



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΧΡΗΣΗ ΒΑΘΙΑΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗ ΓΙΑ ΤΗΝ
ΥΠΟΒΟΗΘΗΣΗ ΑΝΘΡΩΠΩΝ ΜΕ ΠΡΟΒΛΗΜΑΤΑ ΟΡΑΣΗΣ

Ιωάννης Μιχαηλίδης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κολομβάτσος Κωνσταντίνος
Επικουρος καθηγητής

Λαμία έτος



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΧΡΗΣΗ ΒΑΘΙΑΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗ ΓΙΑ ΤΗΝ
ΥΠΟΒΟΗΘΗΣΗ ΑΝΘΡΩΠΩΝ ΜΕ ΠΡΟΒΛΗΜΑΤΑ ΟΡΑΣΗΣ

Ιωάννης Μιχαηλίδης

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κολομβάτσος Κωνσταντίνος
Επίκουρος καθηγητής

Λαμία έτος



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

ON THE USE OF DEEP LEARNING FOR ASSISTING
VISUAL IMPAIRED PEOPLE

Ioannis Michailidis

FINAL THESIS

ADVISOR

Kolomvatsos Konstantinos
Assistant Professor

Lamia year

«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 επεξεργάστηκαν, 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 εργασία έχει ως στόχο την ανάλυση των σύγχρονων τεχνολογιών της επιστήμης των υπολογιστών. Η μηχανική μάθηση και οι υποκατηγορίες της έχουν εισχωρήσει δυναμικά στην καθημερινότητα μας, βελτιώνοντας πολλές φορές όχι μόνο την ταχύτητα των διαδικασιών που έχουμε καθημερινά αλλά συχνά και τον τρόπο ζωής μας. Η χρήση των τεχνολογιών αυτών σε συνδυασμό με άλλα τεχνολογικά μέσα, όπως κάμερες και αισθητήρες, μπορούν να βελτιώσουν την ποιότητα ζωής πολλών ατόμων. Πιο συγκεκριμένα τα άτομα με οπτικά προβλήματα μπορούν να έχουν την δυνατότητα να κερδίσουν χρόνο στις καθημερινές τους διαδρομές, αλλά το σημαντικότερο είναι η ασφάλεια και η αυτοπεποίθηση που μπορεί να τους δοθεί. Παρακάτω θα αναλυθούν κάποια υπάρχοντα συστήματα για την υποβοήθηση των ατόμων με προβλήματα όρασης, καθώς επιπλέον θα παρουσιαστεί προσωπικά δημιουργημένο σύστημα εντοπισμού τους με βάση κάποια συγκεκριμένα χαρακτηριστικά. Η ανάπτυξη της εφαρμογής έγινε με την χρήση της γλώσσας προγραμματισμού Python, στο περιβάλλον του Google Colaboratory και αργότερα σε τοπικό επίπεδο. Για την πραγματοποίηση του χρειάστηκε η επεξεργασία ενός σετ φωτογραφιών που πραγματοποιήθηκε στην εφαρμογή Roboflow, καθώς είναι άμεσα συνδεδεμένη με το μοντέλο εντοπισμού YOLO. Υπάρχει αναλυτική παρουσίαση για τον τρόπο που χρησιμοποιήθηκαν. Τέλος, υπάρχουν κάποιες προτάσεις για εξέλιξη της τεχνολογίας αυτής και την ανάπτυξη παραπλήσιων τεχνολογιών για ποιοτικότερη και συνεχόμενη υποβοήθηση των ατόμων αυτών.

ABSTRACT

The present work aims to analyze modern technologies in computer science. Machine learning and its subcategories have dynamically entered our daily lives, often improving not only the speed of our daily processes but also our lifestyle. The use of these technologies in combination with other technological means such as cameras and sensors can improve the quality of life for many people. More specifically, people with visual impairments can gain time in their daily walking routes, but more importantly, the safety and confidence that can be provided to them is significant. Below, some existing systems for assisting visually impaired people will be analyzed, and a personal created detection system based on specific characteristics will be presented. The development of the application was done using the Python programming language, in the Google Colaboratory environment and later locally. For its implementation, a set of photographs was processed in the Roboflow application as it is directly connected to the YOLO detection model. There is a detailed presentation of how they were used. Finally, there are some suggestions for the development of this technology and the development of similar technologies for the better and more frequent assistance of these individuals.

Table of Contents

ΠΕΡΙΛΗΨΗ	I
ABSTRACT	II
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	2
(1.1 ΠΡΟΛΟΓΟΣ).....	2
(1.2 ΚΙΝΗΤΡΟ)	2
ΚΕΦΑΛΑΙΟ 2 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	4
(2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ)	4
(2.2 ΟΡΙΣΜΟΣ).....	4
(2.3 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ).....	4
(2.4 ΤΥΠΟΙ ΚΑΙ ΜΟΝΤΕΛΑ).....	5
(2.5 ΕΦΑΡΜΟΓΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ).....	8
ΚΕΦΑΛΑΙΟ 3 ΒΑΘΙΑ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	11
(3.1 ΕΙΣΑΓΩΓΗ).....	11
(3.2 ΟΡΙΣΜΟΣ).....	11
(3.3 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ).....	12
(3.4 DEEP LEARNING).....	12
(3.5 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ).....	13
(3.6 ΣΧΟΛΙΑ).....	16
ΚΕΦΑΛΑΙΟ 4 ΠΡΟΤΕΙΝΟΜΕΝΟ ΜΟΝΤΕΛΟ.....	17
(4.1 ΓΕΝΙΚΑ).....	17
(4.2 ΤΡΟΠΟΙ ΧΡΗΣΗΣ).....	17
(4.2.1 ΤΡΟΠΟΣ ΧΡΗΣΗΣ ROBOFLOW).....	17
(4.2.2 ΤΡΟΠΟΣ ΧΡΗΣΗΣ ΤΟΥ YOLO).....	19
(4.2.3 GOOGLE COLABORATORY & PYTHON).....	20
(4.2.3.1 ΥΛΟΠΟΙΗΣΗ ΣΕ GOOGLE COLABORATORY).....	20
(4.2.3.2 ΥΛΟΠΟΙΗΣΗ ΣΕ ΤΟΠΙΚΟ ΕΠΙΠΕΔΟ ΜΕ PYTHON).....	23
(4.3 ΥΠΑΡΧΟΝΤΑ ΣΥΣΤΗΜΑΤΑ ΥΠΟΒΟΗΘΗΣΗΣ).....	27
ΚΕΦΑΛΑΙΟ 5 ΠΕΙΡΑΜΑΤΙΚΗ ΑΠΟΤΙΜΗΣΗ	30
ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΕ GOOGLE COLABORATORY.....	31
ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΤΗ ΠΡΑΞΗ.....	37

ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ 41

(6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ)..... 41

(6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ)..... 42

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ..... 43

ΒΙΒΛΙΟΓΡΑΦΙΑ..... 44

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 ΠΡΟΛΟΓΟΣ

Οι περισσότεροι καθημερινά μετακινούμαστε στις πόλεις, είτε ως πεζοί είτε με οχήματα, διασχίζοντας το οδικό δίκτυο με σχετική ευκολία χάρις τους φωτεινούς σηματοδότες ή τις σημάσεις (κάθετες και οριζόντιες). Δεν ισχύει όμως το ίδιο για όλες τις ομάδες ανθρώπων. Υπάρχουν συνάνθρωποι μας, που αυτό που θεωρείται αυτονόητο και δεδομένο για κάποιους δεν είναι αυτονόητο για άλλους. Αναφερόμαστε στους ανθρώπους με προβλήματα όρασης, που μπορεί να είναι από μερική απώλεια όρασης έως και ολική. Τα αίτια των οπτικών διαταραχών ποικίλουν, μπορεί να είναι εκ γενετής ή να δημιουργηθούν κατά την πορεία της ζωής του ανθρώπου. Υπάρχουν άνθρωποι σε όλο το κόσμο με προβλήματα όρασης που δεν τους εξασφαλίζεται η σωστή και ασφαλής διάσχιση του οδικού δικτύου με τα προαναφερόμενα μέσα, ως αποτέλεσμα αυτό έχει να διακινδυνεύετε η σωματική τους ακεραιότητα. [1] Σύμφωνα με τον Π.Ο.Υ. ο αριθμός των ατόμων με ολικό πρόβλημα στην όραση τους ανέρχεται στους 40 με 45 εκατομμύρια παγκοσμίως, στην Ελλάδα ο αριθμός αυτός πλησιάζει τις 250 χιλιάδες. Έτσι λοιπόν, η αναγκαιότητα παροχής βοήθειας σε αυτούς τους ανθρώπους, με τα τεχνολογικά μέσα που διαθέτουμε κρίνεται πλέον απαραίτητη. Μας δίνεται η δυνατότητα να παρέχουμε ουσιαστική βοήθεια χωρίς να επεμβούμε στην καθημερινότητα τους και χωρίς να τους στιγματίσουμε. Υπάρχουν αρκετοί τρόποι για να βοηθηθούν και να εξυπηρετηθούν, εξασφαλίζοντας έτσι την σωματική τους ακεραιότητα, όχι μόνο κατά την παραμονή τους σε οδικό δίκτυο αλλά και στην καθημερινότητα τους. Μία προτεινόμενη λύση είναι η χρήση καμερών στους φωτεινούς σηματοδότες και όπου ακόμα είναι αναγκαίο να τοποθετηθούν, για τον εντοπισμό των ανθρώπων με προβλήματα όρασης και την ορθή καθοδήγηση τους. Με την βοήθεια κατάλληλων μοντέλων ανίχνευσης, καθώς και την χρήση ηχητικών ειδοποιήσεων (και όχι μόνο) που θα τους προσφέρει την ασφαλέστερη διέλευση των δρόμων.

1.2 ΚΙΝΗΤΡΟ

Με κίνητρο την ασφαλέστερη αλλά και καλύτερη διαβίωση των ανθρώπων με προβλήματα όρασης, θεωρήθηκε αναγκαίο η δημιουργία προγράμματος ανίχνευσης και εντοπισμού τους σε εξωτερικούς χώρους εντός πόλεως, τουλάχιστον κατά την προσέλευση τους στις διασταυρώσεις οδικών δικτύων. Αυτό θα έχει ως αποτέλεσμα την ενίσχυση της σωματικής και της ψυχικής τους υγείας, καθώς όχι μόνο θα έχουν την δυνατότητα να διασχίζουν το οδικό δίκτυο με μεγαλύτερη ασφάλεια αλλά και χωρίς φόβο, για το αν τυχόν παραβιαστεί η σήμανση και θέσουν τον εαυτό τους σε κίνδυνο. Το πρόβλημα που θα λυθεί μέσω αυτής της εφαρμογής είναι η ελλιπής «προστασία» σε ανθρώπους με προβλήματα όρασης που αδυνατούν να κινούνται μόνοι τους χωρίς κάποιον εξοπλισμό. Ειδικότερα σε μεγάλα αστικά κέντρα όπου η κυκλοφορία οχημάτων είναι αυξημένη. Με το σκεπτικό αυτό πραγματοποιήθηκε η δημιουργία κατάλληλου προγράμματος εντοπισμού τους εφόσον πληρούνται

κάποιες προϋποθέσεις, επιπλέον υπάρχει και χρήση ηχητικών ειδοποιήσεων καθ' όλη την διάρκεια ανίχνευσή τους για καλύτερο εντοπισμό.

Στη παρούσα εργασία γίνεται η ανάλυση και η χρήση του μοντέλου ανίχνευσης και εντοπισμού YOLO, με προσαρμοσμένο σύνολο δεδομένων (custom dataset), την χρήση της εφαρμογής Roboflow και των λειτουργιών που έχουν ενσωματωθεί. Για την πραγματοποίηση δημιουργίας προσαρμοσμένων δεδομένων χρησιμοποιήθηκαν classifications, τα οποία δεν προϋπήρχαν μέσα στα classes του μοντέλου YOLO, καθώς και το custom dataset που αποτελείται από ένα πακέτο εικόνων με τα κατάλληλα δεδομένα στις εικόνες αυτές, τα οποία έχουν επεξεργαστεί για την καλύτερη αποτίμηση των δεδομένων.

ΚΕΦΑΛΑΙΟ 2 Μηχανική Μάθηση

2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ [2]

Η μηχανική μάθηση πρωτοεμφανίστηκε περίπου την δεκαετία του 1950 ξεκινώντας την δημιουργία των πρώτων προτύπων και αλγορίθμων για την αυτόνομη εκμάθηση του υπολογιστή χωρίς ρητές εντολές, από τον **Arthur Lee Samuel** έναν Αμερικανό προγραμματιστή. Πιο συγκεκριμένα, το 1959 δημιουργείται το Πεδίο μελέτης της μηχανικής μάθησης από τον ίδιο. Πριν από αυτό το γεγονός υπήρξε ήδη ένα βιβλίο του Καναδού ψυχολόγου **Donald Hebb** με το τίτλο «*The Organization of Behavior*» το 1949, που αναφερόταν σε ένα θεωρητικό νευρωνικό δίκτυο. Ειδικότερα, για το πώς οι υπολογιστές θα μπορούν να αλληλοεπιδρούν μεταξύ τους. Από την αρχή της δημιουργίας της πληροφορικής απασχόλησε πολλούς επιστήμονες και επιστημονικά πεδία η δημιουργία υπολογιστών με την ικανότητα να μαθαίνουν χωρίς απόλυτα αμετάβλητες προγραμματιστικές εντολές, και την επικοινωνία των υπολογιστών μέσω των νευρωνικών δικτύων. Το 1981 υπήρξε αναφορά για την εκμάθηση ενός νευρωνικού δικτύου που μπορεί να αναγνωρίσει 40 χαρακτήρες από ένα τερματικό (terminal) υπολογιστή. Έκτοτε υπήρξαν περαιτέρω βελτιώσεις αλλά και αρκετές σκέψεις για αυτή την δυνατότητα των υπολογιστών. Από τον ορισμό του Tom M. Mitchell που ανέφερε ότι «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P , αν η επίδοσή του σε εργασίες της κλάσης T , όπως αποτιμάται από το μέτρο P , βελτιώνεται με την εμπειρία E », μέχρι τον Alan Turing με το ερώτημα «μπορούν οι μηχανές να κάνουν ό,τι εμείς(ως σκεπτόμενες οντότητες) μπορούμε να κάνουμε;».

2.2 ΟΡΙΣΜΟΣ [3]

Η μηχανική μάθηση είναι ένα σημαντικό κομμάτι της πληροφορικής. Είναι το κομμάτι που εστιάζει στην εκπαίδευση των υπολογιστών πάνω σε μοτίβα για την συνεχή βελτίωση των αποτελεσμάτων. Η εκπαίδευση δεν γίνεται από αλγορίθμους οι οποίοι είναι ρητά προγραμματισμένοι, αλλά έχουν την «ελευθερία να προσαρμόζονται» στο πρόβλημα. Για να επιτευχθεί αυτό υπάρχει η ανάγκη για πληθώρα δεδομένων, καθώς όσο περισσότερα (ιδίου τύπου) δεδομένα περιέχονται τόσο πιο αποδοτικό και ακριβές μπορεί να γίνει ένα μοντέλο. Επιπλέον θα πρέπει να υπάρξει και κατάλληλος αριθμός επαναλήψεων της εκμάθησης των δεδομένων για την καλύτερη αφομοίωση τους.

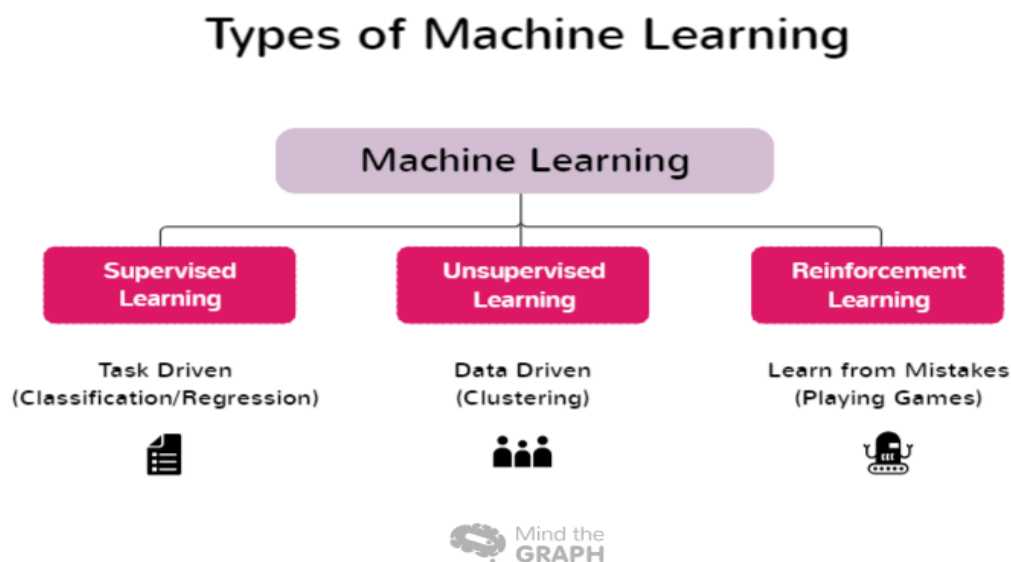
2.3 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ

Η μηχανική μάθηση υπάρχει στην ζωή μας αρκετά χρόνια, όμως μόνο προσφάτως έγινε ευρέως γνωστή σε ακόμα περισσότερους ανθρώπους με την ανάπτυξη της τεχνητής νοημοσύνης (Artificial intelligence / AI), όντας ένα από τα αρχικά στάδια της τεχνητής νοημοσύνης. Με την μηχανική μάθηση μπορούμε να επιτύχουμε σωρεία λύσεων σε προβλήματα που θα απαιτούσε από τους ανθρώπους αρκετό χρονικό διάστημα για τον υπολογισμό τους ή την εύρεση των πληροφοριών. Όπως για παράδειγμα είναι ο εντοπισμός μοτίβων, η πρόβλεψη αποτελεσμάτων κ.α.

πολύπλοκα και σύνθετα προβλήματα. Για την επίτευξη των αποτελεσμάτων αυτών χρειάζονται και οι κατάλληλες πληροφορίες (δεδομένα) που θα αντληθούν από τις εκάστοτε βάσεις δεδομένων που θα παρέχονται στην εφαρμογή. Σε όσα περισσότερα δεδομένα έχει πρόσβαση ο αλγόριθμος τόσο καλύτερα αποτελέσματα θα μας προσφέρει ως προς την ακρίβεια. Με την βελτίωση των βάσεων δεδομένων θα μπορούν να επιτευχθούν καλύτερα και γρηγορότερα αποτελέσματα. Με αυτό τον τρόπο θα χρειάζονται λιγότεροι υπολογιστικοί πόροι για κάθε λειτουργία του, άρα θα μπορούν να πραγματοποιηθούν περισσότερες διεργασίες και κατ' επέκταση περισσότερες έρευνες και αποτελέσματα. Η ανάγκη μειώσεων υπολογιστικών πόρων είναι «ζωτικής σημασίας» αφού από αυτό εξαρτάται η πίεση που θα υπάρχει στο συνολικό σύστημα και ο χρόνος ζωής τους, καθώς και η συνεχής και αδιάκοπη λειτουργία τους. Οι υψηλές θερμοκρασίες επηρεάζουν αρνητικά την λειτουργία, όχι μόνο των υπολογιστών αλλά και γενικότερα όλων των ηλεκτρικών κυκλωμάτων. Έτσι η βελτίωση των αλγορίθμων της μηχανικής μάθησης θα παρουσιάσει όχι μόνο βελτίωση του χρόνου του κώδικα, αλλά θα έχει και συνολική βελτίωση. Κάθε διεργασία θα πρέπει να βελτιστοποιείται όσο το δυνατόν περισσότερο.

