

Ατομική Διπλωματική Εργασία

**ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΓΕΩΤΟΠΟΘΕΤΗΣΗΣ ΜΕ
ΧΡΗΣΗ ΜΗΧΑΝΙΚΗΣ ΟΡΑΣΗΣ**

Λουκία Ιωάννου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάϊος 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση Συστήματος Γεωτοποθέτησης με Χρήση Μηχανικής Όρασης

Λουκία Ιωάννου

Επιβλέπων Καθηγητής
Δημήτρης Ζεϊναλιπούρ

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του
Πανεπιστημίου Κύπρου

Μάϊος 2021

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή Δρ. Δημήτρη Ζεϊναλιπούρ για τη συνεχή υποστήριξη και τη βοήθειά του καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας από την επιλογή του θέματος μέχρι τη συγγραφή της. Θα ήθελα επίσης να ευχαριστήσω τον κ. Πασχάλη Μπέη του οποίου η βοήθεια και οι γνώσεις ιδιαίτερα σε δύσκολες στιγμές της πορείας της πτυχιακής μου εργασίας ήταν πολύτιμες και απαραίτητες για την επιτυχή ολοκλήρωσή της.

Θα ήθελα επίσης να ευχαριστήσω όλο το διδακτικό προσωπικό του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου αφού μέσα από τα μαθήματα που μου πρόσφερε με βοήθησε να αποκτήσω τα κατάλληλα εφόδια για να ολοκληρώσω με επιτυχία τον τετραετή κύκλο σπουδών Πληροφορικής του Πανεπιστημίου. Χάρη σε αυτά τα εφόδια και τις εμπειρίες, που το Πανεπιστήμιο μου πρόσφερε, πιστεύω ότι θα μπορώ να αντιμετωπίσω τα εμπόδια που πιθανόν να συναντήσω στην επαγγελματική μου πορεία, ενώ παράλληλα θα έχω τις απαραίτητες γνώσεις για μια αξιόλογη σταδιοδρομία στο χώρο αυτό.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένειά μου η οποία στήριξε από την αρχή την επιλογή μου για αυτόν τον κλάδο σπουδών και στη συνέχεια με υποστήριξε σε όλη τη διάρκεια των σπουδών μου. Η οικογένειά μου με ώθησε να αξιοποιήσω στο μέγιστο βαθμό τις δυνατότητές μου ώστε να ολοκληρώσω με μεγάλη επιτυχία τις σπουδές μου αποκτώντας στα τέσσερα χρόνια το πτυχίο της Πληροφορικής.

Περίληψη

Βασικός στόχος της παρούσας διπλωματικής ήταν η υλοποίηση ενός συστήματος γεωτοποθέτησης του χρήστη σε εσωτερικούς χώρους με τη χρήση της τεχνολογίας Μηχανικής Όρασης. Ο χρήστης μέσω του συστήματος αυτού θα έχει τη δυνατότητα εύρεσης της ακριβούς του θέσης μέσα σε ένα εσωτερικό χώρο, καταγράφοντας με την κάμερα της συσκευής του τον περιβάλλοντά του χώρο.

Αρχικά, για την υλοποίηση του συστήματος χρησιμοποιήσαμε το ήδη υλοποιημένο σύστημα YOLO, το οποίο προσαρμόσαμε για να εκτελείται στη συσκευή του χρήστη. Το YOLO είναι ένα σύστημα το οποίο λαμβάνει ως είσοδο μια εικόνα ή ένα frame από κάποιο βίντεο, το επεξεργάζεται και ακολούθως επιστρέφει στον χρήστη την εικόνα με σημάνσεις των αναγνωρισμένων αντικειμένων για τη συγκεκριμένη εικόνα.

Το σύστημα που υλοποιήσαμε αποτελείται από τρία υποσυστήματα: τη βάση δεδομένων, το υποσύστημα του Logger και το υποσύστημα του Navigator. Αρχικά, ο χρήστης χρησιμοποιώντας τον Logger, καταγράφει διάφορες θέσεις και σε κάθε θέση δημιουργείται ένα fingerprint. Το fingerprint περιλαμβάνει τις συντεταγμένες της θέσης, οι οποίες ανακτώνται από την ίδια τη συσκευή ή από το χρήστη. Στο fingerprint μαζί με τις συντεταγμένες της θέσης αποθηκεύονται και τα αντικείμενα που αναγνωρίστηκαν στο συγκεκριμένο σημείο.

