ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΗΜΙΟΥΡΓΙΑ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ ΓΙΑ ΒΟΗΘΕΙΑ ΑΤΟΜΩΝ ΜΕ ΠΡΟΒΛΗΜΑΤΑ ΟΡΑΣΗΣ

ΜΠΟΥΡΛΗΣ ΙΩΑΝΝΗΣ-ΝΙΚΗΦΟΡΟΣ
ΔΡΟΣΟΣ ΣΩΤΗΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΚΟΛΟΜΒΑΤΣΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Λαμία Σεπτέμβριος έτος 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΗΜΙΟΥΡΓΙΑ ΕΞΥΠΝΟΥ ΒΟΗΘΟΥ ΓΙΑ ΒΟΗΘΕΙΑ ΑΤΟΜΩΝ ΜΕ ΠΡΟΒΛΗΜΑΤΑ ΟΡΑΣΗΣ

ΜΠΟΥΡΛΗΣ ΙΩΑΝΝΗΣ-ΝΙΚΗΦΟΡΟΣ
ΔΡΟΣΟΣ ΣΩΤΗΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΚΟΛΟΜΒΑΤΣΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Λαμία Σεπτέμβριος έτος 2024

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

DEVELOPMENT OF A SMART ASSISTANT FOR ASSISTING VISUALLY IMPAIRED INDIVIDUALS

MPOURLIS IOANNIS-NIKIFOROS
DROSOS SOTIRIOS

FINAL THESIS

ADVISOR

KONSTANTINOS KOLOMVATSOS
SUBSTITUTE TEACHER

Lamia September year 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: ……/…../20……

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία εξετάζει την ανάπτυξη ενός έξυπνου βοηθού που έχει σχεδιαστεί για να υποστηρίζει άτομα με προβλήματα όρασης στην καθημερινή τους ζωή. Ο βοηθός αξιοποιεί προηγμένες τεχνολογίες όπως η υπολογιστική όραση για την ανίχνευση και αναγνώριση αντικειμένων και εμποδίων στο περιβάλλον, παρέχοντας ηχητική ανατροφοδότηση στον χρήστη σε πραγματικό χρόνο.

Το σύστημα βασίζεται στο μοντέλο YOLOv8 για την ανίχνευση και αναγνώριση αντικειμένων σε πραγματικό χρόνο, ενώ παράλληλα χρησιμοποιεί τεχνικές επεξεργασίας φωνής για την παροχή ηχητικής ανατροφοδότησης στους χρήστες. Ο βοηθός αξιοποιεί την υπολογιστική ισχύ για να υπολογίζει τις αποστάσεις των ανιχνευμένων αντικειμένων χρησιμοποιώντας την τεχνική της τριγωνικής ομοιότητας.

Η ανάπτυξη και η ενσωμάτωση πολλαπλών νημάτων (multi-threading) επιτρέπει την ταυτόχρονη εκτέλεση διαφόρων διεργασιών, όπως η ανίχνευση αντικειμένων και η αναπαραγωγή ήχου, χωρίς να διαταράσσεται η απόδοση του συστήματος.

Μέσω αυτού του έργου, επιδιώκουμε να προσφέρουμε μια πρακτική λύση που να ανταποκρίνεται στις προκλήσεις που αντιμετωπίζουν τα άτομα με προβλήματα όρασης, βελτιώνοντας την ποιότητα ζωής τους.

ABSTRACT

This thesis explores the development of a smart assistant designed to support individuals with visual impairments in their daily lives. The assistant leverages advanced technologies such as computer vision for detecting and recognizing objects and obstacles in the environment, providing real-time auditory feedback to the user.

The system is based on the YOLOv8 model for real-time object detection and recognition, while also utilizing voice processing techniques to deliver auditory feedback to users. The assistant harnesses computational power to calculate the distances of detected objects using the technique of triangular similarity.

The development and integration of multi-threading allows the simultaneous execution of various processes, such as object detection and audio playback, without disrupting the system's performance.

Through this project, we aim to offer a practical solution that addresses the challenges faced by visually impaired individuals, enhancing their quality of life.

Ευχαριστίες

Θα θέλαμε να εκφράσουμε τις ειλικρινείς μας ευχαριστίες σε όλους όσους συνέβαλαν στην ολοκλήρωση της παρούσας πτυχιακής εργασίας. Ιδιαίτερες ευχαριστίες απευθύνουμε στον επιβλέποντα καθηγητή μας Κ. Κολομβάτσο , για την καθοδήγηση, την υποστήριξη και τις πολύτιμες συμβουλές του καθ' όλη τη διάρκεια της εκπόνησης της εργασίας. Επιπλέον, ευχαριστούμε τις οικογένειές μας και τους φίλους μας για την αδιάκοπη ενθάρρυνση και στήριξή τους, σε όλο αυτό το ταξίδι προς την γνώση.

Table of Contents

ΠΕΡΙΛΗΨΗ I
ABSTRACT III

ΚΕΦΑΛΑΙΟ 1Ο 2

1.1Εισαγωγη 2
1.2 Ιστορικη Αναδρομη 5

ΚΕΦΑΛΑΙΟ 2Ο 7

2.1 Βιβλιογραφικη Ανασκοπηση 7
2.2 Σκοπος της Έρευνας 15
2.2.Α Ακριβης Αναγνωριση Αντικειμενων 15
2.2.Β Υποστηριξη σε Πραγματικο Χρονο 15
2.2.C Ενσωματωση Φωνητικης Ανατροφοδοτησης 16
2.2.D Καινοτομια και Συνεισφορα 16
2.3 Ταχυτητα και Ακριβεια στην Αναγνωριση Εικονας 16
2.3.Α Προσιτοτητα και Ευκολια Χρησης 17
2.3.Β Διαλειτουργικοτητα 17
2.4 Τεχνολογικες Βασεις 18
2.5 Τεχνικες Τριγωνικης Ομοιοτητας 19

ΚΕΦΑΛΑΙΟ 3Ο 20

ΜΕΘΟΔΟΛΟΓΙΑ 20
3.1 Επιλογη και προετοιμασια Dataset 20
3.2 Επιλογη Μοντελου 20
3.3 Προετοιμασια Κωδικα 22
3.3.Α Φορτωση και προετοιμασια μοντελου 22
3.3.Β Προετοιμασια περιβαλλοντος Ανιχνευσης 23
3.3.C Ανιχνευση και Αναγνωριση Αντικειμενων σε Πραγματικο Χρονο 25
3.3.D Φωνητικη Ανατροφοδοτηση και Ενσωματωση στο Συστημα 26
3.3.Ε Υπολογισμος Αποστασης Αντικειμενων 28
3.3.F Υπολογισμος Ύψους Αντικειμενων σε Pixels 29
3.3.G Υπολογισμος Focal Length 31
3.3.Η Χρηση multi-threading 33
3.4 Εκτελεση Προγραμματος 35
3.4.Α Εκκινηση προγραμματος 35
3.4.Β Walking Mode 35
3.4.Β Search Mode 38

ΚΕΦΑΛΑΙΟ 4Ο 41

ΑΠΟΤΕΛΕΣΜΑΤΑ 41
4.1 WALKING MODE 41
4.2 SEARCH MODE 47
4.3 ΓΕΝΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ 50

ΚΕΦΑΛΑΙΟ 5Ο 53

ΣΥΜΠΕΡΑΣΜΑΤΑ 53

ΒΙΒΛΙΟΓΡΑΦΙΑ 56

ΚΕΦΑΛΑΙΟ 1ο

1.1Εισαγωγή

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας, το 2018, 217 εκατομμύρια άνθρωποι παγκοσμίως είχαν από μέτρια έως σοβαρή οπτική αναπηρία και 36 εκατομμύρια είχαν ολική τύφλωση. Βάσει εκτιμήσεων μέχρι το 2020, ο αριθμός των ατόμων με οπτική αναπηρία θα αγγίξει τα 250 εκατομμύρια, ενώ εκείνων με ολική τύφλωση τα 75 εκατομμύρια (Bourne et al., 2017). Αξίζει να σημειωθεί στο σημείο αυτό ότι, το μεγαλύτερο ποσοστό των ατόμων με ολική τύφλωση, ζει σε χώρες με χαμηλό οικονομικό επίπεδο, και είναι άνω των 50 ετών. Ωστόσο και στις αναπτυγμένες χώρες, η γήρανση σε συνδυασμό με την μείωση των εισοδημάτων, έχει πλήξει ιδιαίτερα τα άτομα με οπτική αναπηρία.(Budrionis An, Plikynas D., 2020).

Τα προβλήματα που αντιμετωπίζουν τα άτομα με προβλήματα όρασης, είναι στις περισσότερες των περιπτώσεων, μείζονα λειτουργικά με σημαντικές ψυχοκοινωνικές προεκτάσεις. Η όραση, αποτελεί μία από τις αισθήσεις, μέσω της οποίας αναπτύσσονται και ολοκληρώνονται βασικές ανθρώπινες δεξιότητες . Η εκπαίδευση των ατόμων προκειμένου να αμβλύνουν την αισθητηριακή απώλεια με τα υπόλοιπα αισθητηριακά συστήματα (π.χ. ακοή, αφή), συχνά δεν επαρκεί. Η πλοήγηση των ατόμων με προβλήματα όρασης, και η ποιότητα της λειτουργικότητας στην καθημερινότητα, η οποία εν μέρει εξαρτάται και από την ταχύτητα αντίδρασης και αλληλεπίδρασης με το περιβάλλον, αποτελούν σημαντικούς παράγοντες για την επίτευξη της κοινωνικής ένταξής του και της αυτονόμησης. Οι παράγοντες αυτοί, βελτιώνονται μεν με την φυσική αποκατάσταση αλλά δεν μπορούν να φτάσουν σε επίπεδα που να προσομοιάζουν τα φυσιολογικά

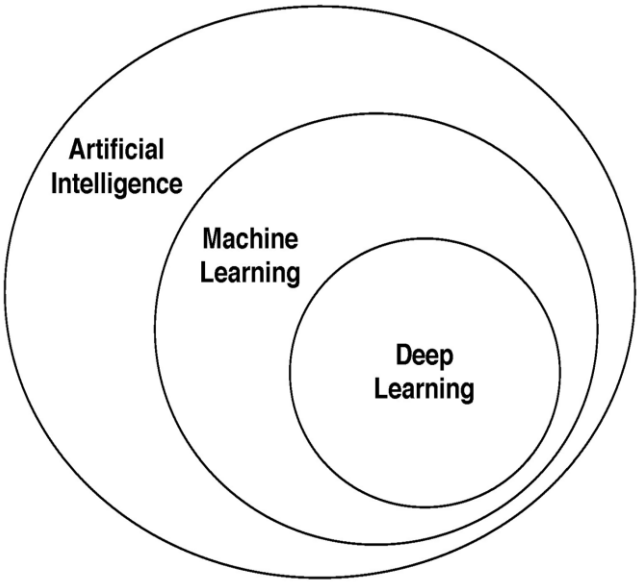
Μερικά από τα πιο σημαντικά λειτουργικά ελλείμματα των ατόμων με προβλήματα όρασης είναι, η ικανότητα πλοήγησης σε εσωτερικό και εξωτερικό περιβάλλον, η αναγνώριση αντικειμένων και προσώπων και η αλληλεπίδραση με αυτά, και η δυνατότητα πρόσβασης σε πληροφορίες που συχνά παρουσιάζονται με οπτικές αναπαραστάσεις

Δεδομένου του αισθητηριακού ελλείμματος, τα άτομα αντιμετωπίζουν συχνά σοβαρά ψυχοκοινωνικά προβλήματα με σημαντικές κοινωνικοοικονομικές προεκτάσεις. Οι βοηθητικές συσκευές, παίζουν σημαντικό ρόλο στην καθημερινότητά τους. Για μεγάλο χρονικό διάστημα, λύσεις όπως η καθοδήγηση από ειδικά εκπαιδευμένους σκύλους, η χρήση ανθρώπινων οδηγών (μετακίνηση χέρι-χέρι) και η χρήση ειδικών μπαστουνιών, αποτελούσαν τις μόνες επιλογές. Η γραφή Braille, επέτρεψε την πρόσβαση σε πληροφορίες, ενώ η τεχνολογία υποβοήθησης όπως η μετατροπή κειμένου σε ηχητικό δεδομένο (text-to-speech), βελτίωσε ακόμα περισσότερο της ποιότητα ζωής τους.

Η επανάκτηση της δημοτικότητας του πεδίου της Τεχνητής Νοημοσύνης (AI) ή της Μηχανικής Μάθησης (ML), οδήγησε σε ταχεία αύξηση του αριθμού των ερευνών που σχετίζονται με εφαρμογές σε πραγματικό κόσμο. Το Deep Learning δημιούργησε πολύπλοκους αλγόριθμους, προκαλώντας ανησυχίες σχετικά με την χρηστικότητά του. Οι περιορισμοί αυτοί και η αυξανόμενη αλληλεπίδραση

ανθρώπου και AI, οδήγησαν σε ένα νέο πεδίο που αφορά την Ανθρωποκεντρική Μηχανική Μάθηση (HCML) (Kaluarachchi Th., et al 2021).

Η AI αντιπροσωπεύει ένα ευρύ φάσμα της αυτοματοποιημένης λήψης αποφάσεων, από την λογική έως τα νευρωνικά δίκτυα. Οι αποφάσεις ή οι προβλέψεις που γίνονται με τη χρήση τεχνικών που βασίζονται σε δεδομένα, εμπίπτουν στην ML (υποσύνολο της AI). Το υποσύνολο των τεχνικών της ML, που χρησιμοποιούν Βαθιά Νευρωνικά Δίκτυα, ονομάζεται DL. (*Εικόνα*1)



Εικόνα1. Αναπαράσταση της AI και των υποσυνόλων της.

Η AI έχει συμβάλλει σημαντικά στην βελτίωση της ποιότητα των πληροφοριών που δέχονται τα άτομα με οπτικές διαταραχές, ωστόσο σημαντικός περιοριστικός παράγοντας ευρείας χρήσης των αντίστοιχων τεχνολογιών, είναι το υψηλό οικονομικό κόστος. Για παράδειγμα, οι έξυπνες συσκευές που συνδυάζουν αναγνώριση εικόνας και τεχνητή νοημοσύνη είναι συχνά οικονομικά απρόσιτες για πολλά άτομα με προβλήματα όρασης. Ένας επιπλέον περιορισμός είναι, η πολυπλοκότητα στη χρήση αυτών των συσκευών αυτών. Πολλές από αυτές τις τεχνολογίες απαιτούν εξειδικευμένες γνώσεις ή εκπαίδευση για να χρησιμοποιηθούν αποτελεσματικά, κάτι που δεν είναι πάντα εφικτό για όλους τους χρήστες.

Η παρατηρούμενη αύξηση του αριθμού ατόμων με οπτική αναπηρία και χαμηλά εισοδήματα, κάνει επιτακτική την ανάγκη ανάπτυξης φθηνότερων προγραμμάτων και συσκευών. Αυτό μπορεί να γίνει μέσω της χρήσης των smartphone ως ένα πολυλειτουργικό και πολυαισθητηριακό σύστημα.(Csapo et al, 2015) με την συμβολή της υπολογιστικής όρασης.

Η υπολογιστική όραση αποτελεί ένα από τα πλέον δυναμικά πεδία της τεχνητής νοημοσύνης. Στοχεύει στη δημιουργία συστημάτων, που επιτρέπουν στους υπολογιστές να αναγνωρίζουν και να ερμηνεύουν οπτικές πληροφορίες από τον κόσμο, με τρόπο που προσομοιάζει την ανθρώπινη οπτική λειτουργία. Η δυνατότητα αυτή είναι κεντρική σε πολλές σύγχρονες τεχνολογίες, με εφαρμογές που

επεκτείνονται σε διάφορους τομείς της καθημερινής ζωής και της βιομηχανίας.

Η υπολογιστική όραση είναι το επιστημονικό πεδίο που ασχολείται με την κατανόηση και ανάλυση οπτικών δεδομένων, μέσω υπολογιστικών μεθόδων. Πρόκειται για μια διεπιστημονική περιοχή ,που συνδυάζει στοιχεία από την τεχνητή νοημοσύνη, τη μηχανική μάθηση, τη ρομποτική, και την επεξεργασία εικόνας. Βασική επιδίωξη της υπολογιστικής όρασης, είναι η ανάπτυξη αλγορίθμων και συστημάτων που επιτρέπουν στους υπολογιστές να «βλέπουν», δηλαδή να επεξεργάζονται, να αναλύουν και να κατανοούν εικόνες και βίντεο.

Οι εφαρμογές της υπολογιστικής όρασης είναι πολυάριθμες και επηρεάζουν πολλές πτυχές της σύγχρονης ζωής. Μερικές από τις πιο γνωστές εφαρμογές περιλαμβάνουν:

- Αναγνώριση προσώπου: Χρησιμοποιείται ευρέως σε συστήματα ασφαλείας, για ταυτοποίηση ατόμων, και σε κοινωνικά δίκτυα για τον αυτόματο εντοπισμό και χαρακτηρισμό προσώπων σε φωτογραφίες.
- Αναγνώριση αντικειμένων: Εφαρμογές όπως το YOLO (You Only Look Once), επιτρέπουν σε συστήματα να αναγνωρίζουν και να κατηγοριοποιούν αντικείμενα σε πραγματικό χρόνο, κάτι που είναι κρίσιμο για εφαρμογές όπως η αυτόνομη οδήγηση.
- Αυτόνομη οδήγηση: Η υπολογιστική όραση επιτρέπει στα αυτόνομα οχήματα να «βλέπουν» το περιβάλλον τους, να αναγνωρίζουν πινακίδες, πεζούς και άλλα οχήματα, και να λαμβάνουν αποφάσεις σε πραγματικό χρόνο.
- Ιατρική διάγνωση: Η ανάλυση ιατρικών εικόνων, όπως ακτινογραφίες και μαγνητικές τομογραφίες, μέσω υπολογιστικής όρασης βοηθά στη διάγνωση ασθενειών με μεγαλύτερη ακρίβεια και ταχύτητα.

Η χρήση της υπολογιστικής όρασης σε συστήματα υποστήριξης για άτομα με αναπηρίες, έχει ανοίξει νέους δρόμους στην ανεξαρτησία και την ποιότητα ζωής αυτών των ατόμων. Ιδιαίτερα για άτομα με προβλήματα όρασης, οι τεχνολογίες αυτές παρέχουν τη δυνατότητα αναγνώρισης του περιβάλλοντος και μετατροπής της οπτικής πληροφορίας σε ακουστικά ή απτικά σήματα, βοηθώντας τους να κινούνται και να αντιλαμβάνονται τον κόσμο γύρω τους με μεγαλύτερη ασφάλεια και αυτοπεποίθηση.

Μια σημαντική εφαρμογή είναι τα συστήματα αναγνώρισης προσώπων και αντικειμένων, που μπορούν να βοηθήσουν τους χρήστες να αναγνωρίσουν τα άτομα γύρω τους ή τα αντικείμενα που βρίσκονται μπροστά τους. Αυτά τα συστήματα μπορούν να ενσωματωθούν σε φορητές συσκευές ή γυαλιά επαυξημένης πραγματικότητας, παρέχοντας στους χρήστες πληροφορίες σε πραγματικό χρόνο για το περιβάλλον τους, μέσω φωνητικών οδηγιών.
Επιπλέον, οι τεχνολογίες υπολογιστικής όρασης χρησιμοποιούνται σε συστήματα πλοήγησης για άτομα με προβλήματα όρασης, επιτρέποντάς τους να κινούνται με μεγαλύτερη ακρίβεια σε δημόσιους χώρους. Αυτά τα συστήματα χρησιμοποιούν κάμερες και αισθητήρες για να αντιλαμβάνονται το περιβάλλον, εντοπίζοντας εμπόδια και κατευθύνοντας τον χρήστη με φωνητικές εντολές ή απτικές δονήσεις.

Η υπολογιστική όραση έχει γνωρίσει ραγδαία εξέλιξη τα τελευταία χρόνια, κυρίως χάρη στις προόδους στην μηχανική μάθηση και στις τεχνικές βαθιάς μάθησης (deep

learning). Τα νευρωνικά δίκτυα, όπως τα Convolutional Neural Networks (CNNs), έχουν φέρει επανάσταση στην ανάλυση εικόνας, επιτρέποντας την ανάπτυξη συστημάτων με ακρίβεια που πλησιάζει ή και ξεπερνά την ανθρώπινη ικανότητα αναγνώρισης.
Ωστόσο, παρά τις επιτυχίες αυτές, το πεδίο της υπολογιστικής όρασης αντιμετωπίζει ακόμα σημαντικές προκλήσεις:

- Ταχύτητα και Απόδοση σε Πραγματικό Χρόνο: Η ανάγκη για αλγόριθμους που να μπορούν να λειτουργούν σε πραγματικό χρόνο με ελάχιστη καθυστέρηση είναι κρίσιμη, ειδικά για εφαρμογές όπως η αυτόνομη οδήγηση ή τα συστήματα υποστήριξης για άτομα με αναπηρίες. Η βελτιστοποίηση αυτών των αλγορίθμων ώστε να λειτουργούν με ταχύτητα και ακρίβεια, παραμένει ένα δύσκολο τεχνικό πρόβλημα.
- Αντιμετώπιση Σύνθετων Περιβαλλόντων: Οι αλγόριθμοι υπολογιστικής όρασης πρέπει να είναι ικανοί να αναγνωρίζουν αντικείμενα και πρόσωπα σε ποικίλες συνθήκες φωτισμού και απόστασης. Αυτό απαιτεί την ανάπτυξη πιο ευέλικτων μοντέλων που μπορούν να προσαρμοστούν σε διαφορετικές συνθήκες χωρίς να χάνουν την ακρίβειά τους.
- Επεξήγηση και Διαφάνεια: Όπως σε όλα τα πεδία της τεχνητής νοημοσύνης, η διαφάνεια και η δυνατότητα εξήγησης των αποφάσεων που λαμβάνουν τα συστήματα υπολογιστικής όρασης, είναι κρίσιμη για την αποδοχή και τη σωστή χρήση τους. Αυτό είναι ιδιαίτερα σημαντικό σε ευαίσθητες εφαρμογές, όπως η ιατρική διάγνωση ή η υποστήριξη ατόμων με αναπηρίες, όπου οι αποφάσεις των συστημάτων πρέπει να είναι κατανοητές και να ερμηνεύονται εύκολα από τους χρήστες.

1.2 Ιστορική Αναδρομή

Τα τελευταία 70 χρόνια, έχουν γίνει αρκετές προσπάθειες, ώστε η Τεχνητή Νοημοσύνη (AI) να αναδείξει τις υψηλές προσδοκίες που δημιουργήθηκαν από την τεχνολογική εξέλιξη.

Η AI έχει συμβάλλει στην ταχεία ανάπτυξη πολλών βιομηχανιών, εκτελώντας καθήκοντα, που τυπικά απαιτούσαν την ανθρώπινη νοημοσύνη για να διεκπεραιωθούν. Στόχος της είναι ο προγραμματισμός της νοημοσύνης σε μηχανές, μαθαίνοντας μέσω εμπειριών και προσαρμοζόμενο στις περιβαλλοντικές μεταβολές, ώστε να ενισχύσουν την ανθρώπινη λήψη αποφάσεων και την διαδικασία κατανόησης. (Muthukrishnan M. et al, 2020).

Η Μηχανική Μάθηση (ML), είναι κλάδος της AI, και σχετίζεται με αλγόριθμους που μπορούν να μάθουν σύνθετα τμήματα ενός έργου και να αναπτύξουν μοντέλα πρόληψης μέσα από ένα δείγμα δεδομένων. Εφαρμόσθηκε με επιτυχία σε τομείς, όπως επεξεργασία φυσικής γλώσσας, μηχανική όραση, την ρομποτική και την διαγνωστική (Pierson HA et al, 2017).

Τα πρώτα μοντέλα AI, προσομοίαζαν την λειτουργίας ενός απλού νευρώνα. Με το πέρασμα των δεκαετιών ωστόσο προστέθηκαν πιο σύνθετες λειτουργίες, για να οδηγηθούμε, στη σημερινή εποχή, στα βαθιά νευρωνικά κανάλια (DL). Μια από τις πρώτες δημοσιεύσεις σχετικά με το AI ήταν μία μελέτη των MacGulloch & Piits , το

1943. Στην μελέτη τους περιέγραψαν ένα μοντέλο υπολογιστή μου μάθαινε με διαδικασία παρόμοια με εκείνη των νευρώνων του εγκεφάλου. Το μοντέλο τους αναφέρονταν ως MCP νευρώνας, και λειτουργούσε λαμβάνοντας πληροφορίες Boolean (η λογική Boolean αναφέρεται σε δεδομένα άλγεβρας σχετικά με περιπτώσεις σωστού/λάθους). Το μοντέλο αυτό ήταν αποτελεσματικό για πολύ απλές εργασίες, ωστόσο είχε σημαντικούς περιορισμούς.