2.4 ΤΥΠΟΙ / ΜΟΝΤΕΛΑ [3]

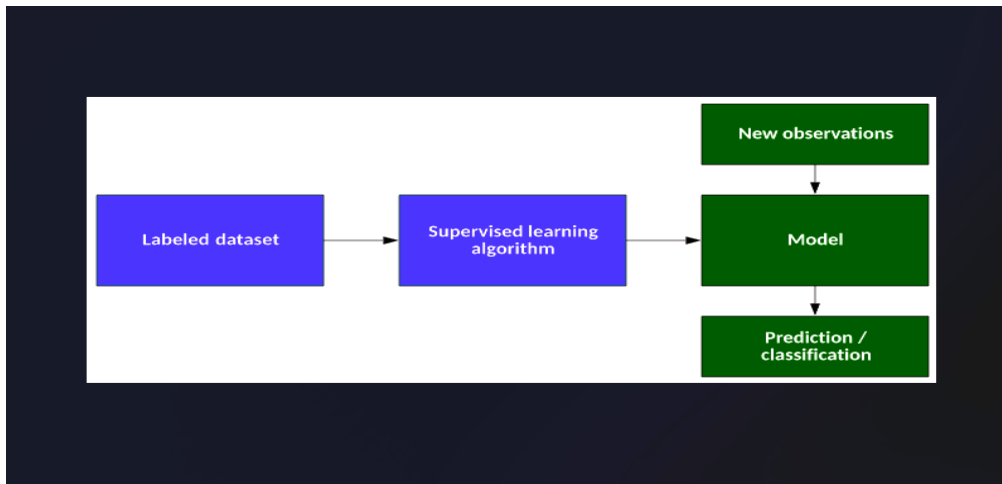
Η μηχανική μάθηση δεν είναι μονοδιάστατη, καθώς υπάρχουν πολλά προβλήματα τα οποία επιζητούμε να επιλύσουμε, έτσι υπάρχουν και διάφορα μοντέλα. Παρακάτω θα αναλύσουμε τα τέσσερα βασικά μοντέλα της μηχανικής μάθησης.



Εικόνα 1 <<https://mindthegraph.com/blog/el/machine-learning-in-science/>>

Στη παραπάνω εικόνα έχουμε τις τρεις βασικές κατηγορίες της μηχανικής μάθησης, αριστερά είναι η επιβλεπόμενη μάθηση, στο κέντρο έχουμε την μη επιβλεπόμενη μάθηση και δεξιά την ενισχυμένη μάθηση.

- **Επιβλεπόμενη μάθηση:** Αυτό το μοντέλο έχει ως κομμάτι εκπαίδευσης τα παραδείγματα που θα του αναθέσουμε εμείς. Δίνοντας του ένα σύνολο δεδομένων με ταμπέλες (labels), παρουσιάζοντας του τα αντικείμενα τα οποία αναζητάμε (κυρίως είναι αριθμοί, γράμματα και σύμβολα στις πιο απλές περιπτώσεις). Μαθαίνει από τα παλαιότερα δεδομένα και τα συνδυάζει με τα επόμενα για να κάνει την πρόβλεψη του με βάση την εμπειρία του. Έχει την δυνατότητα της αναπροσαρμογής για την βελτίωση του. Στη παρακάτω εικόνα βλέπουμε μια τυπική διαδρομή των δεδομένων για την επιβλεπόμενη μάθηση.

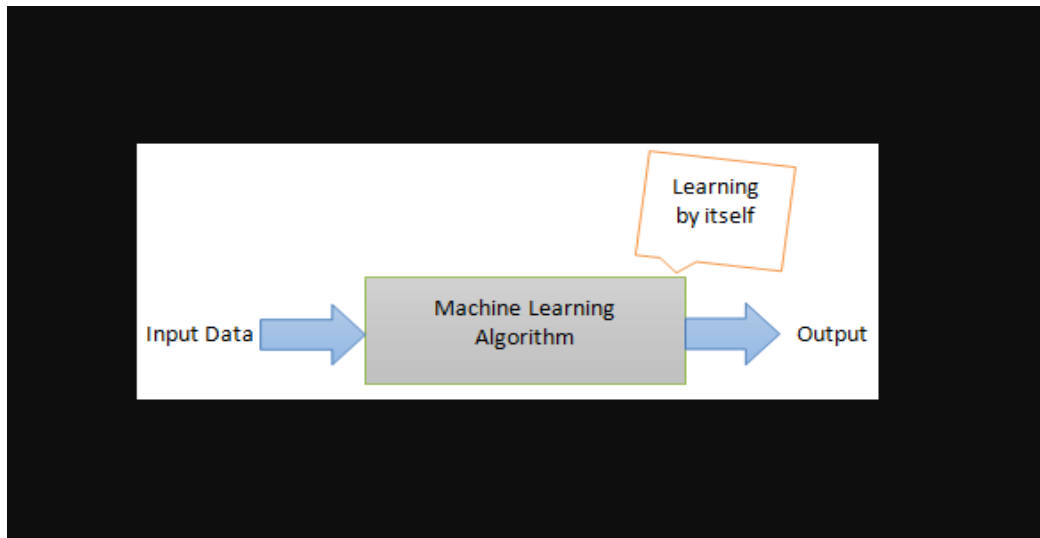


Εικόνα 2

< <https://www.analyticsvidhya.com/blog/2020/04/supervised-learning-unsupervised-learning/> >

Στη παραπάνω εικόνα βλέπουμε μια τυπική διαδρομή των δεδομένων για την επιβλεπόμενη μάθηση.

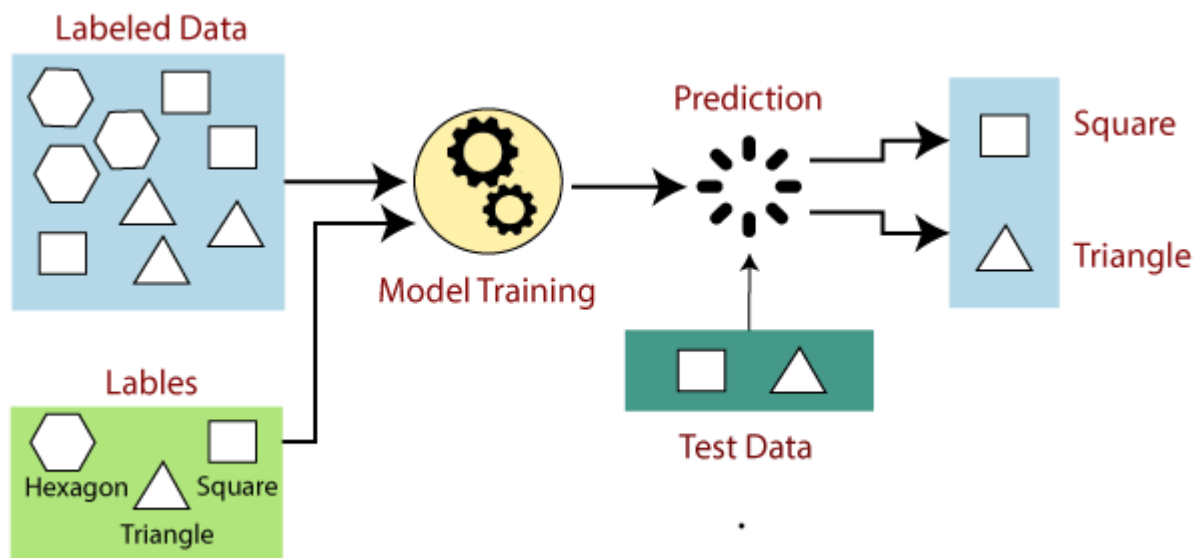
- **Μη επιβλεπόμενη μάθηση:** Σε αυτό το μοντέλο δεν έχουμε τόσο περιοριστικούς κανόνες καθώς «δεν υπάρχει κλειδί απάντησης». Με την χρήση των δεδομένων εισόδου εντοπίζει αυτόνομα επαναλαμβανόμενα μοτίβα μαθαίνοντας από αυτά, κατηγοριοποιώντας τα και τα ομαδοποιεί. Πρακτικά γίνεται μια διαδικασία αυτόματης μάθησης όπου μαθαίνει να εντοπίζει τα δεδομένα και να τα ξεχωρίζει μόνο του.



Εικόνα 3 <<https://www.analyticsvidhya.com/blog/2020/04/supervised-learning-unsupervised-learning/>>

Στη παραπάνω εικόνα απεικονίζεται η ροή των δεδομένων σε μια τυπική διαδικασία για την μη επιβλεπόμενη μάθηση. Καθώς και το σημείο που πραγματοποιεί την εκπαίδευση.

- Ημι-επιβλεπόμενη μάθηση: Έχοντας ένα μικρό σύνολο δεδομένων αυτό το μοντέλο εκπαιδεύεται να αναγνωρίζει όχι μόνο τα γνωστά για αυτών δεδομένα (labels), αλλά έχει την ικανότητα να αναγνωρίζει αντικείμενα χωρίς ετικέτα, δίνοντας του έτσι καλύτερη ακρίβεια όταν υπάρχει επανάληψη των δεδομένων, για τα αντικείμενα που δεν του αναθέσαμε labels, ο αλγόριθμος θα αναθέσει προσωρινά δικά του labels. Είναι σαν ένας συνδυασμός των δύο προηγούμενων μοντέλων.
- Ενισχυμένη μάθηση: Το τελευταίο μοντέλο στην κατηγορία έχει το χαρακτηριστικό ότι δεν δέχεται μονάχα τα δεδομένα, αλλά ενισχύεται και από τις δοκιμές τις οποίες επιχειρεί. Όσο περισσότερες διαφορές υπάρχουν μεταξύ των προσπαθειών, τόσο καλύτερα αποτελέσματα έχει, όμως οι πολλαπλοί λανθασμένοι εντοπισμοί ή απαντήσεις μπορεί να έχουν αρνητικό αντίκτυπο στην ακρίβεια. Ουσιαστικά το πρόγραμμα μαθαίνει από τα λάθη, στρέφεται στην βελτίωση πάνω στα λάθη που έχει εντοπίσει και όχι μόνο. Στην επόμενη εικόνα μπορούμε να δούμε το σύνολο των δεδομένων που δέχεται το πρόγραμμα, το σύνολο των ετικετών που εμείς έχουμε αναθέσει. Ύστερα γίνεται η εκπαίδευση του μοντέλου, δημιουργεί τις προβλέψεις του με βάση τα δεδομένα δοκιμής και τέλος μας δίνει το αποτέλεσμα που το ίδιο βρήκε.



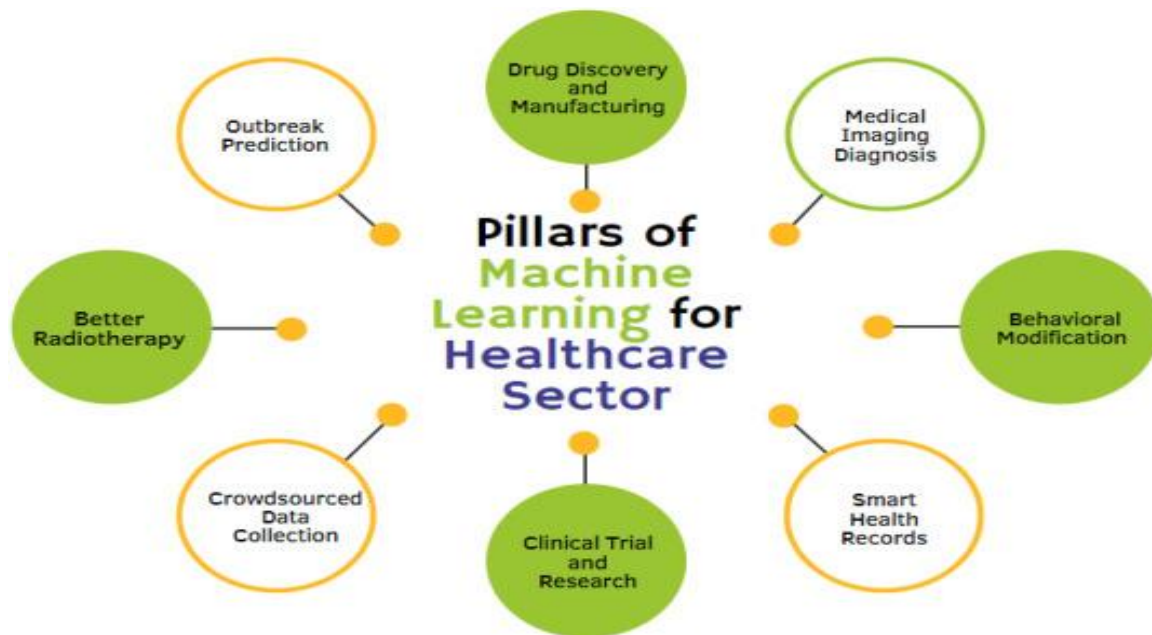
Εικόνα 4 <<https://www.javatpoint.com/supervised-machine-learning>>

Στην παραπάνω εικόνα μπορούμε να δούμε το σύνολο των δεδομένων που δέχεται το πρόγραμμα, το σύνολο των ετικετών που εμείς έχουμε αναθέσει. Ύστερα γίνεται η εκπαίδευση του μοντέλου, δημιουργεί τις προβλέψεις του με βάση τα δεδομένα δοκιμής και τέλος μας δίνει το αποτέλεσμα που το ίδιο βρήκε.

2.5 ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Η μηχανική μάθηση έχει αναδειχθεί σε ένα από τα πιο δυναμικά πεδία της πληροφορικής με πληθώρα εφαρμογών σε διάφορους τομείς, οι οποίοι με την ενσωμάτωση της έχουν βελτιώσει την αποδοτικότητά τους. Ακολουθούν ορισμένες εφαρμογές της μηχανικής μάθησης σε επαγγελματικούς τομείς:

[4] **Υγεία:** Στον τομέα της υγείας, η μηχανική μάθηση χρησιμοποιείται για την ανάλυση ιατρικών εικόνων, την πρόγνωση ασθενειών και την εξατομίκευση θεραπειών. Μπορεί να εντοπίζει πρότυπα που βοηθούν στη διάγνωση ασθενειών όπως ο καρκίνος και οι καρδιακές παθήσεις, παρέχοντας γρήγορες και ακριβείς διαγνώσεις. Ως αποτέλεσμα σημαίνει ότι μπορούν να εξυπηρετηθούν περισσότεροι ασθενείς στο ίδιο χρονικό διάστημα. Εκτενέστερα, μπορούν να δημιουργηθούν εφαρμογές παρακολούθησης ασθενών με χρόνιες παθήσεις για την έγκαιρη επέμβαση είτε των ιδίων είτε ιατρικού προσωπικού.



Εικόνα 5 <<https://www.sciencedirect.com/science/article/pii/S2666603022000069>>

Όπως μπορούμε να δούμε και στην παραπάνω εικόνα, οι τομείς της υγείας που μπορούν να βοηθηθούν από την μηχανική μάθηση ποικίλουν. Ενδεικτικά κάποιοι από τους τομείς.

[5] **Οικονομία και Χρηματοοικονομικά:** Στις χρηματοοικονομικές αγορές, η μηχανική μάθηση χρησιμοποιείται για την ανάλυση δεδομένων και την πρόβλεψη τάσεων της αγοράς. Εφαρμογές περιλαμβάνουν την ανίχνευση απάτης, την αξιολόγηση πιστοληπτικής ικανότητας και τη βελτιστοποίηση χαρτοφυλακίου. Με αυτό τον τρόπο γίνεται προσπάθεια μείωσης λαθών και τον εντοπισμό μεταβλητών που ενδεχόμενος να μην έγιναν άμεσα αντιληπτοί.

[6] **Μάρκετινγκ και Λιανικό Εμπόριο:** Στον τομέα του μάρκετινγκ, η μηχανική μάθηση βοηθά στην ανάλυση της καταναλωτικής συμπεριφοράς και στην εξατομίκευση προσφορών και προτάσεων. Χρησιμοποιείται για την πρόβλεψη των αναγκών των πελατών και τη βελτιστοποίηση της αλυσίδας εφοδιασμού. Είναι γνώριμο στους περισσότερους από εμάς μέσω των διαφημίσεων της Google και την λειτουργία των λεγόμενων cookies σε κάθε εφαρμογή και ιστοσελίδα.

[7] **Προστασία στον Κυβερνοχώρο:** Στην κυβερνοασφάλεια, η μηχανική μάθηση χρησιμοποιείται για την ανίχνευση και την πρόληψη κυβερνοεπιθέσεων. Εντοπίζει ανωμαλίες στα δεδομένα δικτύου και προσδιορίζει πιθανές απειλές σε πραγματικό χρόνο, για την έγκαιρη επέμβαση και διόρθωση.

Οι εφαρμογές της μηχανικής μάθησης συνεχώς επεκτείνονται, επηρεάζοντας θετικά πολλούς τομείς της καθημερινής ζωής και της βιομηχανίας. Οι εξελίξεις σε αυτό το πεδίο υπόσχονται ακόμη περισσότερες καινοτομίες και βελτιώσεις στο μέλλον. Σε αυτή την κατηγορία μπορούμε να προσθέσουμε και τα αυτόνομα οχήματα τα οποία βλέπουμε να εισέρχονται στην ζωή μας όλο και περισσότερο. Αν και μπορεί να μην υπάρχουν διαθέσιμα για το ευρύ κοινό τα πλήρως αυτόνομα οχήματα, υπάρχουν όμως διαθέσιμα οχήματα με πολλούς αυτοματισμούς. Όπως γνωρίζουμε το lane assist είναι ένα από τα πιο γνωστά συστήματα αυτοματισμού που είναι η αρχή για τα αυτόνομα οχήματα. Πιο συγκεκριμένα τα αυτόνομα οχήματα χρησιμοποιούν μηχανική μάθηση για να αναγνωρίζουν αντικείμενα, να αντιλαμβάνονται το περιβάλλον τους και να λαμβάνουν αποφάσεις σε πραγματικό χρόνο. Η τεχνολογία αυτή επιτρέπει την ανάπτυξη αυτοκινήτων χωρίς οδηγό που μπορούν να κινούνται με ασφάλεια στο οδικό δίκτυο.