Στη συνέχεια, μετά την καταγραφή των θέσεων, ένας άλλος χρήστης μπορεί χρησιμοποιώντας τον Navigator να καταγράψει τον περιβάλλοντά του χώρο και να του δοθεί η θέση που βρίσκεται. Καταγράφοντας τον περιβάλλοντα χώρο δημιουργείται ένας προσωρινός πίνακας με τα αντικείμενα που ανιχνεύθηκαν. Χρησιμοποιώντας τον συγκεκριμένο πίνακα, ο Navigator, εκτελεί ένα αλγόριθμο στη βάση δεδομένων οποίος βρίσκει το fingerprint το οποίο μοιάζει περισσότερο με τον πίνακα που μόλις δόθηκε σαν είσοδος. Αφού ολοκληρωθεί η εκτέλεση του αλγορίθμου τότε επιστρέφονται στον χρήστη οι συντεταγμένες της θέσης που αντιστοιχούν στο fingerprint που βρέθηκε. Οι συντεταγμένες της θέσης αυτής εμφανίζονται στον χρήστη σαν ένα pop-up παράθυρο.

Μετά την ολοκλήρωση της υλοποίησης, εκτελέσαμε κάποια πειράματα για να διαπιστώσουμε την ορθότητα της εν λόγω εφαρμογής. Η πειραματική αξιολόγηση

εκτελέστηκε σε χώρο κατοικίας λόγω των περιορισμένων αντικειμένων που αναγνωρίζει το YOLO χωρίς νέα εκπαίδευση. Τα πειράματα έδειξαν ότι η εφαρμογή λειτουργεί με αρκετά μεγάλη επιτυχία και μπορεί εύκολα να επιτύχει ακρίβεια μεγαλύτερη από 90% χωρίς εξειδικευμένη προετοιμασία. Παράλληλα η πειραματική αυτή εφαρμογή έδωσε αρκετά ενδιαφέροντα αλλά και ενθαρρυντικά αποτελέσματα για τη μελλοντική επέκταση και ανάπτυξη του συστήματος για βελτίωση της ακρίβειάς της αλλά και αξιοποίηση για άλλες χρήσεις.

Περιεχόμενα

Ευχαριστίες	iii
Περίληψη	iv
Κεφάλαιο 1: Εισαγωγή	1
1.1 Υποκίνηση Εργασίας	1
1.2 Περίγραμμα Εργασίας	2
1.3 Συνεισφορές	4
Κεφάλαιο 2: Υφιστάμενη Γνώση και Σχετικές Εργασίες	5
2.1 Κυριότεροι Όροι της Διπλωματικής	5
2.2 Κατηγοριοποίηση Συστημάτων Εσωτερικής Πλοήγησης	7
2.3 Σύστημα YOLO	10
2.3.1 Τρόπος Λειτουργίας YOLO	10
2.3.2 Εκπαίδευση Συστήματος YOLO	13
2.4 Άλλα Συστήματα Πλοήγησης Εσωτερικού Χώρου	15
2.4.1 AnyPlace	15
2.4.2 MazeMap	18
Κεφάλαιο 3: Αρχιτεκτονική Συστήματος	20
3.1 Ροή Δεδομένων	20
3.2 Εισαγωγή Δεδομένων	22
3.3 Εμφάνιση Αποτελεσμάτων	24
Κεφάλαιο 4: Υλοποίηση Logger με Χρήση Μηχανικής Όραση	26
4.1 Περιγραφή Logger του AnyPlace	26
4.2 Πρώτη Εκτέλεση του Συστήματος	28
4.3 Γραφικό Περιβάλλον	29
4.4 Επεξεργασία Στιγμιότυπων με Χρήση YOLO	31

