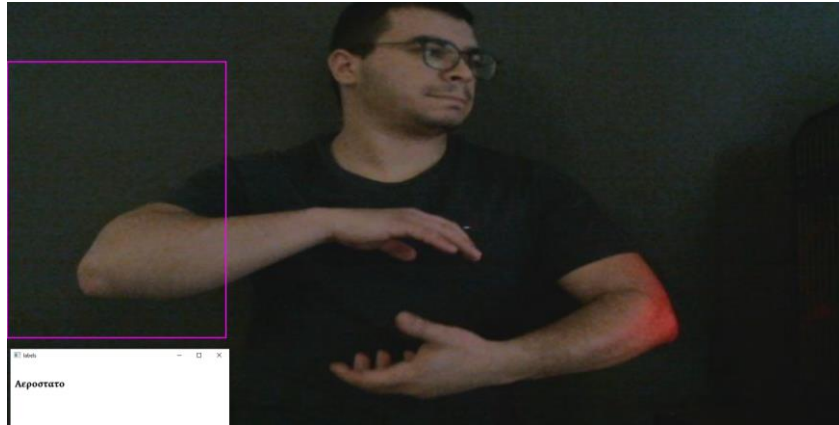
Εικόνα 6.36: Χειρομορφή με λευκό φόντο με καλό φωτισμό



Εικόνα 6.37: Χειρομορφή με μαύρο φόντο με καλό φωτισμό



Εικόνα 6.38: Χειρομορφή με λευκό φόντο με χαμηλό φωτισμό



Εικόνα 6.39: Χειρομορφή με μαύρο φόντο με χαμηλό φωτισμό

7 Επίλογος-Συμπεράσματα.

7.1 Εισαγωγή.

Στην παρούσα πτυχιακή εργασία έγινε η υλοποίηση ενός συστήματος αναγνώρισης της ελληνικής νοηματικής γλώσσας με AI, το οποίο επιτρέπει την αναγνώριση των χειρομορφών σε γράμματα και λέξεις. Μέσω της εκπαίδευσης του προγράμματος AI της ultralytics το YOLOv8 έγινε η επίτευξη αναγνώρισης των χειρομορφών. Επίσης, εξαιτίας της σύστασης του προγράμματος Python που περιέχει το YOLOv8 μπορούν οι χειρομορφές να αναγνωρίζονται ως γράμματα και ως λέξεις ξεχωριστά χωρίς να υπάρχει καμία σύγχυση. Επιπλέον, χρησιμοποιήθηκε η εφαρμογή miniconda για την δημιουργία κατάλληλου περιβάλλοντος για την εκτέλεση των απαραίτητων εργασιών. Επιπρόσθετα έγινε χρήση της εφαρμογής του LabelMe για να πραγματοποιηθεί το annotation των φωτογραφιών και συγκεκριμένα το segmentation. Ακόμη με το Zerotier επετεύχθη η δημιουργία ιδιωτικού δικτύου με καθορισμένες και συγκεκριμένες IP. Παράλληλα χρησιμοποιήθηκε το Raspberry Pi 4 για την χρήση του προγράμματος σε οποιοδήποτε μέρος είναι υπαρκτή η παροχή δικτύου. Ταυτόχρονα για την ρύθμιση του Raspberry Pi 4 ώστε να προβάλλει το πρόγραμμα χρησιμοποιήθηκαν το MJPEG Stream για την σύνδεση της κάμερας του Raspberry Pi 4 με το πρόγραμμα Python. Καθώς και η χρήση της εφαρμογής Moonlight μπορεί να δείχνει την οθόνη της μονάδας υπολογιστή που εκτελείται το πρόγραμμα. Καταλήγοντας από την ενασχόλησή με την πτυχιακή εργασία αποκτήθηκε η κατάλληλη εμπειρία και τεχνογνωσία, για τον σχεδιασμό, την ανάπτυξη και την υλοποίηση ενός συστήματος που αναγνωρίζει την ελληνική νοηματική γλώσσα με ποσοστό επιτυχίας 85-90%.