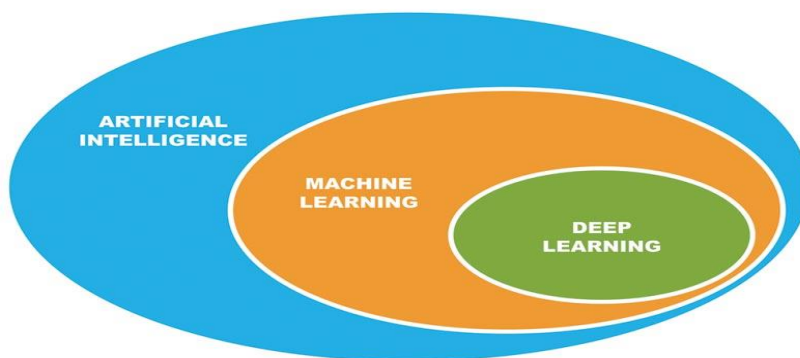
Εικόνα 6 <https://media.cosmotesmartliving.gr/%CE%AD%CF%84%CF%83%CE%B9-%CE%B5%CE%BA%CF%80%CE%B1%CE%B9%CE%B4%CE%B5%CF%8D%CE%BF%CE%BD%CF%84%CE%B1%CE%B9-%CF%84%CE%B1-%CE%B1%CF%85%CF%84%CF%8C%CE%BD%CE%BF%CE%BC%CE%B1-%CE%BF%CF%87%CE%AE%CE%BC%CE%B1/>

Στη παραπάνω εικόνα υπάρχει μια φουτουριστική αναπαράσταση των αυτόνομων οχημάτων. Με σύστημα αυτόματου πιλότου, όπου θα μπορεί με χρήση καμερών και υπολογιστικής όρασης να εντοπίζει άλλα οχήματα και δυνητικά και πεζούς. Έχει την δυνατότητα να ξεκινάει, να κρατάει πορεία και να ελαττώσει ταχύτητα αυτόνομα όποτε το απαιτεί η κατάσταση.

ΚΕΦΑΛΑΙΟ 3 Βαθιά Μηχανική Μάθηση

3.1 ΕΙΣΑΓΩΓΗ

Η βαθιά μηχανική μάθηση (deep learning) που κάποιος θα μπορούσε από το όνομα και μόνο να θεωρήσει πώς είναι μια επέκταση της μηχανικής μάθησης, εν αντι θέση είναι ένα υποσύνολο της. Παρ' όλ' αυτά αν κανείς υπέθετε ότι αυτά τα δυο είναι παρόμοια ως προς τον τρόπο λειτουργίας τους θα έμενε έκπληκτος από τις σημαντικές διαφορές που υπάρχουν μεταξύ τους.



Εικόνα 7 <<https://nowmag.gr/machine-learning/>>

Στο παραπάνω σχήμα μπορούμε να δούμε την τεχνητή νοημοσύνη να έχει ως υποκατηγορία της την μηχανική μάθηση, με την σειρά της η μηχανική μάθηση έχει ως υποκατηγορία την βαθιά μηχανική μάθηση.

3.2 ΟΡΙΣΜΟΣ [8],[9]

Η βαθιά μηχανική μάθηση είναι ένας υποτομέας της μηχανικής μάθησης που χρησιμοποιεί πολυεπίπεδα νευρωνικά δίκτυα για την ανάλυση μεγάλων όγκων δεδομένων και την εκμάθηση πολύπλοκων προτύπων και σχέσεων. Χάρη στην ικανότητά της να ανακαλύπτει αυτόματα σημαντικά χαρακτηριστικά και να εκτελεί μη γραμμικούς μετασχηματισμούς, η βαθιά μηχανική μάθηση έχει επιτύχει εξαιρετικές επιδόσεις σε πληθώρα εφαρμογών. Κάποια βασικά χαρακτηριστικά της είναι, η χρήση νευρωνικών δικτύων για την επίλυση πολύπλοκων διαδικασιών, η υψηλή υπολογιστική δύναμη και γνώση που απαιτείται, καθώς και το backpropagation (οπισθοδιάδοση) [10] με το οποίο ελέγχει παλαιότερα βάρη και τα βελτιώνει για τις μελλοντικές προβλέψεις. Φυσικά υπάρχουν και άλλα χαρακτηριστικά που δεν θα αναφερθούμε στη παρούσα εργασία.

3.3 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ

Αρχικά η βαθιά μηχανική μάθηση δεν περιορίζεται σε μία τοπική βάση δεδομένων αλλά έχει την δυνατότητα να αντλήσει δεδομένα από τα λεγόμενα συνελεκτικά νευρωνικά δίκτυα (CNN / Convolutional neural network). Πήραν την ονομασία αυτή λόγω των παρόμοιων ιδιοτήτων που έχουν με ένα νευρικό δίκτυο εγκεφάλου ενός ανθρώπου. Η βαθιά μηχανική μάθηση μπορεί να εκπαιδευτεί και από τα επαναλαμβανόμενα/αναδρομικά νευρωνικά δίκτυα (RNN / Recurrent neural network). Τα δύο αυτά μοντέλα έχουν πολλαπλές στρώσεις (layers) για την άντληση και επεξεργασία των δεδομένων τους. Αυτές οι δυνατότητες πρόσβασης είναι που προσφέρουν μια τόσο μεγάλη διαφορά από την μηχανική μάθηση καθώς δεν χρειάζονται πολλές επεμβάσεις από τους ανθρώπους για την σωστή εκπαίδευση του εκάστοτε μοντέλου. Αυτό οφείλεται στην πληθώρα δεδομένων που μπορεί να αντληθεί από την πρόσβαση ενός υπολογιστή στα δίκτυα αυτά. Με αυτό τον τρόπο μπορούμε να ενισχύσουμε πολλούς τομείς της πληροφορικής όπως η ρομποτική όραση, με το να κάνουμε τον εντοπισμό αντικειμένων πιο ακριβή με τις πιθανότητες λάθους να μειώνονται με τον αριθμό επαναλήψεων και την πάροδο του χρόνου.

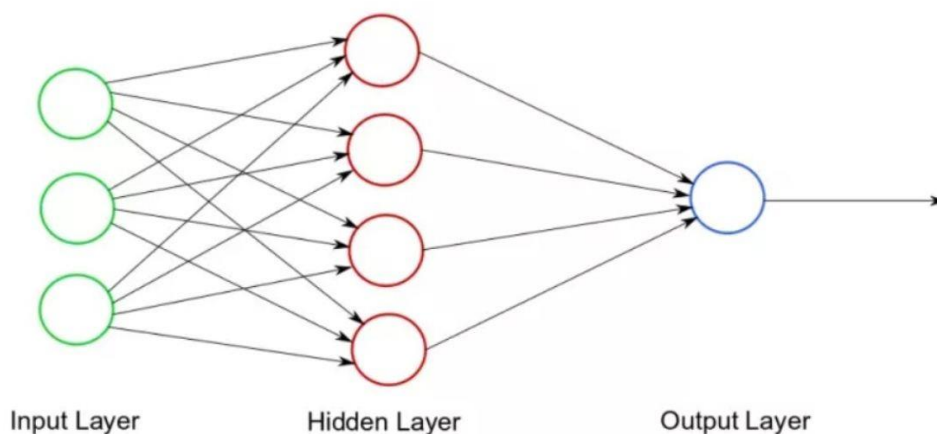
3.4 DEEP LEARNING

Τα μοντέλα deep learning έχουν ενσωματωθεί στην καθημερινότητα μας με απλές αλλά και πιο σύνθετες εντολές. Χαρακτηριστικά αναφερόμαστε στους έξυπνους βοηθούς (smart assistant) Siri, Google, Alexa, Bixby κ.α., που έχουν διευκολύνει την καθημερινότητα της σύγχρονης εποχής, όπως την πραγματοποίηση τηλεφωνικής κλήση με εντολές φωνής, όμως και πιο απαιτητικές όπως την δημιουργία ραντεβού σε επιλεγμένο ιατρό από εμάς. Αν και σε πειραματικό στάδιο για σύνθετα προβλήματα που αφορούν τον απλό κόσμο, το deep learning ενσωματώνεται όλο και περισσότερο στην καθημερινότητα μας. Τα deep learning μοντέλα μπορούν να λειτουργούν πιο αυτόνομα, καθώς δεν βασίζονται απόλυτα σε τοπικούς αλγόριθμους, αλλά στα CNN. Πιο συγκεκριμένα εσωτερικά των CNN υπάρχουν αλγόριθμοι που έχουν σχεδιαστεί για την αναγνώριση συγκεκριμένων μοτίβων, αλλά και πολύπλοκα (ως προς την φύση τους) δεδομένα, σε αντίθεση με την μηχανική μάθηση, όπου οι αλγόριθμοι είναι κυρίως για την εύρεση επαναλαμβανόμενων δεδομένων. Όμως υπάρχουν και αρνητικά τα οποία πρέπει να αναφερθούν. Αρχικά, λόγω των πολύπλοκων δυνατοτήτων που έχουν, είναι και τα ίδια πολύπλοκα ως προς την κατανόηση τους από τους ανθρώπους. Αυτό έχει σαν αποτέλεσμα την ανάγκη για καλή τεχνογνωσία καθώς τα αποτελέσματα μπορεί να είναι αρκετά απρόβλεπτα. Ακόμα ένα μειονέκτημα τους είναι ο τρόπος εκπαίδευσής. Παρόλο που υπάρχουν πολλές αυτοματοποιημένες εντολές και αλγόριθμοι, υπάρχει κίνδυνος για «επιρρεπή υπερπροσαρμογή», δηλαδή την υπερβολική ανάλυση των δεδομένων και την διάσπαση τους σε «υποενότητες». Αυτό έχει ως αποτέλεσμα με την παραμικρή αλλαγή, την εμφάνιση ανακριβούς ή μη επιθυμητού αποτελέσματος.

Όσο αφορά τον τομέα της υγείας [11], υπάρχει αρκετά ισχυρή ενσωμάτωση των deep learning τεχνικών και αλγορίθμων στα μηχανήματα. Πιο συγκεκριμένα, [12] σε τομείς ακτινογραφιών και scanning υπάρχει έντονη χρήση των αλγορίθμων αυτών, καθώς μπορεί να γίνει η ανάλυση των εικόνων και να εντοπιστούν ανωμαλίες που ενδεχομένως να μην είναι εύκολα ορατές. Επιπλέον δίνεται η δυνατότητα να «τεμαχιστούν» κομμάτια ακτινογραφιών για την καλύτερη έρευνα σε ανατομικό επίπεδο. Ακόμα και η βελτίωση των εικόνων βρίσκεται στην φαρέτρα δυνατοτήτων του, έτσι κομμάτια που δεν ήταν εύκολα αντιληπτά μπορούν να εντοπιστούν ευκολότερα από το ιατρικό προσωπικό. Όπως ακόμα και η εύρεση ασθενειών γίνεται ευκολότερη και γρηγορότερη.

3.5 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ [13]

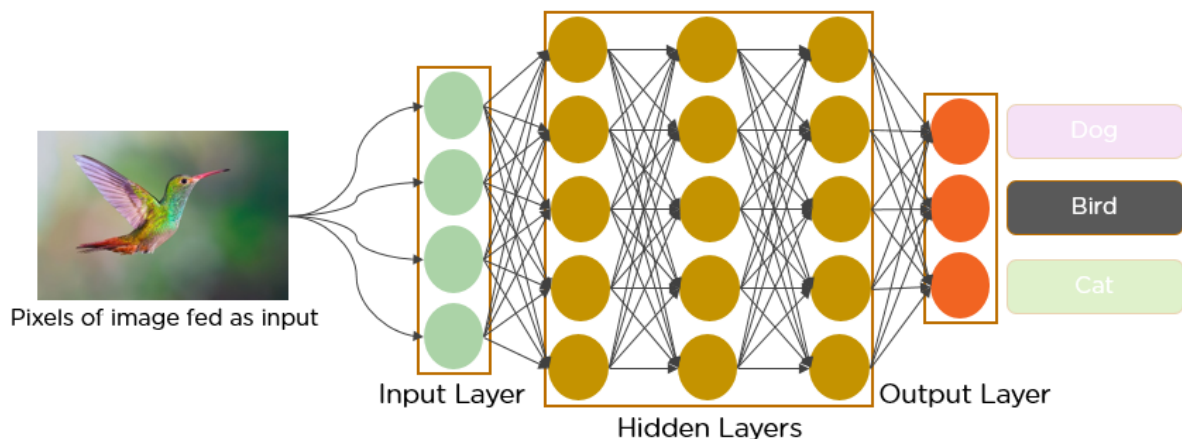
Η αρχή λειτουργίας των νευρωνικών δικτύων είναι οι αλγόριθμοι οι οποίοι εμπεριέχονται στο εσωτερικό τους. Για να πραγματοποιηθεί η λειτουργία των αλγορίθμων αυτών πρέπει να υπάρχει και η κατάλληλη είσοδος δεδομένων. Για αυτό το σκοπό υπάρχει το επίπεδο εισόδου (input layer), το οποίο δέχεται τα δεδομένα και τα επεξεργάζεται για να τα προωθήσει στο επόμενο επίπεδο. Το επόμενο επίπεδο είναι το κρυμμένο επίπεδο (hidden layer), που στην πραγματικότητα μπορεί να υπάρχουν πολλαπλά κρυμμένα επίπεδα στο εσωτερικό του δικτύου. Το κρυμμένο επίπεδο έχει ως ρόλο την μετατροπή των δεδομένων εισόδου ως κάτι αναγνωρίσιμο για το επίπεδο εξόδου (output layer) [14]. Το επίπεδο εξόδου μας δίνει το τελικό αποτέλεσμα που έχει δημιουργήσει ο υπολογιστής μέσω όλων των διαδικασιών επεξεργασίας. Δυο από τα πιο γνωστά μοντέλα είναι το CNN και το RNN που αναφέρθηκαν και παραπάνω.



Εικόνα 8 <<https://bigblue.academy/gr/neuronika-diktua>>

Στο παραπάνω σχεδιάγραμμα μπορούμε να δούμε τον τρόπο με τον οποίο λειτουργεί ένα νευρωνικό δίκτυο. Αρχικά έχουμε στα αριστερά την εισαγωγή των δεδομένων από το επίπεδο εισαγωγής. Στη συνέχεια περνάμε στα πολύπλοκα κρυμμένα επίπεδα και τέλος καταλήγουμε στο τελικό αποτέλεσμα και στο επίπεδο εξόδου.

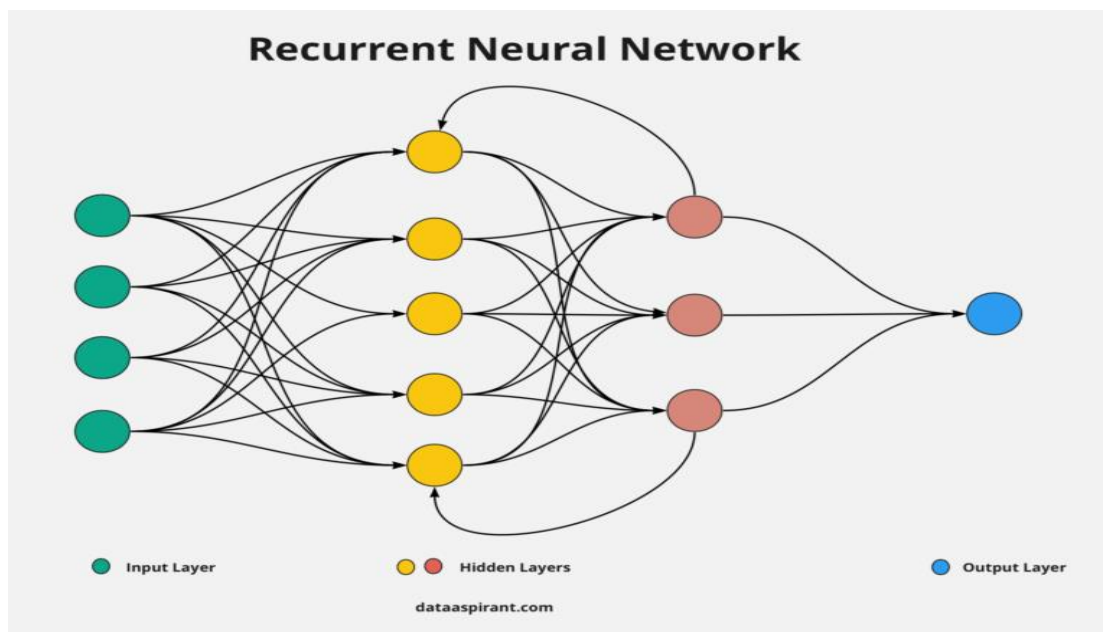
[15] CNN: Ένα νευρωνικό δίκτυο που μπορεί να χρησιμοποιηθεί για την υπολογιστική όραση, αφού μπορεί να επεξεργαστεί «διάφορα δεδομένα όπως είναι οι φωτογραφίες, τα βίντεο, ο ήχος και τα κείμενα». Για να υπάρξει αυτή η επεξεργασία χρειάζεται και η κατάλληλη είσοδος δεδομένων (input layer). Χρησιμοποιείται κυρίως για την ανίχνευση αντικειμένων και προσώπων, και ως πιο σύνθετη χρήση στην αυτόνομη οδήγηση, που ενσωματώνεται σε ολοένα και περισσότερα οχήματα. Ακόμα μπορεί να εφαρμοστεί και σε ιατρικές διαγνώσεις για τον γρηγορότερο εντοπισμό ανωμαλιών καθώς μπορεί να ενισχύει τον εντοπισμό τους σε διάφορες περιπτώσεις.



Εικόνα 9 <<https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>>

Στη από πάνω εικόνα έχουμε τον τρόπο διάσπασης των pixels ως δεδομένα εισόδου και την επεξεργασία τους σε ένα σύστημα CNN. Φτάνοντας στο αποτέλεσμα αναγνώρισης του αντικειμένου, που στην περίπτωση μας είναι ένα πουλί.

[16] RNN: Αυτός ο τύπος δικτύου δημιουργήθηκε για πρόβλεψη. Ένα απλό παράδειγμα αυτού του δικτύου είναι οι προτεινόμενες απαντήσεις που εμφανίζονται στα ψηφιακά (και όχι μόνο) πληκτρολόγια κινητών τηλεφώνων που οι περισσότεροι χρησιμοποιούμε στην καθημερινή μας ζωή. Αυτό συμβαίνει διότι ένα RNN κρατάει την πληροφορία από την προηγούμενη είσοδο που δέχτηκε και προσπαθεί να προβλέψει την επιθυμητή επόμενη είσοδο, με την βοήθεια όλων των δεδομένων που έχει κρατήσει από το παρελθόν εντοπίζει την συχνότητα χρήσης ενός δεδομένου μετά από το δεδομένο που του δώσαμε (στην περίπτωση μας λέξεις). Όμως υπάρχουν και προβλήματα σε αυτό το δίκτυο. Έτσι δημιουργήθηκαν υποκατηγορίες του όπως είναι το LSTM (Long short-term memory) [17], ουσιαστικά με αυτό τον τρόπο αντιμετωπίζεται το πρόβλημα των αποθηκευμένων δεδομένων αφού επεκτείνει την μνήμη που θα κρατάει με τα «memory cells» για ορισμένο χρονικό διάστημα για την καλύτερη λειτουργία του.