4.5 Εισαγωγή Δεδομένων σε Βάση	33
4.6 Υλοποίηση Υποσυστήματος Demo.....	35
4.7 Υλοποίηση Μεθόδου για μη Αναγνώριση Συγκεκριμένων Αντικειμένων	37
Κεφάλαιο 5 : Υλοποίηση Navigator με Χρήση Μηχανική Όραση.....	40
5.1 Γραφικό Περιβάλλον Navigator.....	40
5.2 Τρόπος Εκτέλεσης.....	42
5.3 Περιγραφή Αλγορίθμου για Εύρεση Fingerprint	43
Κεφάλαιο 6: Πειραματική Αξιολόγηση του Συστήματος	48
6.1 Μέθοδος Αξιολόγησης	48
6.2 Περιγραφή Πειράματος 1: Με Χρήση Τυχαίων Αντικειμένων	49
6.3 Περιγραφή Πειράματος 2: Με Χρήση Φωτογραφιών.....	53
6.4 Περιγραφή Πειράματος 3: Με Συνδυασμό Αντικειμένων	54
6.5 Συμπεράσματα από την Πειραματική Εφαρμογή.....	57
Κεφάλαιο 7: Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	60
7.1 Συμπεράσματα.....	60
7.2 Μελλοντικές Επεκτάσεις.....	61
Βιβλιογραφία	65
Παράρτημα Α	A1

Κεφάλαιο 1

Εισαγωγή

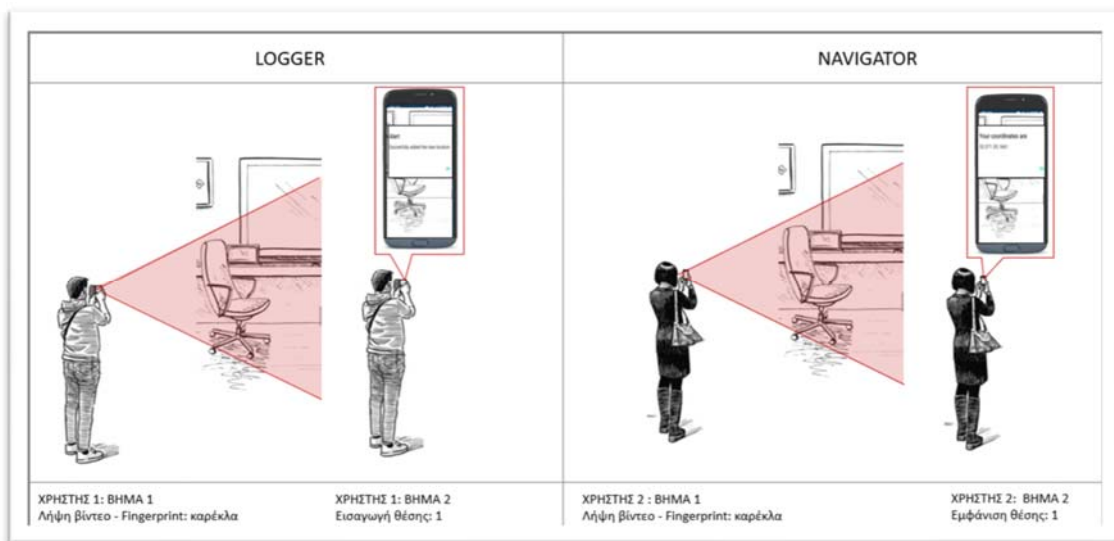
1.1	Υποκίνηση Εργασίας	1
1.2	Περίγραμμα Εργασίας	2
1.3	Συνεισφορές	4

1.1 Υποκίνηση Εργασίας

Πρόσφατες έρευνες έδειξαν ότι οι άνθρωποι περνάνε το μεγαλύτερο μέρος του χρόνου τους (πάνω από το 75%) σε εσωτερικούς παρά σε εξωτερικούς χώρους, με αποτέλεσμα να δημιουργείται έντονα η ανάγκη εντοπισμού της θέσης στην οποία βρίσκονται σε ένα εσωτερικό χώρο. Οι άνθρωποι δηλαδή επιθυμούν να έχουν τη δυνατότητα να γνωρίζουν την ακριβή θέση τους σε ένα εσωτερικό χώρο όπως μπορούν να βρουν τη θέση στην οποία βρίσκονται σε ένα εξωτερικό χώρο. Επιπλέον, είναι γνωστό ότι οι πλείστοι άνθρωποι διαθέτουν τη γνώση χρήσης έξυπνων συσκευών. Κατά συνέπεια φαίνεται ότι με την κατάλληλη χρήση έξυπνων συσκευών θα μπορέσει να επιλυθεί το πρόβλημα της εσωτερικής πλοήγησης με τη δημιουργία ενός συστήματος που θα εντοπίζει τη θέση του χρήστη μέσα σε ένα εσωτερικό χώρο. Ένα τέτοιο σύστημα μπορεί να είναι αρκετά χρήσιμο και σημαντικό σε περιπτώσεις εύρεσης της θέσης του χρήστη σε μεγάλα κτήρια όπως εμπορικά κέντρα, ξενοδοχειακά συγκροτήματα αλλά ακόμη και σε πλοία.