7.2 Προβλήματα και αντιμετώπιση.

Οι δυσκολίες και τα προβλήματα που παρουσιάστηκαν στην υλοποίηση της πτυχιακής εργασίας ήταν αρκετά. Τα περισσότερα καταπολεμήθηκαν, όπως η δημιουργία δεδομένων και το annotation χρειάστηκαν πολλές ώρες καθώς επεξεργάστηκαν 1419 φωτογραφίες, η δημιουργία του κώδικα χωρίς την κατάρρευση του συστήματος, η σωστή χρήση του Raspberry Pi 4 χωρίς να προκαλείται η διακοπή της λειτουργίας του από την επιβάρυνση των εφαρμογών που χρειάζεται να εκτελέσει και η σωστή εκπαίδευση του YOLOv8 καθώς στις αρχές λόγω ενός προβλήματος στον κώδικα δεν γινόταν σωστά η εκπαίδευση του αλλά γενικά χρειαζόταν τουλάχιστον 2 δυο μέρες για να ολοκληρωθεί η εκπαίδευση εξαιτίας του όγκου των δεδομένων. Ωστόσο, ένα πρόβλημα δεν μπόρεσε να επιλυθεί και αυτό ήταν το συστημικό πρόβλημα που εκτελείται το πρόγραμμα. Ο υπολογιστής που πραγματοποιήθηκε η εκπαίδευση του προγράμματος δεν είναι “δυνατός” για να μπορέσουν να προστεθούν περισσότερα δεδομένα και να έχει μεγαλύτερη ευστοχία το πρόγραμμα.

7.3 Συμπεράσματα και πιθανότητες εξέλιξης.

Μελλοντικά με ένα δυνατότερο σύστημα θα μπορούσε να εισαχθούν ακόμα περισσότερες λέξεις και να υπάρξει μεγαλύτερη ευστοχία. Επιπλέον, η χρήση της εργασίας θα μπορούσε να είναι πολυπρόσωπη με την εφαρμογή της να μπορέσει στο μέλλον να υπάρξει σε καταστήματα για την καλύτερη εξυπηρέτηση των ατόμων που χρησιμοποιούν την ΕΝΓ. Επίσης, η χρήση της μελλοντικά είναι αναγκαία στα κέντρα υγείας για να υπάρξει μια άριστη επικοινωνία μεταξύ του ασθενούς που χρησιμοποιεί την ΕΝΓ και του γιατρού. Τα κέντρα ελληνικής νοηματικής γλώσσας θα μπορούσαν να κάνουν χρήση αυτού του συστήματος μελλοντικά. Με την εισαγωγή όλων των λέξεων θα μπορούσε να είναι εργαλείο για την καλύτερη μάθηση και εξάσκηση της ΕΝΓ. Η δημιουργία εφαρμογής για κινητά θα μπορούσε να είναι χρήσιμη έτσι ώστε να γίνει επικοινωνία μεταξύ ενός ατόμου που δεν γνωρίζει καθόλου την ΕΝΓ με άτομο το οποίο επικοινωνεί με αυτή. Τέλος, η επέκταση του συγκεκριμένου συστήματος όχι μόνο στην προσθήκη όλων των λέξεων αλλά και στη μετάφραση της ομιλίας σε νοηματική μέσω βίντεο ή AI avatar να είναι το καλύτερο μελλοντικό σενάριο που θα μπορούσε να πραγματοποιηθεί.

8 Βιβλιογραφία

8.1 Διαδικτυακή Ελληνική.