Εικόνα 10 <<https://dataaspirant.com/how-recurrent-neural-network-rnn-works/>>

Στη παραπάνω εικόνα έχουμε ένα σχεδιάγραμμα για την ροή των δεδομένων σε ένα σύστημα RNN. Παρατηρούμε πως τα δεδομένα έχουν την δυνατότητα να επιστρέψουν σε ένα προηγούμενο κρυμμένο επίπεδο για την βελτίωση τους, πριν δοθεί το τελικό αποτέλεσμα.

Άξιο αναφοράς είναι και τα GAN [18] (generative adversarial network), δηλαδή δίκτυο γενετικών αντιπάλων. Η χρήση αυτών των μοντέλων είναι για την δημιουργία ακριβέστερων και ασφαλέστερων αποτελεσμάτων. Η λειτουργίας έχει ως εξής: δυο (ή περισσότερα) νευρωνικά δίκτυα χρησιμοποιούνται για την δημιουργία ενός αποτελέσματος. Ένα από αυτά προσπαθεί να μπερδέψει τα υπόλοιπα με ψευδή στοιχεία, τα υπόλοιπα προσπαθούν να εντοπίσουν τα λανθασμένα στοιχεία με βάση ενός δοσμένου dataset. Με αυτό τον τρόπο το μοντέλο εκπαιδεύεται στην αναγνώριση λαθών και έχει ένα πιο ασφαλές περιβάλλον, καθώς ο αλγόριθμος θα σταματήσει όταν βρεθεί ένα ρεαλιστικό αποτέλεσμα.

3.6 ΣΧΟΛΙΑ

Γνωρίζοντας καλύτερα τέτοιου είδους εφαρμογές μας επιτρέπεται η ομαλότερη μετάβαση και εξοικείωση με τις τεχνολογίες αυτές, δημιουργώντας τα περιθώρια για περαιτέρω ανάπτυξη εφαρμογών εξυπηρέτησης ατόμων με προβλήματα όρασης, αφού θα μπορέσει να υπάρχει καλύτερη διασύνδεση και χρήση τους. Ως αποτέλεσμα θα υπάρχει και η εξοικείωση όλων με αυτές τις εφαρμογές μετατρέποντας τις σε ένα καθημερινό εργαλείο, χωρίς να υπάρχει φόβος για στιγματισμό των ατόμων που το έχουν ανάγκη. Με την εξοικείωση όλων μας θα μπορέσει να υπάρξει και περαιτέρω ανάπτυξη εφαρμογών για την υποβοήθηση ατόμων, όχι μόνο με προβλήματα όρασης αλλά και με άλλες ανάγκες.

ΚΕΦΑΛΑΙΟ 4 Προτεινόμενο Μοντέλο

4.1 ΓΕΝΙΚΑ

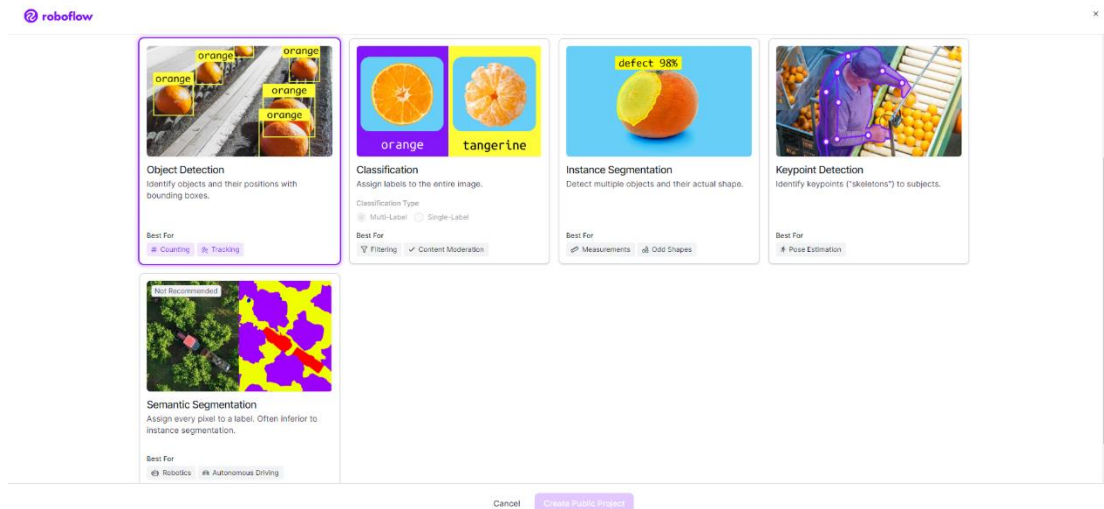
Για την επίτευξη και πραγματοποίηση της εργασίας και του αντίστοιχου προγράμματος για την υποβοήθεια ανθρώπων με προβλήματα όρασης, χρειαζόταν η χρήση μοντέλων ανίχνευσης και εντοπισμού καίριων αντικειμένων που έχουν στην καθημερινότητα τους. Το μοντέλο που χρησιμοποιήθηκε είναι το YOLO και πιο συγκεκριμένα η έκδοση YOLOv8. Το μοντέλο αυτό έχει την δυνατότητα ανίχνευσης και εντοπισμού αντικειμένων, ζώων, ανθρώπων κ.α. . Το μοντέλο YOLO κάνει χρήση μηχανικής μάθησης και κατ' επέκταση deep learning και νευρωνικών δικτύων. Ακόμα χρειάστηκε η διάθεση των δεδομένων στο πρόγραμμα αυτό, τα δεδομένα δημιουργήθηκαν με την χρήση της εφαρμογής Roboflow. Η επιλογή της έγινε λόγω της εύκολη μετατροπής των εικόνων σε δεδομένα για το μοντέλο ανίχνευσης YOLO.

4.2 ΤΡΟΠΟΙ ΧΡΗΣΗΣ

Παρακάτω θα αναπτυχθούν οι τρόποι χρήσης του εκάστοτε προγράμματος για την καλύτερη κατανόηση τους.

4.2.1 ΤΡΟΠΟΣ ΧΡΗΣΗΣ ROBOFLOW

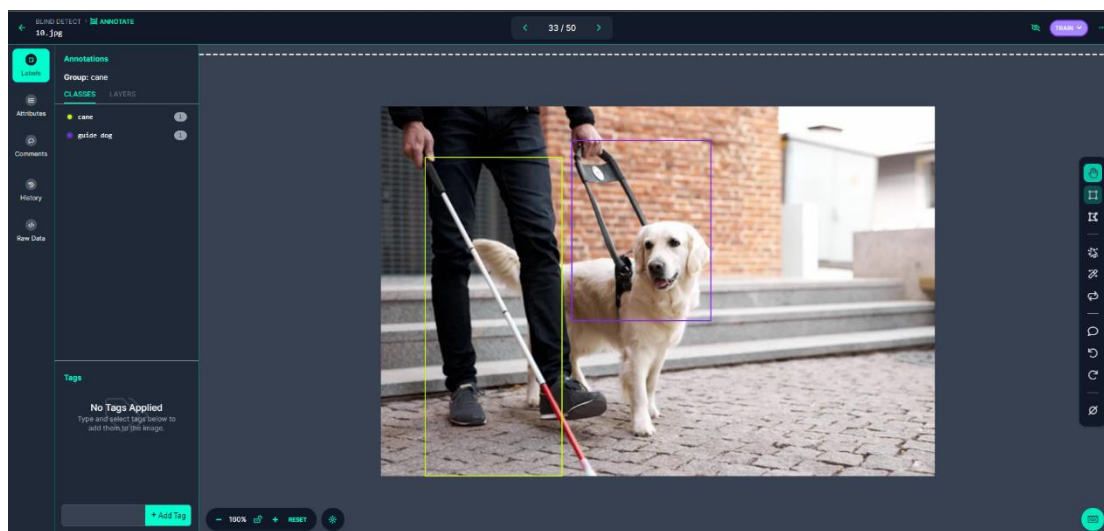
Το Roboflow είναι μια εφαρμογή που σου επιτρέπει να φτιάξεις δικά σου προγράμματα με υπολογιστική όραση ανεξάρτητα από το βάθος γνώσεων που έχεις. Η λειτουργία χρήσης του είναι αρκετά απλή. Δημιουργείς ένα περιβάλλον εργασίας στην εφαρμογή, και επιλέγεις μέσα από κάποιες βασικές επιλογές που σου παρέχει, με τι αντικείμενο θες να ασχοληθείς. Υπάρχουν οι επιλογές Object Detection (εντοπισμός αντικειμένου), Classification, Instance Segmentation, Keypoint Detection και Semantic Segmentation. Για την περίπτωση μας επέλεξα τον εντοπισμό αντικειμένου.



Εικόνα 11 < <https://roboflow.com/> >

Στην εικόνα αυτή έχουμε τις επιλογές που μας δίνονται για την κατηγορίας εντοπισμού που επιθυμούμε να ενασχοληθούμε, να πραγματοποιήσουμε και να ασχοληθούμε.

Εφόσον επιλέξουμε το τύπο που θέλουμε, εισάγουμε τις φωτογραφίες που επιθυμούμε. Κατά την εισαγωγή τους θα μας ζητηθεί να σημειώσουμε (annotate) τα δεδομένα που θέλουμε να γίνουν εντοπισμοί και να τους προσδώσουμε μια ετικέτα (label). Για την δική μας περίπτωση έχουμε δυο ετικέτες cane και Guide-dog.



Εικόνα 12 από το πακέτο εικόνων που δημιουργήσα.

Στην εικόνα παραπάνω έχουμε ένα παράδειγμα επεξεργασίας εικόνας για την εκμάθηση εντοπισμού του ειδικού μπαστουνιού και τους σκύλου-οδηγού. Πρέπει ατομικά κάθε εικόνα να γίνει crop στα κατάλληλα σημεία.

Αφού ολοκληρωθεί και αυτό το στάδιο και δοθούν οι ετικέτες που επιθυμούμε και έχουμε τοποθετήσει κάθε εικόνα στο εκάστοτε set (train, validation, test), μπορούμε με την αυτοματοποιημένη εντολή που παρέχεται να δημιουργήσουμε το dataset μας και να το έχουμε άμεσα διαθέσιμο μέσω της εταιρίας για προσωπική χρήση. Να σημειωθεί ότι το ίδιο dataset μπορεί να χρησιμοποιηθεί και για άλλες εκδόσεις της εταιρίας και όχι μόνο για το v8.

4.2.2 ΤΡΟΠΟΣ ΧΡΗΣΗΣ ΤΟΥ YOLO

Ο βασικός τρόπος χρήση τέτοιων μοντέλων είναι η εισαγωγή δεδομένων και χαρακτηρισμού των αντικειμένων που αναζητούμε ως labels. Στην περίπτωση μας οι πηγές δεδομένων είναι εικόνες που περιέχουν είτε το ένα από τα βασικά στοιχεία είτε και τα δυο ταυτόχρονα. Τα βασικά δεδομένα μας είναι τα λευκά μπαστούνια (μερικές φορές και άλλου χρώματος) που είναι σήμα κατατεθέν των ανθρώπων με προβλήματα όρασης, και οι σκύλοι οδηγοί με χαρακτηρίστηκα σαμαράκια στο σώμα τους και την ειδική λαβή. Οι ειδικοί αλγόριθμοι που υπάρχουν στο μοντέλο μπορούν να εντοπίσουν παρόμοια δεδομένα σε εικόνες οι οποίες δεν έχουν συμπεριληφθεί στην βάση δεδομένων. Πιο χαρακτηριστικά υπάρχουν τρεις κατηγορίες δεδομένων εικόνων. Αρχικά έχουμε το train, που όπως προσδιορίζει και το όνομα του, είναι το βασικό κομμάτι της εκπαίδευσης όπου διαβάζει τα δεδομένα τα οποία έχουμε εισάγει (τις φωτογραφίες), τα επεξεργάζεται και τα αφομοιώνει με βάση τις ετικέτες που εμείς του δώσαμε. Ύστερα είναι το validation, όπου με αυτό το κομμάτι γίνεται η εκτενέστερη εκμάθηση στα δεδομένα και η βελτίωση της ακρίβειας (accuracy) του. Τέλος έχουμε το test όπου είναι ένα πακέτο εικόνων που θα χρησιμοποιηθούν για τον έλεγχο του ορθού εντοπισμού και της ακρίβειας του. Εφόσον έχουμε ετοιμάσει τα δεδομένα μας (τα πακέτα εικόνων) πρέπει να προχωρήσουμε στην πραγματική εκπαίδευση (training) του μοντέλου μας. Λέγοντας πραγματική εκπαίδευση εννοείται η διαδικασία προγραμματισμού με την δυνατότητα το πρόγραμμα να κάνει σωστό εντοπισμό και σε άγνωστα δεδομένα. Έτσι πρέπει να δημιουργήσουμε το δικό μας τελικό μοντέλο με βάση το YOLO. Το περιβάλλον όπου πραγματοποιήθηκε η δημιουργία του μοντέλου είναι το Google Colaboratory και η χρήση της γλώσσα προγραμματισμού Python.

4.2.3 GOOGLE COLABORATORY & PYTHON

Με την χρήση του Google Colaboratory (Google Colab) πραγματοποιήθηκε η εκπαίδευση του μοντέλου που είναι αναγκαία για την επίτευξης λύσης του προβλήματος που αναφέρθηκε στην αρχή. Πρώτα αφού εγκαταστήσουμε την Python στο περιβάλλον μας και τις κατάλληλες εντολές εμφάνισης αποτελεσμάτων, μπορούμε να αρχίσουμε την διαδικασία της εκπαίδευσης.

Επιλέχθηκε η χρήση του Google colaborator, καθώς οι ενέργειες που θα αναλυθούν παρακάτω απαιτούν υψηλή υπολογιστική ισχύς.

4.2.3.1 ΥΛΟΠΟΙΗΣΗ GOOGLE COLABORATORY

Αρχικά ας αναφερθούμε τι είναι το Google Colaboratory και τι δυνατότητες μας προσφέρει. Το Google Colaboratory είναι μια διαδικτυακή πλατφόρμα της Google που μας δίνει την δυνατότητα να εργαστούμε πάνω στη γλώσσα προγραμματισμού Python. Μας επιτρέπεται να έχουμε πρόσβαση απομακρυσμένα σε έναν υπολογιστή μέσα από το ειδικό περιβάλλον που μας παρέχετε, μπορούμε να εισάγουμε σχεδόν όλες τις βιβλιοθήκες καθώς μπορούμε να τις κατεβάσουμε επί τόπου. Τα κυριότερα χαρακτηριστικά του όμως είναι η υπολογιστική ισχύς που μπορούμε να έχουμε από οποιοδήποτε άλλον υπολογιστή (ισχυρό ή καθημερινής χρήσης), καθώς και η συνεργασία που μπορεί να υπάρξει μεταξύ πολλών ατόμων μέσω της δυνατότητας collaborate (συνεργασίας) που προσφέρεται.

Στην παρακάτω εικόνα παρουσιάζονται οι εντολές που χρειάστηκαν να χρησιμοποιηθούν για την υλοποίηση της δημιουργίας του μοντέλου αναγνώρισης μπαστούνιων και σκύλων-οδηγών. Στη πρώτη γραμμή γίνεται η εισαγωγή του os module που μας επιτρέπει να χρησιμοποιήσουμε τα functionalities του operating system όπως ανάγνωση και εγγραφή. Με την δεύτερη γραμμή έχουμε την αποθήκευση τους στην διεύθυνση HOME και στην τρίτη γραμμή την εμφάνιση τους. Στη συνέχεια κάνουμε εγκατάσταση την βιβλιοθήκη ultralytics η οποία είναι κομμάτι της εφαρμογής της Roboflow. Παρακάτω προσθέτουμε την συνάρτηση display έτσι ώστε να μας εμφανίζονται τα αποτελέσματα. Εισάγουμε την συνάρτηση ultralytics και την ενσωματωμένη σε αυτή εφαρμογή YOLO, όπως και την εισαγωγή της Image για την εμφάνιση εικόνων. Στο τελευταίο κομμάτι της πρώτης εικόνας έχουμε την εγκατάσταση της Roboflow και την εισαγωγή του πακέτου εικόνων που δημιουργήσαμε, με αυτοματοποιημένη λειτουργία χάρις την ίδια την εταιρία.

```
[ ] import os
HOME = os.getcwd()
print(HOME)
```

↔ /content

```
[ ] !pip install ultralytics==8.0.196
from IPython import display
display.clear_output()
import ultralytics
ultralytics.checks()
from ultralytics import YOLO
from IPython.display import display, Image
```

↔ Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 29.8/78.2 GB disk)

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key= )
project = rf.workspace("ptyxiaki").project("blind-detect-cpmf7")
version = project.version(2)
dataset = version.download("yolov8")
```

Η παρακάτω γραμμή κώδικα ξεκινάει την διαδικασία εκπαίδευσης για τον εντοπισμό των αντικειμένων που του αναθέσαμε για το πακέτο εικόνων μας. Το μοντέλο που χρησιμοποιείται είναι το yolov8s. Εν συνεχεία θα κάνει 150 επαναλήψεις (epochs) για την αναγνώριση και τον εντοπισμό τους, έτσι ώστε να υπάρχει καλύτερη κατανόηση του τι αναζητά. Θα μετατρέψει τις εικόνες σε μέγεθος ανάλυσης 640x640 pixels. Το patience έχει τη λειτουργία να κάνει το πρόγραμμα να σταματάει όταν εντοπίσει ίδιο ή παρόμοιο αριθμό επίδοσης σε συγκεκριμένο αριθμό επαναλήψεων που θα του ορίσουμε εμείς. Με το να το θέσουμε 0 απενεργοποιείται η λειτουργία αυτή και ολοκληρώνει όλες τις επαναλήψεις.

loading Roboflow project...