Μέσα στο πλαίσιο αυτό έχουν ήδη υλοποιηθεί αρκετά συστήματα εσωτερικής τοποθέτησης (Indoor positioning system -IPS) όπως για παράδειγμα το AnyPlace ή το MazeMap. Εδώ πρέπει να σημειωθεί ότι τα συστήματα, που έχουν ήδη υλοποιηθεί, απαιτούν κάποια υλική υποστήριξη (hardware), όπως για παράδειγμα beacons, το MazeMap και σήματα διαδικτύου Wi-Fi το AnyPlace. Υπάρχουν όμως χώροι, όπως για παράδειγμα τα μεγάλα μεταφορικά πλοία, τα οποία δεν έχουν τη δυνατότητα εγκατάστασης νέου υλικού για τη γεωτοποθέτησή τους μέσα στο πλοίο, με αποτέλεσμα

το προαναφερθέν πρόβλημα να μην έχει ακόμη επιλυθεί. Το ζητούμενο δηλαδή παραμένει η γεωτοποθέτηση του χρήστη σε ένα εσωτερικό χώρο χωρίς όμως τη χρήση κάποιου υλικού εξοπλισμού (hardware), ο οποίος χρειάζεται και παροχή ενέργειας. Η ανάγκη αυτή αποτέλεσε το κίνητρο για την υλοποίηση της εφαρμογής της παρούσας διπλωματικής. Έτσι αξιοποιώντας την κάμερα της έξυπνης συσκευής και με βοήθεια της τεχνολογίας Μηχανικής Όρασης, υλοποιήσαμε μια νέα εφαρμογή, η οποία έχει τη δυνατότητα εύρεσης της θέσης του χρήστη μέσα σε ένα εσωτερικό χώρο, χρησιμοποιώντας μόνο την κάμερα της συσκευής, όπως φαίνεται στο σχήμα 1.1.



Σχήμα 1.1 Παράδειγμα εκτέλεσης του υλοποιημένου συστήματος

Η μέθοδος που θα παρουσιαστεί αναλυτικότερα στη συνέχεια, αποτελεί μια πρωτότυπη προσπάθεια σε διεθνές ίσως επίπεδο αφού σε μια προκαταρκτική έρευνα που κάναμε στο διαδίκτυο δεν εντοπίσαμε κάτι παράλληλο. Παρόμοια συστήματα μπορεί να θεωρηθούν αυτά που χρησιμοποιούν παθητικά συστήματα RFID αλλά αυτά έχουν άλλους περιορισμούς όπως η απόσταση από τον ανιχνευτή.

1.2 Περίγραμμα Εργασίας

Στο πρώτο κεφάλαιο της παρούσας διπλωματικής παρουσιάζεται η υποκίνηση της εργασίας (το πρόβλημα δηλαδή που προσπαθεί να επιλυθεί στην εργασία αυτή), μια μικρή ανασκόπηση του συστήματος αλλά και η συνεισφορά της εργασίας στον τομέα των συστημάτων εσωτερικής γεωπλοήγησης.

Στο δεύτερο κεφάλαιο γίνεται επεξήγηση των κυριότερων όρων που χρησιμοποιούνται εκτενώς σε ακόλουθα κεφάλαια. Γίνεται επίσης κατάταξη του υπό ανάπτυξη συστήματος σε αντίστοιχες κατηγορίες παρόμοιων συστημάτων ανάλογα με συγκεκριμένα κριτήρια που παρουσιάζονται στη σχετική βιβλιογραφία [1]. Στο ίδιο κεφάλαιο γίνεται λεπτομερής περιγραφή του συστήματος YOLO, στο οποίο στηρίχθηκε η υλοποίηση της συγκεκριμένης διπλωματικής και σύντομη περιγραφή άλλων παρόμοιων συστημάτων γεωπλοήγησης σε εσωτερικούς χώρους.