Βικιπαίδεια. (n.d.). Άλαν Τούρινγκ. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Άλαν Τούρινγκ - Βικιπαίδεια](#)

Βικιπαίδεια. (n.d.). Ελληνική Νοηματική Γλώσσα. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Ελληνική νοηματική γλώσσα - Βικιπαίδεια](#)

Βικιπαίδεια. (n.d.). Νευρωνικό Δίκτυο. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [el.wikipedia.org/Νευρωνικό δίκτυο - Βικιπαίδεια](http://el.wikipedia.org/Νευρωνικό_δίκτυο)

Γιωτάκης Αριστείδης, (2023, 7 Μαρτίου). Βαρηκοΐα: Ποιες οι αιτίες όταν δεν ακούτε καλά. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Βαρηκοΐα: Ποιες οι αιτίες όταν δεν ακούτε καλά - Γιωτάκης Αριστείδης Χειρουργός Ωτορινολαρυγγολόγος \(ΩΡΛ\)](#)

Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος. (2018, 26 Σεπτεμβρίου). Δακτυλικό Αλφάβητο. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [ΔΑΚΤΥΛΙΚΟ ΑΛΦΑΒΗΤΟ – Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος](#)

Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος. (2018, 9 Οκτωβρίου). Η εξέλιξη της Ελληνικής Νοηματικής Γλώσσας (ΕΝΓ). Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Η εξέλιξη της Ελληνικής Νοηματικής Γλώσσας \(ΕΝΓ\) – Εκπαιδευτικό Κέντρο Νοηματικής Γλώσσας Κρατύλος](#)

Εργασίες Φοιτητών ΕΤΤΑΠ. (n.d.). Ιστορία της Νοηματικής Γλώσσας. Πανεπιστήμιο Αθηνών. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Ιστορία της Ε.Ν.Γ](#)

ΙΑΣΩ. ΚΟΡΡΕΣ ΓΕΩΡΓΙΟΣ ΦΑΙΔΩΝ (2023, 1 Μαρτίου). Πώς γίνεται η διάγνωση της βαρηκοΐας; Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Πώς γίνεται η διάγνωση της βαρηκοΐας;](#)

Κουκουράβα Βαΐα, (2024, 24 Μαρτίου). Βαρηκοΐα: Συμπτώματα, Αιτίες και Διαχείριση. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Βαρηκοΐα: Συμπτώματα, Αιτίες και Διαχείριση - ΚΟΥΚΟΥΡΑΒΑ](#)

Λαμπροπούλου Ρ. (2022, 11 Ιουλίου). Προσβασιμότητα των Ατόμων με Αναπηρία στα Μέσα Μαζικής Ενημέρωσης. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [imerida_amea_mme_r_lambropoulou_11072022.pdf](#)

Νοηματική Κιβωτός Πολυχώρος. (2022, 20 Ιουνίου). Ιστορία Νοηματικής Γλώσσας. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [ΙΣΤΟΡΙΑ ΝΟΗΜΑΤΙΚΗΣ ΓΛΩΣΣΑΣ](#)

Νοηματική Κιβωτός Πολυχώρος. (2023, 10 Νοεμβρίου). Νοηματική γλώσσα: Η γλώσσα της ισότητας και της ένταξης. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Νοηματική γλώσσα: Η γλώσσα της ισότητας και της ένταξης](#)

Σαπκάς, Dimitris Stefanidis (2020, 19 Σεπτεμβρίου). Ιστορία και Εξέλιξη της Τεχνητής Νοημοσύνης. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Ιστορία και Εξέλιξη της Τεχνητής Νοημοσύνης - Artificial Intelligence \(A.I.\) - Professor G. Sarpkas - Ορθοπεδικός Χειρουργός - Καθηγητής Πανεπιστημίου](#)

Σουλαντίκας Κωνσταντίνος, (2023, 27 Νοεμβρίου). Βαρηκοΐα. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Βαρηκοΐα | Τύποι, Αίτια, Συμπτώματα, Θεραπεία](#)

Σύλλογος Ακοοπροθετιστών Ελλάδος. (n.d.). Τύποι Ακουστικών Βαρηκοΐας. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τύποι Ακουστικών Βαρηκοΐας - Σύλλογος Ακοοπροθετιστών Ελλάδος](#)

Big Blue Academy. (2023, 31 Ιανουαρίου). Κίνδυνοι τεχνητής νοημοσύνης: Αντιμετωπίζοντας τις προκλήσεις της AI. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Ποιοι Είναι οι Βασικοί Κίνδυνοι της Τεχνητής Νοημοσύνης;](#)