```
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=150 imgsz=640 plots=True patience=0 batch=4
```

Αναφορικά υπάρχουν τα μοντέλα yolov8n, yolov8s, yolov8m, yolov8l και yolov8x. Το κάθε επόμενο μοντέλο έχει μεγαλύτερη ανάλυση δεδομένων και μεγαλύτερη ακρίβεια, όμως λόγω των σύνθετων δεδομένων που του παρέχουμε αποφασίστηκε μετά από αρκετές δοκιμές, να γίνει η χρήση ενός λιγότερο ακριβές μοντέλου ως προς την αναγνώριση ώστε να είναι πιο ευέλικτο στην εκμάθησή του.

Παρακάτω υπάρχουν εικόνες σχετικά με την απόδοση του προγράμματος σε κάθε επανάληψη. Κοιτώντας κυρίως το mAP50 μπορούμε να καταλάβουμε πόσο τις εκατό είναι οίγουρο το πρόγραμμα ότι βρήκε τα κατάλληλα στοιχεία. Στην αρχή βλέπουμε ότι βρίσκεται ~37% (0.376) αλλά στην συνέχεια μειώνεται, μέχρι να αποκτήσει αρκετή εξουκείωση και να αυξήσει την απόδοση του πάνω από 70% στο συνδυαστικό κομμάτι. Το τελικό μοντέλο που χρησιμοποιείται έχει συνδυαστική απόδοση ~80%.

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=150 imgsz=640 plots=True patience=0 batch=4
Starting training for 150 epochs...

Epoch 1/150	GPU_mem 1.32G	box_loss 2.059	cls_loss 5.578	dfl_loss 2.288	Instances 7	Size 640:	100% 12/12	[00:04<00:00, 2.43it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 3.10it/s]
	all	13	18	0.627	0.312	0.376	0.0802	
Epoch 2/150	GPU_mem 1.38G	box_loss 1.813	cls_loss 3.372	dfl_loss 1.985	Instances 3	Size 640:	100% 12/12	[00:01<00:00, 7.18it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 6.49it/s]
	all	13	18	0.341	0.469	0.322	0.094	
Epoch 3/150	GPU_mem 1.37G	box_loss 1.558	cls_loss 2.948	dfl_loss 1.705	Instances 5	Size 640:	100% 12/12	[00:02<00:00, 4.85it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 3.55it/s]
	all	13	18	0.62	0.25	0.133	0.0315	
Epoch 4/150	GPU_mem 1.41G	box_loss 1.672	cls_loss 2.78	dfl_loss 1.651	Instances 9	Size 640:	100% 12/12	[00:02<00:00, 4.90it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 9.00it/s]
	all	13	18	0.246	0.281	0.16	0.0899	
Epoch 5/150	GPU_mem 1.39G	box_loss 1.848	cls_loss 2.908	dfl_loss 1.819	Instances 3	Size 640:	100% 12/12	[00:01<00:00, 8.19it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.49it/s]
	all	13	18	0.105	0.781	0.156	0.0722	
Epoch 6/150	GPU_mem 1.4G	box_loss 1.915	cls_loss 2.701	dfl_loss 1.832	Instances 4	Size 640:	100% 12/12	[00:01<00:00, 8.11it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 6.77it/s]
	all	13	18	0.0378	0.281	0.028	0.0143	
Epoch 7/150	GPU_mem 1.4G	box_loss 1.78	cls_loss 2.86	dfl_loss 1.761	Instances 7	Size 640:	100% 12/12	[00:01<00:00, 7.84it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.51it/s]
	all	13	18	0.0378	0.281	0.028	0.0143	
Epoch 8/150	GPU_mem 1.39G	box_loss 1.993	cls_loss 3.069	dfl_loss 1.973	Instances 3	Size 640:	100% 12/12	[00:01<00:00, 8.23it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.49it/s]
	all	13	18	0.611	0.125	0.0651	0.0221	

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=150 imgsz=640 plots=True patience=0 batch=4

Epoch 68/150	GPU_mem 1.41G	box_loss 2.166	cls_loss 2.812	dfl_loss 2.272	Instances 2	Size 640:	100% 12/12	[00:02<00:00, 4.60it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 5.47it/s]
	all	13	18	0.107	0.188	0.103	0.0299	
Epoch 69/150	GPU_mem 1.39G	box_loss 2.098	cls_loss 2.815	dfl_loss 2.199	Instances 2	Size 640:	100% 12/12	[00:01<00:00, 7.27it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.76it/s]
	all	13	18	0.143	0.25	0.137	0.0405	
Epoch 70/150	GPU_mem 1.41G	box_loss 2.153	cls_loss 2.644	dfl_loss 2.243	Instances 4	Size 640:	100% 12/12	[00:01<00:00, 8.22it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.73it/s]
	all	13	18	0.138	0.219	0.163	0.0591	
Epoch 71/150	GPU_mem 1.41G	box_loss 2.115	cls_loss 2.64	dfl_loss 2.154	Instances 3	Size 640:	100% 12/12	[00:01<00:00, 8.69it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.52it/s]
	all	13	18	0.138	0.219	0.163	0.0591	
Epoch 72/150	GPU_mem 1.4G	box_loss 2.16	cls_loss 2.656	dfl_loss 2.231	Instances 6	Size 640:	100% 12/12	[00:01<00:00, 7.01it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 6.78it/s]
	all	13	18	0.242	0.449	0.279	0.164	
Epoch 73/150	GPU_mem 1.41G	box_loss 2.069	cls_loss 2.64	dfl_loss 2.15	Instances 5	Size 640:	100% 12/12	[00:02<00:00, 4.70it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 5.29it/s]
	all	13	18	0.395	0.531	0.416	0.266	
Epoch 74/150	GPU_mem 1.41G	box_loss 2.133	cls_loss 2.584	dfl_loss 2.262	Instances 5	Size 640:	100% 12/12	[00:01<00:00, 7.54it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.76it/s]
	all	13	18	0.653	0.642	0.61	0.241	
Epoch 75/150	GPU_mem 1.41G	box_loss 2.111	cls_loss 2.506	dfl_loss 2.302	Instances 7	Size 640:	100% 12/12	[00:01<00:00, 8.46it/s]
	Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100% 2/2 [00:00<00:00, 10.87it/s]
	all	13	18	0.653	0.642	0.61	0.241	

```
+ Κώδικας + Κείμενο

[ ]
↔
Epoch 147/150 GPU_mem 1.41G box_loss 1.476 cls_loss 1.663 dfl_loss 1.665 Instances 2 Size 640: 100% 12/12 [00:01<00:00, 8.78it/s]
Class all Images 13 Instances 18 Box(P) 0.681 R 0.507 mAP50 0.583 mAP50-95): 100% 2/2 [00:00<00:00, 9.72it/s]
all 13 18 0.681 0.507 0.583 0.337

Epoch 148/150 GPU_mem 1.39G box_loss 1.556 cls_loss 1.733 dfl_loss 1.768 Instances 2 Size 640: 100% 12/12 [00:02<00:00, 5.56it/s]
Class all Images 13 Instances 18 Box(P) 0.717 R 0.531 mAP50 0.608 mAP50-95): 100% 2/2 [00:00<00:00, 7.08it/s]
all 13 18 0.717 0.531 0.608 0.367

Epoch 149/150 GPU_mem 1.4G box_loss 1.521 cls_loss 1.702 dfl_loss 1.729 Instances 2 Size 640: 100% 12/12 [00:02<00:00, 4.72it/s]
Class all Images 13 Instances 18 Box(P) 0.507 R 0.562 mAP50 0.6 mAP50-95): 100% 2/2 [00:00<00:00, 6.16it/s]
all 13 18 0.507 0.562 0.6 0.39

Epoch 150/150 GPU_mem 1.41G box_loss 1.593 cls_loss 1.83 dfl_loss 1.837 Instances 2 Size 640: 100% 12/12 [00:02<00:00, 4.46it/s]
Class all Images 13 Instances 18 Box(P) 0.474 R 0.594 mAP50 0.463 mAP50-95): 100% 2/2 [00:00<00:00, 9.44it/s]
all 13 18 0.474 0.594 0.463 0.303

150 epochs completed in 0.145 hours.
Optimizer stripped from runs/detect/train7/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train7/weights/best.pt, 22.5MB

Validating runs/detect/train7/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126358 parameters, 0 gradients, 28.4 GFLOPs
Class Images Instances Box(P) R mAP50 mAP50-95): 100% 2/2 [00:00<00:00, 6.61it/s]
all 13 18 0.521 0.594 0.709 0.435
cane 13 16 0.497 0.688 0.589 0.316
guide dog 13 2 0.546 0.5 0.828 0.554

Speed: 1.2ms preprocess, 7.6ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train7
💡 Learn more at https://docs.ultralytics.com/modes/train
```

Για την υλοποίηση αυτή χρειάζεται αρκετός χρόνος και οι κατάλληλες επαναλήψεις, έτσι ώστε να μην δημιουργηθεί υπερπροσαρμογή στο μοντέλο μας και υπάρξει μείωση των αποδόσεων. Τέλος για να μπορούμε να χρησιμοποιήσουμε το μοντέλο μας χωρίς περιορισμούς τους οποίους έχει το Google Colab, μπορούμε να το εγκαταστήσουμε στον υπολογιστή με άμεση λήψη από το Colab, καθώς προσωρινά τα δεδομένα μένουν σε ειδικούς φακέλους.

4.2.3.2 ΥΛΟΠΟΙΗΣΗ ΣΕ ΤΟΠΙΚΟ ΕΠΙΠΕΔΟ ΜΕ PYTHON

Για την υλοποίηση της εφαρμογής σε τοπικό επίπεδο χρειάζεται η εγκατάσταση της Python Cuda, καθώς και η ενεργοποίηση της ενισχυμένης χρήσης για να υπάρχει πρόσβαση στην κάρτα γραφικών μέσω της Nvidia. Αφού εγκαταστήσουμε τις κατάλληλες βιβλιοθήκες στη Cuda, μπορούμε να εργαστούμε είτε πάνω στο δικό της περιβάλλον, είτε στο Microsoft visual studio code.

Υπάρχουν δυο υλοποιήσεις όπου εφαρμόστηκαν, η πρώτη είναι ο εντοπισμός των δεδομένων μέσω ζωντανής καταγραφής από κάμερα. Η δεύτερη υλοποίηση για την παρουσίαση, είναι ο εντοπισμός σε βίντεο που του δίνουμε. Στις παρακάτω εικόνες παρουσιάζεται το πρόγραμμα που αναπτύχθηκε για την επεξεργασία βίντεο, την ενσωμάτωση περιγράμματος για κάθε class και την ενσωμάτωση ηχητικών ειδοποιήσεων κατά την διάρκεια εντοπισμού των δυο δεδομένων. Το κάθε ένα έχει τον δικό του χαρακτηριστικό ήχο και χρώμα περιγράμματος.

Πρώτη υλοποίηση

Η πρώτη υλοποίηση είναι ο εντοπισμός με ζωντανή καταγραφή, έχει πιο απλή λογική. Εισάγουμε τις κατάλληλες βιβλιοθήκες και πακέτα, στην περίπτωση μας είναι η YOLO, από την ultralytics για την ενσωμάτωση του μοντέλου μας και η βιβλιοθήκη playsound όπου θα μας επιτρέψει να προσθέσουμε τους ήχους κατά την διάρκεια εντοπισμού. Όπως θα δούμε και παρακάτω έχουμε την εισαγωγή του μοντέλου μας ((last (train3).pt)). Στην από κάτω σειρά έχουμε το results το οποίο θα δεχτεί το οπτικό πεδίο που δείχνει η κάμερα μας και θα ξεκινήσει τον εντοπισμό του μπαστούνιού και τους σκύλους-οδηγούς. Κατά την διάρκεια εντοπισμού θα εμφανίζονται τα εκάστοτε περιγράμματα της κάθε ετικέτας. Δίπλα η εντολή show = True μας επιτρέπει να παρακολουθούμε από ειδικό παράθυρο την διαδικασία. Η εντολή conf (συντομογραφία από το confidence) θα ενημερώσει το σύστημα ότι θέλουμε να κάνει εντοπισμό και παρουσίαση, μόνο ό,τι θεωρεί το ίδιο ότι έχει δυνατότητα να είναι ένα από τα δυο στοιχεία με πιθανότητα άνω του 50%. Ταυτόχρονα θα ενεργοποιείται και το κομμάτι κώδικα εντός της for που όταν γίνεται ο κάθε εντοπισμός και έχει προστεθεί το πλαίσιο, τότε να ηχεί και ο εκάστοτε ήχος (0 είναι η κλάση για το μπαστούνι και 1 είναι η κλάση για τον σκύλο-οδηγό). Οι ήχοι είναι αποθηκευμένοι στο ηλεκτρονικό υπολογιστή στον ίδιο φάκελο με τον κώδικα του μοντέλου.

```
YOLO live.py > ...
1  from ultralytics import YOLO
2  from playsound import playsound
3
4  model = YOLO("last (train3).pt")
5  results = model(source=0 , show=True , conf=0.5 , save=True)
6
7  for r in results:
8      if 0 in r.bboxes.cls: # 0 is cane class
9          playsound('Ding Sound Effect.mp3')
10     elif 1 in r.bboxes.cls: # 1 is guide dog class
11         playsound('Tick Sound.mp3')
12
13
14
15
16
17
18
```

Δεύτερη υλοποίηση

Στη δεύτερη υλοποίηση έχει γίνει επεξεργασία και εντοπισμός σε βίντεο. Τα βίντεο τα οποία έχουν δοκιμαστεί περιέχουν μαστούνια και σκύλο-οδηγό, είτε ταυτόχρονα είτε μεμονωμένα. Για την δημιουργία τελικού βίντεο με περιγράμματα και ήχους θα πρέπει να γίνει διπλή επεξεργασία. Δηλαδή, αρχικά θα γίνει μεμονωμένα η επεξεργασία του βίντεο για τα περιγράμματα όπως βλέπουμε στην παρακάτω εικόνα.

```
Yolo start.py > ...  
1  from ultralytics import YOLO  
2  
3  
4  model = YOLO("last (train3).pt")  
5  results = model(source='video2.mp4' , conf=0.55, show=True, save=True)  
6  
7  
8  
9  
10  
11  
12  
13
```

Έπειτα αφού ολοκληρωθεί η επεξεργασία, βρίσκουμε την διεύθυνση του νέου αποθηκευμένου βίντεο και το προσθέτουμε στον κώδικα. Όμως, για να προστεθεί ήχος πρέπει να προσθέσουμε τις κατάλληλες βιβλιοθήκες οι οποίες περιέχονται στο moniepy.editor, από εκεί θα χρειαστούμε τα videofileclip για την επεξεργασία του βίντεο, compositeaudioclip και το audiofileclip, τα οποία θα μας προσθέσουν τους ήχους στο βίντεο. Ύστερα θα επαναλάβουμε την διαδικασία για την αναγνώριση του βίντεο με τα περιγράμματα (το οποίο μπορούμε και να μην το αποθηκεύσουμε εφόσον υπάρχει προηγουμένως). Αφού ολοκληρωθεί αυτό το στάδιο προσθέτουμε το προηγούμενος επεξεργασμένο βίντεο (video2_box.avi) από το οποίο έχουν αφαιρεθεί όλοι οι ήχοι. Στη συνέχεια ελέγχοντας κάθε frame που γίνεται ο εντοπισμός από το βίντεο, που έχει αποθηκεύει στο results, προστίθενται και οι ήχοι ειδοποίησης. Τέλος γίνεται η τελική επεξεργασία και δημιουργείται το βίντεο με ήχους και περιγράμματα και αποθηκεύετε στην μεταβλητή video_with_sounds, προσδιορίζουμε το όνομα του τελικού βίντεο στο output_path με την ονομασία και την κωδικοποίηση που επιθυμούμε.

Yolo final.py > ...