Στο τρίτο κεφάλαιο γίνεται μια σύντομη παρουσίαση της αρχιτεκτονικής του συστήματος, δηλαδή αναφέρεται ο τρόπος με τον οποίο πραγματοποιείται η επικοινωνία της εφαρμογής με τη βάση δεδομένων. Στο κεφάλαιο αυτό περιγράφεται η ροή των δεδομένων από την εφαρμογή στη βάση, αναφέρεται ο τρόπος εισαγωγής των δεδομένων και ο τρόπος εμφάνισης των αποτελεσμάτων στον χρήστη στο τέλος της διαδικασίας.

Στο τέταρτο κεφάλαιο γίνεται η περιγραφή της υλοποίησης του Logger. Συγκεκριμένα, παρουσιάζεται ο τρόπος που εκτελείται το YOLO μέσα στην εφαρμογή, δηλαδή πώς γίνεται η αναγνώριση των αντικειμένων. Γίνεται επίσης η περιγραφή του γραφικού περιβάλλοντος της εφαρμογής.

Στο πέμπτο κεφάλαιο περιγράφεται η υλοποίηση του Navigator. Σε αυτό το κεφάλαιο γίνεται η λεπτομερής παρουσίαση της υλοποίησης του Navigator, καθώς και του αλγορίθμου ανάκτησης της ακριβούς θέσης του χρήστη από τη βάση δεδομένων.

Μετά δηλαδή την εισαγωγή των fingerprints στη βάση δεδομένων από τον Logger, ο Navigator, ανάλογα με τις πληροφορίες που συλλέγει από την κάμερα του χρήστη, ανακτά από τη βάση δεδομένων το πιο παρόμοιο fingerprint, του οποίου εμφανίζει τις συντεταγμένες στον χρήστη για σκοπούς γεωτοποθέτησής του.

Σημειώνεται ότι το σύστημα της παρούσας διπλωματικής είναι “participatory” που σημαίνει ότι οποιοσδήποτε χρήστης με εγκατεστημένη την εφαρμογή στο κινητό και γνώση του κτηρίου μπορεί να αποθηκεύσει δεδομένα (fingerprint) στη βάση δεδομένων.

Επίσης τα κτήρια που έχουν ειδικά χαρακτηριστικά σε κάθε θέση δεν θα χρειαστούν οποιαδήποτε επιπλέον υποδομή αλλά μόνο την εκπαίδευση του YOLO. Σε κτήρια που οπτικά είναι τα ίδια με ομοιόμορφα χαρακτηριστικά σε πολλές θέσεις θα χρειαστεί να

προστεθούν σε επιλεγμένες θέσεις διαφορετικές φωτογραφίες ή αντικείμενα για να μπορέσει το σύστημα αυτό να λειτουργήσει με αποτελεσματικότητα.

Στο έκτο κεφάλαιο παρουσιάζεται η πειραματική διαδικασία που ακολουθήσαμε για τη διαπίστωση της ορθότητας του συστήματος χωρίς τη χρήση του συστήματος των συντεταγμένων GPS. Επίσης, περιγράφεται η μέθοδος αξιολόγησης που χρησιμοποιήσαμε και ακολουθεί η αναλυτική περιγραφή του τρόπου εκτέλεσης των πειραμάτων. Στο κεφάλαιο αυτό, γίνεται ανάλυση των αποτελεσμάτων που προέκυψαν από την εκτέλεση των πειραμάτων.

Στο έβδομο κεφάλαιο αναλύονται τα συμπεράσματα που προέκυψαν από την υλοποίηση του συστήματος και επίσης παρουσιάζονται κάποια προβλήματα που εντοπίστηκαν στην υλοποίηση. Τέλος αναφέρονται μερικές πιθανές λύσεις ως μελλοντικές επεκτάσεις του συστήματος.