Big Blue Academy. (2022, 3 Μαΐου). Τεχνητή νοημοσύνη: Θετικά και αρνητικά. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τεχνητή Νοημοσύνη: Τα Θετικά και Αρνητικά του AI](#)

Big Blue Academy. (n.d.). Deep Learning vs Machine Learning: Κατανόηση των διαφορών. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Deep Learning vs Machine Learning: Οι Διαφορές](#)

eTwinning Greece. Γεωργία Βλαμάκη (n.d.). Χρήση της Τεχνολογίας και Τεχνητής Νοημοσύνης στην Εκπαίδευση. Ανακτήθηκε 20 Οκτωβρίου 2024 από: welcome.etwinning.gr/archives/349

Greeco. Greeco Team (2023, 1 Δεκεμβρίου). Τι είναι η Τεχνητή Νοημοσύνη και πως λειτουργεί; Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τι είναι η Τεχνητή Νοημοσύνη και πως λειτουργεί; - Greeco](#)

Greek Tech. (2023, 9 Αυγούστου). Η εξέλιξη της τεχνητής νοημοσύνης: Αναδρομή στην ιστορία της Τεχνητής Νοημοσύνης. Ποιες είναι οι προοπτικές της; Ανακτήθηκε 20 Οκτωβρίου

2024 από: [Η εξέλιξη της τεχνητής νοημοσύνης: Αναδρομή στην ιστορία της Τεχνητής Νοημοσύνης.](#)

GSIS. (n.d.). GSIS Services for Citizens. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Ψηφιακή Προσβασιμότητα | General Secretariat for Information Systems and Digital Governance](#)

In.gr. (2021, 19 Οκτωβρίου). Τεχνητή Νοημοσύνη, Γλωσσική Τεχνολογία και Γλωσσικές Υποδομές. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τεχνητή Νοημοσύνη, Γλωσσική Τεχνολογία και Γλωσσικές υποδομές | in.gr](#)

KENG. (n.d.). Λεξικό Ελληνικής Νοηματικής Γλώσσας. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Λεξικό - Κέντρο Ελληνικής Νοηματικής Γλώσσας](#)

MedLabNews.Κωνσταντίνου Λούβρου, (2014, 4 Μαρτίου). Ποιες οι σημαντικότερες αιτίες κώφωσης; Πώς να φροντίσετε τα αυτιά σας για να αποφύγετε την απώλεια ακοής. Ανακτήθηκε 20 Οκτωβρίου 2024 από:[Ποιες οι σημαντικότερες αιτίες κώφωσης; Πώς να φροντίσετε τα αυτιά σας για να αποφύγετε την απώλεια ακοής; | MEDLABNEWS.GR / IATRICA NEA](#)

Newsit. (2023, 12 Αυγούστου). Η τεχνητή νοημοσύνη στην καθημερινότητά μας: Οι τομείς όπου το AI αλλάζει τα δεδομένα. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Η τεχνητή νοημοσύνη στην καθημερινότητά μας - Οι τομείς όπου το AI αλλάζει τα δεδομένα](#)

Open Standards. Γιάννης Παπαβασιλείου (2020, 2 Δεκεμβρίου). Γιατί η προσβασιμότητα στο διαδίκτυο είναι τόσο σημαντική και τι μπορείς να κάνεις για να την επιτύχεις; Open Standards. Ανακτήθηκε 20 Οκτωβρίου 2024 από: <https://openstandards.ellak.gr/2020/12/02/giati-i-prosvasimotita-sto-diadiktio-ine-toso-simantiki-ke-ti-mporis-na-kanis-gia-na-tin-epitichis/>

PC Steps. Γιάννης Φυτίλης (2018, 11 Απριλίου). Τι Είναι το Machine Learning Και Πώς Έφερε Επανάσταση στο AI. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τι Είναι το Machine Learning Και Πώς Έφερε Επανάσταση στο AI | PCsteps.gr](#)

PCLT. (2024, 3 Φεβρουαρίου.). Τι είναι το νευρωνικό δίκτυο. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τι είναι το νευρωνικό δίκτυο :: PcLT](#)