Το 1958 ο Rosenblatt παρουσίασε μία πιο εξελιγμένη μορφή AI, στο οποίο έδωσε την ονομασία Αναγνωριστής (Percepton). Το μοντέλο του ήταν πιο ευέλικτο και έγινε ένας από τους ακρογωνιαίους λίθους των σύγχρονων νευρικών δικτύων.

Το 1950 ο Alan Turing, στη μελέτη με τίτλο "Computing Machinery and Intelligence" έθεσε το ερώτημα σχετικά με το αν οι μηχανές μπορούν να μιμηθούν την ανθρώπινη νοημοσύνη και εισήγαγε ένα κριτήριο αξιολόγησης , μέσω του οποίου εξετάζεται το μοντέλο νοημοσύνης.

Στο συνέδριο του Dartmouth το 1956, η AI αναγνωρίστηκε και καθορίστηκε ουσιαστικά ο στόχος της . Από την εποχή εκείνη έως και το 1990, η AI τέθηκε αρκετές φορές υπό αμφισβήτηση και οι συνεχιζόμενες έρευνες συχνά αποτύγχαναν να εκπληρώσουν τις υψηλές προσδοκίες που είχαν δημιουργηθεί. Το ενδιαφέρον αναζωπυρώθηκε στα μέσα της δεκαετίας του 1990 καθώς η υπολογιστική ισχύ αυξήθηκε και μπορούσε να υποστηρίξει την ανάπτυξη νευρικών δικτύων. Οι δυνατότητες του AI παρουσιάστηκαν το 1997, όταν η IBM ανέπτυξε το supercomputer Deep Blue, τον πρώτο υπολογιστή που αναμετρήθηκε σε μία παρτίδα σκακιού με τον παγκόσμιο πρωταθλητή Kasparov, προκαλώντας για ακόμα μια φορά το ενδιαφέρον του κοινού.

Την τελευταία δεκαετία, υπήρξαν δύο σημαντικές βελτιώσεις που επέτρεψαν την πρόοδο των νευρικών δικτύων: η αποθήκευση δεδομένων και οι μονάδες γραφικής επεξεργασίας (GPU). Τα δεδομένα πλέον αποθηκεύονται ευκολότερα και με χαμηλότερο κόστος. Η ML πραγματοποιείται με αλγόριθμους που βασίζονται στα διαθέσιμα δεδομένα. βΑυτό οδήγησε τους Hinton et al, να παρουσιάσουν στο κοινό την εφαρμογή αναγνώρισης φωνής. Επιπλέον με τα GPUs και την βελτίωση των υπολογιστών, αντιμετωπίστηκαν πολλοί περιορισμοί που εμπόδιζαν την απόδοση των νευρικών δικτύων, φτάνοντας ως το σήμερα, που το AI καθημερινά καλύπτει σημαντικούς τομείς της ανθρώπινης καθημερινότητας.

ΚΕΦΑΛΑΙΟ 2ο

2.1 Βιβλιογραφική Ανασκόπηση

Η ανασκόπησης της βιβλιογραφίας έγινε σε δύο βάσεις δεδομένων, Google Scholar και Scopus, για έρευνες που διεξήχθησαν από το 2008 έως το 2023, και αφορούσαν μεθόδους και αλγόριθμους για την υποβοήθηση ατόμων με προβλήματα όρασης. Η αναζήτηση έγινε με τις εξής λέξεις-κλειδιά: Visually Impaired People (VIP), Computer Vision (CV), computer vision techniques, electronic travel aid, object recognition . Η επιλογή μελετών έγινε με τα εξής κριτήρια:

- Να είναι στην Αγγλική Γλώσσα
- Να αφορούν άτομα με προβλήματα όρασης
- Να είναι επικεντρωμένα στην ανάπτυξη και χρήση τεχνικών που χρησιμοποιούν την Υπολογιστική Όραση

Από την αναζήτησή μας βρέθηκαν 55 μελέτες, από τις οποίες οι 25 πληρούσαν τα κριτήρια επιλογής. Στον ακόλουθο πίνακα βλέπουμε τις μελέτες ανά χρονολογία, καθώς και τον σκοπό της πρότασής τους.

α/α	**Συγγραφέας- Τίτλος**	**Μελέτη- σκοπός**
1	Ivanchenko V., (2008): Computer Vision-Based Clear Path Guidance for Blind Wheelchair Users	Αυτή η μέθοδος περιγράφει την τεχνική καθοδήγησης Clear Path για άτομα με προβλήματα όρασης ενώ κινούνται σε αναπηρικό καροτσάκι. Ένα τυπικό (μη τροποποιημένο) λευκό μπαστούνι, χρησιμοποιείται εμπρός και πίσω. Το σύστημα παρακολουθεί συνεχώς τη θέση του μπαστουνιού και ειδοποιεί με ηχητικό σήμα, εάν εντοπιστεί κάποια ιδιαιτερότητα του εδάφους προς την κατεύθυνση που δείχνει το μπαστούνι
2	Hend S. Al Khalifa (2008): Utilizing QR Code and Mobile Phones for Blinds and Visually Impaired People	Χρησιμοποιώντας κωδικούς QR (δισδιάστατο γραμμωτό κώδικα) που επικολλώνται σε ένα αντικείμενο και σαρώνονται χρησιμοποιώντας ένα τηλέφωνο με κάμερα εξοπλισμένο με λογισμικό ανάγνωσης QR. Ο αναγνώστης αποκωδικοποιεί τον γραμμωτό κώδικα σε μια διεύθυνση URL και κατευθύνει το πρόγραμμα περιήγησης του τηλεφώνου να ανακτήσει ένα αρχείο ήχου από τον

		ιστό που περιέχει λεκτική περιγραφή του αντικειμένου. Θα πρέπει να υπάρχει σύνδεση στο διαδίκτυο.
3	Bighamy JP, et al (2010): VizWiz::LocateIt - Enabling Blind People to Locate Objects in Their Environment	Εφαρμογή για κινητά στην πλατφόρμα iOS, επομένως είναι μια δαπανηρή λύση. Γίνεται επεξεργασία εικόνας και εξαρτάται από τη σύνδεση στο διαδίκτυο και την τεχνολογία μηνυμάτων κινητής τηλεφωνίας. Εξαρτάται επίσης από την διαθεσιμότητα πληροφοριών.
4	Jing Su et al (2010): Timbremap: Enabling the Visually-Impaired to Use Maps on Touch- Enabled Devices	Εφαρμογή διάταξης/χάρτη. Λειτουργεί σε εσωτερικούς χώρους. Κάθε όροφος χαρτογραφείται ψηφιακά, και θα πρέπει να μπορεί να μεταφορτωθεί ή να είναι προσβάσιμος από την εφαρμογή.
5	Manduchi R et al. (2010): Blind guidance using mobile computer vision: a usability study	Εύρεση διαδρομής σε εσωτερικούς χώρους. Εξαρτάται από ειδικούς χρωματικούς δείκτες, τοποθετημένους σε καίρια σημεία του περιβάλλοντος, οι οποίοι μπορούν να ανιχνευθούν από ένα κανονικό τηλέφωνο με κάμερα. Βασίζεται στην κάμερα και την επεξεργασία εικόνας
6	Tian Y et al (2010): Improving Computer Vision-Based Indoor Wayfinding for Blind Persons with Context Information	Αλγόριθμος ανίχνευσης εισόδων βασισμένος σε εικόνες με βάση τα γενικά και σταθερά χαρακτηριστικά των θυρών (πλευρές και γωνίες) αντί για χαρακτηριστικά εμφάνισης (χρώμα, υφή κ.λπ.). Ο προτεινόμενος αλγόριθμος ανίχνευσης αξιολογείται σε μια βάση δεδομένων θυρών που συλλέγονται από μια ποικιλία περιβαλλόντων με ένα ευρύ φάσμα χρωμάτων, υφών, φωτισμού, κλίμακας και όψεων
7	Bellotti R. et al (2011): , LodeStar: a mobile device to enhance visually impaired people experience of cultural and naturalistic places	Λειτουργεί σε εξωτερικό περιβάλλον και στηρίζεται σε διαδραστική σχέση ανθρώπου-υπολογιστή. Δίνει επιπρόσθετες πληροφορίες σχετικά με την θέση στο χώρο των αντικειμένων σε σχέση με το άτομο και των αντικειμένων σε σχέση με τα

		υπόλοιπα αντικείμενα
8	Jose J. et al (2011): The Smart Vision local navigation aid for blind and visually impaired persons	Μικρό, φθηνό και ευκολοφόρετο βοήθημα πλοήγησης για τυφλούς και άτομα με προβλήματα όρασης. Η λειτουργικότητά του καλύπτει βασική πλοήγηση του χρήστη προς έναν προορισμό την πλοήγηση σε μονοπάτια, πεζοδρόμια και διαδρόμους, με αποφυγή στατικών καθώς και κινούμενων εμποδίων. Η πλοήγηση ισχύει τόσο για εσωτερικούς όσο και για εξωτερικούς χώρους.
9	Brihault A. et al (2011): Fusion of Artificial Vision and GPS to Improve Blind Pedestrian Positioning	Πρόκειται για μια εφαρμογή που βασίζεται σε GPS και λειτουργεί ακόμη και σε αστικό περιβάλλον όπου τα σήματα GPS υποβαθμίζονται. Η εκτιμώμενη θέση είναι τότε συμβατή με την υποβοηθούμενη πλοήγηση για τυφλούς. Αυτή η εφαρμογή, μπορεί να απαντήσει στις ανάγκες των τυφλών παρέχοντάς τους σχετική επίγνωση του περιβάλλοντος (εντοπισμός αντικειμένων ενδιαφέροντος), κατά μήκος της διαδρομής.
10	Bhargava B. et al (2011): A Mobile-Cloud Pedestrian Crossing Guide for the Blind	Συνεργατική προσέγγιση mobile cloud για εξωτερική πλοήγηση με επίγνωση περιβάλλοντος. Χρησιμοποιεί την υπολογιστική ισχύ των πόρων που διατίθενται από τους παρόχους cloud computing για επεξεργασία εικόνας σε πραγματικό χρόνο
11	Schauertey B. et al (2012): An assistive vision system for the blind that helps find lost things	Συνδυασμός ανίχνευσης αντικειμένων που βασίζονται σε χρώμα και SIFT προς πιθανές θέσεις αντικειμένων-στόχων. Πολύ έξυπνη εφαρμογή για να μάθετε που βρίσκονται χαμένα αντικείμενα, ιδιαίτερα σε μη αναμενόμενες θέσεις.
12	Johnsen AD et al (2012): Making Touch-Based Mobile Phones Accessible for the Visually Impaired	Πιθανές τεχνολογίες είναι οι αναγνώστες οθόνης και η απτική ανάδραση. Αυτή η λύση βασίζεται σε συνδυασμό ηχητικών και απτικών σημάτων. Είναι βασισμένο

		σε Android που επιδεικνύει τη νέα διεπαφή χρήστη, επιτρέποντας στα άτομα με προβλήματα όρασης να αλληλεπιδρούν απρόσκοπτα με ένα έξυπνο τηλέφωνο. Ο χρήστης λαμβάνει πληροφορίες όταν πατάει ή σύρει το δάχτυλο στην οθόνη.
13	Matusiak K. et al (2013): Object recognition in a mobile phone application for visually impaired users	Τεχνική αναγνώρισης χρωμάτων και αντικειμένων. Είναι μια εφαρμογή για κινητά στην πλατφόρμα Android. Φτηνό σε κόστος και εξαρτάται μόνο από βάση δεδομένων για κινητά. Θα πρέπει να περιλαμβάνει τεχνητή νοημοσύνη για μελλοντική βελτίωση.
14	Sivan & Darsan (2016): Computer Vision based assistive technology for blind and visually impaired people	Αλγόριθμος που προτείνεται για αναγνώριση εισόδων με ανίχνευση άκρων. εμφανίζει καλή αξιοπιστία.
15	Mocanu et al (2017): Seeing without sight- an automatic cognition system dedicated to blind and visually impaired people	Το YOLO v1, προσδιορίζει αντικείμενα και εμπόδια με mAP πάνω από 89% για εμπόδια όλων των ειδών.
16	Balasuriya B. et al (2017): Learning platform for visually impaired children through artificial intelligence and computer vision	Εφαρμογή για τη δημιουργίας μιας πλατφόρμας μάθησης για παιδιά με προβλήματα όρασης συνδυάζοντας βαθιά δίκτυα, επαναλαμβανόμενα νευρωνικά δίκτυα και μοντέλα ομιλίας για την παροχή διαδραστικής μαθησιακής εμπειρίας για συγκεκριμένα περιβάλλοντα. Εμφανίζει μειονεκτήματα σε πραγματικά περιβάλλοντα
17	Jiang H (2017): computer vision and text recognition for assisting visually impaired people using android smartphone	Ένα σύστημα που χρησιμοποιεί υπάρχουσες τεχνολογίες, όπως η οπτική αναγνώριση χαρακτήρων (OCR) και η μετατροπή κειμένου σε ομιλία (TTS) που είναι διαθέσιμες σε smartphone Android, και τις χρησιμοποιεί για να αναγνωρίζει αυτόματα κείμενα και πινακίδες στο περιβάλλον και να βοηθά τους χρήστες να πλοηγούνται. Το προτεινόμενο σύστημα χρησιμοποιεί έναν συνδυασμό

		υπολογιστικής όρασης και συνδεσιμότητας στο Διαδίκτυο σε ένα smartphone Android όχι μόνο για την αναγνώριση σημείων, αλλά και για την ανακατασκευή προτάσεων και τη μετατροπή τους σε ομιλία
18	Daliya D. et al (2018): Computer vision technique for scene captioning to provide assistance to visually impaired	Εφαρμογή για λεζάντες εικόνων σε πραγματικό χρόνο, χρησιμοποιώντας ένα μοντέλο βαθιάς μάθησης VGG16-LSTM με βοήθεια υπολογιστικής όρασης. Το πλαίσιο έχει αναπτυχθεί σε έ υπολογιστή μονής πλακέτας Raspberry Pi 4B, με δυνατότητες μονάδας επεξεργασίας γραφικών. Η εφαρμογή επιτρέπει την αυτοματοποιημένη παραγωγή σχετικών λεζάντων για φωτογραφίες που λαμβάνονται σε πραγματικό χρόνο από μια μονάδα κάμερας NoIR. Σχόλια των χρηστών δείχνουν σημαντική βελτίωση στην κατανόηση του οπτικού περιεχομένου
19	Elmannai W. et al (2018): Computer vision-based framework for supporting the mobility of the visually impaired	Φορητή συσκευή για την υποστήριξη ατόμων με προβλήματα όρασης (VI). Βασίζεται στην ενσωμάτωση τεχνικών που χρησιμοποιούν αισθητήρες και υπολογιστική όραση, ώστε να εισαχθεί μια αποτελεσματική και οικονομική οπτική συσκευή. Το ποσοστό ακρίβειας φτάνει το 98%. Χρησιμοποιήθηκαν διάφοροι αλγόριθμοι υπολογιστικής όρασης, όπως Oriented FAST και Rotated BRIEF (ORB), (KNN), Random sample consensus (RANSAC) και K-mean. Στοχεύει στην αποφυγή εμποδίων. Έλεγχος σε πραγματικό χρόνο έδειξε ότι η συγκεκριμένη προσέγγιση αποφυγής εμποδίων μπορεί να βοηθήσει τους χρήστες με προβλήματα όρασης να αποφύγουν το 100% των ανιχνευμένων αντικειμένων.

20	Talebi M., Vafaei A., (2018): Vision-based entrance detection in outdoor scenes	Μια μέθοδος για την ανίχνευση εισόδων κτιρίων σε εξωτερικούς χώρους με ανιχνευτή τμήματος γραμμής. Η ακρίβειά του στον προσδιορισμό εισόδου φτάνει το 93,2% για το ImageNet Dataset
21	Afif M et al (2020): Indoor objects detection and recognition for an ICT mobility assistance of visually impaired people	Το YOLO v3, προσδιορίζει εμπόδια κατά την κίνηση σε διάφορα περιβάλλοντα με ποσοστό ακρίβειας 73,19%
22	Duh PJ, Sung YC (2020): V-Eye: a vision-based navigation system for the visually impaired	Μέθοδος εντοπισμού (VB-GPS) και τεχνικές τμηματοποίησης εικόνας για την επίτευξη καλύτερης κατανόησης σκηνής με μία μόνο κάμερα. Το σύστημα μπορεί να παρέχει αξιόπιστα ακριβείς τοποθεσίες και πληροφορίες προσανατολισμού (με διάμεσο σφάλμα περίπου 0,27m και 0,95°). Η υποστήριξη πλοήγησης σε εσωτερικά και εξωτερικά περιβάλλοντα. Το V-eye βοηθά στην πλοήγηση, βελτιώνει την ευαισθητοποίηση των ατόμων με προβλήματα όρασης για τα εμπόδια, ενισχύει την αντίληψη του χώρου, καθώς και το αίσθημα ασφάλειας και ανεξαρτησίας κατά την βάδιση.
23	Joshi R (2020): Efficient multi-object detection and smart navigation using artificial intelligence for visually impaired people	Μια πλήρως αυτόματη υποστηρικτική τεχνολογία βασισμένη στην τεχνητή νοημοσύνη για την αναγνώριση διαφορετικών αντικειμένων και ακουστικών εισροών παρέχονται στον χρήστη σε πραγματικό χρόνο, γεγονός που δίνει καλύτερη κατανόηση στο άτομο με προβλήματα όρασης για το περιβάλλον του. Εκτός από τις τεχνικές που βασίζονται στην υπολογιστική όραση για την αναγνώριση αντικειμένων, ένας αισθητήρας μέτρησης απόστασης είναι ενσωματωμένος για να κάνει τη συσκευή πιο ολοκληρωμένη

		αναγνωρίζοντας εμπόδια κατά την πλοήγηση από το ένα μέρος στο άλλο. Η μέση ακρίβεια αυτής της προτεινόμενης μεθόδου είναι 95,19% και 99,69% για την ανίχνευση και αναγνώριση αντικειμένων, σε πραγματικό χρόνο.
24	Li X., Cui H.(2020): A computer vision-based approach to make all intersection-related pedestrian signals accessible for the visually impaired	Προσέγγιση βασισμένη στην υπολογιστική όραση, που ονομάζεται Cross-Safe, παρέχει ακριβή και προσβάσιμη καθοδήγηση σε άτομα με προβλήματα όρασης όταν περνούν διασταυρώσεις, ως μέρος μιας μεγαλύτερης ενοποιημένης έξυπνης φορητής συσκευής. Το Cross-Safe αξιοποιεί υπερσύγχρονες τεχνικές βαθιάς μάθησης για ανίχνευση και αναγνώριση σήματος πεζών σε πραγματικό χρόνο. Μια φορητή μονάδα GPU, η Nvidia Jetson TX2, παρέχει φορητούς οπτικούς υπολογιστές και ένας ψηφιακός βοηθός παρέχει ακριβή φωνητική καθοδήγηση.
25	Mahmud S et al (2020): A vision based voice controlled indoor assistant robot for visually impaired people	Σχεδιασμός και ανάπτυξη ενός προσωπικού βοηθού ρομπότ, το οποίο ελέγχεται από φωνητικές εντολές για την εύρεση και τον προσδιορισμό της σχετικής θέσης αντικειμένου σε εσωτερικό περιβάλλον με αλγόριθμο παράγοντα συσχέτισης (CRF) που αναπτύχθηκε σε αυτή τη μελέτη. Το προτεινόμενο βοηθητικό ρομπότ είναι ένα ημι-ανθρωποειδές ρομπότ που αποτελείται από πολλές κάμερες. Οι κάμερες χρησιμοποιούνται για αυτόνομη κίνηση, ανίχνευση αντικειμένων, μέτρηση απόστασης και σχεδιασμό κίνησης. Επιπλέον, το ρομπότ ενημερώνει τον χρήστη για τα αποτελέσματα των ενεργειών του αυξάνει τη χρησιμότητα του

		ρομπότ. Τα πειραματικά αποτελέσματα δείχνουν μεγάλη ακρίβεια, γεγονός που θα κάνει τα άτομα με προβλήματα όρασης να αισθάνονται το εσωτερικό περιβάλλον ασφαλές, βολικό και άνετο.

Πίνακας1. Βιβλιογραφική Ανασκόπηση.

Κατά την βιβλιογραφική επισκόπηση, βασική επιδίωξη ήταν η μελέτη προτάσεων που στόχευαν στην διευκόλυνση της καθημερινότητας των ατόμων με προβλήματα όρασης. Από τις 25 μελέτες, οι 9 ήταν προτάσεις διευκόλυνσης της πλοήγησης σε εξωτερικά και εσωτερικά περιβάλλοντα. Συγκεκριμένα, μία μελέτη (Ivanchenko V. et al 2008), αφορούσε την πλοήγηση τυφλών που κινούνται με αναπηρικό αμαξίδιο, οπότε αναφέρεται σε άτομα με επιπρόσθετη κινητική αναπηρία, μία κατηγορία σημαντική δεδομένου ότι, όπως αναφέρθηκε στην εισαγωγή, το ποσοστό ηλικιωμένων με προβλήματα όρασης και ενδεχόμενα συνοδά προβλήματα κινητικότητας βαίνει αυξανόμενο, ενώ υπάρχουν και οι περιπτώσεις ατόμων με νευροκινητικές διαταραχές και συνοδά προβλήματα όρασης.

Οι εφαρμογές των Jings et al (2010) και των Manduchi R. et al (2010), αφορούν την πλοήγηση σε εσωτερικό χώρο, δίνοντας την δυνατότητα εξερεύνησης περίπλοκων διατάξεων μετά από χαρτογράφηση του χώρου. Σε 5 μελέτες προτείνονται εφαρμογές που υποβοηθούν την πλοήγηση των ασθενών σε εσωτερικούς και εξωτερικούς χώρους (Jose J. (2011), Duh PJ, Sung YC (2020), Joshi R. (2020), Li X., Cui H. (2020), Mahmud S (2020). Ιδιαίτερα στην εφαρμογή των Li X, Cui H., ιδιαίτερη σημασία δίνεται στην ανίχνευση των σημάτων των πεζών και την διευκόλυνση διέλευσης από διασταυρώσει σε πραγματικό χρόνο με ακριβή φωνητική καθοδήγηση. Στην εργασία των Mahmud S et al (2020), προτάθηκε ο σχεδιασμός και η ανάπτυξη ενός προσωπικού βοηθού ρομπότ που να ελέγχεται από φωνητικές εντολές για την εύρεση και τον προσδιορισμό της σχετικής θέσης των αντικειμένων σε εσωτερικό περιβάλλον. Πρόκειται για ημι-ανθρωποειδές ρομπότ που χρησιμοποιεί πολλές κάμερες. Οι κάμερες χρησιμοποιούνται για επαρκή, αυτόνομη κίνηση, ανίχνευση αντικειμένων, μέτρηση απόστασης και σχεδιασμό κίνησης. Επιπλέον το ρομπότ ενημερώνει τον χρήστη για τα αποτελέσματα των ενεργειών του. Τα πειραματικά αποτελέσματα δείχνουν μεγάλη ακρίβεια στην μετακίνηση σε εσωτερικούς χώρους, γεγονός που θα κάνει τα άτομα με προβλήματα όρασης να αισθάνονται ασφαλή και λειτουργικά ανεξάρτητα. Πρόκειται για μία από τις πλέον ολοκληρωμένες προτάσεις που ωστόσο έχουν υψηλό κόστος.

Σε τρία άρθρα η προτεινόμενη εφαρμογή αφορούσε την ανίχνευση εισόδων (Tian Y et al (2020), Sivah & Darsan (2016), Talebi M (2018), 7 άρθρα στην αναγνώριση αντικειμένων και αποφυγή εμποδίων (Elmannai W et al (2018), Afif M. (2020), Bigham JP (2010), Hend S, Al Khalifa (2008), Bellotti R (2011), Matusiak K et al (2013), Mocanu et al (2017). Σε μία έρευνα προτάθηκε πλατφόρμα εκπαίδευσης για τυφλά παιδιά ηλικίας 6-14 παιδιών ώστε να εκπαιδευτούν στην αναγνώριση αντικειμένων του περιβάλλοντος δίνοντας λεπτομερείς περιγραφές και συσχετίσεις

με ηχητικές οδηγίες (Balasuriya B et al (2017)). Τέλος, σε πρόταση των Jiang H et al (2017), εξετάσθηκε η ανάπτυξη εφαρμογής που στηρίζεται στην αναγνώριση χαρακτήρων (OCR) και τηνβ μετατροπή κειμένου σε ομιλία. Η συγκεκριμένη εφαρμογή αναγνωρίζει αυτόματα κείμενα και πινακίδες στο περιβάλλον και βοηθά του χρήστες και στην πλοήγηση.