1.3 Συνεισφορές

Το σύστημα που υλοποιήθηκε στη συγκεκριμένη διπλωματική αποτελεί μια καινοτόμα εφαρμογή. Με τη βοήθεια δηλαδή της εφαρμογής αυτής μπορεί να εντοπιστεί η θέση ενός χρήστη χωρίς την αξιοποίηση υφιστάμενων υλικών υποδομών όπως Wi-Fi κλπ.

Σημειώνεται ότι σύστημα αυτό, παρόλο ότι αναπτύχθηκε αρκετά και έφτασε σε ένα καλό σημείο, μπορεί να τύχει περισσότερης βελτίωσης, όπως θα επεξηγηθεί στα επόμενα κεφάλαια. Μπορεί δηλαδή το σύστημα αυτό να αξιοποιηθεί από άλλους ερευνητές οι οποίοι θα το αναπτύξουν περαιτέρω δημιουργώντας ένα σύστημα εσωτερικής πλοήγησης με χρήση Μηχανικής Όρασης και όχι κάποιων υλικών υποδομών.

Τέλος, σημειώνεται ότι έχει γίνει πειραματική εκτέλεση της εφαρμογής για εξέταση της ορθότητάς της τόσο με χρήση αντικειμένων όσο και με τη χρήση φωτογραφιών με αρκετή επιτυχία όπως περιγράφεται στο κεφάλαιο 6. Είναι δηλαδή φανερό ότι ακόμη και στην παρούσα μορφή η εφαρμογή αυτή μπορεί να αξιοποιηθεί με τη χρήση ενός περιορισμένου αριθμού αντικειμένων για αναγνώριση και εύρεση της θέσης ενός χρήστη.

Κεφάλαιο 2

Υφιστάμενη Γνώση και Σχετικές Εργασίες

2.1	Κυριότεροι Όροι της Διπλωματικής	5
2.2	Κατηγοριοποίηση Συστημάτων Εσωτερικής Πλοήγησης	7
2.3	Σύστημα YOLO	10
	2.3.1 Τρόπος Λειτουργίας YOLO	11
	2.3.2 Εκπαίδευση Συστήματος YOLO	13
2.4	Άλλα Συστήματα Πλοήγησης Εσωτερικού Χώρου	15
	2.4.1 AnyPlace	15
	2.4.2 MazeMap	18

Το συγκεκριμένο κεφάλαιο έχει ως στόχο την επεξήγηση των κυριότερων όρων, οι οποίοι χρησιμοποιούνται εκτενώς στην παρούσα πτυχιακή εργασία. Γίνεται επίσης προσπάθεια ταξινόμησης του υλοποιημένου συστήματος της διπλωματικής σε ομάδες (κατηγορίες) σύμφωνα με το άρθρο των Zeinalipour-Yazti, et al. 2017 [1]. Στη συνέχεια ακολουθεί η περιγραφή του συστήματος YOLO, στο οποίο στηρίχθηκε η υλοποίηση του συστήματος της πτυχιακής. Τέλος, επιλέγεται να γίνει περιγραφή δύο συστημάτων εσωτερικής πλοήγησης το AnyPlace και το MazeMap, τα οποία ακολουθούν δύο εντελώς διαφορετικές μεθοδολογίες αλλά χρησιμοποιούν αρκετά γνωστές τεχνολογίες για τη γεωτοποθέτηση ή τη γεωπλοήγηση.

2.1 Κυριότεροι Όροι της Διπλωματικής

Η παρούσα διπλωματική εργασία υλοποιεί ένα καινοτόμο σύστημα γεωτοποθέτησης ενός χρήστη μέσα σε έναν εσωτερικό χώρο, χρησιμοποιώντας την τεχνολογία του Μηχανικής Όρασης (computer vision). Αφού αναφερόμαστε σε σύστημα πλοήγησης εσωτερικού