Pride.gr. (2024, 1 Ιουλίου). 10 πράγματα που πρέπει να ξέρεις για τη νοηματική γλώσσα. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Νοηματική Γλώσσα: Δέκα πράγματα που πρέπει να ξέρεις | Pride.gr](#)

SAP. (n.d.). Τι είναι το Machine Learning; Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Μηχανική Εκπαίδευση | Ορισμός, τύποι και παραδείγματα | SAP](#)

SecDigital. (2023, Μάιος). Accessibility Guide. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Οδηγός Προσβασιμότητας](#)

Techblog. Λορένα Σιούτη (2014, 8 Ιουνίου.). MotionSavvy: A Tablet App That Reads Sign Language. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [MotionSavvy, το app για tablet που διαβάζει την νοηματική | Techblog.gr](#)

Tecnobits. Σεμπάστιαν Βιδάλ (2023, 9 Ιουλίου). Τι είναι η βαθιά μάθηση και πώς μπορεί να εφαρμοστεί; Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Τι είναι το Deep Learning και πώς μπορεί να εφαρμοστεί; ▷ ↩](#)

8.2 Διαδικτυακή Ξενόγλωσση

AirGPU. (n.d.). Parsec vs Moonlight: Game Streaming Comparison. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Parsec vs Moonlight: Which Tool Is Better for Remote Gaming? - airgpu](#)

Anaconda. (n.d.). Miniconda Documentation. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Miniconda — Anaconda documentation](#)

Conda. (n.d.). Conda Documentation. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Conda Documentation — conda-docs documentation](#)

Data School. (2024, 7 January). Conda vs Anaconda vs Miniconda: Understanding the Differences. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [What are conda, Anaconda, and Miniconda? 🐼](#)

Day Interpreting. Seldean Smith (2023, 3 November). The Future of Sign Language Interpretation: Technology and Accessibility. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Future of Sign Language Interpretation - Day Interpreting Blog](#)

Dev Kit (2024, January 28). YOLOv8 Segmentation: A Comprehensive Guide. Dev Kit. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [YOLOv8 Segmentation: A Comprehensive Guide](#)

DNMTechs. Auden Lily (2022, 9 March). Anaconda vs Miniconda: A Comparison in Python 3 Programming. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Anaconda vs. Miniconda: A](#)

[Comparison in Python 3 Programming - DNMTEchs - Sharing and Storing Technology Knowledge](#)

DZone. Tomasz Waraksa (2021, 9 September). Creating Secure Private Networks with ZeroTier VPN. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Creating Secure Private Networks With ZeroTier VPN - DZone](#)

Electromaker. Moe Long (2019, 10 September). Best Uses for a Raspberry Pi 4. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Best Uses for a Raspberry Pi 4 - Top Raspberry Pi 4 Applications](#)

Font Library. (n.d.). Gentium Plus. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Gentium Plus](#)

GeeksforGeeks. anshul_aggarwal(2023, 22 September). Introduction to Visual Studio. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Introduction to Visual Studio - GeeksforGeeks](#)

GitHub. (n.d.). iiiypuk/rpi-icon: Raspberry Pi Icon Project. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [GitHub - iiiypuk/rpi-icon: Rasberry Pi multi-res ICO without \(r\) logo](#)

HandTalk. (n.d.). HandTalk: Helping bridge communication for the deaf community. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Hand Talk: your website accessible in ASL](#)

HubSpot. Madhu Murali (2023, 30 January). What is Visual Studio? Ανακτήθηκε 20 Οκτωβρίου 2024 από: [What is Visual Studio? All You Need to Know](#)

Innovatiana. Nanobaly (2024, 3 September). LabelMe: How to Use the Image Annotation Tool. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Master Data Labeling: discover the LabelMe tool](#)

Juan R. Terven Instituto Politecnico Nacional CICATA-Qro &Diana M. Cordova-Esparza Universidad Autónoma de Querétaro Facultad de Informática (2024, 29 February). A Survey of Self-Supervised Learning for Image Generation. arXiv. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [\[2304.00501\] A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS](#)