2.2 Σκόπος της Έρευνας

Η έρευνα αυτή στοχεύει στην ανάπτυξη ενός έξυπνου βοηθού που θα επιφέρει σημαντικές βελτιώσεις στην καθημερινή ζωή των ατόμων με προβλήματα όρασης, αξιοποιώντας τις πιο σύγχρονες τεχνολογίες. Με τη συνεχή ανάπτυξη τεχνητής νοημοσύνης και την αυξανόμενη επεξεργαστική ισχύ των συσκευών, είναι πλέον εφικτό να δημιουργηθούν λύσεις που θα επιτρέψουν στα άτομα με οπτικές αναπηρίες να διατηρήσουν ή και να αυξήσουν την αυτονομία τους, μειώνοντας την ανάγκη για εξωτερική βοήθεια. Οι κύριοι στόχοι της έρευνας περιγράφονται παρακάτω, καθώς και η σημασία τους για την επίτευξη του τελικού σκοπού.

2.2.α Ακριβής Αναγνώριση Αντικειμένων

Ένας από τους βασικότερους στόχους της έρευνας είναι η ανάπτυξη αλγορίθμων που θα επιτρέπουν στον έξυπνο βοηθό να αναγνωρίζει αντικείμενα με υψηλή ακρίβεια. Οι υπάρχουσες τεχνολογίες αναγνώρισης εικόνας έχουν σημειώσει μεγάλη πρόοδο, αλλά η εφαρμογή τους για την υποστήριξη ατόμων με προβλήματα όρασης απαιτεί ακόμη μεγαλύτερη ακρίβεια και ταχύτητα. Ο αλγόριθμος YOLO v.8 (You Only Look Once version 8) αποτελεί μια εξέλιξη της τεχνολογίας αναγνώρισης εικόνας, η οποία μπορεί να προσφέρει τόσο ταχύτητα όσο και ακρίβεια στην αναγνώριση αντικειμένων. Αυτός ο αλγόριθμος λειτουργεί σε πραγματικό χρόνο, επιτρέποντας στο σύστημα να παρέχει άμεσα ανατροφοδότηση για τα αντικείμενα που εντοπίζονται στο περιβάλλον του χρήστη.

Η ακρίβεια στην αναγνώριση αντικειμένων είναι κρίσιμη, καθώς μικρά σφάλματα μπορεί να οδηγήσουν σε επικίνδυνες καταστάσεις, όπως η σύγκρουση με εμπόδια ή η αδυναμία ανίχνευσης σημαντικών στοιχείων του περιβάλλοντος. Για το λόγο αυτό, ο αλγόριθμος YOLO v.8 επιλέχθηκε για την έρευνα αυτή, καθώς είναι γνωστός για την ικανότητά του να επεξεργάζεται εικόνες και βίντεο σε υψηλή ανάλυση και να εντοπίζει αντικείμενα με μεγάλη ακρίβεια.

2.2.b Υποστήριξη σε Πραγματικό Χρόνο

Η δεύτερη βασική επιδίωξη της έρευνας είναι η ανάπτυξη ενός συστήματος που θα παρέχει συνεχή υποστήριξη σε πραγματικό χρόνο. Η έννοια της υποστήριξης σε πραγματικό χρόνο είναι εξαιρετικά σημαντική για τα άτομα με προβλήματα όρασης, καθώς η καθυστέρηση στην παροχή πληροφοριών μπορεί να μειώσει την αποτελεσματικότητα του συστήματος και να αυξήσει τον κίνδυνο ατυχημάτων. Ο έξυπνος βοηθός θα πρέπει να είναι ικανός να αντιλαμβάνεται και να αναλύει το περιβάλλον του χρήστη ακαριαία, προκειμένου να παρέχει άμεσες και ακριβείς οδηγίες.

Για να επιτευχθεί αυτός ο στόχος, το σύστημα θα πρέπει να ενσωματώνει εξελιγμένες τεχνολογίες ανάλυσης δεδομένων και αναγνώρισης εικόνας, με δυνατότητες γρήγορης επεξεργασίας. Η έρευνα θα επικεντρωθεί επίσης στην εξασφάλιση ότι οι πληροφορίες που παρέχονται στο χρήστη είναι αξιόπιστες και επικαιροποιημένες, χωρίς να επιβαρύνουν το σύστημα με περιττές καθυστερήσεις ή ανακρίβειες.

2.2.c Ενσωμάτωση Φωνητικής Ανατροφοδότησης

Ο τρίτος κύριος στόχος της έρευνας είναι η ενσωμάτωση ενός συστήματος φωνητικής ανατροφοδότησης που θα καθοδηγεί τον χρήστη στην πλοήγηση του περιβάλλοντος. Η φωνητική ανατροφοδότηση είναι ζωτικής σημασίας για τα άτομα με προβλήματα όρασης, καθώς τους επιτρέπει να αντιλαμβάνονται το περιβάλλον τους με ένα εναλλακτικό τρόπο, αξιοποιώντας την ακοή. Ο έξυπνος βοηθός θα πρέπει να χρησιμοποιεί τεχνολογίες επεξεργασίας φυσικής γλώσσας για να παρέχει καθαρές, κατανοητές και ακριβείς φωνητικές οδηγίες.

Επιπλέον, το σύστημα φωνητικής ανατροφοδότησης θα πρέπει να προσαρμόζεται στις ανάγκες του κάθε χρήστη, παρέχοντας προσωποποιημένες οδηγίες ανάλογα με το επίπεδο της όρασης, την εμπειρία του χρήστη και τις συνθήκες του περιβάλλοντος. Αυτή η προσαρμοστικότητα θα συμβάλει στην αποτελεσματική χρήση του συστήματος σε διάφορα περιβάλλοντα και καταστάσεις, καθιστώντας τον έξυπνο βοηθό έναν απαραίτητο σύντροφο για τα άτομα με προβλήματα όρασης.

2.2.d Καινοτομία και Συνεισφορά

Το σύστημα που αναπτύσσεται στο πλαίσιο αυτής της έρευνας διακρίνεται για την καινοτομία του και τη συνεισφορά του στην υποστήριξη ατόμων με προβλήματα όρασης. Σε μια εποχή όπου η τεχνολογία εξελίσσεται ραγδαία, η πρόκληση δεν είναι μόνο η ανάπτυξη νέων λύσεων, αλλά η δημιουργία εργαλείων που είναι πραγματικά χρήσιμα και προσβάσιμα σε όσους τα χρειάζονται περισσότερο. Η προσέγγιση που υιοθετείται εδώ περιλαμβάνει την αξιοποίηση των πιο προηγμένων τεχνολογιών, όπως το YOLO v.8, σε συνδυασμό με την έμφαση στη χρηστικότητα και την προσβασιμότητα, με στόχο την ανάπτυξη ενός συστήματος που πραγματικά θα βελτιώσει την ποιότητα ζωής των ατόμων με οπτικές αναπηρίες.

2.3 Ταχύτητα και Ακρίβεια στην Αναγνώριση Εικόνας

Μία από τις πιο σημαντικές καινοτομίες αυτού του συστήματος είναι η ενσωμάτωση του αλγορίθμου YOLO v.8 (You Only Look Once version 8) για την αναγνώριση εικόνας. Ο αλγόριθμος αυτός αποτελεί την τελευταία εξέλιξη στην τεχνολογία αναγνώρισης αντικειμένων και είναι γνωστός για την ικανότητά του να εντοπίζει και να αναγνωρίζει αντικείμενα με εξαιρετική ταχύτητα και ακρίβεια. Ενώ προηγούμενες εκδόσεις και άλλες τεχνολογίες αναγνώρισης αντικειμένων συχνά

υστερούν σε ταχύτητα ή ακρίβεια, το YOLO v.8 καταφέρνει να συνδυάζει και τα δύο, κάνοντάς το ιδανικό για χρήση σε συστήματα που απαιτούν άμεση ανατροφοδότηση.

Η χρήση του YOLO v.8 επιτρέπει στο σύστημα να αναγνωρίζει αντικείμενα σε πραγματικό χρόνο, κάτι που είναι κρίσιμο για την αποτελεσματική υποστήριξη ατόμων με προβλήματα όρασης. Οι παλαιότερες τεχνολογίες μπορεί να είχαν καθυστερήσεις ή να απαιτούσαν μεγαλύτερους πόρους για την επεξεργασία των εικόνων, περιορίζοντας την απόδοσή τους σε πραγματικές συνθήκες. Αντιθέτως, το YOLO v.8 λειτουργεί σε πλατφόρμες με χαμηλή κατανάλωση ενέργειας, διατηρώντας την απόδοση σε υψηλά επίπεδα ακόμα και σε φορητές συσκευές.

2.3.a Προσιτότητα και Ευκολία Χρήσης

Ένας άλλος τομέας στον οποίο διαπρέπει το σύστημα αυτό είναι η προσβασιμότητα και η ευκολία χρήσης. Παρά την πολυπλοκότητα της τεχνολογίας που χρησιμοποιείται, το σύστημα έχει σχεδιαστεί με επίκεντρο τον χρήστη, λαμβάνοντας υπόψη τις ανάγκες των ατόμων με προβλήματα όρασης. Αυτό σημαίνει ότι ο έξυπνος βοηθός είναι όχι μόνο λειτουργικός, αλλά και εύκολος στη χρήση. Η διεπαφή του συστήματος είναι σχεδιασμένη ώστε να απαιτεί ελάχιστη εκπαίδευση και να μπορεί να χρησιμοποιηθεί από άτομα με διαφορετικά επίπεδα τεχνολογικής εξοικείωσης.

Επιπλέον, έχει δοθεί έμφαση στην προσιτότητα, ώστε το σύστημα να είναι οικονομικά βιώσιμο για μια ευρύτερη ομάδα χρηστών. Σε αντίθεση με άλλες προηγμένες τεχνολογίες που συχνά συνοδεύονται από υψηλό κόστος, το συγκεκριμένο σύστημα στοχεύει να είναι οικονομικά προσιτό, διασφαλίζοντας ότι τα οφέλη του μπορούν να επεκταθούν σε όσο το δυνατόν περισσότερους χρήστες. Αυτή η προσιτότητα, σε συνδυασμό με την ευκολία χρήσης, καθιστά το σύστημα ιδανικό για καθημερινή χρήση από άτομα με προβλήματα όρασης, προσφέροντας τους αυτονομία και ανεξαρτησία.

2.3.b Διαλειτουργικότητα

Τέλος, μια σημαντική συνεισφορά της έρευνας είναι η διαλειτουργικότητα του συστήματος. Το σύστημα έχει σχεδιαστεί ώστε να ενσωματώνεται εύκολα με άλλες συσκευές και τεχνολογίες, προσφέροντας μια ολοκληρωμένη εμπειρία στον χρήστη. Αυτή η διαλειτουργικότητα επιτρέπει στο σύστημα να συνεργάζεται με άλλες υπάρχουσες λύσεις υποστήριξης, όπως οι έξυπνες συσκευές που φοριούνται (wearables) ή τα υπάρχοντα συστήματα φωνητικής ανατροφοδότησης, επεκτείνοντας τη λειτουργικότητά του και αυξάνοντας την ευελιξία του.

Για παράδειγμα, το σύστημα μπορεί να συνδυαστεί με υπάρχουσες εφαρμογές κινητών τηλεφώνων, επιτρέποντας στο χρήστη να λαμβάνει πληροφορίες από πολλαπλές πηγές μέσω μιας ενοποιημένης πλατφόρμας. Αυτό διευκολύνει την ενοποίηση των υπηρεσιών και βελτιώνει την εμπειρία του χρήστη, δημιουργώντας ένα οικοσύστημα τεχνολογιών που συνεργάζονται για την καλύτερη υποστήριξη των

ατόμων με προβλήματα όρασης.

2.4 Τεχνολογικές Βάσεις

Το YOLO (You Only Look Once) αποτελεί έναν από τους πιο διαδεδομένους αλγορίθμους για την αναγνώριση αντικειμένων σε πραγματικό χρόνο, ιδιαίτερα δημοφιλής σε εφαρμογές υπολογιστικής όρασης που απαιτούν ταχύτητα και ακρίβεια. Ο αλγόριθμος αυτός βασίζεται στη δυνατότητά του να εντοπίζει αντικείμενα σε μία μόνο σάρωση της εικόνας, σε αντίθεση με άλλες μεθόδους που διασπούν την εικόνα σε πολλαπλά τμήματα ή πλαίσια. Αυτή η τεχνική καθιστά το YOLO εξαιρετικά αποδοτικό, επιτρέποντας την ανίχνευση αντικειμένων σε πραγματικό χρόνο, χωρίς σημαντικές καθυστερήσεις.

Η τελευταία έκδοση του αλγορίθμου, το YOLO v.8, έχει επιφέρει σημαντικές βελτιώσεις τόσο στην ταχύτητα όσο και στην ακρίβεια της αναγνώρισης. Συγκεκριμένα, το YOLO v.8 ενσωματώνει εξελιγμένες τεχνικές βαθιάς μάθησης, οι οποίες βελτιώνουν την ικανότητα του αλγορίθμου να αναγνωρίζει αντικείμενα ακόμη και σε σύνθετα ή δυσμενή περιβάλλοντα. Επίσης, η χρήση του YOLO v.8 σε βοηθητικές τεχνολογίες για άτομα με προβλήματα όρασης είναι ιδιαίτερα κρίσιμη, καθώς επιτρέπει την ακριβή ανίχνευση αντικειμένων σε συνθήκες πραγματικού χρόνου, διευκολύνοντας έτσι την αυτονομία και την ασφάλεια των χρηστών.

Ο αλγόριθμος YOLO επεξεργάζεται ολόκληρη την εικόνα ταυτόχρονα, χρησιμοποιώντας ένα ενιαίο νευρωνικό δίκτυο για να ανιχνεύσει πολλαπλά αντικείμενα σε διαφορετικές κλίμακες και τοποθεσίες. Αυτό διασφαλίζει ότι η αναγνώριση γίνεται με μια μοναδική, ολιστική προσέγγιση, χωρίς να απαιτείται η ανάλυση της εικόνας σε μικρότερα τμήματα. Αυτή η προσέγγιση προσφέρει σημαντική βελτίωση στην απόδοση σε εφαρμογές που απαιτούν γρήγορη και αξιόπιστη αναγνώριση αντικειμένων.

Το OpenCV (Open Source Computer Vision Library) είναι μια βιβλιοθήκη ανοιχτού κώδικα που χρησιμοποιείται ευρέως για την ανάπτυξη εφαρμογών υπολογιστικής όρασης και επεξεργασίας εικόνας. Παρέχει ένα πλούσιο σύνολο εργαλείων και μεθόδων για την ανάλυση εικόνας, γεγονός που την καθιστά ιδανική για την ανάπτυξη συστημάτων αναγνώρισης αντικειμένων. Το OpenCV υποστηρίζει πολλαπλές γλώσσες προγραμματισμού, όπως C++, Python και Java, ενώ παρέχει δυνατότητες για την ενσωμάτωση με τεχνολογίες μηχανικής μάθησης και τεχνητής νοημοσύνης.

Μία από τις βασικές λειτουργίες της βιβλιοθήκης OpenCV είναι η επεξεργασία εικόνας, όπου περιλαμβάνονται εργαλεία για την αποθορυβοποίηση, την ανίχνευση ακμών, και τις μετατροπές γεωμετρίας εικόνας. Η βιβλιοθήκη αυτή είναι απαραίτητη για την προετοιμασία των εικόνων πριν την εφαρμογή των αλγορίθμων αναγνώρισης, καθώς η βελτίωση της ποιότητας της εικόνας μπορεί να συμβάλλει σημαντικά στη βελτίωση της ακρίβειας της αναγνώρισης.

Το OpenCV σε συνδυασμό με τον αλγόριθμο YOLO δημιουργεί ένα ισχυρό σύστημα

για την επεξεργασία και την αναγνώριση εικόνας. Το OpenCV λειτουργεί ως η βάση πάνω στην οποία αναλύονται και βελτιστοποιούνται οι εικόνες, ενώ το YOLO εκτελεί την αναγνώριση αντικειμένων, κάνοντας το σύστημα ταυτόχρονα γρήγορο και αποδοτικό.

2.5 Τεχνικές Τριγωνικής Ομοιότητας

Οι τεχνικές τριγωνικής ομοιότητας αποτελούν βασικό εργαλείο για την αναγνώριση αντικειμένων και προσώπων σε διαφορετικές συνθήκες, όπως όταν υπάρχουν αλλαγές στη γωνία θέασης ή στο φωτισμό. Η τριγωνική ομοιότητα μετρά την ομοιότητα μεταξύ διανυσμάτων που αναπαριστούν τα χαρακτηριστικά των εικόνων, επιτρέποντας την ακριβή αντιστοίχιση χαρακτηριστικών ακόμη και όταν οι εικόνες δεν είναι ακριβώς ταυτόσημες.

Αυτή η μέθοδος είναι κρίσιμη για την αναγνώριση αντικειμένων σε συστήματα υπολογιστικής όρασης, διότι τα αντικείμενα συχνά εμφανίζονται υπό διαφορετικές γωνίες ή φωτιστικές συνθήκες στην πραγματική ζωή. Η τριγωνική ομοιότητα μπορεί να βοηθήσει στην αποφυγή σφαλμάτων κατά την αναγνώριση αντικειμένων που έχουν υποστεί μετατροπές ή παραμορφώσεις. Για παράδειγμα, αν ένα αντικείμενο φαίνεται μερικώς κρυμμένο ή παραμορφωμένο από τη γωνία θέασης, η χρήση τριγωνικών τεχνικών επιτρέπει στο σύστημα να αναγνωρίζει το αντικείμενο, ακόμη κι αν δεν είναι απόλυτα ορατό.

Οι τεχνικές αυτές χρησιμοποιούνται συχνά σε εφαρμογές που απαιτούν ακριβή αναγνώριση προσώπων ή αντικειμένων, ειδικά σε συστήματα που στοχεύουν στην υποστήριξη ατόμων με προβλήματα όρασης. Η ικανότητα να αναγνωρίζονται αντικείμενα ανεξαρτήτως της γωνίας ή του φωτισμού είναι σημαντική για τη λειτουργικότητα και την αξιοπιστία αυτών των συστημάτων σε καθημερινές συνθήκες.

ΚΕΦΑΛΑΙΟ 3ο

ΜΕΘΟΔΟΛΟΓΙΑ

3.1 Επιλογή και προετοιμασία Dataset

Για την ανάπτυξη του συστήματος, χρησιμοποιήθηκε το προ-εκπαιδευμένο μοντέλο YOLOv8, το οποίο έχει εκπαιδευτεί πάνω στο COCO dataset. Το Microsoft COCO (Common Objects in Context) dataset είναι ένα από τα πιο ευρέως χρησιμοποιούμενα σύνολα δεδομένων στην υπολογιστική όραση, περιλαμβάνοντας πάνω από 80 κατηγορίες καθημερινών αντικειμένων, όπως άνθρωποι, οχήματα, ζώα, κ.λπ. Το COCO dataset επιλέχθηκε λόγω της ποικιλίας και της πληρότητάς του, παρέχοντας ένα ισχυρό θεμέλιο για την ανάπτυξη ενός συστήματος γενικής χρήσης που μπορεί να ανιχνεύσει και να αναγνωρίσει πολλά διαφορετικά αντικείμενα.
Το μοντέλο YOLOv8 είναι ήδη βελτιστοποιημένο για τα δεδομένα του COCO και ως εκ τούτου δεν απαιτήθηκε περαιτέρω προσαρμογή ή προ επεξεργασία των δεδομένων. Η εκπαίδευση και η δοκιμή του μοντέλου είχαν πραγματοποιηθεί από τους δημιουργούς του, καθιστώντας το έτοιμο για χρήση σε εφαρμογές πραγματικού χρόνου. Αυτή η προσέγγιση επέτρεψε την άμεση ενσωμάτωση του μοντέλου στο σύστημα, εστιάζοντας στην ανάπτυξη των επιμέρους λειτουργιών του.
Οι κατηγορίες των καθημερινών αντικειμένων που βρίσκονται στο COCO Dataset παρουσιάζονται με την μορφή φωτογραφίας παρακάτω (*Εικόνα*2).

person	fire hydrant	elephant	skis	wine glass	broccoli	dining table	toaster
bicycle	stop sign	bear	snowboard	cup	carrot	toilet	sink
car	parking meter	zebra	sports ball	fork	hot dog	tv	refrigerator
motorcycle	bench	giraffe	kite	knife	pizza	laptop	book
airplane	bird	backpack	baseball bat	spoon	donut	mouse	clock
bus	cat	umbrella	baseball glove	bowl	cake	remote	vase
train	dog	handbag	skateboard	banana	chair	keyboard	scissors
truck	horse	tie	surfboard	apple	couch	cell phone	teddy bear
boat	sheep	suitcase	tennis racket	sandwich	potted plant	microwave	hair drier
traffic light	cow	frisbee	bottle	orange	bed	oven	toothbrush

Εικόνα2. Coco Dataset Classes.

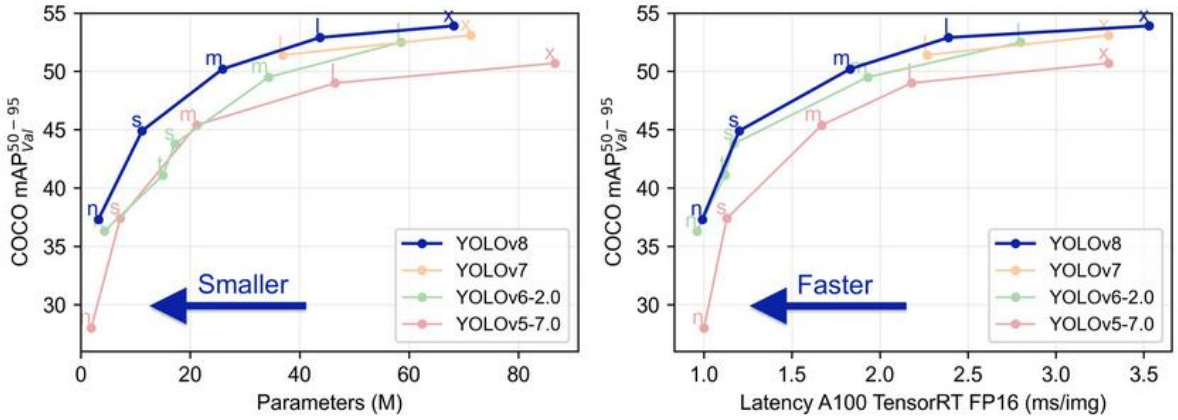
Το MS COCO (Microsoft Common Objects in Context) είναι ένα dataset μεγάλης κλίμακας για ανίχνευση αντικειμένων, τμηματοποίηση, ανίχνευση σημείων-κλειδιών και δημιουργία περιγραφών. Το dataset αποτελείται από 328.000 εικόνες. Το μοντέλο YOLOv8 χρησιμοποιεί την έκδοση COCO Dataset του 2017, η οποία περιλαμβάνει 41000 εικόνες για την εκπαίδευση και δοκιμασία του.

3.2 Επιλογή Μοντέλου

Η επιλογή του κατάλληλου μοντέλου για την ανίχνευση και αναγνώριση αντικειμένων ήταν μια κρίσιμη απόφαση κατά την ανάπτυξη του συστήματος

υποστήριξης για άτομα με προβλήματα όρασης. Για την ανάπτυξη της συγκεκριμένης πτυχιακής εργασίας, επιλέχθηκε το **YOLOv8 (You Only Look Once, version 8) της Ultralytics**, το οποίο είναι γνωστό για την υψηλή του απόδοση σε πραγματικό χρόνο και την ακρίβεια στην αναγνώριση αντικειμένων.

Το YOLOv8 είναι η πιο πρόσφατη έκδοση στη σειρά ανίχνευσης αντικειμένων (Object Detection) σε πραγματικό χρόνο YOLO, προσφέροντας κορυφαίες επιδόσεις όσον αφορά την ακρίβεια και την ταχύτητα. Βασιζόμενο στις εξελίξεις των προηγούμενων εκδόσεων YOLO, το YOLOv8 εισάγει νέα χαρακτηριστικά και βελτιστοποιήσεις που το καθιστούν ιδανική επιλογή για διάφορες εργασίες ανίχνευσης αντικειμένων σε ένα ευρύ φάσμα εφαρμογών(*Εικόνα*3).