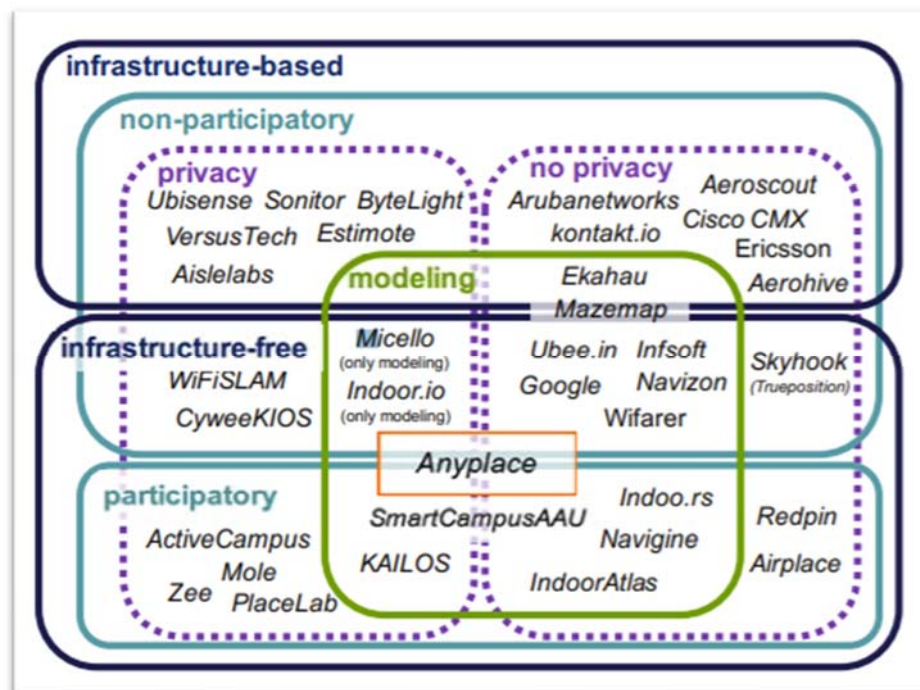
χώρου τότε αναπόφευκτα χρησιμοποιούμε τους όρους fingerprint και localization accuracy.

Συγκεκριμένα όταν χρησιμοποιούμε τον όρο fingerprint εννοούμε ένα «αντικείμενο» το οποίο αποτελεί την ταυτότητα (αποτύπωμα) ενός συγκεκριμένου σημείου σε ένα χώρο. Τα συστήματα πλοήγησης εξωτερικού χώρου χρησιμοποιούν τις συντεταγμένες GPS. Στη δική μας περίπτωση, όταν αναφερόμαστε σε κάποιο εσωτερικό χώρο εννοούμε μια εισαγωγή σε βάση δεδομένων, η οποία περιλαμβάνει κάποιες συντεταγμένες (χ , ψ). Οι συντεταγμένες αυτές μπορεί είτε να σχετίζονται με κάποιο σημείο του Wi-Fi είτε να αναφέρονται σε συντεταγμένες GPS. Στην περίπτωση του συστήματος της παρούσας διπλωματικής ένα fingerprint περιλαμβάνει τις συντεταγμένες χ , ψ του GPS και τον αριθμό των αντικειμένων που εντοπίστηκαν μέσω της κάμερας του χρήστη κατά τη δημιουργία του fingerprint. Διευκρινίζεται ότι ένα fingerprint δημιουργείται από το υποσύστημα του Logger. Ένας χρήστης δηλαδή χρησιμοποιώντας την υπό υλοποίηση εφαρμογή κινείται σε κάποιο χώρο και δημιουργεί τα διαφορά fingerprints. Στη συνέχεια, ένας άλλος χρήστης ο οποίος δεν γνωρίζει τι προηγήθηκε και δεν είναι ειδικός στο συγκεκριμένο τομέα μπορεί να χρησιμοποιεί την εφαρμογή με mode τον Navigator και να περιηγηθεί μέσα σε έναν εσωτερικό χώρο. Ο Navigator στη συνέχεια εκτελεί τον υλοποιημένο αλγόριθμο γεωτοποθέτησης. Ο αλγόριθμος κάνει αναζήτηση στη συλλογή των fingerprints. Μετά το τέλος της αναζήτησης στη συλλογή των fingerprints, το σύστημα παρέχει στον χρήστη τις συντεταγμένες του fingerprint που ταιριάζει περισσότερο.