LabelMe. (n.d.). LabelMe: Image Annotation Tool. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Labelme - AI Image Annotation & Dataset Creation](#)

Lingvano. (n.d.). Learn American Sign Language (ASL) Online. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Learn American Sign Language online - Get started for free!](#)

Matplotlib. (n.d.). Matplotlib: Python plotting library. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Matplotlib — Visualization with Python](#)

McKinney, W. (2022, January 23). Pandas basics. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Python for Data Analysis, 3E - 5 Getting Started with pandas](#)

MMYOLO. (n.d.). YOLOv8 Description. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Algorithm principles and implementation with YOLOv8 — MMYOLO 0.6.0 documentation](#)

Movavi. Gayane Sargsyan (2019, 5 June). MJPEG Codec: What It Is and How It Works. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [What is Motion JPEG codec? | MJPEG Converter by Movavi](#)

Microsoft. (2024, 19 June). Get Started with Visual Studio IDE. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [What is the Visual Studio IDE? | Microsoft Learn](#)

Microsoft. (n.d.). Visual Studio: The Most Powerful IDE for Developers. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Visual Studio: IDE and Code Editor for Software Developers and Teams](#)

Moonlight. (n.d.). Moonlight Streaming: GitHub Repository. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Moonlight Game Streaming Project · GitHub](#)

Moonlight. (n.d.). Moonlight Streaming: Play Your Games from Anywhere. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Moonlight Game Streaming: Play Your PC Games Remotely](#)

Muhammad Yaseen Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan (2024, September 5). Text-to-Image Generation with Visual-Contextual Modulation. arXiv. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [\[2408.15857\] What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector](#)

Numpy Developers. (n.d.). What is NumPy? Ανακτήθηκε 20 Οκτωβρίου 2024 από: [What is NumPy? — NumPy v2.1 Manual](#)

OpenCV Team. (2019. 6 March). About OpenCV. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [About - OpenCV](#)

Pandas Developers. (n.d.). Getting started with Pandas. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Getting started — pandas 2.2.3 documentation](#)

Parker, J. (2023, July 21). Moonlight Gaming: Stream Your Games from Anywhere. Android Authority. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Moonlight gaming tool: What is it and why does every PC gamer need it?](#)

PiCockpit. Adam (2023, 8 September). Everything You Need to Know About the Raspberry Pi 4. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Everything You Need to Know About the Raspberry Pi 4 | PiCockpit](#)

Python Software Foundation. (n.d.). Python. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Welcome to Python.org](#)

Python Software Foundation. (n.d.). Python Community Logos. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [The Python Logo | Python Software Foundation](#)

PyTorch. PyTorch Contributors. (n.d.). PyTorch Documentation: Stable version. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [PyTorch documentation — PyTorch 2.4 documentation](#)

Racheluo (2024, 20 March). Exploring LabelMe: An Open Source Image Annotation Tool. Medium. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Exploring Labelme — An Open Source Image Annotation Tool | by Racheluo | Medium](#)

Raspberry Pi. (n.d.). Raspberry Pi 4 Model B. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Buy a Raspberry Pi 4 Model B – Raspberry Pi](#)

Raspberry Pi. (n.d.). Raspberry Pi 4 Model B Specifications. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Raspberry Pi 4 Model B specifications – Raspberry Pi](#)

Raspberry Pi Foundation. (n.d.). Raspberry Pi: Credit Card Sized Computer. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Teach, learn, and make with the Raspberry Pi Foundation](#)

Raspberry Pi Foundation. (n.d.). Getting Started with Raspberry Pi. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Getting started - Raspberry Pi Documentation](#)

Real Python. (n.d.). Real Python: Python Programming Tutorials. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Python Tutorials – Real Python](#)

SignWriting. (n.d.). SignWriting: Credits. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [SignWriting For Sign Languages](#)

TechCult. Henry Quill (2023, 20 October). Is ZeroTier Safe? Review. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Is ZeroTier Safe? All You Need to Know About It – TechCult](#)