```
1 from ultralytics import YOLO
2 from moviepy.editor import VideoFileClip, CompositeAudioClip, AudioFileClip
3
4 # Initialize model
5 model = YOLO("last (train3).pt")
6
7 # Process video and get results
8 results = model(source='video2.mp4' , conf=0.55, show=True, save=True)
9
10 # Load the original video to get the frame rate
11 video = VideoFileClip('video2_box.avi')
12 frame_rate = video.fps
13 # Extract events with times and corresponding sound files
14 events = []
15 for frame_id, r in enumerate(results):
16     if any(cls == 0 for cls in r.bboxes.cls): # Check if class 0 (cane class) is detected cane sound effect
17         events.append((frame_id / frame_rate, 'Ding Sound Effect.mp3'))
18     if any(cls == 1 for cls in r.bboxes.cls): # Check if class 1 (guide dog class) is detected dog sound effect
19         events.append((frame_id / frame_rate, 'Tick Sound.mp3'))
20
21
22 # Function to add sound effects at specific times
23 def add_sound_effects(video, events):
24     audio_clips = []
25
26     for event_time, sound_file in events:
27         sound = AudioFileClip(sound_file).set_start(event_time)
28         audio_clips.append(sound)
29
30     new_audio = CompositeAudioClip(audio_clips)
31     return video.set_audio(new_audio)
32
33 # Add the sound effects to the video
34 video_with_sounds = add_sound_effects(video, events)
35
36 # Save the new video with sound effects
37 output_path = 'video2_with_sounds.mp4'
38 video_with_sounds.write_videofile(output_path, codec='libx264', audio_codec='aac')
```

ΣΥΜΠΕΡΑΣΜΑ

Το μοντέλο αυτό μπορεί να χρησιμοποιηθεί για ζωντανό εντοπισμό, αλλά και για εντοπισμό σε βίντεο με τον κατάλληλο προγραμματισμό. Αυτό μας επιτρέπει την χρήση σε κάμερες κυκλοφορίας για την καλύτερη εξυπηρέτηση των πολιτών και την καλύτερη δυνατή ασφάλεια τους. Σε συνδυασμό με τα ηχητικά σήματα που μπορούμε να αναμεταδώσουμε, η ευκολία στην μετακίνηση, η εξοικονόμηση ενέργειας και υπολογιστικών πόρων από τη μη συνεχή λειτουργία όλων των μηχανημάτων, μπορεί να βελτιωθεί.

4.3 ΥΠΑΡΧΟΝΤΑ ΣΥΣΤΗΜΑΤΑ ΥΠΟΒΟΗΘΗΣΗΣ [19-26]

Με την υπάρχουσα ευκαιρία είναι σκόπιμο να αναλύσουμε κάποια άλλα συστήματα υποβοήθησης ατόμων με προβλήματα όρασης που υπάρχουν ήδη και χρησιμοποιούνται από το ευρύ κοινό. Το πιο ευρέως και γνωστό σύστημα είναι οι ηχητικοί σηματοδότες που βρίσκονται κυρίως σε φωτεινούς σηματοδότες. Η κύρια χρήση τους (και η μοναδική) είναι η εκπομπή δύο διαφορετικών τύπων ηχητικού σήματος, ένα για την περίπτωση όπου οι πεζοί έχουν πράσινη σήμανση και έχουν δικαίωμα να διασχίσουν το οδικό δίκτυο, και ένα για την περίπτωση όπου οι πεζοί έχουν κόκκινη σήμανση και δεν έχουν δικαίωμα να διασχίσουν το οδικό δίκτυο. Αν και στις ευρωπαϊκές (και όχι μόνο) μεγαλουπόλεις παρατηρούμε να υπάρχουν σε αρκετές διασταυρώσεις οι ειδικοί ηχητικοί σηματοδότες, στην χώρα μας είναι ελάχιστοι και μόνο πρόσφατα φαίνεται να αυξάνονται και να χρησιμοποιούνται. Παρόλο όμως την μειωμένη χρήση τους υπάρχουν επαγγελματίες που φέρνουν καινούργιες ιδέες και υλοποιήσεις για την καλύτερη εξυπηρέτηση των ατόμων με προβλήματα όρασης. Αναφερόμενος στα «έξυπνα» φανάρια που τοποθετήθηκαν στη Θεσσαλονίκη (αν και 6 στον αριθμό τους το έτος 2018), όπου όχι μόνο εκπέμπουν ηχητικό σήμα, αλλά ακόμα έχουν και ανάγλυφη υφή βοηθώντας να αντιληφθούν το πότε ο φωτεινός σηματοδότης είναι πράσινος ή κόκκινος. Ακόμα υπάρχει και η δυνατότητα με το πάτημα ενός κουμπιού να γίνει εκφώνηση για την τοποθεσία του φαναριού στην πόλη και την ενημέρωση του χρήστη για την ακριβή του τοποθεσία του. Όλα αυτά μπορούν να συνδυαστούν και με την χρήση Bluetooth για την αναμετάδοση των πληροφοριών σε κινητή συσκευή τηλεφώνου.

Αλλάζοντας χώρα και πιο συγκεκριμένα στην Ρωσία και την Ρωσική Ομοσπονδία έχουμε την δυνατότητα να μελετήσουμε το πρόγραμμα «Talking city». Το οποίο δεν παρέχει απλώς ενημερώσεις για το που βρίσκονται οι άνθρωποι μέσα στην πόλη αλλά έχει εξελιχθεί περαιτέρω των βασικών ενεργειών. Συγκεκριμένα στην Αγία Πετρούπολη το 2008 αναπτύχθηκε ένα ραδιοφωνικό σύστημα που βοήθησε τους ανθρώπους με περιορισμένη όραση και γενικά τους ΑμεΑ. Το πιλοτικό πρόγραμμα ονομάστηκε «Talking City» (πόλη που μιλάει). Μεγάλη συμμετοχή υπήρξε από άτομα με προβλήματα όρασης. Το 2011 ξεκίνησε η υλοποίηση και η χρήση του συστήματος. Το 2014 με μεγάλη επιτυχία χρησιμοποιήθηκε σε μέσα μεταφοράς/ υποδομές μεταφορών (λεωφορεία, πλοία, αεροπλάνα κ.λπ.) στους Ολυμπιακούς και Παραολυμπιακούς αγώνες. Το 2022 το σύστημα «Talking City» έχει εφαρμοστεί σε 32 περιφέρειες της Ρωσικής Ομοσπονδίας και η αρμόδια εταιρία παράγει συστήματα 3^{ης} γενιάς. Περισσότερο από 4000 μεταφορικά μέσα, 4200 φωτεινοί σηματοδότες πεζών και πάνω από 1500 υποδομές (όπως τράπεζες, εστίαση κ.λπ.) ήταν εξοπλισμένα με τέτοιου είδους συστήματα. Επιπλέον ο εξοπλισμός αυτός υπάρχει στα ΜΜΜ και στις στάσεις αυτών για την καλύτερη ενημέρωση των πολιτών.

Η χρήση του «Talking City» μπορεί να πραγματοποιηθεί με την χρήση εφαρμογής μέσω smartphone ή εξειδικευμένες συνδρομητικές συσκευές. Ο χρήστης έχει πληροφορίες για το πού είναι τοποθετημένος εξοπλισμός, επίσης έχει την δυνατότητα μέσω ενεργοποίησης ηχητικού σήματος, να εντοπίσει που βρίσκεται ακριβώς το αντικείμενο το οποίο περιέχει τον αντίστοιχο εξοπλισμό για την εύρεση του, μέσω ηχείου ή ακουστικού. Η αναμετάδοση γίνεται μέσω ραδιοφωνικών καναλιών, χωρίς να ακουστεί σε άλλους ή να μεταδοθεί σε άλλη συσκευή. Τέλος μεγάλο πλεονέκτημα είναι ότι λειτουργεί χωρίς την χρήση διαδικτύου.

Σε άλλη χώρα και συγκεκριμένα στην Ολλανδία υπάρχει μία εταιρία όπου δημιούργησε κάτι πρωτοφανές. Η εταιρία ονομάζεται Envision AI και ασχολείται με τεχνολογίες που ενσωματώνουν τεχνητή νοημοσύνη στα προϊόντα τους. Το πιο χαρακτηριστικό προϊόν είναι τα έξυπνα γυαλιά τους. Τα συγκεκριμένα γυαλιά έχουν την δυνατότητα να αναγνωρίζουν κείμενα και να τα εκφωνούν στους χρήστες τους με ειδικό πάτημα ενός κουμπιού, ακόμα μπορεί να κάνει μετάφραση κειμένων, μέχρι και να διαβάσει ένα συγκεκριμένο κομμάτι από όλο το κείμενο (όπως μπορούμε να δούμε στο βίντεο που έχουν, μπορεί με την φωνητική εντολή του χρήστη να αναγνωρίσει την ημερομηνία λήξης του πακέτου με τοίχλες). Η εταιρία έχει δημιουργήσει και ακουστικά οστών (όπως τα ονομάζει) καθώς μεταδίδουν ειδικά κύματα για να ενημερώσει τον χρήστη τους για κάποιες πληροφορίες, χωρίς όμως να απομονώνει τα αυτιά τους και να μην έχουν ηχητική επαφή με το περιβάλλον. Υπάρχουν ακόμα και το πληκτρολόγιο με την γραφή Braille για να μπορούν να πληκτρολογούν σε ηλεκτρονικούς υπολογιστές, καθώς και οι οθόνες Braille που με ειδικά cells, που περιέχονται στην οθόνη, μπορούν να αναγνώσουν το κείμενο. Τέλος υπάρχει μια πιο προσιτή λύση (από οικονομικής άποψης), Audio Labeler. Είναι μια συσκευή όπου δίνει ταμπέλες στα αντικείμενα στο γύρω περιβάλλον που αναγνωρίζονται με ειδικά scanners. Είναι ένα πολύ χρήσιμο προϊόν, αφού δίνει την δυνατότητα στα άτομα με προβλήματα όρασης να γνωρίζουν τι αντικείμενα υπάρχουν στο χώρο και να έχουν μια καλύτερη αντίληψη. Όλα αυτά έρχεται να τα συμπληρώσει και η ηλεκτρονική εφαρμογή για smartphone.



Εικόνα 13 <https://www.letsenvision.com/glasses/home> (Αι γυαλιά με κάμερα)

Σε αυτή την εικόνα βλέπουμε τα έξυπνα γυαλιά της εταιρίας Envision AI και την κάμερα/αισθητήρα που παρέχουν, καθώς και τον τρόπο χρήσης αφής.

ΚΕΦΑΛΑΙΟ 5 Πειραματική Αποτίμηση [27-31]

Ο εντοπισμός ατόμων με προβλήματα όρασης είναι πλέον κάτι το αναγκαίο για την κοινωνία μας. Εφόσον υπάρχουν οι τεχνολογίες που μπορούν να χρησιμοποιηθούν για την διευκόλυνσή της κυκλοφορίας τους, οφείλουμε να τους δώσουμε όλα τα «εργαλεία» που διαθέτουμε. Για να γίνει όμως η διάθεση του στο ευρύ κοινό πρέπει να υπάρχουν κάποιες προϋποθέσεις. Κάποιες από τις σημαντικότερες είναι, να είναι όσο πιο ακριβές ως προς τον εντοπισμό γίνεται. Να είναι αποδοτικό για να μην επιβαρύνει περαιτέρω από το προβλεπόμενο τα συστήματα. Ο εντοπισμός να μπορεί να γίνει και κάτω από δυσοίωνες καιρικές συνθήκες. Για να επιτευχθούν αυτά απαιτείται να υπάρχουν καλά αποτελέσματα στο πειραματικό στάδιο τους. Έτσι παρακάτω θα αναλύσουμε τις πειραματικές επιδόσεις και θα παραθέσουμε κάποιες εικόνες από τα αποτελέσματα. Όμως, για να έχουμε καλά αποτελέσματα απαιτείται η δοκιμή επαναλαμβανόμενα μέχρις ότου να έχουμε το μέγιστο δυνατό αποτέλεσμα. Στην παρακάτω εικόνα που αναφερθήκαμε και προηγουμένως έχουμε το mAP50 όπου είναι το ποσοστό επίδοσης αναγνώρισης των δύο ετικετών στα «εύκολα» αναγνωρίσιμα αντικείμενα. Με τον όρο «εύκολα αναγνωρίσιμα», εννοούνται τα αντικείμενα τα οποία δεν απαιτούν υπερβολική προσπάθεια από τον υπολογιστή για να αναγνωριστούν. Παρατηρούμε ότι όσο προχωράει το μοντέλο αυξάνει την βεβαιότητά του άρα βελτιώνεται (στο τέλος υπάρχει μια μείωση που σημαίνει πως σε αυτή την επανάληψη είχαμε αστοχίες). Για να θεωρηθεί το μοντέλο άριστο θα πρέπει θεωρητικά να πλησιάζει όσο περισσότερο γίνεται την μονάδα δηλαδή το 1.0. Αυτό μπορεί να θεωρηθεί και αδύνατον, μιας και ο υπολογιστής μαθαίνει καλά αλλά είναι ελάχιστες οι φορές που θα λειτουργήσει τέλεια σε όλες τις περιπτώσεις. Αν λοιπόν υπάρξει το θεωρητικό 100% επιτυχία στα δείγματα τα οποία έχουμε, είναι βέβαιο πως σε ένα πραγματικό σενάριο θα υπάρχουν αντικείμενα τα οποία παρομοιάζουν με αυτά που έχουμε θέσει, για παράδειγμα ένα μπαστούνι στήριξης ενός ηλικιωμένου ανθρώπου. Στη δική μας περίπτωση έχουμε μία τελική βεβαιότητα (στη φωτογραφία) 0.709 περίπου 71%. Εάν επαναλάβουμε την εκπαίδευση με τα ίδια δεδομένα μπορεί να υπάρξει ακόμα και βελτίωση των επιδόσεων. Όπως αναφέρθηκα και παραπάνω το μοντέλο μας έχει ~80% επιτυχία κατά την διάρκεια εκπαίδευσης του.

```

+ Κώδικας + Κείμενο
[ ]
Epoch 147/150 GPU_mem 1.41G box_loss 1.476 cls_loss 1.663 dfl_loss 1.665 Instances 2 Size 640: 100% 12/12 [00:01<00:00, 8.78it/s]
Class all Images 13 Instances 18 Box(P) 0.681 R 0.507 mAP50 0.583 mAP50-95): 100% 2/2 [00:00<00:00, 9.72it/s]
all 13 18 0.681 0.507 0.583 0.337

Epoch 148/150 GPU_mem 1.39G box_loss 1.556 cls_loss 1.733 dfl_loss 1.768 Instances 2 Size 640: 100% 12/12 [00:02<00:00, 5.56it/s]
Class all Images 13 Instances 18 Box(P) 0.717 R 0.531 mAP50 0.608 mAP50-95): 100% 2/2 [00:00<00:00, 7.08it/s]
all 13 18 0.717 0.531 0.608 0.367

Epoch 149/150 GPU_mem 1.4G box_loss 1.521 cls_loss 1.702 dfl_loss 1.729 Instances 2 Size 640: 100% 12/12 [00:02<00:00, 4.72it/s]
Class all Images 13 Instances 18 Box(P) 0.507 R 0.562 mAP50 0.6 mAP50-95): 100% 2/2 [00:00<00:00, 6.16it/s]
all 13 18 0.507 0.562 0.6 0.39

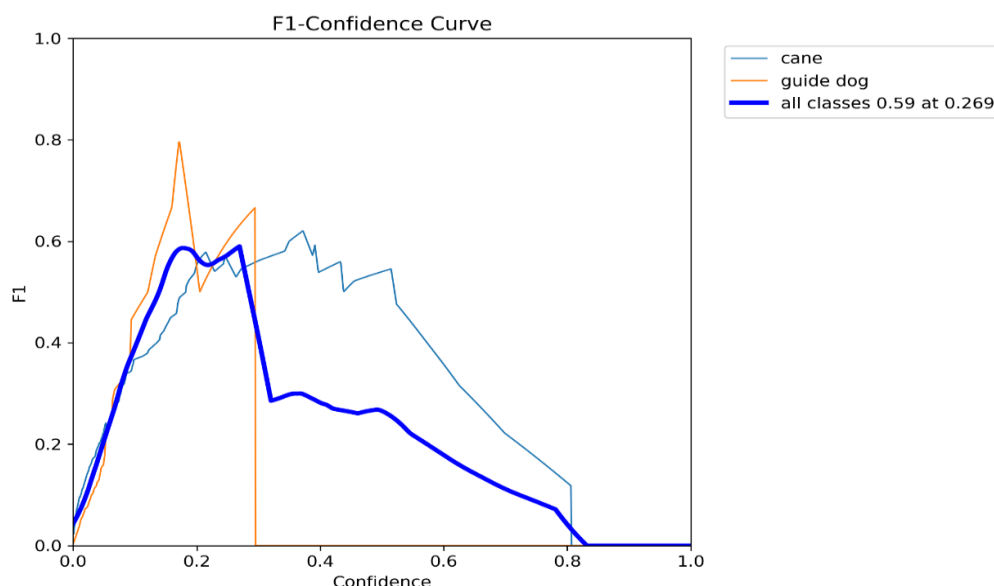
Epoch 150/150 GPU_mem 1.41G box_loss 1.593 cls_loss 1.83 dfl_loss 1.837 Instances 2 Size 640: 100% 12/12 [00:02<00:00, 4.46it/s]
Class all Images 13 Instances 18 Box(P) 0.474 R 0.594 mAP50 0.463 mAP50-95): 100% 2/2 [00:00<00:00, 9.44it/s]
all 13 18 0.474 0.594 0.463 0.303

150 epochs completed in 0.145 hours.
Optimizer stripped from runs/detect/train7/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train7/weights/best.pt, 22.5MB