Εικόνα3. Αναπαράσταση μοντέλων YOLO.

Η σειρά YOLOv8 προσφέρει μια ποικιλία μοντέλων, το καθένα εξειδικευμένο για συγκεκριμένες εργασίες στην υπολογιστική όραση.

Κάθε παραλλαγή της σειράς YOLOv8(*Εικόνα*4) είναι βελτιστοποιημένη για την αντίστοιχη εργασία της, εξασφαλίζοντας υψηλή απόδοση και ακρίβεια. Επιπλέον, αυτά τα μοντέλα είναι συμβατά με διάφορους λειτουργικούς τρόπους, όπως Εξαγωγή Συμπερασμάτων, Επαλήθευση, Εκπαίδευση και Εξαγωγή, διευκολύνοντας τη χρήση τους σε διάφορα στάδια ανάπτυξης και εφαρμογής.

Model	Filenames	Task	Inference	Validation	Training	Export
YOLOv8	yolov8n.pt yolov8s.pt yolov8m.pt yolov8l.pt yolov8x.pt	Detection	✅	✅	✅	✅
YOLOv8-seg	yolov8n-seg.pt yolov8s-seg.pt yolov8m-seg.pt yolov8l-seg.pt yolov8x-seg.pt	Instance Segmentation	✅	✅	✅	✅
YOLOv8-pose	yolov8n-pose.pt yolov8s-pose.pt yolov8m-pose.pt yolov8l-pose.pt yolov8x-pose.pt yolov8x-pose-p6.pt	Pose/ Keypoints	✅	✅	✅	✅
YOLOv8-obb	yolov8n-obb.pt yolov8s-obb.pt yolov8m-obb.pt yolov8l-obb.pt yolov8x-obb.pt	Oriented Detection	✅	✅	✅	✅
YOLOv8-cls	yolov8n-cls.pt yolov8s-cls.pt yolov8m-cls.pt yolov8l-cls.pt yolov8x-cls.pt	Classification	✅	✅	✅	✅

Εικόνα4. Παραλλαγές του YOLO v.8.

Στην συγκεκριμένη εργασία, η παραλλαγή του μοντέλου YOLOv8 που χρησιμοποιήσαμε είναι το **yolov8n.pt**

3.3 Προετοιμασία Κώδικα

3.3.α Φόρτωση και προετοιμασία μοντέλου

Η πρώτη κρίσιμη ενέργεια στον κώδικα είναι η φόρτωση του προ-εκπαιδευμένου μοντέλου YOLOv8n μέσω της βιβλιοθήκης ultralytics(*Εικόνα*5). Το μοντέλο αυτό είναι υπεύθυνο για την ανίχνευση και αναγνώριση αντικειμένων στις εικόνες που λαμβάνονται από την κάμερα.

Η σύνταξη του κώδικα πραγματοποιήθηκε σε γλώσσα προγραμματισμού Python, και συγκεκριμένα με το πρόγραμμα PyCharm IDE καθώς αποτελεί ένα πολύ βολικό εργαλείο για την σύνταξη μεγάλων προγραμμάτων και εφαρμογών.

Αρχικά, πραγματοποιήθηκε φόρτωση της βιβλιοθήκης ultralytics που διαθέτει το μοντέλο YOLOv8, και στη συνέχεια φόρτωση του προ-εκπαιδευμένου μοντέλου μέσω κατάλληλης συνάρτησης της βιβλιοθήκης(*Εικόνα*6).

```
from ultralytics import YOLO  # Import YOLO model from ultralytics
```

Εικόνα5. Φόρτωση μοντέλου YOLO μέσω της ultralytics library.

```
model = YOLO("yolov8n.pt")  # Load the YOLOv8n model with pre-trained weights
```

Εικόνα6. Εκκίνηση μοντέλου YOLO.

3.3.b Προετοιμασία περιβάλλοντος Ανίχνευσης

Προτού αρχίσει η διαδικασία ανίχνευσης, ορίζονται τα πραγματικά ύψη των αντικειμένων με βάση τον μέσο όρο(*Εικόνα*7), που θα ανιχνευθούν καθώς είναι απολύτως απαραίτητα για τον υπολογισμό της απόστασης τους από την κάμερα του συστήματος.

```
# Heights of Yolov8 class objects in real life
# to calculate the distance from camera using triangle similarity
real_heights = [  # Real-world average heights of objects corresponding to YOLO class IDs
    1.75,  # person (average human height)
    1.0,  # bicycle
    1.5,  # car
    1.2,  # motorcycle
    3.0,  # airplane (height at fuselage)
    3.2,  # bus
    3.0,  # train
    3.5,  # truck
    2.5,  # boat (small boat)
    2.0,  # traffic light
    1.0,  # fire hydrant
    2.0,  # stop sign
    1.5,  # parking meter
    0.8,  # bench
    0.3,  # bird
    0.25,  # cat
    0.5,  # dog
    1.6,  # horse
    1.0,  # sheep
    1.5,  # cow
    4.0,  # elephant
    2.0,  # bear
    1.5,  # zebra
    5.0,  # giraffe
    0.4,  # backpack
    1.0,  # umbrella
    0.3,  # handbag
    0.6,  # tie
```

Εικόνα7. Πραγματικά Μεγέθη των Αντικειμένων.

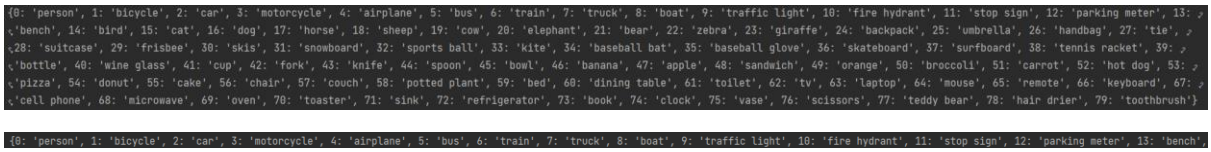
Έχουν οριστεί σε μορφή θέσεων σε έναν ενιαίο πίνακα, για την καλύτερη διαχείριση και οργάνωση τους, και η κάθε θέση του πίνακα αντιστοιχεί στις θέσεις που έχουν τα ίδια τα αντικείμενα στο YOLOv8 μοντέλο με την μορφή ενός Python Dictionary.

Η πληροφορία σχετικά με την αντιστοίχιση και την μορφή σύνταξης των κλάσεων των αντικειμένων στο προ-εκπαιδευμένο μοντέλο, μπορεί να φανερωθεί με τον ακόλουθο κώδικα(*Εικόνα*8).

```
from ultralytics import YOLO

model = YOLO('yolov8n.pt') # Load the YOLOv8n model with pre-trained weights
print(model.names) # Print the YOLOv8 Class Names' Dictionary
```

Εικόνα8. Εκτύπωση των κλάσεων του μοντέλου YOLO.

Εάν εκτελεστεί επιτυχώς ο παραπάνω κώδικας, εμφανίζεται το αποτέλεσμα στο Console(*Εικόνα*9).



Εικόνα9. Python dictionary των κλάσεων.

Σε μορφή κειμένου, εμφανίζεται το ακόλουθο κείμενο:

{0: 'person', 1: 'bicycle', 2: 'car', 3: 'motorcycle', 4: 'airplane', 5: 'bus', 6: 'train', 7: 'truck', 8: 'boat', 9: 'traffic light', 10: 'fire hydrant', 11: 'stop sign', 12: 'parking meter', 13: 'bench', 14: 'bird', 15: 'cat', 16: 'dog', 17: 'horse', 18: 'sheep', 19: 'cow', 20: 'elephant', 21: 'bear', 22: 'zebra', 23: 'giraffe', 24: 'backpack', 25: 'umbrella', 26: 'handbag', 27: 'tie', 28: 'suitcase', 29: 'frisbee', 30: 'skis', 31: 'snowboard', 32: 'sports ball', 33: 'kite', 34: 'baseball bat', 35: 'baseball glove', 36: 'skateboard', 37: 'surfboard', 38: 'tennis racket', 39: 'bottle', 40: 'wine glass', 41: 'cup', 42: 'fork', 43: 'knife', 44: 'spoon', 45: 'bowl', 46: 'banana', 47: 'apple', 48: 'sandwich', 49: 'orange', 50: 'broccoli', 51: 'carrot', 52: 'hot dog', 53: 'pizza', 54: 'donut', 55: 'cake', 56: 'chair', 57: 'couch', 58: 'potted plant', 59: 'bed', 60: 'dining table', 61: 'toilet', 62: 'tv', 63: 'laptop', 64: 'mouse', 65: 'remote', 66: 'keyboard', 67: 'cell phone', 68: 'microwave', 69: 'oven', 70: 'toaster', 71: 'sink', 72: 'refrigerator', 73: 'book', 74: 'clock', 75: 'vase', 76: 'scissors', 77: 'teddy bear', 78: 'hair drier', 79: 'toothbrush'}

Το οποίο αναφέρει τις θέσεις και τις ονομασίες των αντικειμένων (Εικ5) του μοντέλου YOLOv8 σε μορφή Python Dictionary.

Αν παρατηρήσουμε τον πίνακα που ορίσαμε στον κώδικα μας και το συγκεκριμένο λεξικό, θα παρατηρήσουμε πως οι θέσεις μνήμης και στις δύο διαφορετικές μορφές αποθήκευσης δεδομένων, ταυτίζονται με αποτέλεσμα να διευκολύνει τις ενέργειες του μοντέλου όπως αναφέραμε πιο πάνω.

3.3.c Ανίχνευση και Αναγνώριση Αντικειμένων σε Πραγματικό Χρόνο

Το κύριο μέρος της ανίχνευσης αντικειμένων υλοποιείται με τη χρήση του YOLOv8 και της βιβλιοθήκης OpenCV για την επεξεργασία εικόνων σε πραγματικό χρόνο από την κάμερα.

Το OpenCV είναι μια τεράστια βιβλιοθήκη ανοιχτού κώδικα για την υπολογιστική όραση, τη μηχανική μάθηση και την επεξεργασία εικόνας. Σήμερα, διαδραματίζει σημαντικό ρόλο σε λειτουργίες σε πραγματικό χρόνο, οι οποίες είναι πολύ σημαντικές στα σύγχρονα συστήματα. Χρησιμοποιώντας το, μπορεί κανείς να επεξεργαστεί εικόνες και βίντεο. Στη παρούσα εργασία, χρησιμοποιείται για να επιτρέψει την πρόσβαση στην κάμερα του συστήματος, με τη μορφή καταγραφής βίντεο ή ζωντανής μετάδοσης για να πραγματοποιηθεί η Ανίχνευση και Αναγνώριση αντικειμένων σε πραγματικό χρόνο.

Η παρούσα βιβλιοθήκη προστέθηκε στον κώδικα μας με την παρακάτω σύνταξη(*Εικόνα*10).

```
import cv2  # Import OpenCV for image and video processing
```

Εικόνα10. Φόρτωση του OpenCV library.

Η πιο απλή και γρήγορη αναπαράσταση κώδικα για την καταγραφή βίντεο μέσα από την κάμερα του συστήματος μας, με την χρήση του OpenCV είναι με την μορφή του παρακάτω κώδικα(*Εικόνα*11).

```python
import cv2

cam = cv2.VideoCapture(0)

while True:
    ret, frame = cam.read()

    cv2.imshow('Camera', frame)

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

cam.release()
cv2.destroyAllWindows()
```

Εικόνα11. Απλό Παράδειγμα OpenCV.

Η δική μας περίπτωση ακολουθεί την ίδια δομή της συνεχούς επανάληψης, αλλά υπάρχουν αρκετές τροποποιήσεις στο εσωτερικό της για την αναγνώριση και ανίχνευση των αντικειμένων που χρειαζόμαστε.

3.3.d Φωνητική Ανατροφοδότηση και Ενσωμάτωση στο Σύστημα

Η φωνητική ανατροφοδότηση είναι βασική λειτουργία του συστήματος και υλοποιείται με τη χρήση των βιβλιοθηκών **pyttsx3** και **gTTS**. Όταν ανιχνεύεται ένα αντικείμενο, το όνομα και η απόσταση του αντικειμένου μετατρέπονται από κείμενο σε ομιλία και ανακοινώνονται στον χρήστη μέσω ηχείων ή ακουστικών.

Η συγκεκριμένη διαδικασία πραγματοποιείται μέσω της συνάρτησης text_to_speech() (*Εικόνα*12) και έχει σαν μοναδικό όρισμα το κείμενο που επιθυμούμε να ακουστεί στον χρήστη.

```python
def text_to_speech(text):
    # Convert the given text to speech and play it using pygame

    # tts_lock is commented out, could be used for thread safety if uncommented
    tts = gTTS(text, lang='en')  # Convert the given text to speech in English using gTTS
    audio_file = BytesIO()  # Create an in-memory binary stream to hold the audio data
    tts.write_to_fp(audio_file)  # Write the generated speech audio to the in-memory stream
    audio_file.seek(0)  # Reset the stream position to the beginning

    pygame.init()  # Initialize pygame

    pygame.mixer.music.load(audio_file)  # Load the audio data from the in-memory stream into pygame's mixer
    pygame.mixer.music.play()  # Play the audio
    while pygame.mixer.music.get_busy():  # Wait until the audio playback is complete
        pygame.time.Clock().tick(10)  # Slow down the loop to avoid excessive CPU usage
```

Εικόνα12. Συνάρτηση Text To Speech.

Η φωνητική ανατροφοδότηση πραγματοποιείται και μέσα από τον χρήστη, καθώς χρειάζεται να επικοινωνήσει με τον έξυπνο βοηθό για να του αναφέρει την επιλογή που επιθυμεί να ακολουθήσει ο βοηθός. Η συγκεκριμένη διαδικασία πραγματοποιείται με την χρήση του μικροφώνου σαν πηγή και καταγράφει την φωνή του χρήση σε μορφή κειμένου ώστε να επεξεργαστεί από το ίδιο το πρόγραμμα.

Η συγκεκριμένη διαδικασία πραγματοποιείται μέσω της συνάρτησης speech_to_text() (*Εικόνα*13).

```python
def speech_to_text():
    recognizer = sr.Recognizer()  # Create a Recognizer object for speech recognition
    with sr.Microphone() as source:  # Use the microphone as the audio source
        print("Listening...")  # Notify that the program is listening for speech
        recognizer.adjust_for_ambient_noise(source)  # Adjust the recognizer for ambient noise
        # source, timeout, phrase time limit
        audio = recognizer.listen(source, 10, 3)  # Listen to the audio input with a timeout and phrase time limit

    try:
        text = recognizer.recognize_google(audio)  # Try to recognize the speech using Google's API
        print("You said:", text)  # Print the recognized text
        return text  # Return the recognized text
    except sr.UnknownValueError:  # Handle the case where the speech was not understood
        print("Could not understand audio")  # Notify that the audio was not understood
        return ""  # Return an empty string
    except sr.RequestError as e:  # Handle API request errors
        print("Could not request results; {0}".format(e))  # Print the error message
        return ""  # Return an empty string
```

Εικόνα13. Συνάρτηση Speech To Text.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την συγκεκριμένη λειτουργία είναι(*Εικόνα*14).

1. gTTS (Google Text-To-Speech): Μετατροπή κειμένου σε φωνή.
2. Pygame: Επιτρέπει την δυνατότητα αναπαραγωγής
3. IO: Επιτρέπει την διαχείριση δυαδικών σειρών στην μνήμη.

```
from gtts import gTTS  # Import Google Text-to-Speech for converting text to speech
import pygame  # Import pygame for playing audio
from io import BytesIO  # Import BytesIO for handling binary streams in memory
```

Εικόνα14. Χρήσιμες Βιβλιοθήκες.

3.3.e Υπολογισμός Απόστασης Αντικειμένων

Ο υπολογισμός της απόστασης των ανιχνευμένων αντικειμένων από την κάμερα βασίζεται στην αρχή της **τριγωνικής ομοιότητας**. Η τριγωνική ομοιότητα χρησιμοποιείται συχνά για τον υπολογισμό της απόστασης αντικειμένων με τη βοήθεια κάμερας, όταν το πραγματικό μέγεθος του αντικειμένου είναι γνωστό και το εικονικό του μέγεθος στην εικόνα μπορεί να μετρηθεί.

Η τριγωνική ομοιότητα βασίζεται στο γεγονός ότι δύο τρίγωνα είναι όμοια όταν οι αντίστοιχες γωνίες τους είναι ίσες. Στο συγκεκριμένο πρόβλημα, η κάμερα, το ανιχνευμένο αντικείμενο και η εικόνα του αντικειμένου σχηματίζουν δύο όμοια τρίγωνα. Με βάση αυτή την ομοιότητα, μπορούμε να υπολογίσουμε την απόσταση του αντικειμένου από την κάμερα, εφόσον γνωρίζουμε το πραγματικό ύψος του αντικειμένου (**H_real**), το ύψος του αντικειμένου στην εικόνα (**H_image**), και την εστιακή απόσταση της κάμερας (**focal Length**)

Η απόσταση υπολογίζεται με τον παρακάτω τύπο:

$$D = \frac{H_{real} * focalLength}{H_{image}}$$

Εξίσωση1. Υπολογισμός απόσταασης.

Όπου,

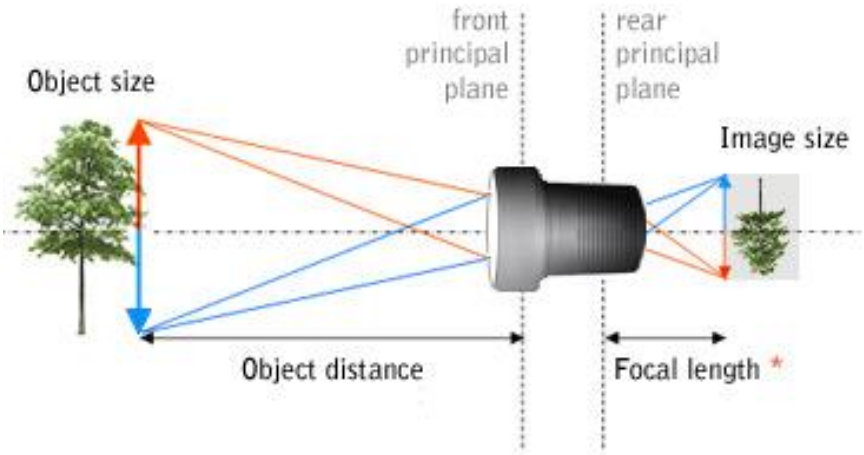
- H_{real}: Το πραγματικό ύψος του αντικείμενου στην κανονική/καθημερινή ζωή, μετριέται σε μέτρα (meters).

Αυτή η μεταβλητή καλύπτεται από τον πίνακα που εξηγήσαμε στην αρχή του κεφαλαίου σχετικά με τα αληθινά ύψη των αντικειμένων.

- H_{image}: Το ύψος του αντικειμένου κατά την διάρκεια της καταγραφής μέσω της κάμερας, μετριέται σε pixels.

- focalLength: Η εστιακή απόσταση της κάμερας (σε pixels).

Η εστιακή απόσταση(*Εικόνα*15) ενός οπτικού συστήματος είναι ένας μετρητής που δείχνει πόσο έντονα το σύστημα, σε αυτή την περίπτωση η κάμερα, συγκλίνει ή αποκλίνει το φως.



Εικόνα15. Distance Calculation from Camera.

3.3.f Υπολογισμός Ύψους Αντικειμένων σε Pixels

Όπως προ-αναφέρθηκε, στη συγκεκριμένη εργασία, για την ανίχνευση αντικειμένων σε πραγματικό χρόνο χρησιμοποιείται το μοντέλο YOLOv8. Μετά την ανίχνευση ενός αντικειμένου, το μοντέλο επιστρέφει ένα **bounding box** για κάθε αντικείμενο που εντοπίζει στην εικόνα. Το bounding box είναι ένα ορθογώνιο πλαίσιο που περιβάλλει το ανιχνευμένο αντικείμενο, και χρησιμοποιείται για τον προσδιορισμό της θέσης και του μεγέθους του αντικειμένου στην εικόνα.

Ένα **bounding box** ορίζεται από τις συντεταγμένες των δύο αντίθετων κορυφών του:

- **(x1, y1)**: Η πάνω αριστερή γωνία του πλαισίου.
- **(x2, y2)**: Η κάτω δεξιά γωνία του πλαισίου.

Με τη βοήθεια των συντεταγμένων αυτών, μπορούμε να υπολογίσουμε:

- Το **ύψος** του αντικειμένου στην εικόνα σε pixels.
- Το **πλάτος** του αντικειμένου στην εικόνα σε pixels.

Ο υπολογισμός του ύψους του αντικειμένου βασίζεται στη διαφορά των συντεταγμένων **y** του bounding box με τον ακόλουθο τύπο:

$$H_{image} = y2 - y1$$

Εξίσωση2. Υπολογισμός ύψους bounding box.

- **y2**: Η κατακόρυφη συντεταγμένη της κάτω δεξιάς γωνίας του πλαισίου.
- **y1**: Η κατακόρυφη συντεταγμένη της πάνω αριστερής γωνίας του πλαισίου.

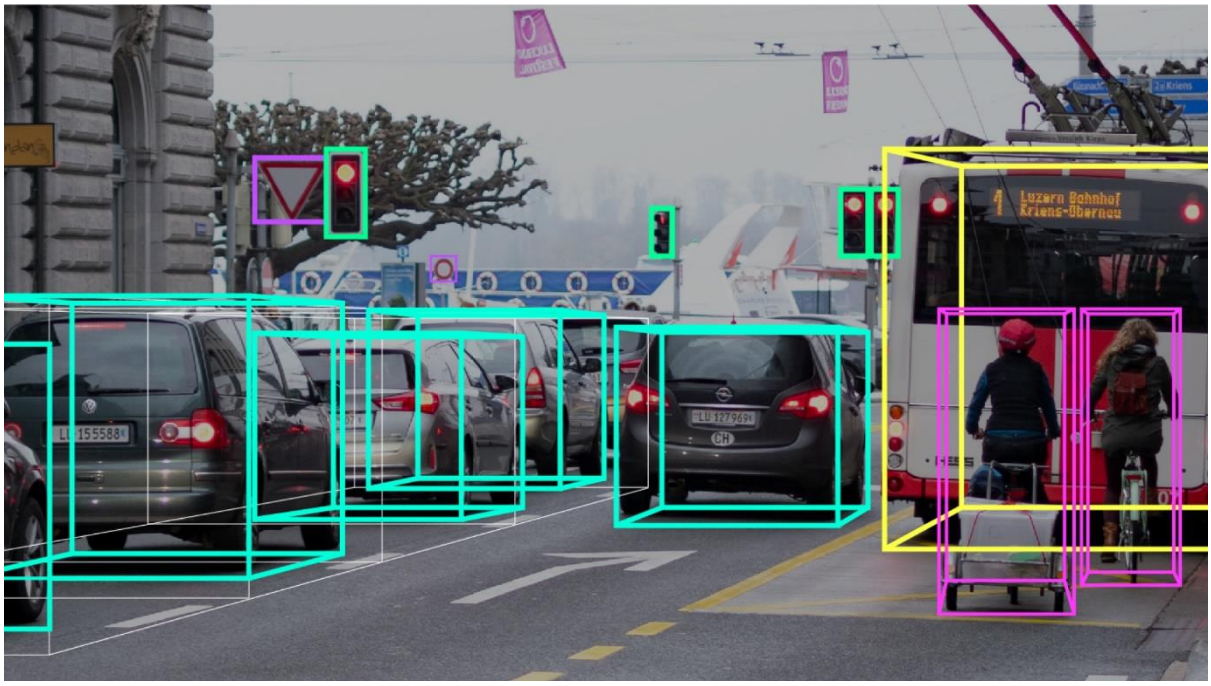
Σε αυτή την έκδοση του κώδικα, η λειτουργία HandleBoundingBoxesHeight() διαχειρίζεται πολλαπλά αντικείμενα, αποθηκεύοντας το ύψος σε pixels και τον τύπο κλάσης (class ID) για κάθε αντικείμενο που ανιχνεύεται. Το σύστημα καταγράφει αυτά τα δεδομένα για κάθε αντικείμενο και επιστρέφει τις πληροφορίες για μετέπειτα χρήση(*Εικόνα*16).

```python
def HandleBoundingBoxesHeight(predictions):
    names = model.names  # Get the list of class names from the YOLO model
    Bounding_Box_heights_in_pixels = []  # List to store the heights of all detected objects
    class_ids = []  # List to store the class IDs for all detected objects

    for r in predictions:  # Iterate through the predictions

        if stop_flag:  # Check if the stop flag is set, and if so, break the loop
            break

        for box in r.boxes:  # Iterate through the bounding boxes in each prediction
            detected_class_id = int(box.cls)  # Get the class ID for the detected object
            detected_name = names[detected_class_id]  # Get the class name using the class ID
            detected_names_Set.add(detected_name)  # Add the detected class name to the set

            # Store the class ID and name for later use
            class_ids.append(detected_class_id)
            detected_names_Set.add(detected_name)  # Add the detected class name to the set
            print(f"Detected: {detected_name}")  # Print the detected class name

            x1, y1, x2, y2 = box.xyxy[0]  # Get the coordinates of the bounding box

            # Calculate height in pixels by subtracting the top y-coordinate from the bottom y-coordinate
            height_in_pixels = y2 - y1
            Bounding_Box_heights_in_pixels.append(height_in_pixels)  # Store the height for this object

            # Debugging: Print the bounding box height
            print(f"Bounding Box Height (in pixels): {height_in_pixels}")

    return Bounding_Box_heights_in_pixels, class_ids  # Return the list of all bounding box heights and corresponding class IDs
```

Εικόνα16. Συνάρτηση υπολογισμού ύψους bounding box για καθε κλάση.