TensorFlow. (n.d.). Learn TensorFlow: Tutorials and resources. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Introduction to TensorFlow](#)

TutorialsPoint. (n.d.). Python Tutorial. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Python Tutorial](#)

Ultralytics. (2015). Ultralytics Documentation: YOLOv8. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Home - Ultralytics YOLO Docs](#)

Ultralytics github. (n.d.). ultralytics/ultralytics: Official YOLOv8 repository. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [GitHub - ultralytics/ultralytics: Ultralytics YOLO11](#) 🚀

Van der Plas, J. (n.d.). Python Data Science Handbook. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Python Data Science Handbook | Python Data Science Handbook](#)

V7 Labs. Hmrishav Bandyopadhyay (2022, 14 October). LabelMe Guide: A Comprehensive Overview for Image Annotation. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Labeling with LabelMe: Step-by-step Guide \[Alternatives + Datasets\]](#)

Waskom, M. (2020). Introduction to Seaborn. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [An introduction to seaborn — seaborn 0.13.2 documentation](#)

Wikipedia. (n.d.). File: Visual Studio Icon 2022.svg. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [File:Visual Studio Icon 2022.svg - Wikipedia](#)

Wikipedia. (n.d.). Motion JPEG. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Motion JPEG - Wikipedia](#)

Wikipedia. (n.d.). Thomas Hopkins Gallaudet. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Thomas Hopkins Gallaudet - Wikipedia](#)

Wikipedia. (n.d.). Visual Studio. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Visual Studio - Wikipedia](#)

Wlink Technologies. (n.d.). About ZeroTier: Simplifying Networking. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [ZeroTier Technology](#)

W3C. (2023, 21 September). Web Content Accessibility Guidelines (WCAG) 2.1. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [Web Content Accessibility Guidelines \(WCAG\) 2.1](#)

YOLOv8. (n.d.). YOLOv8: The state-of-the-art real-time object detection model. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [YOLOv8: A New State-of-the-Art Computer Vision Model](#)

YOLOv8. Jane Torres (2024, March 19). YOLOv8 Architecture Explained. YOLOv8. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [YOLOv8 Architecture Explained: Key Features | YOLOv8](#)

YOLOv8. Jane Torres (2024, March 22). YOLOv8 Improvements. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [YOLOv8 Improvements: Key Architectural Enhancements](#)

ZeroTier. Adam Ierymenko (2023, 10 June). ZeroTier Review: Everything You Need to Know About ZeroTier in 2023. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [ZeroTier Review: Everything You Need to Know About ZeroTier in 2023 - ZeroTier](#)

ZeroTier. (2024, 15 April). ZeroTier: Software-Defined Networking Made Simple. Ανακτήθηκε 20 Οκτωβρίου 2024 από: [ZeroTier | Global Networking Solution for IoT, SD-WAN, and VPN](#)