Validating runs/detect/train7/weights/best.pt...
Ultralytics YOLOv8.0.196 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126358 parameters, 0 gradients, 28.4 GFLOPs
Class Images Instances Box(P) R mAP50 mAP50-95): 100% 2/2 [00:00<00:00, 6.61it/s]
all 13 18 0.521 0.594 0.709 0.435
cane 13 16 0.497 0.688 0.589 0.316
guide dog 13 2 0.546 0.5 0.828 0.554
Speed: 1.2ms preprocess, 7.6ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to runs/detect/train7
🔗 Learn more at https://docs.ultralytics.com/modes/train

```

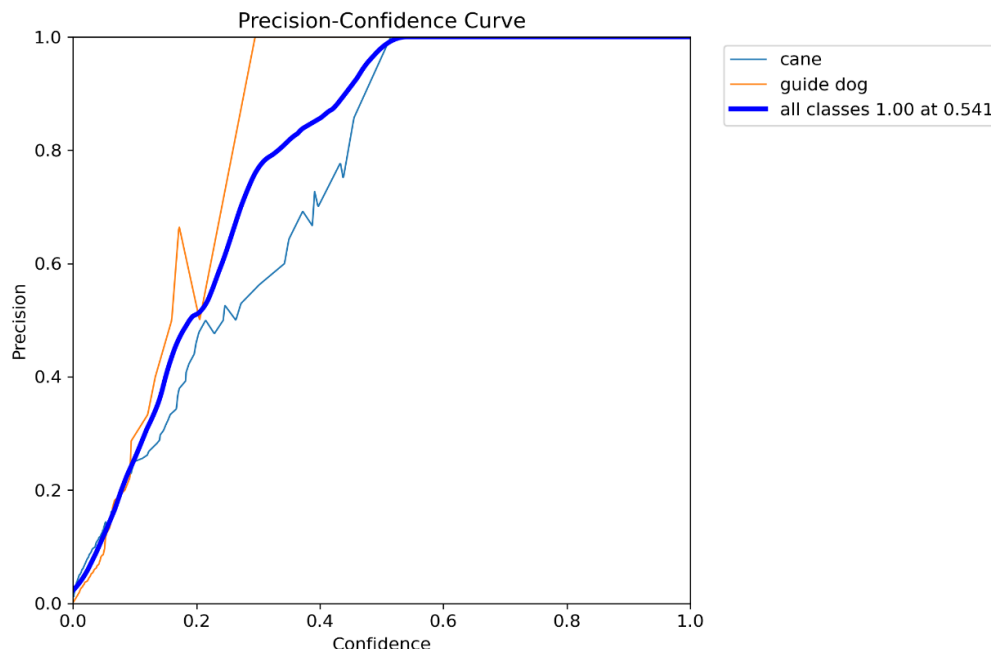
Στην παρακάτω εικόνα(Εικόνα Α) θα δούμε ένα διάγραμμα που μας παρουσιάζει την αυτοπεποίθηση (confidence) του μοντέλου που εκπαιδεύσαμε. Μας δείχνει τα δύο classes ξεχωριστά καθώς και τον μέσο όρο τους. Το σχεδιάγραμμα αυτό μας παρουσιάζει πόσο «σίγουρο» είναι το μοντέλο μας ότι εντόπισε σωστά τα αντικείμενα μας. Όσο υψηλότερα είναι η καμπύλη τόσο καλύτερα τα αποτελέσματα. Η τιμή F1 μας δίνει την αποτελεσματικότητα του μοντέλου μας.



Εικόνα Α από τα αποτελέσματα στο Google Colab

Παρατηρούμε στο από πάνω διάγραμμα (εικόνα Α) την αυξομείωση ανάμεσα στην αυτοπεποίθηση και την αποτελεσματικότητα.

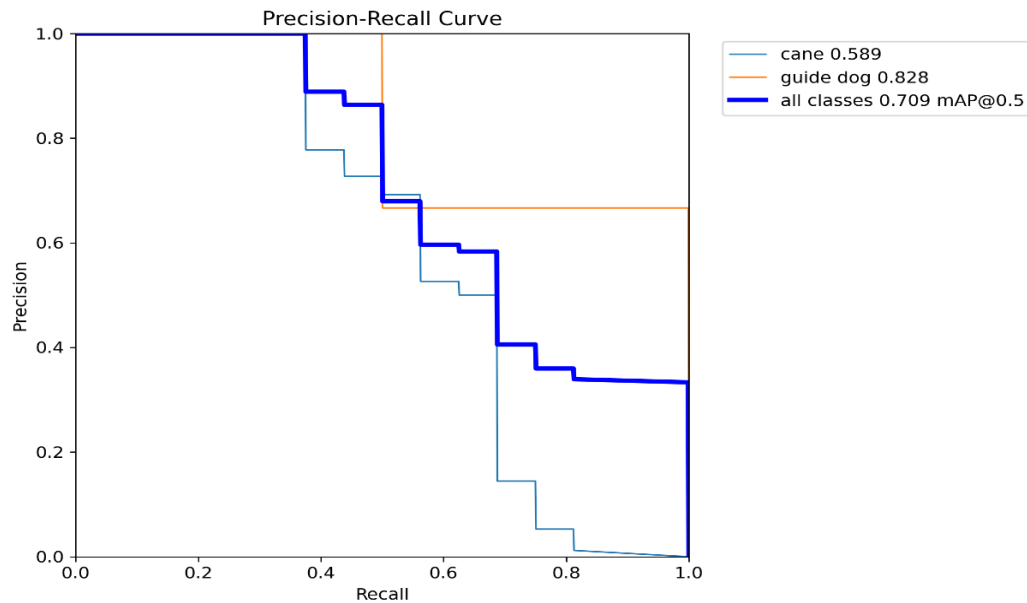
Η επόμενη εικόνα (Εικόνα Β) έχει να κάνει με την καμπύλη της ακρίβειας και της αυτοπεποίθησης. Παρατηρούμε ότι όσο αυξάνεται η ακρίβεια των εντοπισμών τόσο αυξάνεται και η αυτοπεποίθηση του μοντέλου μας.



Εικόνα Β από τα αποτελέσματα στο Google Colab

Παρατηρούμε στο παραπάνω διάγραμμα την ταυτόχρονη αύξηση της ακρίβειας και της αυτοπεποίθησης

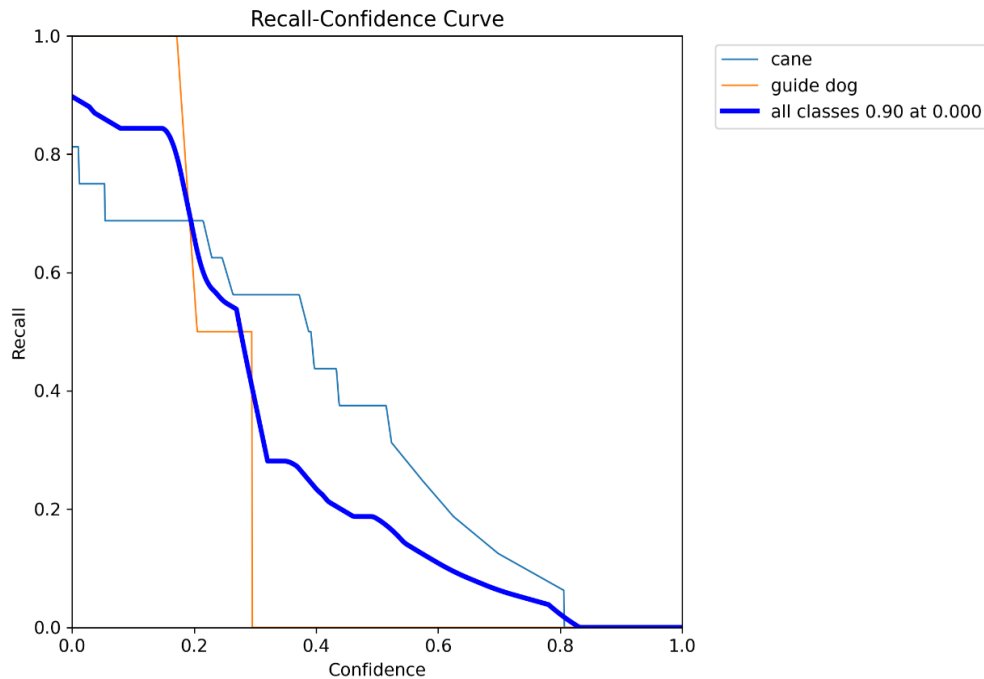
Σε αυτή την εικόνα(Εικόνα Γ) έχουμε την καμπύλη της ακρίβειας και της ανάκλησης. Μπορούμε να παρατηρήσουμε ότι όσο αυξάνονται οι ανακλήσεις, μειώνεται η ακρίβεια του μοντέλου μας. Αυτό δεν φαίνεται λογικό, καθώς θεωρητικά θα έπρεπε να αυξάνεται, όμως επειδή μας δείχνει τις τιμές σε διάφορα σημεία της υλοποίησης αυτό είναι αποδεκτό καθώς μπορούν να μπουν παραπάνω παράγοντες στον εντοπισμό που δεν έχουν συναντηθεί ή ανακαλυφθεί νωρίτερα από το πρόγραμμα μας.



Εικόνα Γ από τα αποτελέσματα στο Google Colab

Στη παραπάνω εικόνα έχουμε την καμπύλη της ακρίβειας με τον αριθμό επαναλήψεων.

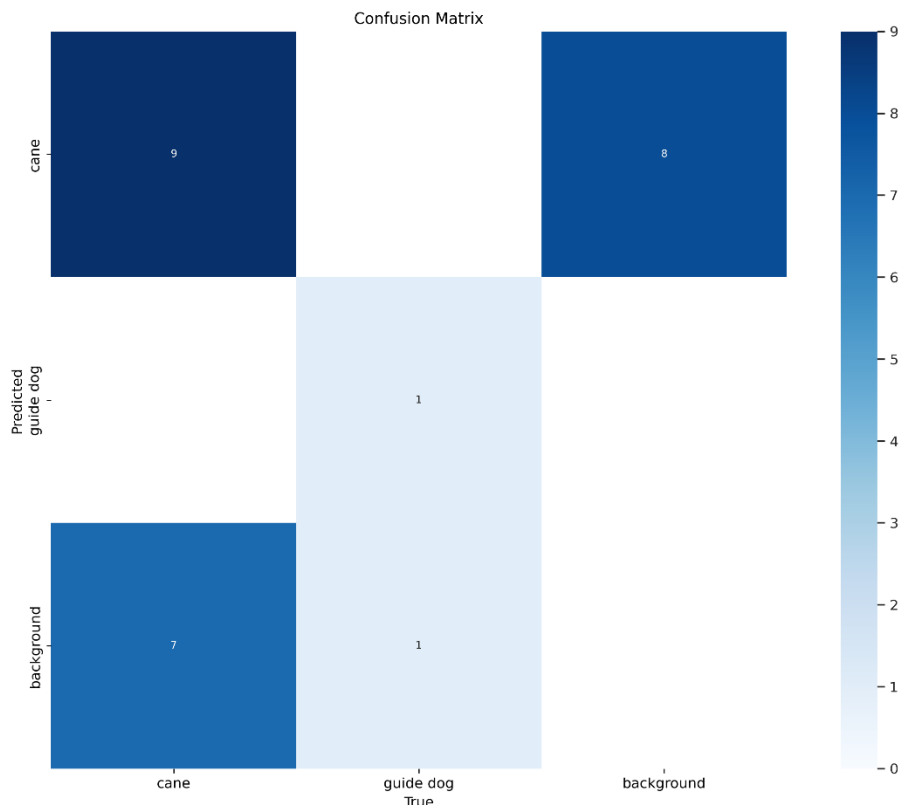
Τέλος έχουμε την καμπύλη της ανάκλησης και αυτοπεποίθησης (Εικόνα Δ). Παρατηρούμε ότι όσο περισσότερες ανακλήσεις γίνονται έχουμε μειωμένη αυτοπεποίθηση, ενώ όσο μειώνονται οι ανακλήσεις αυξάνεται και αυτοπεποίθηση του μοντέλου μας. Υπάρχει ένα σημείο όπου, η αυτοπεποίθηση έχει υψηλό σκορ σε ένα συγκεκριμένο αριθμό ανακλήσεων και μετά από αυτό τον αριθμό ξεκινά να μειώνεται. Αυτό μπορεί να οφείλεται όπως και πριν σε λάθη του μοντέλου, δηλαδή τον λανθασμένο εντοπισμό ή τον εντοπισμό περαιτέρω αντικειμένων.



Εικόνα Δ από τα αποτελέσματα στο Google Colab

Τελικώς στη παραπάνω απεικόνιση έχουμε την καμπύλη των αριθμών των επαναλήψεων με την αυτοπεποίθηση.

Επιπλέον υπάρχουν και γραφήματα που μας ενημερώνουν σχετικά με την ακρίβεια των δεδομένων που εντοπίστηκαν σε αριθμούς. Για παράδειγμα στο παρακάτω γράφημα (γράφημα1) έχουμε τον εντοπισμό εννέα μπαστούνιων σε πρώτη φάση, όπου από αυτά τα επτά είναι αληθής, καθώς έχουμε και άλλα οκτώ μπαστούνια που εντοπίστηκαν στο background. Από την άλλη για τους σκύλους-οδηγούς έχουμε μεγαλύτερη ακρίβεια, καθώς υπάρχει ένας εντοπισμός αρχικά και είναι αληθής. Λέγοντας ότι είναι πιο ακριβής αυτό ισχύει μόνο στους μαθηματικούς όρους από αυτή την οπτική γωνιά, καθώς μπορεί να έχουμε 100% επιτυχία αλλά το δείγμα είναι ελάχιστο σε σχέση με τα μπαστούνια. Αυτό μας προκαλεί σκέψεις για το αν το μοντέλο μας θα είναι αρκετά αποτελεσματικό.

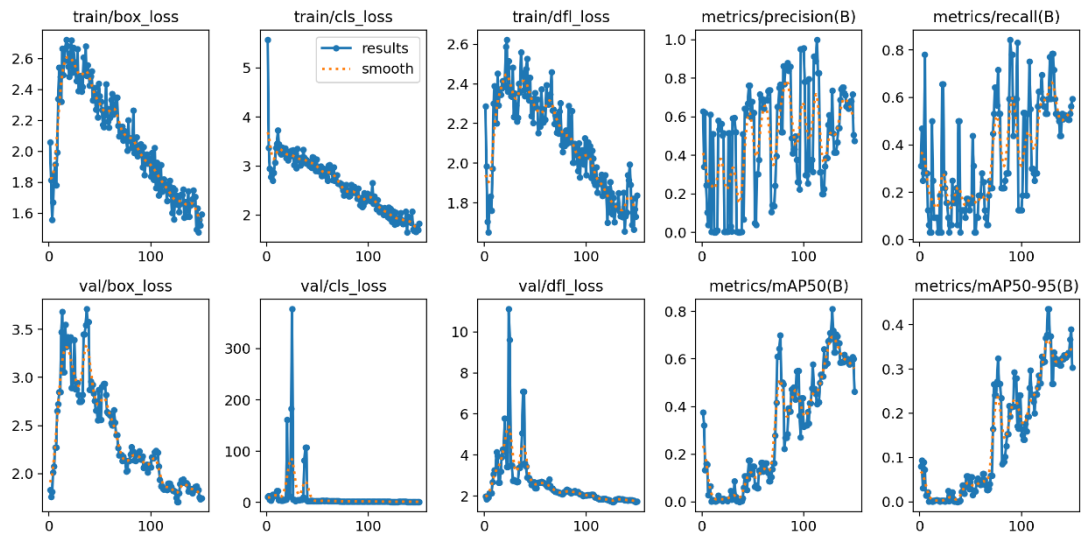


Γράφημα 1 από τα αποτελέσματα στο Google Colab

Στο παραπάνω γράφημα έχουμε τον αριθμό των εντοπισμένων σκύλων-οδηγών, των μπαστούνιων καθώς και το ποσοστό αλήθειας

Στο παρακάτω γράφημα (γράφημα2) θα δούμε κάποια γραφήματα, που το καθένα γράφημα περιέχει δικά του στοιχεία για τις επιδόσεις του προγράμματος. Αριστερά και ως πρώτο έχουμε την απώλεια του κουτιού (περιγράμματος) `train/box_loss`, που όπως βλέπουμε στον άξονα Y όσο περισσότερες επαναλήψεις πραγματοποιούνται τόσο λιγότερη απώλεια υπάρχει. Δίπλα `train/cls_loss` το οποίο ελέγχει την πρόβλεψη για το περίγραμμα και κάνει τις απαιτούμενες διορθώσεις. Τρίτο στη σειρά `train/dfl_loss` το οποίο μας ενημερώνει για την διαφορά στις ισορροπίες ανάμεσα στα δεδομένα και την βοήθεια που μας δίνει το σύστημα για να εξισορροπήσει αυτή την διαφορά. Τέταρτο έχουμε το `metrics/precision (B)`, σε αυτό το σχεδιάγραμμα αποτυπώνεται η ακρίβεια του προγράμματος μας σε κάθε επανάληψη (δηλαδή σε κάθε πέρασμα του σετ φωτογραφιών), παρατηρούμε πως με το πέρασμα των επαναλήψεων το ποσοστό επιτυχίας (είναι η κάθετη στήλη όπου 1.0 είναι το 100%) με μία μείωση του προς το τέλος. Αυτό το φαινόμενο πιθανώς οφείλετε στην υπερπροσαρμογή του προγράμματος. Πέμπτο έχουμε `metrics/recall (B)`, με την βοήθεια αυτού του σχεδιαγράμματος έχουμε την δυνατότητα να ελέγξουμε το ποσοστό αληθείας του προγράμματος μας. Με το όρο ποσοστό αληθείας εννοείται το ποσοστό που έγινε πρόβλεψη και αποδείχθηκε αληθής, δηλαδή ήταν αληθώς θετική (true positive). Όπως παρατηρείται στο σχεδιάγραμμα το ποσοστό επιτυχίας μένει ελάχιστα παραπάνω από το 0.8 (εστί 80%). Έκτο στην σειρά (πρώτο στη δεύτερη σειρά) έχουμε το `val/box_loss`, στο συγκεκριμένο έχουμε την δυνατότητα παρακολούθησης της επιτυχίας του προγράμματος να προβλέψει σωστά την σωστή τοποθέτηση του περιβλήματος (κουτιού) στις εικόνες και βίντεο κατά την διάρκεια εντοπισμού των αντικειμένων. Όσο χαμηλότερη η γραμμή, τόσο καλύτερη πρόβλεψη και προσαρμογή στο άγνωστο έχουμε. Σε

αντίθεση με το `train/box_loss` όπου ο έλεγχος γίνεται στην κατάσταση της εκπαίδευσης, εδώ γίνεται κατά την διάρκεια του `validation` (επικύρωσης). Έβδομο είναι το `val/cls_loss`, στο παρόν σχεδιάγραμμα έχουμε την ίδια συμπεριφορά όπως στο `train/cls_loss` και έχουμε την διόρθωση του περιγράμματος στην κατάσταση της επικύρωσης. Όγδοο το `val/df_l_loss`, ξανά όπως και στο `train/df_l_loss` έχουμε την ενημέρωση στην ισορροπία και την αναπροσαρμογή τους για την εξομάλυνση των δεδομένων στη κατάσταση `validation`. Τέλος τα `metric/mAP50(B)` & `metric/mAP50-95(B)` παρουσιάζουν το ποσοστό επιτυχίας για τις προβλέψεις με `confidence` 50% ενώ το δεύτερο με `confidence` από 50% έως 95%, έτσι όσο πιο υψηλά η γραμμή του διαγράμματος τόσο μεγαλύτερη ακρίβεια υπάρχει.



Γράφημα 2 από τα αποτελέσματα στο Google Colab

Στα παραπάνω γραφήματα έχουμε τις απεικονίσεις για την εκπαίδευση, για τις απώλειες, τις ακρίβειες, καθώς το ίδιο ισχύει και για το κομμάτι της επικύρωσης.

Στις παρακάτω φωτογραφίες θα δούμε κάποια παραδείγματα εντοπισμού των μπαστούνιων και των σκύλων-οδηγών.

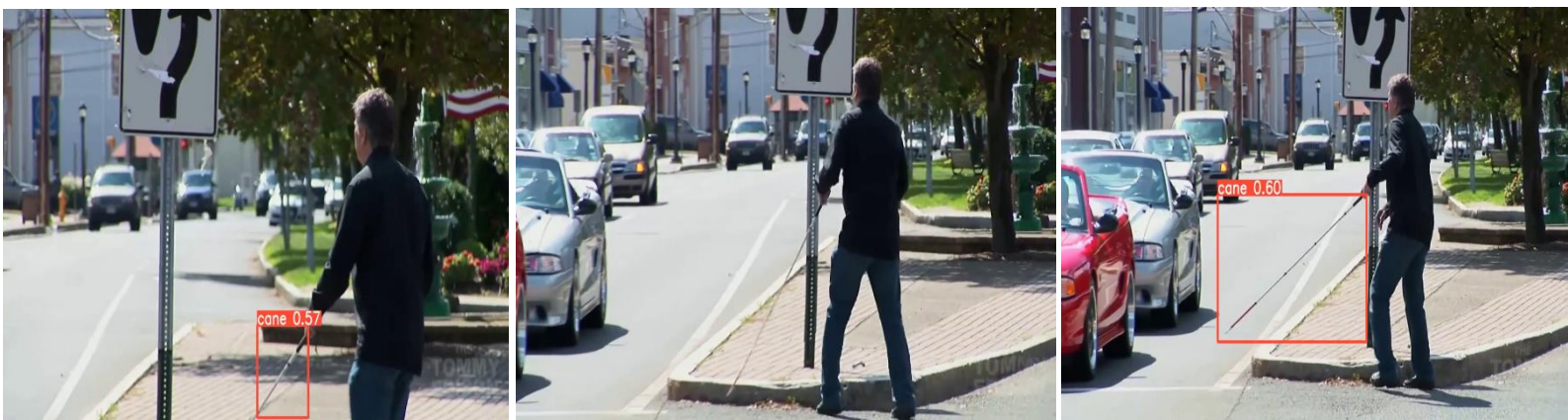
Στις τρεις πρώτες φωτογραφίες μπορούμε να δούμε τον εντοπισμό του μπαστούνιού σε μια αρκετά κοντινή λήψη. Παρατηρούμε πώς στα δύο `frames` γίνεται σωστά ο εντοπισμός του, αλλά σε ένα δεν πραγματοποιείται. Αυτό μπορεί να συμβεί για αρκετούς λόγους, κάποιοι από αυτούς είναι η αυτοπεποίθηση (`confidence`) του μοντέλου μας, στην περίπτωση μας έχουμε εντοπισμό όταν το `confidence` είναι μεγαλύτερο του 0.55 (55%). Ένας άλλος λόγος μπορεί να είναι η γρήγορη κίνηση του αντικειμένου με αποτέλεσμα να μην προλαβαίνει ο υπολογιστής να διαβάσει κάθε `frame` ή να μην το διαβάσει σωστά. Ακόμα μπορούμε να υποθέσουμε και ότι στο δοσμένο `set` φωτογραφιών δεν υπήρχαν αρκετά παραδείγματα για την συγκεκριμένη γωνία λήψης.



Πηγή εικόνων: Επεξεργασμένα βίντεο με το πρόγραμμα εντοπισμού.

Στις παραπάνω εικόνες έχουμε αριστερά τον εντοπισμό του μπαστουνιού, στο κέντρο την απώλεια εντοπισμού του και αριστερά τον εντοπισμό του ξανά.

Σε αυτές τις φωτογραφίες μπορούμε να δούμε ένα αντίστοιχο αποτέλεσμα. Όμως στην πρώτη φωτογραφία παρατηρούμε ότι γίνεται εντοπισμός μπαστουνιού χωρίς να εμφανίζεται ολόκληρο στην εικόνα.



Πηγή εικόνων: Επεξεργασμένα βίντεο με το πρόγραμμα εντοπισμού.

Στις τρεις παραπάνω εικόνες βλέπουμε αριστερά τον εντοπισμό (έστω και μερικώς από το μπαστούνι), στο κέντρο έχουμε την απώλεια και δεξιά τον εντοπισμό του ξανά.

Στις επόμενες φωτογραφίες μπορούμε να δούμε ακόμα ένα παράδειγμα σωστού εντοπισμού στα αριστερά μας, αλλά και ένα εντοπισμό μπαστουνιού στήριξης στα δεξιά μας, που όπου όπως προαναφερθήκαμε και νωρίτερα μπορεί να συμβεί κάτι τέτοιο. Θα πρέπει να λαμβάνουμε υπόψιν μας τέτοια σφάλματα και να τα βελτιώνουμε όσο το δυνατόν περισσότερο.



Πηγή εικόνων: Επεξεργασμένα βίντεο με το πρόγραμμα εντοπισμού.

Στις δύο εικόνες από πάνω παρατηρούμε αριστερά τον εντοπισμό ενός μπαστουνιού για τυφλούς και δεξιά τον λανθασμένο εντοπισμό μπαστουνιού ενός ηλικιωμένου ατόμου.

Παρακάτω υπάρχουν φωτογραφίες εντοπισμού των σκύλων-οδηγών που όπως γίνεται εύκολα αντιληπτό και παρατηρούμε ότι υπάρχουν σωστοί αλλά και λάθος εντοπισμοί. Και σε αυτή την περίπτωση μπορεί η γωνία λήψης να μην είναι η ιδανική, ή κάποια στοιχεία να έχουν παρόμοια χαρακτηριστικά με τα δεδομένα από το set που δώσαμε, με αποτέλεσμα ο υπολογιστής να μπερδεύεται.



Πηγή εικόνων: Επεξεργασμένα βίντεο με το πρόγραμμα εντοπισμού.

Στις εικόνες αυτές έχουμε τον σωστό εντοπισμό του σκύλου-οδηγού στα αριστερά αλλά και τον λανθασμένο εντοπισμό ενός άλλου αντικειμένου στα δεξιά.

Τέλος, μπορούμε να δούμε πως το μοντέλο μας είναι ικανό να ξεχωρίσει τους σκύλους συντροφιάς από τους σκύλους οδηγούς, κάτι το οποίο κρίνεται απαραίτητο για την λειτουργία που θέλουμε να του δώσουμε. Πάλι όμως έχουμε λανθασμένο εντοπισμό, καθώς το πεζοδρόμιο σε συνδυασμός με τον δρόμο από αυτή την γωνία λήψης προκαλούν σύγχυση στο πρόγραμμα μας, κάνοντας το να εντοπίζει μπαστούνι.



Πηγή εικόνων: Επεξεργασμένα βίντεο με το πρόγραμμα εντοπισμού.

Στις παραπάνω εικόνες φαίνεται ο ορθός εντοπισμός του σκύλου-οδηγού και η παράβλεψη ενός σκύλου συντροφιάς, ενώ δεξιά φαίνεται ο λανθασμένος εντοπισμός του πεζοδρομίου ως μπαστούνι.

Συμπερασματικά, μπορούμε να αντλήσουμε σαν γενικό συμπέρασμα ότι δεν υπάρχει επιτυχία 100%, καθώς αυτό θα ήταν ένα ουτοπικό σενάριο. Εάν θεωρήσουμε και εφαρμόσουμε το σενάριο απόλυτης επιτυχίας θα δημιουργηθούν μεταγενέστερα προβλήματα με μαθηματική ακρίβεια. Το καλύτερο δυνατό σενάριο είναι να τα υπολογίσουμε, έστω με ένα μικρό ποσοστό σφάλματος, έτσι ώστε να υπάρχει η δυνατότητα αναπροσαρμογής και βελτίωσης του προγράμματος, χωρίς να θέσουμε τις ζωές των ανθρώπων σε κίνδυνο, με την δυνατότητα πρόβλεψης σφάλματος και έγκαιρης επέμβασης.

ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα και Μελλοντικές Επεκτάσεις

Σε αυτό το κεφάλαιο όντας και τελευταίο θα γίνει η παρουσίαση κάποιων συμπερασμάτων και ύστερα οι δυνατότητες επέκτασης προγραμμάτων και τεχνολογιών παρόμοιας φιλοσοφίας.