Ο υπολογισμός του ύψους σε pixels είναι ζωτικής σημασίας για τον υπολογισμό της απόστασης του αντικειμένου από την κάμερα μέσω της τεχνικής της **τριγωνικής ομοιότητας**. Όσο μεγαλύτερο είναι το ύψος σε pixels, τόσο πιο κοντά βρίσκεται το αντικείμενο στην κάμερα. Αντίθετα, όσο μικρότερο είναι το ύψος σε pixels, τόσο πιο μακριά βρίσκεται το αντικείμενο.

Ένα παράδειγμα εικόνας που να αναπαριστά τα Bounding Boxes βρίσκεται παρακάτω:



Εικόνα17. Παράδειγμα απεικόνισης Bounding Boxes.

Ο κώδικας υπολογίζει το ύψος κάθε αντικειμένου ξεχωριστά, και αυτές οι τιμές χρησιμοποιούνται για την ακριβή εκτίμηση της απόστασης των αντικειμένων με βάση την τεχνική της τριγωνικής ομοιότητας.

Η υλοποίηση αυτή διαχειρίζεται πολλαπλά αντικείμενα σε πραγματικό χρόνο, διασφαλίζοντας ότι κάθε αντικείμενο που ανιχνεύεται επεξεργάζεται με ακρίβεια, ενώ η απόσταση κάθε αντικειμένου υπολογίζεται αυτόνομα.

3.3.g Υπολογισμός Focal Length

Το focal length (εστιακή απόσταση) είναι μια από τις πιο σημαντικές παραμέτρους μιας κάμερας και παίζει καθοριστικό ρόλο στον υπολογισμό της απόστασης των αντικειμένων από την κάμερα. Η εστιακή απόσταση επηρεάζει το πεδίο όρασης της κάμερας, καθώς και το μέγεθος των αντικειμένων που καταγράφονται στην εικόνα. Στη συγκεκριμένη πτυχιακή εργασία, το focal length χρησιμοποιείται για την εκτίμηση της απόστασης των αντικειμένων με τη χρήση της τριγωνικής ομοιότητας.

Το focal length ορίζεται ως η απόσταση μεταξύ του κέντρου του φακού και του αισθητήρα της κάμερας, όταν το αντικείμενο είναι σε εστίαση. Μετριέται συνήθως σε χιλιοστά (mm) στις αναλογικές φωτογραφικές μηχανές και στις κάμερες κινητών, ή όπως στην δική μας περίπτωση, σε pixels στις ψηφιακές κάμερες και τις εφαρμογές υπολογιστικής όρασης.

Το focal length διαφέρει ανάλογα με τον τύπο και το μοντέλο της κάμερας. Οι παράγοντες που επηρεάζουν την εστιακή απόσταση είναι:

- Το μέγεθος του φακού της κάμερας.
- Η απόσταση μεταξύ του φακού και του αισθητήρα.

- Ο τύπος της κάμερας (π.χ., DSLR, webcam, κάμερα κινητού).

Ο υπολογισμός του focal length για μια κάμερα μπορεί να γίνει μέσω της διαδικασίας calibration, δηλαδή ρύθμιση της κάμερας, είτε θεωρητικά είτε πρακτικά. Στην υπολογιστική όραση, το focal length μπορεί να μετρηθεί σε pixels και υπολογίζεται κατά το calibration της κάμερας, χρησιμοποιώντας γνωστά αντικείμενα και αποστάσεις.

Ο θεωρητικός τύπος για τον υπολογισμό της εστιακής απόστασης είναι ο εξής:

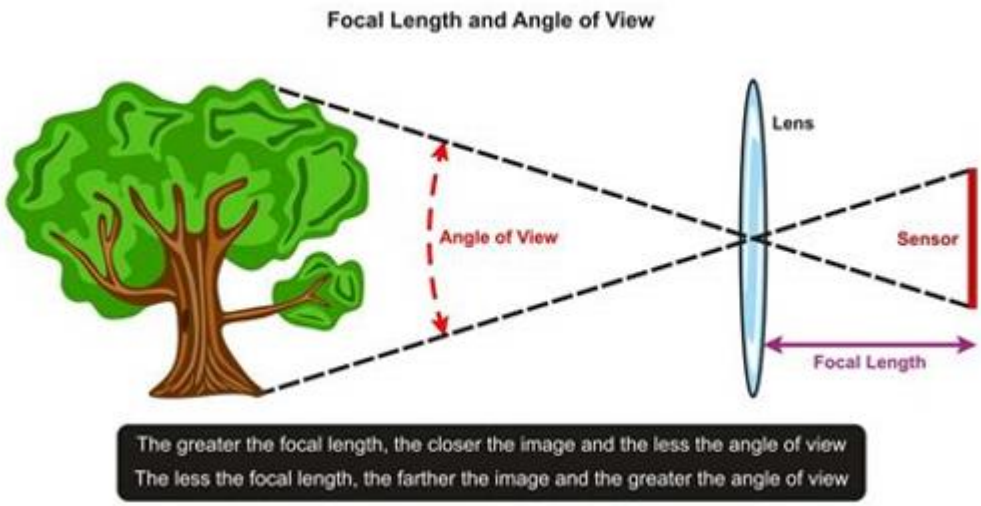
$$focalLength = \frac{D_{real} * H_{image}}{H_{real}}$$

Εξίσωση3. Υπολογισμός focal length.

Οπου:

- **D_real**: Η πραγματική απόσταση του αντικειμένου από την κάμερα (π.χ., 2 μέτρα).
- **H_image**: Το ύψος του αντικειμένου στην εικόνα (σε pixels).
- **H_real**: Το πραγματικό ύψος του αντικειμένου (π.χ., 1.75 μέτρα για έναν άνθρωπο).

Στο πλαίσιο της παρούσας πτυχιακής, το focal length παίζει καθοριστικό ρόλο στον υπολογισμό της απόστασης των αντικειμένων από την κάμερα. Χρησιμοποιώντας την τριγωνική ομοιότητα, το focal length επιτρέπει την ακριβή εκτίμηση της απόστασης κάθε ανιχνευμένου αντικειμένου*(Εικόνα18)*, εφόσον γνωρίζουμε το πραγματικό του ύψος.



Εικόνα18. Focal Length Visualization.

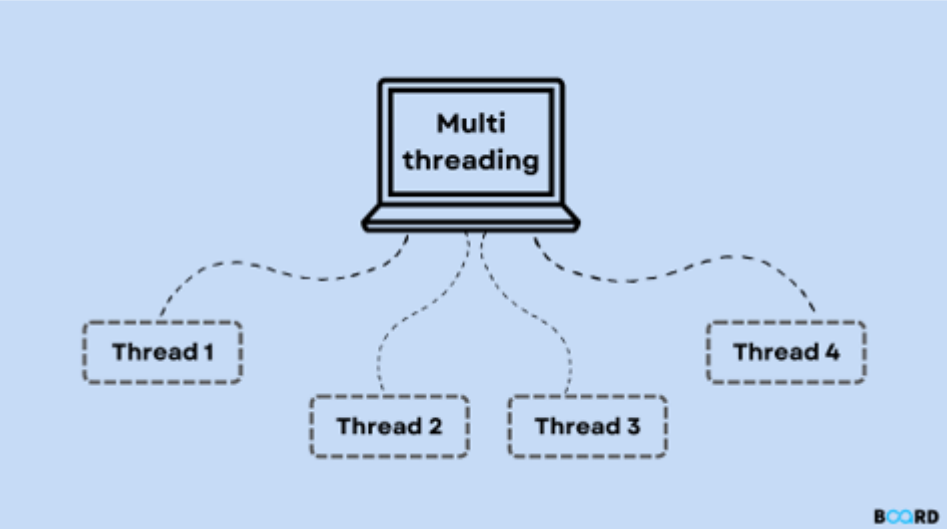
Στα πλαίσια του κώδικα του συστήματος μας, έχουμε ορίσει στο Focal Length την τιμή των 200 pixels*(Εικόνα19*).

```
focalLength = 200  # Pre-determined focal length of the camera in pixels (for distance calculation)
```

Εικόνα19. Αρχικοποίηση του Focal Length στα 200 pixels.

3.3.h Χρήση multi-threading

Το multi-threading(Εικόνα20) είναι μια τεχνική στον προγραμματισμό που επιτρέπει την εκτέλεση πολλαπλών εργασιών ταυτόχρονα (παράλληλα), αξιοποιώντας περισσότερους από έναν επεξεργαστές ή πυρήνες του επεξεργαστή. Στη συγκεκριμένη εφαρμογή, η χρήση του multi-threading ήταν κρίσιμη για τη διασφάλιση της ομαλής λειτουργίας του συστήματος σε πραγματικό χρόνο, χωρίς καθυστερήσεις ή διακοπές στη φωνητική ανατροφοδότηση και την ανίχνευση αντικειμένων.



Εικόνα20. Multi-Threading Visualization.

Η ανάγκη για multi-threading προέκυψε από την ταυτόχρονη εκτέλεση πολλών λειτουργιών:

- Ανίχνευση αντικειμένων μέσω του μοντέλου YOLOv8.
- Φωνητική ανατροφοδότηση στον χρήστη σε πραγματικό χρόνο.
- Αναγνώριση φωνητικών εντολών για την αλληλεπίδραση με τον χρήστη.
- Ανάλυση εικόνας και προβολή αποτελεσμάτων μέσω της κάμερας.

Στον κώδικα, το **multi-threading**(*Εικόνα*21) χρησιμοποιήθηκε για να διασφαλίσουμε ότι οι διαφορετικές διεργασίες, όπως η αναγνώριση αντικειμένων και η φωνητική ανατροφοδότηση, εκτελούνται παράλληλα, χωρίς να διακόπτουν η μία την άλλη.

```
import threading  # Import threading for running multiple threads concurrently
```

Εικόνα21. Φόρτωση των Threads.

Όταν δουλεύουμε με OpenCV, όπως στην δική μας περίπτωση, για να διαβάσουμε την κάμερα και να επεξεργαστούμε κάθε frame, η επεξεργασία μπορεί να πάρει χρόνο, ειδικά όταν περιλαμβάνει ανίχνευση αντικειμένων με μοντέλα όπως το YOLOv8. Αυτό σημαίνει ότι, αν όλες οι διεργασίες γίνονται σειριακά (σε ένα μόνο thread), το σύστημα μπορεί να φαίνεται ότι "παγώνει" μέχρι να ολοκληρωθεί κάθε διεργασία σε ένα frame πριν προχωρήσει στο επόμενο.

Στον κώδικα που ακολουθεί, χρησιμοποιούνται τα **threads** για να εκτελούνται ταυτόχρονα η φωνητική ανατροφοδότηση των ανιχνευμένων αντικειμένων και η ανίχνευση νέων αντικειμένων(*Εικόνα*22).

```
# This is a function that announces the detected class names
# Because the Multi-threading was announcing only 1 class name
# The detected class name is being added in a Set to be stored
# and once it is being announced by the text_to_speech() functions, it pops.

def announce_detected_names():
    global announced_thread_running  # Indicate that the function modifies the global variable

    announced_thread_running = True  # Set the thread running flag to True

    while detected_names_Set and not stop_tts_flag:  # Continue as long as there are names in the set
        with Threadlock:  # Acquire the thread lock to ensure thread-safe operations
            if detected_names_Set:  # Check if there are still names to announce
                detected_class_name = detected_names_Set.pop()  # Pop a name from the set
                text_to_speech(detected_class_name)
    announced_thread_running = False  # Set the thread running flag to False when done
```

Εικόνα22. Συνάρτηση ανακοίνωσης ανιχνεύσιμων αντικειμένων.

- **Συνάρτηση announce_detected_names()**: Αυτή η συνάρτηση εκτελείται σε ένα ξεχωριστό thread και ανακοινώνει τα ανιχνευμένα αντικείμενα. Χρησιμοποιεί το **Threadlock** για να διασφαλίσει ότι οι λειτουργίες ανακοινώσεων εκτελούνται με ασφάλεια χωρίς διακοπή από άλλες διεργασίες.
- **Thread Creation and Execution**: Το thread που δημιουργείται για την εκτέλεση της συνάρτησης **announce_detected_names()** εξασφαλίζει ότι η φωνητική ανατροφοδότηση μπορεί να συμβεί παράλληλα με την ανίχνευση αντικειμένων, χωρίς να καθυστερεί η μία διεργασία την άλλη.
- **Threadlock**: Χρησιμοποιείται για να αποτραπούν τυχόν ασυγχρονίες (race conditions) μεταξύ διαφορετικών threads που ενδέχεται να προσπαθούν να τροποποιήσουν τα ίδια δεδομένα ταυτόχρονα.

Χωρίς το multi-threading, το σύστημα θα εκτελούσε αυτές τις λειτουργίες σειριακά, δηλαδή η μία μετά την άλλη, γεγονός που θα οδηγούσε σε καθυστερήσεις, καθώς η ανίχνευση αντικειμένων θα έπρεπε να περιμένει την ολοκλήρωση της φωνητικής

ανατροφοδότησης. Με το **multi-threading**, το σύστημα μπορεί να ανιχνεύει αντικείμενα και να παρέχει φωνητική ανατροφοδότηση ταυτόχρονα.

3.4 Εκτέλεση Προγράμματος

3.4.a Εκκίνηση προγράμματος

Η εκκίνηση του προγράμματος γίνεται αυτόματα μόλις ο χρήστης δώσει τη φωνητική εντολή ή επιλέξει να ξεκινήσει το πρόγραμμα χειροκίνητα. Κατά την εκκίνηση, το σύστημα καλεί το κύριο πρόγραμμα, το οποίο εμφανίζει ένα μήνυμα καλωσορίσματος και περιμένει τη φωνητική εντολή του χρήστη για να καθορίσει σε ποιο mode θα προχωρήσει (Walking Mode ή Search Mode).

Η εφαρμογή παρέχει δύο βασικές λειτουργίες: το **Walking Mode** και το **Search Mode**. Κάθε λειτουργία επιτρέπει στο χρήστη να λαμβάνει πληροφορίες σχετικά με τα αντικείμενα στο περιβάλλον του, καθώς και να ανιχνεύει και να αναζητά συγκεκριμένα αντικείμενα μέσω της κάμερας, με την υποστήριξη του μοντέλου ανίχνευσης αντικειμένων **YOLOv8**(*Εικόνα*23).

```python
# AI Welcomes the user
text_to_speech("Hello! My name is Chloe. How Can I help you today?")  # Welcome message using text-to-speech

# Speech To Text
while True:  # Start an infinite loop to continuously listen for commands
    speech_text = speech_to_text()  # Convert the user's speech to text

    if "walk" in speech_text:  # If the user says "walk"
        walking_mode()  # Activate the walking mode
    elif "find" in speech_text:  # If the user says "find"
        lower_class_names = [name.lower() for name in model.names.values()]  # Convert all object class names from the YOLO model to 
        found_class = False  # Flag to track if a matching object class is found
        words = speech_text.lower().split()  # Split the user's speech into individual words and convert them to lowercase
        print("words: ", words)  # Debugging: Print the list of words from the speech for reference

        for i in range(len(words)):  # Iterate over each word in the command
            # Check if the current word matches any single-word class name in the YOLO model
            if words[i] in lower_class_names:
                temp = words[i]  # Store the matched word (class name) in the 'temp' variable
                found_class = True  # Set the 'found_class' flag to True since we found a match
                updated_search_mode(temp)  # Activate the search mode for the detected object class
                break  # Exit the loop once a match is found

            # Check for compound-word matches (two adjacent words)
            if i < len(words) - 1:  # Ensure there are at least two words left to check
                combined_words = f"{words[i]} {words[i + 1]}"  # Combine the current word with the next word
                if combined_words in lower_class_names:  # Check if the combined two-word phrase matches a class name
                    temp = combined_words  # Store the combined class name in 'temp'
                    found_class = True  # Set the 'found_class' flag to True since we found a match
                    updated_search_mode(temp)  # Activate the search mode for the detected object class
                    break  # Exit the loop once a match is found
```

Εικόνα23. Φωνητικό μενού χρήστη.

3.4.b Walking Mode

Το Walking Mode επιτρέπει στο χρήστη να λαμβάνει συνεχείς ενημερώσεις για τα αντικείμενα που βρίσκονται γύρω του ενώ κινείται. Η εφαρμογή ανιχνεύει

αντικείμενα σε πραγματικό χρόνο μέσω της κάμερας, χρησιμοποιώντας το μοντέλο YOLOv8, και ανακοινώνει φωνητικά τα αντικείμενα που εντοπίζει, καθώς και την απόστασή τους από τον χρήστη.

Όταν ο χρήστης δίνει φωνητική εντολή για την ενεργοποίηση του **Walking Mode**, η εφαρμογή ξεκινά την καταγραφή βίντεο(*Εικόνα*24) από την κάμερα και πραγματοποιεί ανίχνευση αντικειμένων σε πραγματικό χρόνο.

```
text_to_speech("Entering Walking Mode")  # Announce that the walking mode is being entered

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)  # Initialize video capture from the default camera
```

Εικόνα24. Εντολή καταγραφής βίντεο από την κάμερα.

Η κάμερα ανοίγει με τη χρήση του OpenCV και ξεκινάει να καταγράφει τα frames που θα χρησιμοποιηθούν για την ανίχνευση αντικειμένων. Το πρόγραμμα περιμένει μέχρι να λάβει ένα frame από την κάμερα.

Αφού λάβουμε ένα frame από την κάμερα, το πρόγραμμα προσπαθεί να ανιχνεύσει αντικείμενα χρησιμοποιώντας το **YOLOv8**(*Εικόνα*25).

```
while True:  # Start an infinite loop to continuously process frames from the camera
    ret, frame = cap.read()  # Capture a frame from the camera

    # if frame is read correctly ret is True
    while ret == False:  # If frame capture fails (ret is False)
        print("Can't receive frame. Retrying ...")  # Print a message indicating the failure
        cap.release()  # Release the camera resource
        cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)  # Reinitialize the video capture
        ret, frame = cap.read()  # Attempt to capture the frame again

    if not ret or frame is None:  # If ret is False or the frame is None (not captured correctly)
        print("Debug: Failed to capture frame, ret:", ret)  # Print a debug message
        continue  # Skip to the next iteration of the loop

    if frame is None or frame.size == 0:  # Check if the frame is invalid (None or empty)
        print("Invalid frame detected, skipping...")  # Print a message indicating an invalid frame
        continue  # Skip to the next iteration of the loop
```

Εικόνα25. Συνθήκες λήψης σωστού frame.

Με τη χρήση του **YOLOv8**, το σύστημα ανιχνεύει τα αντικείμενα σε κάθε frame που λαμβάνεται από την κάμερα. Τα αντικείμενα που ανιχνεύονται εμφανίζονται με bounding boxes με τον ακόλουθο τρόπο(*Εικόνα*26).

```
model_track = model.predict(source=frame)  # Use the model to make predictions on the current frame

result = model(frame, iou=0.4)[0]  # Get the detection results with an Intersection over Union (IoU) threshold of 0.4

detections = sv.Detections.from_ultralytics(result)  # Convert the detection results into a usable format
```

Εικόνα26. Εντολές ανίχνευσης αντικειμένων εντός του frame.

Αφού εντοπιστεί ένα αντικείμενο, το σύστημα χρησιμοποιεί την τριγωνική ομοιότητα για να υπολογίσει την απόστασή του από την κάμερα, βασιζόμενο στο ύψος του αντικειμένου σε pixels και το πραγματικό του ύψος με τον ακόλουθο κώδικα(*Εικόνα*27).

```
detections = sv.Detections.from_ultralytics(result)  # Convert the detection results into a usable format

height_in_pixels, class_ids = HandleBoundingBoxesHeight(model_track)  # Calculate the height of the detected objects in pixels

labels = []  # Initialize an empty list to store detection labels
# Iterate over each detected object and calculate its distance
for i, height in enumerate(height_in_pixels):
    objectClassID = class_ids[i]  # Use the correct class ID for each object
    if objectClassID is not None and isinstance(objectClassID, int) and 0 <= objectClassID < len(real_heights):
        # Calculate distance for the object using its real height and bounding box height
        distance = calculate_distance(real_heights[objectClassID], focalLength, height)
        steps = distance / 0.76  # Convert distance to steps
        detection_message = f"WM: {names[objectClassID]}, {distance:.1f} m, {steps:.1f} steps"  # Create a detection message
        labels.append(detection_message)  # Add the detection message to the labels list
        detected_names_Set.add(
            f"{names[objectClassID]} detected at {distance:.1f} m or {steps:.1f} steps")  # Add to set
```

Εικόνα27. Εντολές υπολογισμού απόστασης από την κάμερα.

Το σύστημα ανακοινώνει φωνητικά τα αντικείμενα που ανιχνεύθηκαν και την απόστασή τους μέσω του **multi-threading**, ώστε να μην καθυστερεί η ανίχνευση των επόμενων αντικειμένων.
Η διαδικασία της φωνητικής ανατροφοδότησης είναι υλοποιημένη με τη χρήση ενός **Set** (σύνολο) που αποθηκεύει τα ονόματα των ανιχνευμένων αντικειμένων, τα οποία στη συνέχεια ανακοινώνονται στον χρήστη. Χάρη στη χρήση του **multi-threading**, η ανατροφοδότηση εκτελείται ανεξάρτητα από την κύρια διαδικασία ανίχνευσης, διασφαλίζοντας ότι το πρόγραμμα δεν θα καθυστερήσει κατά την αναγνώριση νέων αντικειμένων.

Για να μην διακοπεί η ανίχνευση αντικειμένων κατά την αναπαραγωγή της φωνητικής ανατροφοδότησης, η διαδικασία αυτή εκτελείται σε ξεχωριστό **thread**. Αυτό επιτρέπει στο σύστημα να συνεχίζει την ανίχνευση αντικειμένων και την προβολή των αποτελεσμάτων χωρίς καθυστέρηση(*Εικόνα*28).

```
if detected_names_Set and not announced_thread_running:  # If there are detected names and the announcement thread is not running
    tts_thread = threading.Thread(target=announce_detected_names, daemon=True)  # Create a new thread to announce the detected names
    tts_thread.start()  # Start the thread
else:
    detected_names_Set.clear()  # Clear the set of detected names if no announcement is made
```

Εικόνα28. Εκκίνηση ξεχωριστού thread.

Αν το **Set** δεν είναι κενό και το thread των ανακοινώσεων δεν τρέχει ήδη, δημιουργείται ένα νέο thread για να ανακοινώσει τις ανιχνεύσεις.

Εάν δεν ανιχνευτούν αντικείμενα σε ένα frame ή αν έχουν ανακοινωθεί όλα τα αντικείμενα που ανιχνεύθηκαν, το **Set** εκκαθαρίζεται. Αυτό αποτρέπει την

ανακοίνωση αντικειμένων που δεν ανιχνεύονται πλέον και διασφαλίζει ότι το **Set** παραμένει έτοιμο να δεχτεί νέες ανιχνεύσεις.

Η συνάρτηση **announce_detected_names()** που αναφέραμε πιο πάνω, ανακοινώνει τα ονόματα των αντικειμένων από το **Set** χρησιμοποιώντας φωνητική ανατροφοδότηση. Καθώς διατρέχει το **Set**, κάθε αντικείμενο που ανιχνεύεται ανακοινώνεται και διαγράφεται από το σύνολο.

Τα αποτελέσματα της ανίχνευσης προβάλλονται στην οθόνη με τις ετικέτες που περιλαμβάνουν το όνομα του αντικειμένου, την απόσταση και τα βήματα(*Εικόνα*29):

```
frame = box_annotator.annotate(scene=frame, detections=detections)  # Annotate the frame with bounding boxes
frame = label_annotator.annotate(scene=frame, detections=detections, labels=labels)  # Annotate the frame with detection labels
```

Εικόνα29. Αποτελέσματα εντοπισμού εντός του frame.