9 Παράρτημα Α: Κώδικας Python για την ανίχνευση της Ελληνικής Νοηματικής Γλώσσας

```
# Βιβλιοθήκες
import cv2
import pandas as pad
import numpy as num
import time
from PIL import Image, ImageDraw, ImageFont
from ultralytics import YOLO

# Φόρτιση του μοντέλου YOLO
model = YOLO('best1.pt')

# Φόρτιση των ετικετών απο το txt
with open("asl_classes.txt", "r", encoding="utf-8") as my_file:
    data = my_file.read()
class_list = data.split("\n")

# Τα γράμματα και οι λέξεις που ανιχνεύονται
characters_in_roi = ['Α', 'Β', 'Γ', 'Δ', 'Ε', 'Ζ', 'Η', 'Θ', 'Ι', 'Κ',
'Λ', 'Μ', 'Ν', 'Ξ', 'Ο', 'Π', 'Ρ', 'Σ', 'Τ', 'Υ', 'Φ', 'Ψ', 'Ω', 'Χ']
characters_out_roi = ['Ελικόπτερο', 'Αεροπλانو', 'Αεροστατο',
'Αυτοκινητο']

# Η κάμερα που χρησιμοποιείται εξ αποστάσεως με mjpeg online από το
Raspberry Pi
cap = cv2.VideoCapture('http://10.242.225.69:8080/?action=stream')
count = 0
area = [(0, 150), (500, 150), (500, 780), (0, 780)]

# Το fontpath για την εμφάνιση των ελληνικών χαρακτήρων
font_path = "C:\\Users\\giann\\nohmatikh\\GentiumBookPlus-Bold.ttf"
font = ImageFont.truetype(font_path, 24)

# Η λειτουργία για να σημειώνει την ανίχνευση
def draw_text(image, text, position, font, color):
    pil_image = Image.fromarray(image)
    draw = ImageDraw.Draw(pil_image)
    draw.text(position, text, fill=color, font=font)
    return num.array(pil_image)

# Η σημειώσει των ανιχνεύσιμων χαρακτήρων
detected_characters = {}
while True:
    ret, frame = cap.read()
    if not ret:
        break
    count += 1
    if count % 3 != 0:
        continue
    frame = cv2.resize(frame, (1920, 1080))

# Εκτέλεσή του εκπαιδευμένου YOLO
results = model.predict(frame, conf=0.80)
a = results[0].boxes.data.cpu().numpy()
```

```

px = pad.DataFrame(a).astype("float")
current_text = ""

# Η διαδικασία για την ανίχνευση των γραμμάτων που βρίσκονται μέσα στο
roi
for result in results:
    for box in result.boxes.data:
        x1, y1, x2, y2 = box[0], box[1], box[2], box[3]
        c = class_list[int(box[5])]
        result = cv2.pointPolygonTest(num.array(area, num.int32),
(int(x1), int(y2)), False)
        if result >= 0 and any(char in c for char in
characters_in_roi):
            for char in characters_in_roi:
                if char in c and (char not in detected_characters or
time.time() - detected_characters[char] >= 10):
                    cv2.rectangle(frame, (int(x1), int(y1)),
(int(x2), int(y2)), (0, 0, 255), 2)
                    cv2.circle(frame, (int(x1), int(y2)), 4, (255, 0,
0), -1)
                    frame = draw_text(frame, char, (int(x1), int(y1)
- 10), font, (0, 0, 255))
                    current_text += char
                    detected_characters[char] = time.time()

# Η διαδικασία για την ανίχνευση των λέξεων που βρίσκονται έξω στο roi
for result in results:
    for box in result.boxes.data:
        x1, y1, x2, y2 = box[0], box[1], box[2], box[3]
        c = class_list[int(box[5])]
        result = cv2.pointPolygonTest(num.array(area, num.int32),
(int(x1), int(y2)), False)
        if result < 0 and any(char in c for char in
characters_out_roi):
            for char in characters_out_roi:
                if char in c and (char not in detected_characters or
time.time() - detected_characters[char] >= 10):
                    cv2.rectangle(frame, (int(x1), int(y1)),
(int(x2), int(y2)), (0, 0, 255), 2)
                    cv2.circle(frame, (int(x1), int(y2)), 4, (255, 0,
0), -1)
                    frame = draw_text(frame, char, (int(x1), int(y1)
- 10), font, (0, 0, 255))
                    current_text += char
                    detected_characters[char] = time.time()

# Εμφάνισή των ανιχνεύσιμων χαρακτήρων και διακοπή του προγράμματος με το
πλήκτρο ESC
frame = draw_text(frame, current_text, (10, 30), font, (255, 255,
255))
display_window = num.ones((300, 500, 3), dtype=num.uint8) * 255
y_offset = 30
for char, det_time in detected_characters.items():
    if time.time() - det_time <= 10:
        display_window = draw_text(display_window, char, (10,
y_offset), font, (0, 0, 0))
        y_offset += 30
cv2.imshow("labels", display_window)
cv2.polylines(frame, [num.array(area, num.int32)], True, (255, 0,
255), 2)

```

```
cv2.imshow("greek_asl", frame)
if cv2.waitKey(1) & 0xFF == 27:
    break
cap.release()
cv2.destroyAllWindows()
```