6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Υπάρχουν αρκετά συμπεράσματα τα οποία έχουν προκύψει κατά την διάρκεια δημιουργίας του προγράμματος, αλλά καθώς και στη συγγραφή της εν λόγω εργασίας. Αρχικά, για την ενασχόληση μας με τέτοιου είδους προγράμματα θα πρέπει να έχουμε επιλέξει ποιο μοντέλο εντοπισμού θα χρησιμοποιήσουμε, καθώς όπως αναφερθήκαμε και νωρίτερα υπάρχουν αρκετές επιλογές. Δεύτερον, θα πρέπει να αποκτήσουμε γνώσεις, σε προγραμματιστικό επίπεδο, ικανές να ανταπεξέλθουμε σε συνήθη εμπόδια, καθώς και την καλή κατανόηση της μηχανικής μάθησης και των υποκατηγοριών της. Τρίτον, είναι πολύ σημαντικό να δημιουργήσουμε ένα επαρκές σετ φωτογραφιών και βίντεο για τον εκτενή έλεγχο της ακρίβειας του προγράμματος μας, εφόσον αν δεν έχουμε επαρκή ποσότητα, καθαρότητα και ορθά παραδείγματα εικόνων το πρόγραμμα μας δεν θα αποδώσει όσο επιθυμούμε. Τέταρτον, έχουμε τον αριθμό των επαναλήψεων που θα πρέπει να του αναθέσουμε να κάνει, έτσι ώστε να μην δημιουργήσουμε το φαινόμενο της υπερπροσαρμογής, αλλά και να μην έχουμε το πρόβλημα της ελλιπής προσαρμογής και εκμάθησης.

Επιπλέον, παρατηρήθηκε η δυσκολία αποφυγής σφάλματος ως προς τον εντοπισμό των αντικειμένων, κάτι το οποίο είναι αναγκαίο να περιοριστεί όσο το δυνατόν περισσότερο, και την συνεχή βελτίωση του. Ακόμα, δεν μπορούμε να παραβλέψουμε τον λανθασμένο εντοπισμό των αντικειμένων που θέσαμε (όπως για παράδειγμα το μπαστούνι ενός ηλικιωμένου ατόμου, το πρόβλημα αυτό μπορεί να επιλυθεί και με άλλους τρόπους εκτός της βελτίωσης εντοπισμού. Ένας τρόπος είναι να θεωρήσουμε και αυτά τα μπαστούνια ως αντικείμενα της κλάσης μας και να ενεργοποιηθεί ο μηχανισμός μας κατά τον εντοπισμό τους, αυτό θα γίνει για την αποφυγή αλλαγής πληροφοριών στο πακέτο εικόνων και την πιθανή δημιουργία αλλού προβλήματος.

Παρατηρήσαμε στις παραπάνω εικόνες, στην ενότητα 4, κάποια αποτελέσματα ποσοστού σιγουριάς του προγράμματος μας να είναι ~71% και να έχουμε επιτύχει ποσοστό σιγουριάς ~80% με την βελτίωση του. Το πρόγραμμα μας θα πρέπει να έχει όσο το δυνατόν μεγαλύτερο ποσοστό επιτυχίας. Κατά την άποψη μου ένα ποσοστό ~95% είναι απόλυτα ικανοποιητικό και μπορεί να ενεργήσει χωρίς πολλές παρεμβάσεις από τον άνθρωπο. Ένα τέτοιο ποσοστό επιτυχίας όμως σημειώνεται στο σετ το οποίο εμείς αναρτήσαμε, έτσι θα πρέπει το σετ φωτογραφιών μας να είναι ολοκληρωμένο, για την μεγαλύτερη δυνατή κάλυψη συμβάντων και οπτικών γωνιών που μπορεί να υπάρξει σε ένα φυσικό περιβάλλον. Φυσικά κανένα σύστημα δεν είναι αλάνθαστο, είναι αναγκαίο να υπάρχει μια μερική επίβλεψη για την αποφυγή και την πρόληψη σφαλμάτων.

6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στην εποχή μας, όπου η τεχνολογία βρίσκεται στην ακμή της, και συνεχίζει να ακμάζει ολοένα και περισσότερο έχουμε την δυνατότητα να βοηθήσουμε άτομα τα οποία χρήζουν βοήθειας. Με αυτή την σκέψη έγινε και η δημιουργία αυτής της εφαρμογής, για την βοήθεια των ατόμων με προβλήματα όρασης. Το πρόγραμμα αυτό είναι μόνο η αρχή για το τι μπορούμε σαν άνθρωποι και επιστημονικός κλάδος να προσφέρουμε σε αυτούς τους ανθρώπους, και φυσικά σε ακόμα περισσότερους. Οι επεκτάσεις της εφαρμογής αυτής μπορούν να είναι ποικίλες.

Μια εύκολη λύση θα είναι να εμπλουτίσουμε τον ήδη υπάρχον εξοπλισμό, ο οποίος είναι το μπαστούνι. Θα μπορούσαμε να δημιουργήσουμε έξυπνα μπαστούνια, όπου με την υποβοήθηση παρόμοιων προγραμμάτων θα μπορούσαν να εντοπίζουν όχι μόνο την προσέλευση σε μια διασταύρωση, αλλά και τα τυχόν εμπόδια που θα προκύψουν στην διαδρομή του ατόμου αυτού. Επιπλέον, με ένα ειδικό ακουστικό ή smart band/smartwatch (wearables) θα έχουμε την δυνατότητα να τον ειδοποιεί ότι πλησιάζει σε ένα αντικείμενο, αναλόγως το μέγεθος του να εκπέμπει και ένα αντίστοιχο ήχο ή δόνηση, ανάλογα το αξεσουάρ που θα επιλέξει ο καθένας. Ακόμα παραπάνω θα μπορούσαν να ενσωματώνουν υπέρυθρες ακτίνες για τον εντοπισμό των εμποδίων κάτι το οποίο δεν θα απαιτούσε μεγάλες μπαταρίες για την λειτουργία των καμερών. Επιπλέον, θα μπορούσαμε να κάνουν χρήση ειδικών (έξυπνων) γυαλιών τα οποία θα περιέχουν κάμερες και αισθητήρες για την προειδοποίηση τους σε περίπτωση εμποδίων, κάτι το οποίο όπως και με τις κάμερες και τους αισθητήρες στο μπαστούνι θα λειτουργούν με ζωντανό εντοπισμό και θα προσαρμόζονται ανάλογα την κατάσταση.

Από την άλλη πλευρά οι σκύλοι-οδηγοί μπορούν να εξοπλιστούν με έξυπνα κολάρα που περιλαμβάνουν GPS, αισθητήρες περιβάλλοντος και μικρές κάμερες. Με αυτά τα κολάρα θα μπορούν να στέλνονται σήματα/ειδοποιήσεις στα αξεσουάρ του ατόμου(wearables) και να τον ειδοποιεί για τυχόν εμπόδια ή διασταυρώσεις. Επιπλέον, τα δεδομένα μπορούν να αποστέλλονται σε εξειδικευμένες υπηρεσίες για άμεση υποστήριξη και συμβουλές εάν βρεθούν σε δυσμενείς καταστάσεις.

Σε συνδυασμό με την άμεση υποστήριξη και τα «τεχνολογικά» γυαλιά θα ήταν εφικτό να δημιουργήσουμε ένα «κουμπί πανικού» (panic button) το οποίο να υπάρχει είτε στα ίδια τα γυαλιά είτε ως επιπλέον αξεσουάρ. Με το κουμπί αυτό θα μπορεί ο χρήστης να αιτηθεί σύνδεση με εργαζομένους στην άμεση υποστήριξη και να τους παραχωρήσει πρόσβαση στις κάμερες των γυαλιών. Με αυτό τον τρόπο όταν ένα άτομο με προβλήματα όρασης, που δεν είναι εξοικειωμένο με το μπαστούνι και να προσανατολίζεται με αυτό με αποτέλεσμα να έχει «χαθεί» θα μπορεί ο εργαζόμενος απομακρυσμένα να τον καθοδηγεί. Αυτό φυσικά θα πρέπει να γίνει με συνδυασμό τεχνολογίας GPS για την καλύτερη και ακριβέστερη εξυπηρέτηση.

Με τα τεχνολογικά αυτά μέσα έχουμε την δυνατότητα να βελτιώσουμε την καθημερινότητα πολλών ατόμων παγκοσμίως. Όμως δεν πρέπει να ξεχνάμε ότι υπάγονται στους ίδιους νόμους με όλους τους άλλους πολίτες τις χώρας τους, έτσι

έχουν και τα ίδια δικαιώματα. Ως επιστημονικός κλάδος οφείλουμε να σεβαστούμε τα προσωπικά δεδομένα και την ελευθερία επιλογών του καθενός, και να μην υπάρξουν φαινόμενα (ακραίες περιπτώσεις) κατασκοπίας ή παρακολούθησης χωρίς την συγκατάθεση του εκάστοτε ατόμου ξεχωριστά. Φυσικά επειδή μιλάμε για μια τεχνολογία που έχει πρόσβαση στην ζωή του ατόμου (μέσω των καμερών και μικροφώνων), η συγκατάθεση θα πρέπει να γίνεται ανά τακτά χρονικά διαστήματα (πιθανώς και ημερήσια συγκατάθεση).

Η τεχνολογία είναι ένα πολύτιμο «όπλο» που έχουμε στην φαρέτρα μας για την αντιμετώπιση πολλών δυσκολιών. Όμως δεν πρέπει να ξεχνάμε ότι η τεχνολογία είναι για να μας εξυπηρετεί όσο δεν παραβιάζονται θεμελιώδη δικαιώματα μας όπως είναι το δικαίωμα στην ιδιωτικότητα, και να μην παραβιάζεται ανεξαρτήτως την κατάσταση.

Τεχνητή Νοημοσύνη(AI) και Μηχανική Μάθηση

Η χρήση τεχνητής νοημοσύνης (AI) και μηχανικής μάθησης μπορεί να βελτιώσει δραματικά τις δυνατότητες των συστημάτων καθοδήγησης. Οι AI αλγόριθμοι μπορούν να αναλύουν συνεχώς τα δεδομένα από τους αισθητήρες, να προβλέπουν και να αποφεύγουν κινδύνους και να προσαρμόζουν τις οδηγίες με βάση τις συνθήκες του περιβάλλοντος. Ταυτόχρονα, η μηχανική μάθηση μπορεί να προσαρμόζει τις λειτουργίες του μπαστουνιού ή του κολάρου ανάλογα με τις ανάγκες και τις προτιμήσεις του χρήστη. Έτσι έχουμε μια άμεση λύση καθώς ένας προσωπικός βοηθός AI θα μπορεί να ενσωματωθεί σε κάθε ένα αντικείμενο ξεχωριστά, δεν θα χρειάζεται επέμβαση του ανθρώπου για πιο απλές διεργασίες. Παράλληλα σε συνδυασμό με το «κουμπί πανικού» θα μπορούσαν να υπάρχουν πολλές λειτουργίες, όπως παραδείγματος χάριν με ένα πάτημα να γινόταν σύνδεση με τους ηλεκτρονικούς χάρτες (πού όλοι έχουμε στα smartphones μας) και να δινόταν αυτόματα η εντολή για καθοδήγηση προς την οικία του ατόμου. Επιπλέον με την βοήθεια των ακουστικών να υπάρχει ορθή φωνητική καθοδήγηση. Ακόμα θα μπορούσαν να ενσωματώσουμε τις AI λειτουργίες στα ειδικά γυαλιά οράσεως για την καθοδήγηση ή και την εκφώνηση γραπτών κειμένων σε πινακίδες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- 1.<https://www.ert.gr/teleytaia-nea/15-%CE%BF%CE%BA%CF%84%CF%89%CE%B2%CF%81%CE%AF%CE%BF%CF%85-%CF%80%CE%B1%CE%B3%CE%BA%CF%8C%CF%83%CE%BC%CE%B9%CE%B1-%CE%B7%CE%BC%CE%AD%CF%81%CE%B1-%CE%BB%CE%B5%CF%85%CE%BA%CE%BF%CF%8D-%CE%BC%CF%80%CE%B1/>
- 2.https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE_%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7
- 3.<https://www.sap.com/greece/products/artificial-intelligence/what-is-machine-learning.html>
- 4.<https://www.foreseemed.com/blog/machine-learning-in-healthcare>
- 5.<https://corporatefinanceinstitute.com/resources/data-science/machine-learning-in-finance/>
- 6.https://mailchimp.com/resources/machine-learning-in-marketing-guide/?ds_c=DEPT_AOC_Google_Search_ROW_EN_NB_Acquire_Broad_DSA-Rsrc_T5&ds_kids=p80322580036&ds_a_lid=dsa-2227026702184&ds_cid=71700000119083212&ds_agid=58700008729598297&gad_source=1&gclid=CjwKCAjwnqK1BhBvEiwAi7o0X4SEo4nuoy1Pk5qI_7y2ep7oO87Ghm8pifRvs4yv5Fsv5YnhtvU3cBoCqioQAvD_BwE&gclsrc=aw.ds
- 7.<https://www.cisco.com/c/en/us/products/security/machine-learning-security.html#~how-ml-works>
- 8.<https://greeco.gr/business/techniti-noimosyni/vathia-mathisi/>
- 9.<https://eclass.upatras.gr/modules/document/file.php/PT187/11.DEEP%20LEARNING%2C%20MACHINE%20LEARNING.pdf>
- 10.<https://brilliant.org/wiki/backpropagation/>
- 11.<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7778711/>
- 12.<https://medium.com/@usama.6832/introduction-to-deep-learning-in-healthcare-519336d651b1>
- 13.<https://bigblue.academy/gr/neuronika-diktua>
- 14.<https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning>

15. <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
16. <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/>
17. <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/>
18. <https://www.geeksforgeeks.org/generative-adversarial-network-gan/>
19. <https://ypodomes.com/ichitika-systimata-topothetithikan-se-fanaria-tis-patras-gia-atoma-me-provlimata-orasis/>
20. <https://speakingcity.org/>
21. <https://play.google.com/store/apps/details?id=com.stp.speakingcity&hl=ru>
(ρωσική έκδοση)
22. <https://play.google.com/store/apps/details?id=com.stp.speakingcity>
23. <https://speakingcity.org/about/>
24. <https://social-tech.ru/support/members/info-tex/govoryashhij-gorod.html>
25. <https://www.maty.gr/%CF%84%CF%85%CF%86%CE%BB%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CE%B1-%CF%80%CF%81%CF%8C%CF%83%CE%B2%CE%B1%CF%83%CE%B7/1926-%CF%84%CE%B1-%C2%AB%CE%AD%CE%BE%CF%85%CF%80%CE%BD%CE%B1%C2%BB-%CF%85%CF%80%CE%B5%CF%81%CF%83%CF%8D%CE%B3%CF%87%CF%81%CE%BF%CE%BD%CE%B1-%CF%86%CE%B1%CE%BD%CE%AC%CF%81%CE%B9%CE%B1-%CF%80%CE%BF%CF%85-%CE%BC%CE%B9%CE%BB%CE%AC%CE%BD%CE%B5,-%CE%B3%CE%B9%CE%B1-%CF%84%CF%85%CF%86%CE%BB%CE%BF%CF%8D%CF%82-%CF%83%CF%84%CE%B7-%CE%B8%CE%B5%CF%83%CF%83%CE%B1%CE%BB%CE%BF%CE%BD%CE%AF%CE%BA%CE%B7.html>
26. <https://www.letsenvision.com/blog/top-5-assistive-technology-devices-for-people-who-are-blind-or-have-low-vision-2023>
27. <https://roboflow.com/about>
28. <https://docs.ultralytics.com/guides/yolo-performance-metrics/#class-wise-metrics>
29. <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>

30. <https://github.com/ultralytics/ultralytics/issues/7444>
31. <https://github.com/ultralytics/ultralytics/issues/10375>