Τέλος, το πρόγραμμα τερματίζεται αν ο χρήστης πατήσει το πλήκτρο «X» (*Εικόνα*30).

```
if (cv2.waitKey(1) & 0xFF == ord('x')):  # Check if the 'x' key is pressed
    stop_tts_flag = True  # Set the stop flag to True to stop the TTS thread
    exit()  # Exit the program
```

Εικόνα30. Τερματισμός προγράμματος.

3.4.b Search Mode

Η διαδικασία του **Search Mode** ξεκινά μόλις ο χρήστης δώσει φωνητική εντολή για την αναζήτηση ενός συγκεκριμένου αντικειμένου. Το πρόγραμμα επιβεβαιώνει την κατηγορία του αντικειμένου που δόθηκε από τον χρήστη και, αν το αντικείμενο υπάρχει στο μοντέλο **YOLOv8**, ξεκινά η διαδικασία αναζήτησης. Η κάμερα ανοίγει με τη χρήση του **OpenCV** για την καταγραφή βίντεο, και το πρόγραμμα περιμένει για τη λήψη frames(*Εικόνα*31).

```
text_to_speech("Entering Search Mode")  # Announce that the search mode is being entered

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)  # Initialize video capture from the default camera
```

Εικόνα31. Εντολή καταγραφής βίντεο από την κάμερα.

Πριν προχωρήσει η αναζήτηση, το σύστημα ελέγχει αν η κατηγορία του αντικειμένου που δόθηκε από τον χρήστη υπάρχει στο μοντέλο **YOLOv8**.
Εάν η κατηγορία του αντικειμένου δεν υπάρχει, το σύστημα ενημερώνει τον χρήστη ότι το αντικείμενο δεν βρέθηκε και το πρόγραμμα τερματίζεται. Σύμφωνα με αυτό(*Εικόνα*32).

```
if temp in list(model.names.values()):  # Check if the search term is a valid class in the YOLO model
    text_to_speech(f"Searching for {temp}")  # Announce what object is being searched for

    # Detect the YOLOv8 class ID by using the index of the list
    if list(model.names.values()).index(temp) >= 0 and (list(model.names.values()).index(temp)) <= len(list(model.names.values())):
        # Ensure the index is valid
        print("Found Class ", temp, list(model.names.values()).index(temp))  # Print the found class and its index
        class_index = list(model.names.values()).index(temp)  # Store the class index for use in detection
else:
    print("Parameter not found in the dataset. Current model needs to be trained further.")  # Inform the user if the class is not found
    text_to_speech("Invalid Searching Argument")  # Announce that the search term is invalid
    exit()  # Exit the program
```

Εικόνα32. Έλεγχος εγκυρότητας ζητούμενης κλάσης.

Το **YOLOv8** χρησιμοποιείται για την ανίχνευση μόνο των αντικειμένων που ανήκουν στην κατηγορία που ζήτησε ο χρήστης. Αυτό γίνεται με την εφαρμογή της μεθόδου **predict()** για την κατηγορία που ταιριάζει το αντικείμενο αναζήτησης. Η συγκεκριμένη διαδικασία υλοποιείται με το συγκεκριμένο κομμάτι κώδικα(*Εικόνα*33).

```
while True:  # Start an infinite loop to continuously process frames from the camera
    ret, frame = cap.read()  # Capture a frame from the camera

    distances = []  # Initialize an empty list to store the calculated distances
    labels = []  # Initialize an empty list to store detection labels
    current_detections = set()  # Initialize a set to store current detections

    pred = model.predict(source=frame, classes=class_index)  # Predict using YOLO model for the specific class

    # Run YOLOv8 on the frame with an Intersection over Union (IoU) threshold of 0.3
    results = model(frame, iou=0.3)[0]

    detections = sv.Detections.from_ultralytics(results)  # Convert the detection results into a usable format
```

Εικόνα33. Ανίχνευση αντικειμένων εντός του frame.

Όπως και στο **Walking Mode**, η απόσταση του αντικειμένου υπολογίζεται χρησιμοποιώντας την τριγωνική ομοιότητα. Με βάση το ύψος του αντικειμένου σε pixels και το πραγματικό ύψος, το σύστημα υπολογίζει την απόσταση σε μέτρα και τα βήματα που απαιτούνται για να φτάσει ο χρήστης στο αντικείμενο(*Εικόνα*34).

```
    heights_in_pixels, class_ids = HandleBoundingBoxesHeight(pred) # Calculate the height of the detected objects in pixels
    labels = []  # Initialize an empty list to store detection labels

    # Run YOLOv8 on the frame with an Intersection over Union (IoU) threshold of 0.3
    results = model(frame, iou=0.3)[0]

    detections = sv.Detections.from_ultralytics(results)  # Convert the detection results into a usable format

    # Iterate over each detected object and calculate its distance
    for i, height_in_pixels in enumerate(heights_in_pixels):
        objectClassID = class_ids[i]  # Get the correct class ID for each object
        if objectClassID == class_index:  # Only process the target class
            if objectClassID is not None and 0 <= objectClassID < len(real_heights):
                # Calculate the distance for the object using its real height and bounding box height
                distance = calculate_distance(real_heights[objectClassID], focalLength, height_in_pixels)
                steps = distance / 0.76  # Convert distance to steps
                detection_message = f"SM: {temp}, {distance:.1f} m, {steps:.1f} steps"  # Create a detection message
                labels.append(detection_message)  # Add the detection message to the labels list
                detected_names_Set.add(f"{temp} detected at {distance:.1f} m or {steps:.1f} steps")  # Add to set
```

Εικόνα34. Υπολογισμός απόστασης από την κάμερα.

Επίσης, Όπως και στο **Walking Mode**, η ανίχνευση των αντικειμένων που αναζητούνται ανακοινώνεται φωνητικά στον χρήστη μέσω ενός ξεχωριστού thread για να μην καθυστερεί η ανίχνευση των επόμενων αντικειμένων(Εικόνα35).

```
# Check if there are detected names and the announcement thread is not running
if detected_names_Set and not announced_thread_running:
    # Add the detection message to the set of detected names
    detected_names_Set.add(f"{temp} detected at {distance:.1f} m or {steps: .1f} steps")
    tts_thread = threading.Thread(target=announce_detected_names, daemon=True)  # Create a thread to announce detected names
    tts_thread.start()  # Start the thread
else:
    detected_names_Set.clear()  # Clear the set of detected names if no announcement is made
```

Εικόνα35. Εκκίνηση ξεχωριστού thread.

Τα αποτελέσματα της ανίχνευσης προβάλλονται στην οθόνη με τις ετικέτες που περιλαμβάνουν το όνομα του αντικειμένου, την απόσταση και τα βήματα(*Εικόνα*36).

```
frame = box_annotator.annotate(scene=frame, detections=detections)  # Annotate the frame with bounding boxes
frame = label_annotator.annotate(scene=frame, detections=detections, labels=labels)  # Annotate the frame with detection labels

cv2.imshow("Camera", frame)  # Display the annotated frame in a window
```

Εικόνα36. Αποτελέσματα εντοπισμού εντός του frame.

Τέλος, όπως και στο Walking Mode, Το πρόγραμμα τερματίζεται αν ο χρήστης πατήσει το πλήκτρο «**X**» (*Εικόνα*37).

```
    # Exit the loop and program if the 'x' key is pressed
    if cv2.waitKey(1) & 0xFF == ord('x'):
        stop_tts_flag = True # Set the stop flag to True to stop the TTS thread
        exit()  # Exit the program
```

Εικόνα37. Τερματισμός προγράμματος.

ΚΕΦΑΛΑΙΟ 4ο

Αποτελέσματα

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα αποτελέσματα της εφαρμογής του συστήματος ανίχνευσης αντικειμένων σε πραγματικό χρόνο, με χρήση του YOLOv8. Τα αποτελέσματα βασίζονται τόσο σε ποιοτική ανάλυση της λειτουργίας του συστήματος όσο και σε ποσοτικές μετρήσεις της απόδοσης. Τα αποτελέσματα περιλαμβάνουν screenshots από τη λειτουργία του προγράμματος.

Η κάμερα (webcam) που χρησιμοποιείται για τις δοκιμές, είναι μια webcam με διαστάσεις 640x480 pixels, καθώς αυτή είναι η διαθέσιμη κάμερα του εξοπλισμού στον οποίο αναπτύχθηκε ο παρών κώδικας.

Η μέτρηση της απόστασης των αντικειμένων από την κάμερα **δεν** είναι ακριβής, κυρίως λόγω των περιορισμών της κάμερας και της αβεβαιότητας όσον αφορά το πραγματικό μέγεθος των αντικειμένων στην πραγματική ζωή.

Στη συγκεκριμένη υλοποίηση, η απόσταση υπολογίζεται με βάση την τεχνική της τριγωνικής ομοιότητας, η οποία απαιτεί τη γνώση του πραγματικού ύψους των αντικειμένων. Ωστόσο, επειδή δεν είναι εφικτό να γνωρίζουμε το πραγματικό ύψος κάθε αντικειμένου, χρησιμοποιούμε τις μέσες τιμές ύψους (π.χ., για ανθρώπους, αυτοκίνητα κ.λπ.), γεγονός που μπορεί να προκαλέσει ανακρίβειες στην εκτίμηση της απόστασης.

Γνωρίζοντας το πραγματικό ύψος ενός αντικειμένου σε πραγματικό χρόνο μπορεί να είναι δύσκολο χωρίς προηγούμενη γνώση σχετικά με το αντικείμενο ή ένα εξωτερικό σύστημα μέτρησης όπως ένας αισθητήρας βάθους (Depth Sensor) ή οι κάμερες RealSense της Intel.

Επιπλέον, η ποιότητα της κάμερας, ο φωτισμός του χώρου στον οποίον γίνεται η εξέταση, η ανάλυση της εικόνας, καθώς και το εστιακό μήκος, επηρεάζουν τη δυνατότητα ακριβούς μέτρησης της απόστασης.

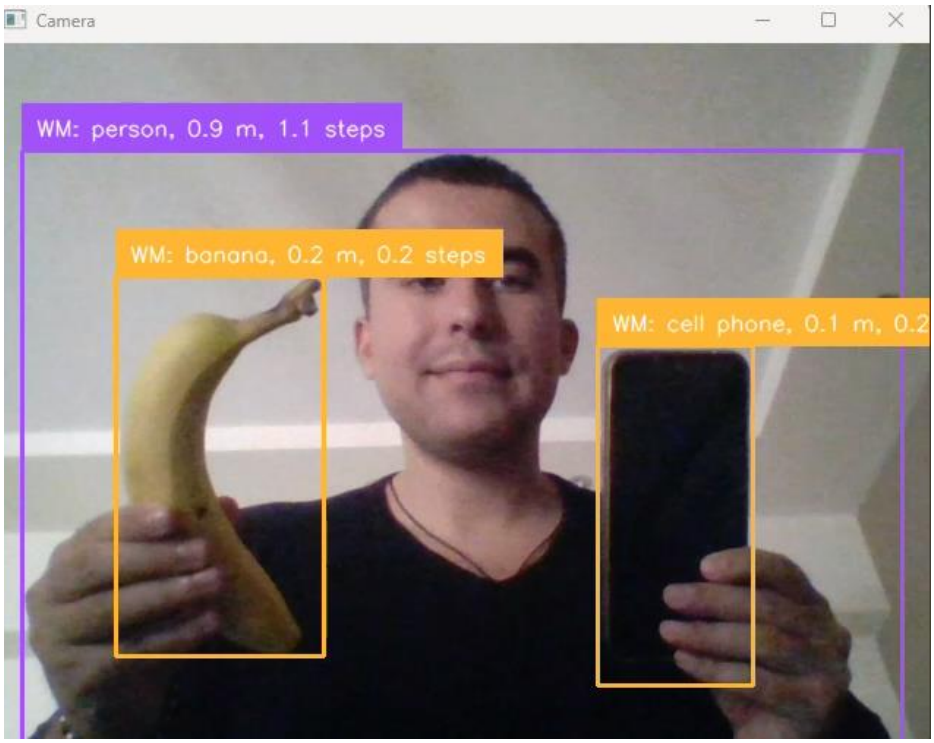
4.1 Walking Mode

Όπως προαναφέραμε, σε αυτή τη λειτουργία, το σύστημα ανιχνεύει τα αντικείμενα που βρίσκονται στο περιβάλλον του χρήστη σε πραγματικό χρόνο, ενώ παράλληλα παρέχει φωνητική ανατροφοδότηση για τα αντικείμενα που εντοπίζονται. Τα αντικείμενα εντοπίζονται μέσω του μοντέλου YOLOv8, και η κάμερα χρησιμοποιείται για την ανίχνευση αυτών, με τα αποτελέσματα να προβάλλονται στην οθόνη και να ανακοινώνονται στον χρήστη.

Τα αντικείμενα που ανιχνεύονται προβάλλονται στην οθόνη με bounding boxes γύρω από αυτά, και παράλληλα, το όνομά τους και η απόστασή τους από τον χρήστη ανακοινώνονται φωνητικά. Αυτό επιτρέπει στον χρήστη να γνωρίζει την ακριβή θέση των αντικειμένων και να κινείται με ασφάλεια στο περιβάλλον του.

Όταν το σύστημα εισέρχεται στο Walking Mode, ξεκινά να λαμβάνει συνεχώς frames από την κάμερα και να τα επεξεργάζεται με στόχο την ανίχνευση αντικειμένων. Τα αντικείμενα που ανιχνεύονται προβάλλονται στην οθόνη με bounding boxes γύρω από αυτά, και παράλληλα, το όνομά τους και η απόστασή τους από τον χρήστη ανακοινώνονται φωνητικά. Αυτό επιτρέπει στον χρήστη να γνωρίζει την ακριβή θέση των αντικειμένων και να κινείται με ασφάλεια στο περιβάλλον του.

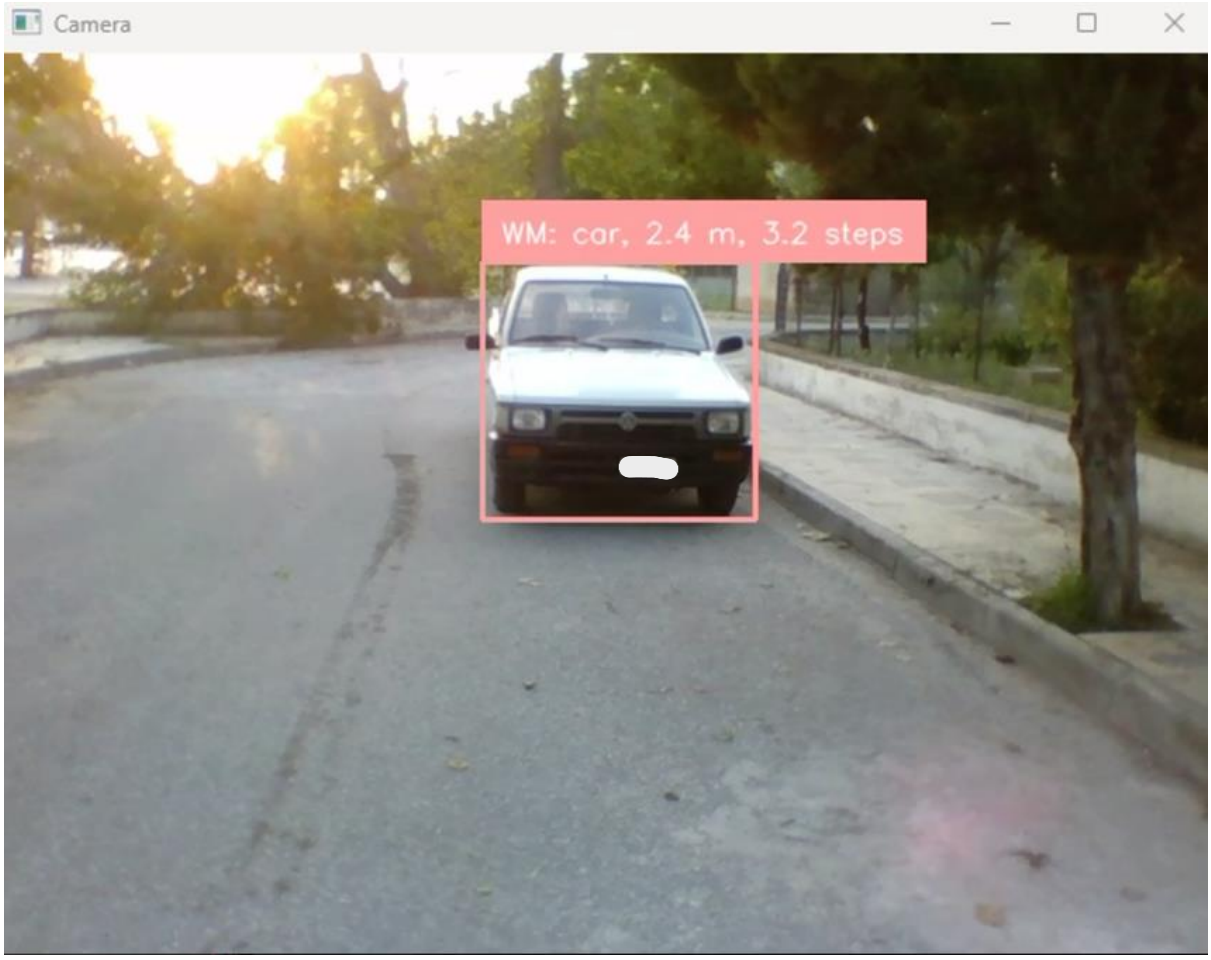
Παρακάτω, έχουμε τοποθετήσει ένα screenshot από την χρήση του συγκεκριμένου Mode, όπου ανιχνεύει το άτομο, σε καθαρό φωτισμό, κρατώντας 2 αντικείμενα (μια μπανάνα και ένα κινητό τηλέφωνο) , αντικείμενα τα οποία βρίσκονται στα classes του YOLOv8 Dataset(*Παράδειγμα*0).



Παράδειγμα0. Walking Mode σε Ανοιχτό Φωτισμό.

Ένας πολύ καθοριστικός παράγοντας για την ανάπτυξη του μοντέλου, είναι η ταχύτητα επεξεργασίας του frame. Καθώς, όπως προαναφέραμε, το μοντέλο έχει αναπτυχθεί για να λειτουργήσει στην πραγματική ζωή, χρειάζεται να αναγνωρίσει και να πραγματοποιήσει αποφάσεις σε πολύ άμεσο χρόνο, για να εξυπηρετήσει τον χρήστη.

Στην συνέχεια παρουσιάζεται η λειτουργία του μοντέλου στον εξωτερικό χώρο. Αυτή η δυνατότητα του μοντέλου να λειτουργεί αποτελεσματικά σε εξωτερικούς χώρους ενισχύει την πρακτική εφαρμογή του σε καθημερινές καταστάσεις, όπως η πλοήγηση και η αναγνώριση αντικειμένων για άτομα με προβλήματα όρασης(*Παράδειγμα1*).



Παράδειγμα1. Walking Mode σε εξωτερικό χόρο.

Παρακάτω, έχει σχεδιαστεί ένα γράφημα μέσω της βιβλιοθήκης **Matplotlib**(*Γράφημα1*) το οποίο απεικονίζει τον χρόνο επεξεργασίας του κάθε frame, καθώς και τον **μέσο χρόνο επεξεργασίας**(*Εικόνα*38).

Για την εξαγωγή των στατιστικών περιεχομένων, πραγματοποιήθηκαν κάποιες τροποποιήσεις σε τμήματα κώδικα, τα οποία δεν απαιτούνται για την ομαλή χρήση του μοντέλου(*Εικόνα*39).

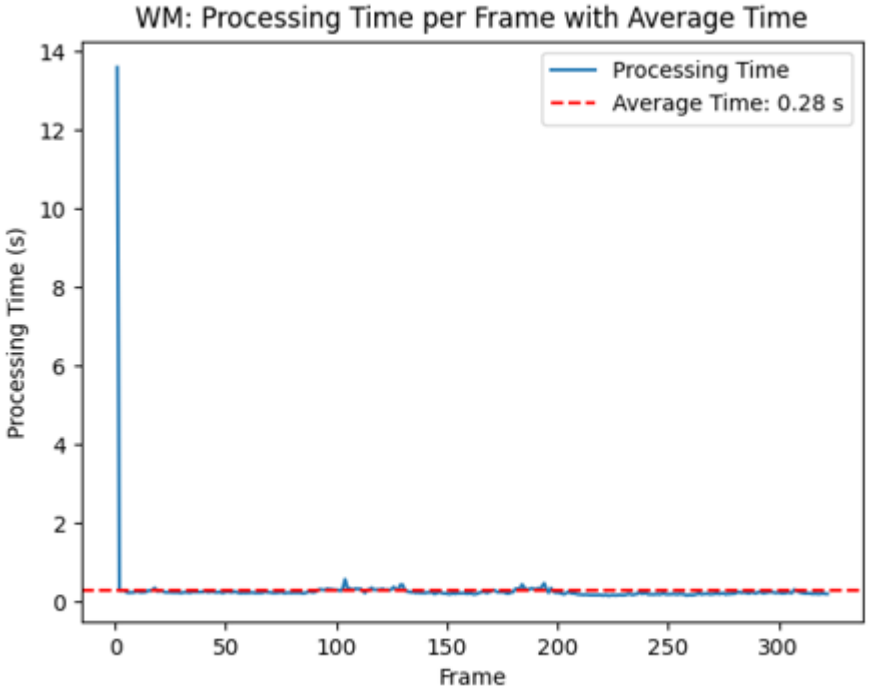
Η τροποποίηση του κώδικα που πραγματοποιήθηκε στην συνάρτηση του Walking Mode.

```
print("Model Tracking", model_track)  # Print the model tracking results for debugging
print("Specific detections = ", detections.class_id, "$$")  # Print the specific detections for debugging
end_time = time.time()
processing_time = end_time - start_time
processing_times.append(processing_time)
print(f"Current frame: {frame_count}")
print(f"Time to process frame: {processing_time} seconds")


frame_count += 1
```

Εικόνα38. Υπολογισμός χρόνου επεξεργασίας του frame.

```python
# Save processing times and frame numbers in a DataFrame
df = pd.DataFrame({
    'Frame': range(1, len(processing_times) + 1),
    'Processing Time (s)': processing_times
})

# Calculate the average processing time
average_processing_time = sum(processing_times) / len(processing_times)

# Save the DataFrame to a CSV file (optional)
df.to_csv('frame_processing_times.csv', index=False)

# Plot the frame processing times
plt.plot(df['Frame'], df['Processing Time (s)'], label='Processing Time')

# Plot the average processing time as a horizontal line
plt.axhline(y=average_processing_time, color='r', linestyle='--',
            label=f'Average Time: {average_processing_time:.2f} s')

plt.xlabel('Frame')
plt.ylabel('Processing Time (s)')
plt.title('WM: Processing Time per Frame with Average Time')
plt.legend()
plt.show()
```

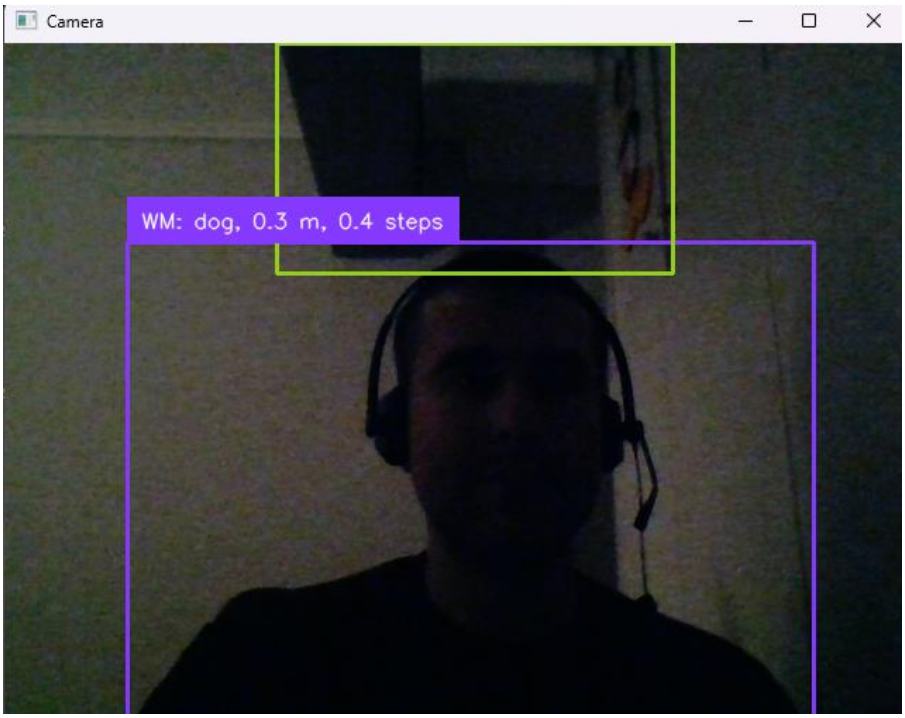
Εικόνα39. Εξαγωγή γραφήματος αναπαράστασης χρόνου επεξεργασίας.



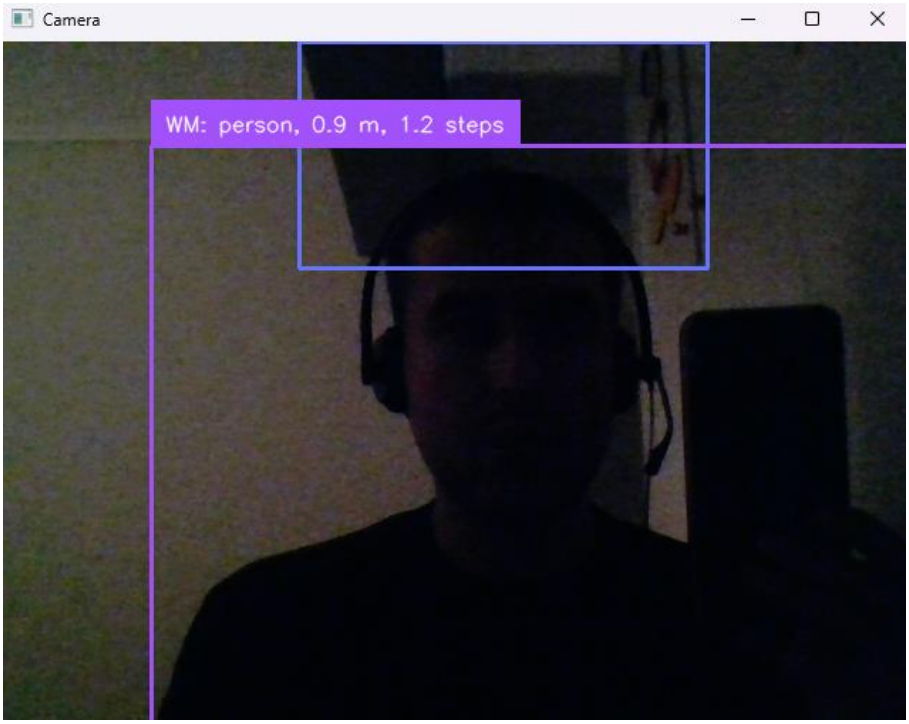
Γράφημα1. Μέσος Χρόνος Επεξεργασίας του Frame σε Ανοιχτό Φωτισμό.

Έναν πολύ σημαντικό ρόλο για την ανίχνευση των αντικειμένων, είναι ο φωτισμός του χώρου.

Παρά την υψηλή ακρίβεια του μοντέλου YOLOv8 σε βέλτιστες συνθήκες φωτισμού, παρατηρήθηκε σημαντική μείωση στην απόδοσή του όταν το περιβάλλον φωτισμού ήταν χαμηλό όπως στα ακόλουθα παραδείγματα (*Παράδειγμα*2)(*Παράδειγμα*3).



Παράδειγμα2. Walking Mode σε Χαμηλό Φωτισμό.

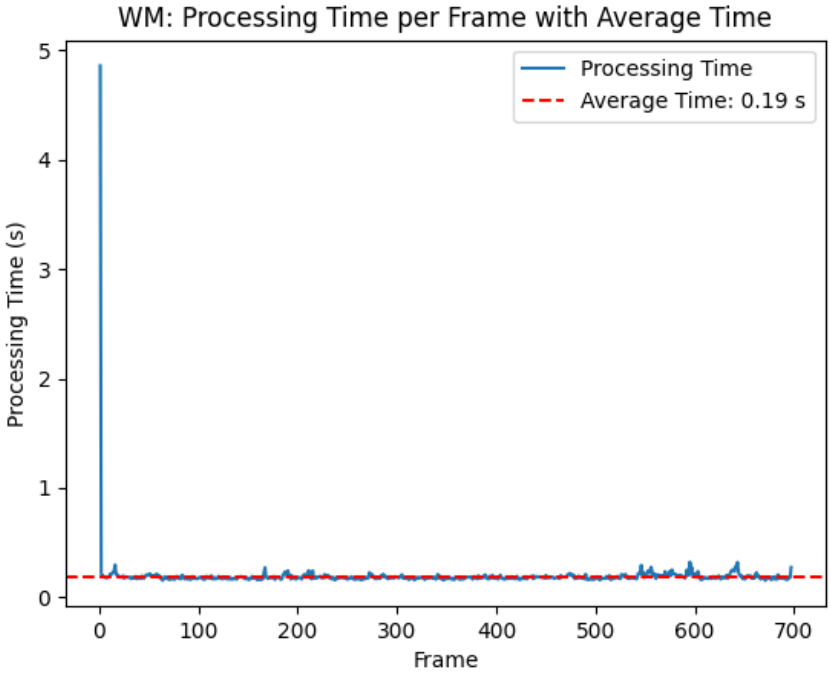


Παράδειγμα3. Walking Mode σε Χαμηλό Φωτισμό.

Αυτό είναι εμφανές από τη σύγκριση των αποτελεσμάτων σε συνθήκες φυσικού φωτισμού και χαμηλού φωτισμού. Σε χαμηλό φωτισμό, το μοντέλο δεν καταφέρνει

να αναγνωρίσει με ακρίβεια αντικείμενα που σε φωτεινές συνθήκες αναγνωρίζει χωρίς πρόβλημα. Για παράδειγμα, αντικείμενα όπως τον άνθρωπο στο παραπάνω παράδειγμα ή μικρότερα αντικείμενα (πχ. Κινητό τηλέφωνο) συχνά δεν ανιχνεύονται σωστά, ενώ η απόσταση που υπολογίζεται για αντικείμενα που βρίσκονται σε μεγαλύτερη απόσταση παρουσιάζει μεγαλύτερα σφάλματα.

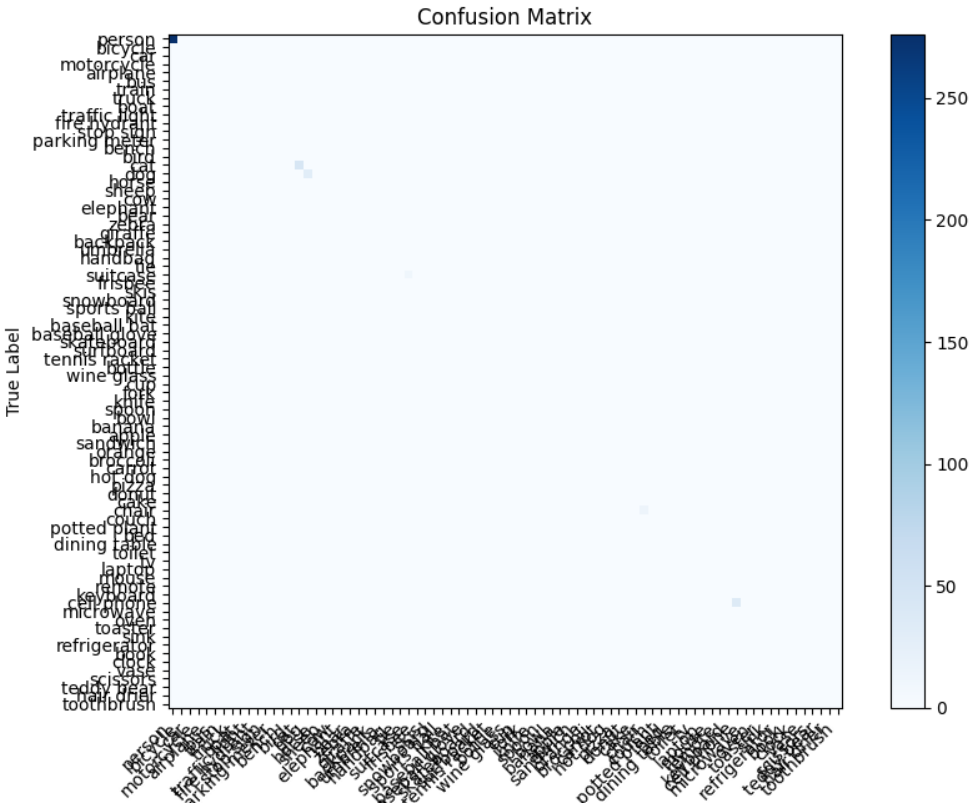
Παρακάτω, έχει σχεδιαστεί ένα γράφημα μέσω της βιβλιοθήκης **Matplotlib** το οποίο απεικονίζει τον χρόνο επεξεργασίας του κάθε frame, καθώς και τον **μέσο χρόνο επεξεργασίας** σε περιβάλλον χαμηλού φωτισμού(*Γράφημα*2)**.**



Γράφημα2. Μέσος Χρόνος Επεξεργασίας του Frame σε Χαμηλό Φωτισμό

Μπορούμε εύκολα να διαπιστώσουμε πως ο χρόνος επεξεργασίας του frame, δεν επηρεάζεται από το επίπεδο φωτισμού στον χώρο καθώς το μοντέλο είναι ήδη έτοιμο και προ-εκπαιδευμένο για να μπορέσει να ανιχνεύσει αντικείμενα.

Παρακάτω, με κατάλληλη τροποποίηση του κώδικα, έχουμε σχεδιάσει το Confusion Matrix για το συγκεκριμένο Mode(*Γράφημα*3).

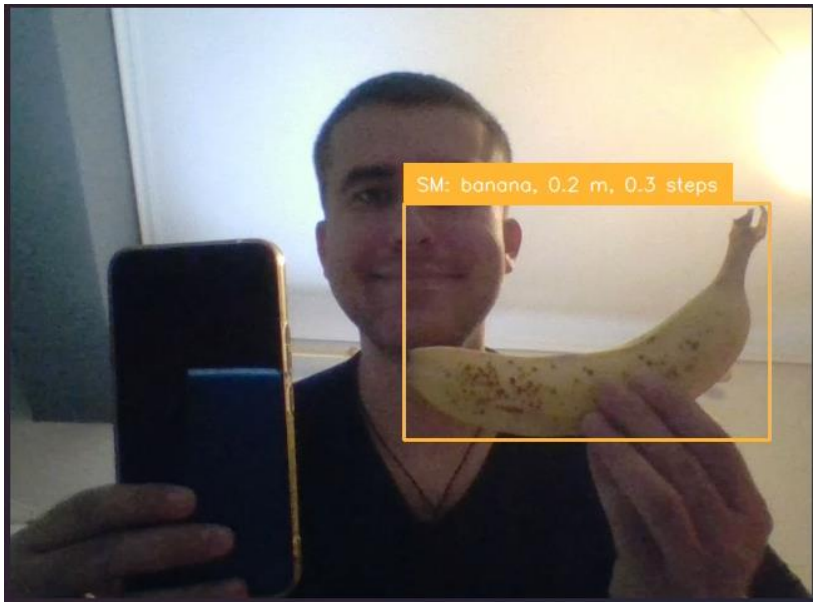


Γράφημα3. Confusion Matrix

4.2 Search Mode

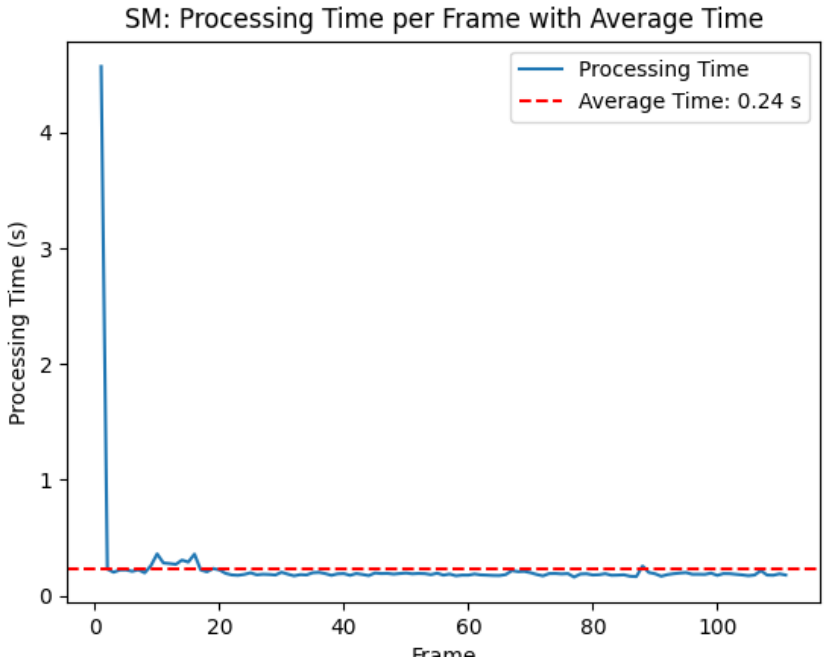
Όπως προαναφέραμε, όταν το σύστημα εισέρχεται στο Search Mode, ξεκινά να λαμβάνει συνεχώς frames από την κάμερα και να τα επεξεργάζεται με στόχο την ανίχνευση αντικειμένων που δίνονται από τον χρήστη χωρίς να εντοπίζει άλλα αντικείμενα. Τα αντικείμενα που ανιχνεύονται προβάλλονται στην οθόνη με bounding boxes γύρω από αυτά, και παράλληλα, το όνομά τους και η απόστασή τους από τον χρήστη ανακοινώνονται φωνητικά όπως και στο Walking Mode.

Παρακάτω, έχει τοποθετηθεί ένα screenshot με αρκετά αντικείμενα στο εύρος της κάμερας (Άνθρωπος, κινητό τηλέφωνο, μπανάνα), σε καθαρό φωτισμό, αλλά έχει ζητηθεί από τον χρήστη, να ανιχνευτεί μόνο το αντικείμενο της μπανάνας(*Παράδειγμα*4).



Παράδειγμα4. Search Mode σε Ανοιχτό Φωτισμό

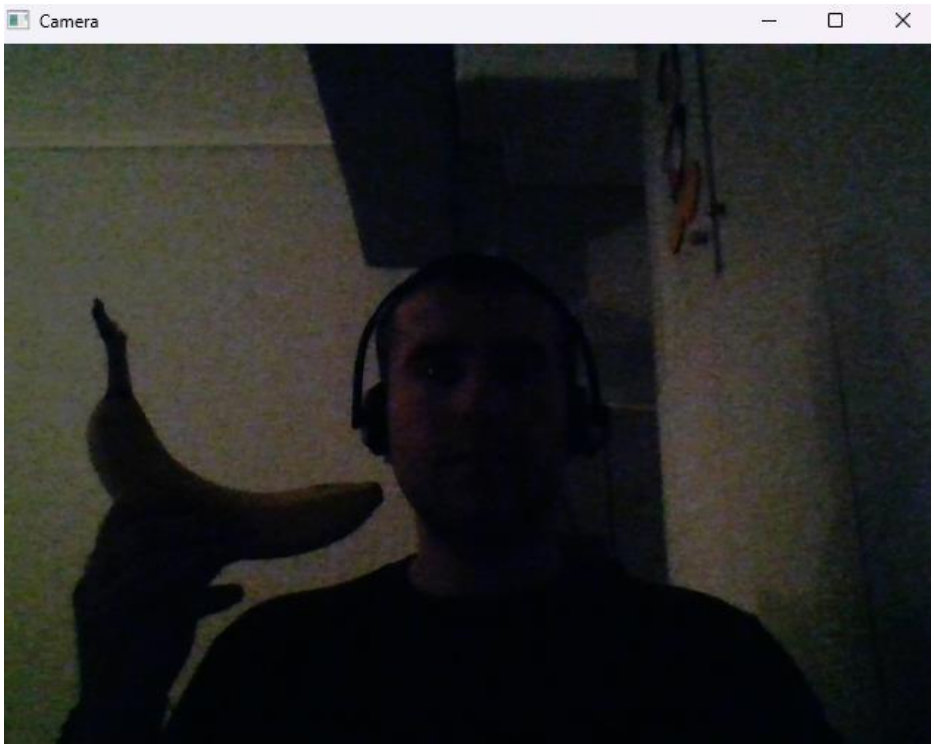
Παρακάτω, έχει σχεδιαστεί ένα γράφημα μέσω της βιβλιοθήκης **Matplotlib** το οποίο απεικονίζει τον χρόνο επεξεργασίας του κάθε frame, καθώς και τον **μέσο χρόνο επεξεργασίας** σε περιβάλλον φυσιολογικού φωτισμού(*Γράφημα4*).



Γράφημα4. Μέσος Χρόνος Επεξεργασίας του Frame σε Ανοιχτό Φωτισμό

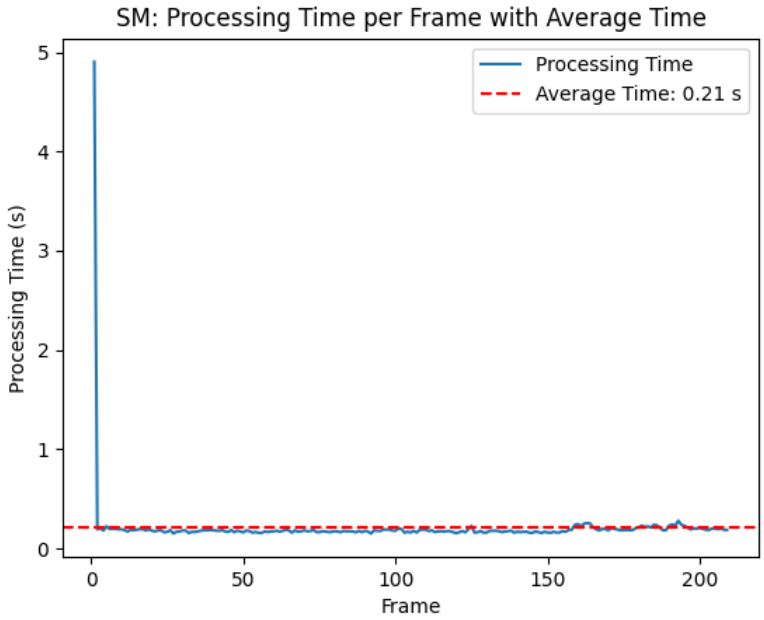
Όπως και στην περίπτωση του Walking Mode, παρατηρήθηκε σημαντική μείωση στην απόδοσή του όταν το περιβάλλον φωτισμού ήταν χαμηλό. Κατά την αναζήτηση του ίδιου αντικειμένου (μπανάνα) , το μοντέλο δεν μπορεί να ανιχνεύσει το αντικείμενο, με αποτέλεσμα να μην το σημειώνει με κάποιο Bounding Box, και

φυσικά δεν υπάρχει φωνητική ανατροφοδότηση όπως στο ακόλουθο παράδειγμα(*Παράδειγμα*5).



Παράδειγμα5. Search Mode σε Χαμηλό Φωτισμό

Παρακάτω, έχει σχεδιαστεί ένα γράφημα μέσω της βιβλιοθήκης **Matplotlib** το οποίο απεικονίζει τον χρόνο επεξεργασίας του κάθε frame, καθώς και τον **μέσο χρόνο επεξεργασίας** σε περιβάλλον χαμηλού φωτισμού(*Γράφημα*5)**.**



Γράφημα5. Μέσος Χρόνος Επεξεργασίας του Frame σε Χαμηλό Φωτισμό

Η εξαγωγή του παραπάνω γραφήματος πραγματοποιήθηκε μέσω του κώδικα στην ακόλουθη Εικόνα(*Εικόνα*40).

```
# Save processing times and frame numbers in a DataFrame
df = pd.DataFrame({
    'Frame': range(1, len(processing_times) + 1),
    'Processing Time (s)': processing_times
})


# The code below was used to extract the model statistics and has been commented out since it serves us no further

# # Calculate the average processing time
average_processing_time = sum(processing_times) / len(processing_times)

# Save the DataFrame to a CSV file
df.to_csv('frame_processing_times.csv', index=False)

# Plot the frame processing times
plt.plot(df['Frame'], df['Processing Time (s)'], label='Processing Time')

# Plot the average processing time as a horizontal line
plt.axhline(y=average_processing_time, color='r', linestyle='--',
            label=f'Average Time: {average_processing_time:.2f} s')

plt.xlabel('Frame')
plt.ylabel('Processing Time (s)')
plt.title('SM: Processing Time per Frame with Average Time')
plt.legend()
plt.show()
```

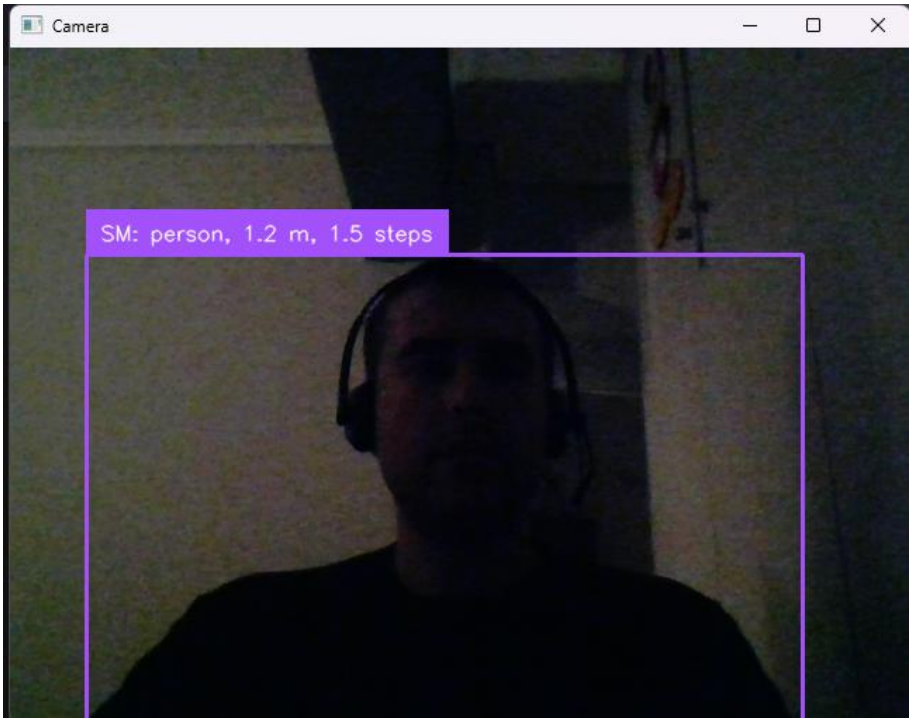
Εικόνα40. Εξαγωγή γραφήματος αναπαράστασης χρόνου επεξεργασίας.

4.3 Γενικές παρατηρήσεις

Οι χαμηλές συνθήκες φωτισμού επηρεάζουν άμεσα την ικανότητα του μοντέλου να ανιχνεύει αντικείμενα, καθώς η ανάλυση των καρέ που παράγονται από την κάμερα γίνεται λιγότερο καθαρή, γεγονός που δυσχεραίνει την αναγνώριση αντικειμένων.

Επιπλέον, η μείωση της φωτεινότητας προκαλεί αύξηση του θορύβου στην εικόνα(*Παράδειγμα*6), κάτι που μπορεί να οδηγήσει σε λανθασμένες προβλέψεις ή ακόμα και σε μη αναγνώριση αντικειμένων.

Η μείωση της ακρίβειας στις προβλέψεις υπό αυτές τις συνθήκες υποδεικνύει ότι για βελτιωμένη απόδοση του μοντέλου σε περιβάλλοντα με χαμηλό φωτισμό, θα πρέπει να εξεταστεί η χρήση τεχνικών προ επεξεργασίας εικόνας όπως η ενίσχυση φωτεινότητας ή η εφαρμογή πιο εξελιγμένων αλγορίθμων για τη μείωση του θορύβου.



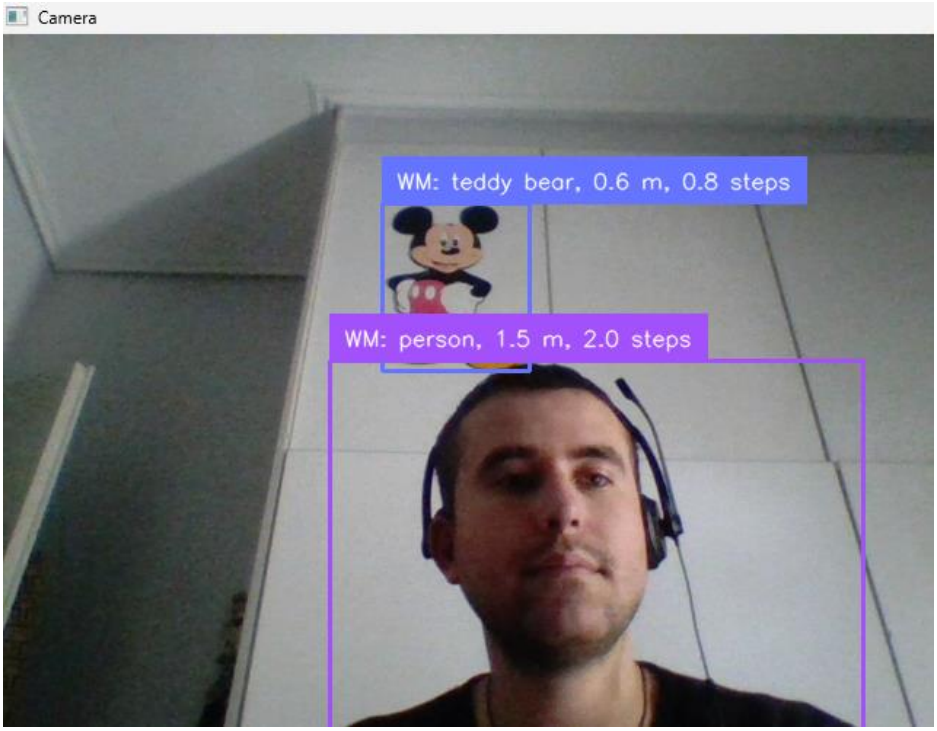
Παράδειγμα6. Χαμηλός φωτισμός.

Επιπροσθέτως, διαπιστώθηκε ότι η απόδοση της ανίχνευσης μπορεί να επηρεαστεί σημαντικά από τη γωνία λήψης της κάμερας σε σχέση με το αντικείμενο. Συγκεκριμένα, όταν η κάμερα βρίσκεται σε γωνία όπου τα χαρακτηριστικά του αντικειμένου δεν είναι εμφανή(*Παράδειγμα7*)(*Παράδειγμα8*) ή είναι μερικώς καλυμμένα, το μοντέλο παρουσιάζει μειωμένη ακρίβεια στην αναγνώριση και τον εντοπισμό του αντικειμένου.

Αυτό το πρόβλημα γίνεται πιο εμφανές σε περιπτώσεις όπου η γωνία ανίχνευσης είναι από το πλάι ή από πίσω σε αντικείμενα όπως τα πρόσωπα, τα οποία απαιτούν χαρακτηριστικά προσώπου για ακριβή αναγνώριση. Αντικείμενα με πολύπλοκα σχήματα ή συμμετρίες, όπως καρέκλες ή καναπέδες, μπορεί επίσης να ανιχνεύονται λανθασμένα ή καθόλου αν η γωνία λήψης τα κρύβει εν μέρει από το οπτικό πεδίο της κάμερας.

Παρόλο που το μοντέλο YOLOv8 είναι ιδιαίτερα αποδοτικό σε γενικές συνθήκες ανίχνευσης, οι περιορισμοί αυτοί αποδεικνύουν ότι η ακρίβεια εξαρτάται από τη θέση της κάμερας και την οπτική επαφή με τα αντικείμενα. Αυτή η αδυναμία επιβεβαιώθηκε σε αρκετές δοκιμές πεδίου όπου αντικείμενα που βρίσκονται σε περίεργες γωνίες ή είναι μερικώς καλυμμένα, ανιχνεύονταν λανθασμένα ή καθόλου.

Ακολουθούν μερικά παραδείγματα από screenshots κατά την διάρκεια δοκιμής:



Παράδειγμα7. Λανθασμένη εκτίμηση.

Παράδειγμα8. Λανθασμένη εκτίμηση απόστασης.

Στην παραπάνω περίπτωση, η γωνία της κάμερας δεν επιτρέπει στο μοντέλο να ανιχνεύσει σωστά την κατανομή των αντικειμένων στον χώρο αναζήτησης, με αποτέλεσμα να εμφανίζει λανθασμένα την απόσταση των αντικειμένων από την κάμερα.

ΚΕΦΑΛΑΙΟ 5ο

Συμπεράσματα

Η παρούσα πτυχιακή εργασία επικεντρώθηκε στην ανάπτυξη ενός έξυπνου βοηθού για άτομα με προβλήματα όρασης, αξιοποιώντας τεχνολογίες υπολογιστικής όρασης και ανίχνευσης αντικειμένων σε πραγματικό χρόνο. Χρησιμοποιήθηκε το μοντέλο YOLOv8, το οποίο επέδειξε εξαιρετικές δυνατότητες ανίχνευσης σε διάφορες συνθήκες φωτισμού και περιβάλλοντος, ενώ παράλληλα η ενσωμάτωση φωνητικής ανατροφοδότησης βελτίωσε την αλληλεπίδραση του συστήματος με τον χρήστη.

Αναλύοντας τα αποτελέσματα της εφαρμογής, μπορούμε να συμπεράνουμε ότι η τεχνολογία αυτή μπορεί να προσφέρει πρακτική υποστήριξη στην καθημερινότητα των ατόμων με προβλήματα όρασης. Η ικανότητα του συστήματος να ανιχνεύει και να αναγνωρίζει αντικείμενα όπως κινητά τηλέφωνα, καρέκλες, φώτα κυκλοφορίας, και άλλα, επιτρέπει στον χρήστη να πλοηγείται πιο εύκολα στο περιβάλλον του, παρέχοντας μια μορφή "εικονικής όρασης".

Ωστόσο, η απόδοση του συστήματος εξαρτάται από εξωτερικούς παράγοντες, όπως ο φωτισμός και η ποιότητα της κάμερας. Σε χαμηλό φωτισμό, η ακρίβεια των προβλέψεων μειώνεται σημαντικά, γεγονός που καθιστά το σύστημα λιγότερο αξιόπιστο σε σκοτεινές συνθήκες. Αυτό το ζήτημα θα μπορούσε να αντιμετωπιστεί με τη χρήση πιο προηγμένων τεχνικών προεπεξεργασίας εικόνας ή τη χρήση κάμερας με καλύτερη απόδοση σε συνθήκες χαμηλού φωτισμού.

Η συνδυασμένη χρήση των τεχνολογιών YOLO v.8, OpenCV, και των τεχνικών τριγωνικής ομοιότητας δημιουργεί ένα ισχυρό σύστημα αναγνώρισης αντικειμένων και προσώπων σε πραγματικό χρόνο, προσφέροντας ταχύτητα και ακρίβεια στην αναγνώριση, ακόμη και σε απαιτητικές συνθήκες. Η χρήση αυτών των τεχνολογικών βάσεων είναι ζωτικής σημασίας για την ανάπτυξη αποτελεσματικών βοηθητικών συστημάτων για άτομα με προβλήματα όρασης, επιτρέποντάς τους να αλληλεπιδρούν με το περιβάλλον τους με περισσότερη αυτονομία και ασφάλεια.

Όσον αφορά τις δυνατότητες μελλοντικής επέκτασης, η μεταφορά αυτής της τεχνολογίας σε κινητές συσκευές θα μπορούσε να προσφέρει μεγαλύτερη ευελιξία και χρηστικότητα. Θεωρητικά, η υλοποίηση του συστήματος σε κινητές συσκευές, όπως smartphones, είναι εφικτή, καθώς οι σύγχρονες συσκευές διαθέτουν ισχυρούς επεξεργαστές και ικανοποιητικές δυνατότητες υπολογιστικής όρασης μέσω των καμερών τους. Παράλληλα, η τεχνολογία φωνητικής ανατροφοδότησης θα μπορούσε εύκολα να ενσωματωθεί στα λειτουργικά συστήματα των κινητών, προσφέροντας μια ολιστική εμπειρία στον χρήστη.

Η μεγαλύτερη πρόκληση στην υλοποίηση σε κινητές συσκευές θα ήταν η βελτιστοποίηση των μοντέλων υπολογιστικής όρασης για να λειτουργούν σε πραγματικό χρόνο με περιορισμένους πόρους υπολογιστικής ισχύος και μπαταρίας. Η χρήση ελαφρύτερων μοντέλων όπως το YOLOv8n (nano) είναι ένα πρώτο βήμα προς αυτή την κατεύθυνση. Επιπλέον, οι βελτιώσεις στη διάρκεια ζωής της μπαταρίας, η χρήση αποτελεσματικότερων αλγορίθμων και η αξιοποίηση cloud υποδομών θα μπορούσαν να βελτιώσουν την απόδοση και την αξιοπιστία της εφαρμογής σε κινητά.

Συνοψίζοντας, το σύστημα αυτό μπορεί να αποτελέσει μια σημαντική βοήθεια για τα άτομα με προβλήματα όρασης, παρέχοντας μια πρακτική λύση για την ανίχνευση αντικειμένων και την πλοήγηση στο περιβάλλον. Με μελλοντικές βελτιώσεις και προσαρμογές, η εφαρμογή του σε κινητές συσκευές θα μπορούσε να επεκτείνει τις δυνατότητές του και να καταστήσει το σύστημα πιο προσιτό και χρηστικό στην καθημερινότητα των χρηστών.

Θετικά:

Ακρίβεια Ανίχνευσης Αντικειμένων: Το μοντέλο YOLOv8 απέδειξε την ικανότητά του να ανιχνεύει αντικείμενα σε πραγματικό χρόνο με μεγάλη ακρίβεια. Η δυνατότητα να εντοπίζει καθημερινά αντικείμενα, όπως κινητά τηλέφωνα, καρέκλες, και φώτα κυκλοφορίας, ενισχύει την αποτελεσματικότητα του συστήματος στην υποστήριξη ατόμων με προβλήματα όρασης.

Πολυπλοκότητα και Απλότητα Χρήσης: Η φωνητική ανατροφοδότηση καθιστά το σύστημα εύκολο στη χρήση, ακόμα και από άτομα με ελάχιστη τεχνογνωσία, καθώς ο χρήστης μπορεί να αλληλοεπιδράσει με το σύστημα μέσω φωνητικών εντολών.

Πραγματικός Χρόνος Εκτέλεσης: Το σύστημα έχει σχεδιαστεί για να λειτουργεί σε πραγματικό χρόνο, δίνοντας στον χρήστη τη δυνατότητα να λάβει άμεσα πληροφορίες για τα αντικείμενα που βρίσκονται γύρω του.

Επεκτασιμότητα και Δυνατότητα Προσαρμογής: Το σύστημα μπορεί να προσαρμοστεί για διάφορες εφαρμογές πέρα από την ανίχνευση αντικειμένων για άτομα με προβλήματα όρασης, όπως σε ρομποτικά συστήματα ή αυτόνομα οχήματα, προσφέροντας δυνατότητες επέκτασης.

Ευελιξία Μοντέλου: Η χρήση του YOLOv8 παρέχει ευελιξία στη χρήση ελαφρύτερων μοντέλων (όπως το YOLOv8n), τα οποία είναι πιο κατάλληλα για εφαρμογές σε κινητές συσκευές.

Αρνητικά:

Απόδοση σε Συνθήκες Χαμηλού Φωτισμού: Ένα από τα βασικά αρνητικά είναι η απόδοση του συστήματος σε συνθήκες χαμηλού φωτισμού. Το μοντέλο ανίχνευσης αντιμετωπίζει δυσκολίες στην ακριβή αναγνώριση αντικειμένων

όταν ο φωτισμός είναι ανεπαρκής, γεγονός που μειώνει την αξιοπιστία του συστήματος σε δύσκολες περιβαλλοντικές συνθήκες.

Ανακρίβεια Υπολογισμού Απόστασης: Παρά την αποτελεσματικότητα του συστήματος στην ανίχνευση αντικειμένων, η εκτίμηση της απόστασης μέσω της τριγωνικής ομοιότητας ενδέχεται να παρουσιάζει αποκλίσεις, κυρίως όταν το πραγματικό ύψος ενός αντικειμένου διαφέρει από την εκτιμώμενη τιμή.

Απαιτήσεις Υλικού: Αν και το μοντέλο έχει βελτιστοποιηθεί για να τρέχει σε πραγματικό χρόνο, οι απαιτήσεις σε υλικό (hardware) είναι σχετικά υψηλές, κάτι που μπορεί να περιορίσει την εφαρμογή του σε πιο αδύναμα συστήματα ή κινητές συσκευές με χαμηλότερες επιδόσεις.

Εξάρτηση από την Κάμερα: Η ποιότητα της κάμερας παίζει σημαντικό ρόλο στην ακρίβεια του συστήματος. Οι χαμηλής ποιότητας κάμερες μπορεί να προκαλέσουν αστάθεια στην ανίχνευση, ιδίως σε συνθήκες με μεγάλο εύρος δυναμικού φωτισμού.

Έλλειψη Προσαρμοσμένης Εκπαίδευσης: Το σύστημα βασίζεται σε προκαθορισμένα δεδομένα του μοντέλου YOLOv8, τα οποία δεν έχουν εκπαιδευτεί για συγκεκριμένα περιβάλλοντα ή συνθήκες, κάτι που μπορεί να επηρεάσει την ακρίβεια της ανίχνευσης σε εξειδικευμένα περιβάλλοντα.

Με την ανάλυση αυτών των θετικών και αρνητικών, καταλήγουμε στο συμπέρασμα ότι το σύστημα αποτελεί μια καλή βάση για υποστήριξη ατόμων με προβλήματα όρασης, αλλά υπάρχουν περιθώρια για βελτιώσεις, ιδίως σε θέματα ακρίβειας, φωτισμού και υλικών απαιτήσεων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

2. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

3. Bradski, G. (2000). The OpenCV library. Dr. Dobb's Journal of Software Tools.

4. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

5. Lowe, D. G. (1999). Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision (pp. 1150-1157). IEEE.

6. World Health Organization. (2021). World report on vision. Geneva: World Health Organization.

7. Sánchez, J., & Aguayo, F. (2007). Mobile computing to aid the visually impaired. Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 259-260).

8. Manduchi, R., & Kurniawan, S. (2011). Mobility-related accidents experienced by people with visual impairment. A survey on the effectiveness of existing assistive technologies. Disability and Rehabilitation: Assistive Technology, 6(5), 358-366.

9. Bigham, J. P., & Cavender, A. C. (2009). Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1829-1838).

10. Lannizzotto, G., & La Rosa, F. (2014). Assistive technologies for the visually impaired: An innovative and easy-to-use ICT platform. IEEE International Conference on Systems, Man, and Cybernetics (pp. 2690-2695).

11. Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

12. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

13. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

14. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

15. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification

with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

16. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

17. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

18. Ishimaru, H., Kunze, K., & Bulling, A. (2018). A Survey on Computer Vision-Based Recognition for Smart Eyewear. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2(4), 1-29.

19. Russakovsky, O., Deng, J., Su, H., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115, 211-252.

20. Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep Learning Face Representation by Joint Identification-Verification. Advances in Neural Information Processing Systems, 27, 1988-1996.

21. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

22. Glenn Jocher, et al. (2023). YOLO by Ultralytics. GitHub Repository.

23. Newell, A., & Deng, J. (2017). Associative embedding: End-to-end learning for joint detection and grouping. Advances in Neural Information Processing Systems, 30, 2274-2284.

24. Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer-Verlag London.

25. Khoshdelniat, S., & Daliri, M. R. (2021). Deep Learning for Object Detection and Scene Perception in Assistive Technology: A Review. IEEE Access, 9, 23943-23965.

26. Muthukrishnan M., Maleki F., et al (2020) " Brief History of Artificial Intelligence", Neuroimag Clin N Am 30, 393-399, 1052- 5149/20/Elsevier Inc, doi.org/10.1016/j.nic.2020.07.004

27. McCulloch WS, Pitts W (1943): A logical calculus of the ideas immanent in nervous activity", The Bulletin of Mathematical Biophysics, 115-33

28. Rosenblatt F (1958): The perception: a probablistic motel for information storage and organization in the brain", Psycho Rev 65(6) 386-408

29. Turing AM (1950): Computing machinery and intelligence, Mind, LIX (236):433-60

30. McCorduck P (2004): Machines who think, 2nd ed, Natick (MA): A k Peters, Ltd

31.Schaeffer J., Plaat A (1997): Kasparov versus deep blue: the rematch, J Int Comput Games Assoc, 20:95-101

32. Hinton GE, Osindero S, Teh Y-W (2006); A fast learning algorithm for deep belief nets, Neural Comput, 18(7): 1527-54

33. Pierson HA., Gashler MS., (2017) Deep learning in robotics: a review of recent research, Act Robot, 31(16):821-35

34. Kuriakose B, S.R., Sandnes F E, Tools and technologies for blind and visually impaired navigation support: a review. IETE Technical Review, 2022. 39(1): p. 3-18.

35. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

36. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

37. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

38.He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

39. Kumar, D., & Maji, S. (2020). Vision-based Assistive Technologies for the Blind and Visually Impaired. Journal of Visualized Experiments (JoVE).

40. Microsoft. (2021). Seeing AI: App to Assist the Visually Impaired.

41. Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

42. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

43. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

.

44. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

45. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

46. Ballard, D. H., & Brown, C. M. (1982). Computer Vision. Prentice-Hall.

47. Fang, H., Cheng, Z., & Xiang, S. (2019). A Survey of the Application of Deep Learning in Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, 20(9), 3263-3283.

48. Budrionis An., Plikynas D., et al (2020): "Smartphone-based computer vision travelling aids for blind and visual impaired individuals: a systematic review", J Assistive Technology DOI: 10.1080/10400435.2020.1743381

49. Bourne, R. R. A., Flaxman, S. R., Braithwaite, T., Cicinelli, M. V., Das, A., Jonas, J. B., Keeffe, J., Kempen, J. H., Leasher, J., Limburg, H., Naidoo, K., Pesudovs, K., Resnikoff, S., Silvester, A., Stevens, G. A., Tahhan, N., Wong, T. Y., Taylor, H. R., & Vision Loss Expert Group. (2017). Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: A systematic review and meta-analysis. The Lancet. Global Health, 5(9), e888–e897.

50. Csapó, Á., & Wersényi, G. (2013). Overview of Auditory Representations in Human-machine Interfaces. ACM Comput. Surv., 46(2), 19:1–19:23.

51. Volodymyr Ivanchenko, James Coughlan, WilliamGerrey and Huiying Shen, Computer Vision-BasedClear Path Guidance for Blind Wheelchair Users, Proceedings of the 10th international ACMSIGACCESS conference on Computers andaccessibility, Pages 291-292, ISBN: 978-1-59593- 976-0, 2008.

52. Hend S. Al-Khalifa, Utilizing QR Code andMobile Phones for Blinds and Visually ImpairedPeople, ICCHP '08 Proceedings of the 11thinternational conference on Computers Helping International Journal of Scientific Research Engineering & Technology (IJSRET) Volume 2 Issue 8 pp 498-503 November 2013, ISSN 2278 -0882, IJSRET @ 2013 People with Special Needs, Pages 1065 – 1069, ISBN: 978-3-540-70539-0, 2008.

53. Jerey P. Bighamy, Chandrika Jayant H, Andrew Miller, Brandyn Whitez, and Tom Yeh, VizWiz::LocateIt - Enabling Blind People to Locate Objects in Their Environment, In Proceedings of the Workshop on Computer Vision Applications for the Visually Impaired (CVAVI 2010). San Francisco, California, 2010.

54. Jing Su, Alyssa Rosenzweig, Ashvin Goel, Eyal de Lara and Khai N. Truong, Timbremap: Enabling the Visually-Impaired to Use Maps on Touch- Enabled Devices, MobileHCI '10: Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, Pages 17-26, ISBN:

978-1-60558-835-3, 2010.

55. Roberto Manduchi, Sri Kurniawan and Homayoun Bagherinia, Blind guidance using mobile computer vision: a usability study, Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility, Pages 241-242, ISBN: 978-1-60558-881-0, 2010.

56. Y. Tian, C. Yi, and A. Arditi, Improving Computer Vision-Based Indoor Wayfinding for Blind Persons with Context Information, in Proc. ICCHP (2), pp.255-262, 2010.

57. F. Bellotti, R. Berta, A. DeGloria and M. Margarone, LodeStar: a mobile device to enhance visually impaired people experience of cultural and naturalistic places, in Proceedings of Re- Thinking Technology in Museums Conference, University of Limerick, Ireland. 26-27, 2011.

58. João José, Miguel Farrajota, João M.F. Rodrigues and J.M. Hans du Buf, The SmartVision local navigation aid for blind and visually impaired persons, international Journal of Digital Content Technology and its Applications Vol.5 No.5, 2011.

59. Brilhault, A., Kammoun, S., Gutierrez, O., Truillet, P., Jouffrais, C., Fusion of Artificial Vision and GPS to Improve Blind Pedestrian Positioning, New Technologies, Mobility and Security (NTMS), 4th IFIP International Conference on , Page(s): 1 – 5, 2011.

60. Bharat Bhargava, Pelin Angin and Lian Duan, A Mobile-Cloud Pedestrian Crossing Guide for the Blind, Department of Computer Science, Purdue University, 2011.

61. Boris Schauertey, Manel Martinezy, AngelaConstantinescuz and Rainer Stiefelhagen, AnAssistive Vision System for the Blind that Helps Find Lost Things, ICCHP'12 Proceedings of the13th international conference on Computers Helping People with Special Needs - Volume Part II, Pages 566-572, ISBN: 978-3-642-31533-6, 2012.

62. Alexander Dreyer Johnsen, Tor-MortenGrønli, Bendik Bygstad, Making Touch-BasedMobile Phones Accessible for the VisuallyImpaired, Norwegian School of IT, Publication: Norsk informatikkonferanse, ISBN: 9788232100132, 2012.

63. K. Matusiak, P.Skulimowski, and P. Strumitto, Object recognition in a mobile phone applicationfor visually impaired users, in Human SystemInteraction (HSI), The 6th International Conference, Sopot, Poland, Pages- 479 – 484, ISSN: 2158-2246, Print ISBN: 978-1-4673-5635- 0, 2013.

64. Afif, M., Ayachi, R., Pissaloux, E., Said, Y., & Atri, M. (2020). Indoor objects detection and recognition for an ICT mobility assistance of visually impaired

people. Multimedia Tools and Applications, 79 (41), 31645–31662.

65. Balasuriya, B., Lokuhettiarachchi, N., Ranasinghe, A., Shiwantha, K., & Jayawardena, C. (2017). Learning platform for visually impaired children through artificial intelligence and computer vision, 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA).

66. Dahiya, D., Issac, A., Dutta, M. K., Říha, K., & Kříž, P. (2018). Computer Vision Technique for Scene Captioning to Provide Assistance to Visually Impaired. 41st International Conference on Telecommunications and Signal Processing (TSP).

67. Duh, P.-J., Sung, Y.-C., Chiang, L.-Y. F., Chang, Y.-J., & Chen, K.-W. (2020). V-Eye: A Vision-based Navigation System for the Visually Impaired. IEEE Transactions on Multimedia.

68. Elmannai, W., & Elleithy, K. M. (2018). Computer Vision-Based Framework for Supporting the Mobility of the Visually Impaired. Bridgeport, CT: University of Bridgeport.

69. Jiang, H., Gonnot, T., Yi, W.-J., & Saniie, J. (2017). Computer vision and text recognition for assisting visually impaired people using Android smartphone. IEEE International Conference on Electro Information Technology (EIT).

70. Joshi, R. C., Yadav, S., Dutta, M. K., & Travieso-Gonzalez, C. (2020). Efficient Multi-Object Detection and Smart Navigation Using Artificial Intelligence for Visually Impaired People. Entropy, 22 (9), 941.

71. Li, X., Cui, H., Rizzo, J.-R., Wong, E., & Fang, Y. (2020). ross-Safe: A Computer Vision-Based Approach to Make All Intersection-Related Pedestrian Signals Accessible for the Visually Impaired. Advances in Intelligent Systems and Computing.

72. Cham Mahmud, S., Sourave, R. H., Islam, M., Lin, X., & Kim, J.-H. (2020). A Vision based Voice Controlled Indoor Assistant Robot for Visually Impaired People. IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS).

73. Mocanu, B., Tapu, R., & Zaharia, T. (2017). Seeing Without Sight — An Automatic Cognition System Dedicated to Blind and Visually Impaired People. International Conference on Computer Vision Workshops (ICCVW).

74. Jiang, H., Gonnot, T., Yi, W.-J., & Saniie, J. (2017). Computer vision and text recognition for assisting visually impaired people using Android smartphone. IEEE International Conference on Electro Information Technology (EIT).

75. Joshi, R. C., Yadav, S., Dutta, M. K., & Travieso-Gonzalez, C. (2020). Efficient Multi-Object Detection and Smart Navigation Using Artificial Intelligence for Visually Impaired People. Entropy, 22 (9), 941.