Ένας άλλος όρος που χρησιμοποιείται συχνά, όταν αναφερόμαστε σε συστήματα γεωτοποθέτησης, είναι το localization accuracy, δηλαδή το ποσοστό επιτυχίας του συστήματος για την εύρεση της ακριβούς θέσης στην οποία βρίσκεται ο τελικός χρήστης. Τα συστήματα τα οποία προσπαθούν να βρουν τη θέση του χρήστη σε εσωτερικό χώρο δεν έχουν κάποιο ακριβή τρόπο για να βρουν τη θέση αυτή, και έτσι χρησιμοποιούνται διάφοροι αλγόριθμοι με αποτέλεσμα να υπάρχει η πιθανότητα λάθους. Ο όρος localization accuracy δηλώνει πόσες φορές το σύστημα βρήκε τη σωστή θέση του χρήστη με κάποια τυπική απόκλιση, δηλαδή μέσα σε κάποια όρια. Αυτό συμβαίνει γιατί σε αρκετές περιπτώσεις ο τελείως ακριβής ορισμός της θέσης του χρήστη χωρίς σήμα GPS είναι αρκετά δύσκολο να επιτευχθεί.

2.2 Κατηγοριοποίηση Συστημάτων Εσωτερικής Πλοήγησης

Κάθε σύστημα πλοήγησης διαφέρει ως προς τα χαρακτηριστικά του από άλλα συστήματα με αποτέλεσμα να γίνονται διάφορες κατατάξεις των συστημάτων με βάση διάφορα κριτήρια (Σχήμα 2.1). Σύμφωνα με το άρθρο των Zeinalipour-Yazti, et al. 2017 [1] με θέμα Internet-based navigation services τα διάφορα συστήματα πλοήγησης εσωτερικού χώρου μπορούν να χωριστούν σε διαφορετικές κατηγορίες, α) ανάλογα με τον τρόπο που υλοποιείται το κάθε σύστημα, β) ανάλογα με τον τρόπο που δημιουργούνται τα fingerprints, γ) ανάλογα με τον τρόπο που γίνεται η οργάνωση των δεδομένων και τέλος δ) ανάλογα με το αν ο χρήστης διατηρεί την ιδιωτικότητά του κατά τη χρήση του συγκεκριμένου συστήματος. Οι διάφορες κατηγορίες των συστημάτων πλοήγησης σε εσωτερικό χώρο περιγράφονται στη συνέχεια (Σχήμα 2.1).



Σχήμα 2.1: Διάγραμμα Venn κατηγοριοποίησης γνωστών συστημάτων πλοήγησης [1]

Infrastructure-based/Infrastructure-free

Η βασική τεχνολογία για τα συγκεκριμένα συστήματα είναι ο τύπος του hardware που επιτρέπει τη διαδικασία εντοπισμού. Πιο αναλυτικά, τα συστήματα χωρίζονται σε δύο τουλάχιστον διαφορετικές κατηγορίες ανάλογα με το αν χρησιμοποιούν ή όχι κάποιο infrastructure για τη γεωτοποθέτηση του χρήστη. Συγκεκριμένα χωρίζονται σε: α) infrastructure- based και β) infrastructure- free. Η κατηγοριοποίηση αυτή σχετίζεται με τη γενική ιδέα της διαδικασίας γεωτοποθέτησης του χρήστη, αν δηλαδή η εφαρμογή θα χρησιμοποιήσει κάποιο hardware όπως beacons [2], antennas κλπ. ή θα χρησιμοποιήσει κάποια υπηρεσία η οποία δεν απαιτεί hardware [3, 4, 5, 6]. Το σύστημα που υλοποιήθηκε στην παρούσα διπλωματική ανήκει στην κατηγορία των infrastructure-free συστημάτων αφού για τη γεωτοποθέτηση του χρήστη κατά το logging των fingerprints χρησιμοποιείται η υπηρεσία GPS σε συνδυασμό με τη χρήση της τεχνολογίας μηχανικής όρασης και όχι beacons ή χρήση συρμάτων.

Participatory/Non-participatory

Ένα άλλο κριτήριο ταξινόμησης των συστημάτων εσωτερικής πλοήγησης βασίζεται στο αν η βάση δεδομένων γεωγραφικής θέσης έχει συμπληρωθεί από ειδικούς ή μη ειδικούς, δηλαδή με συμμετοχικό τρόπο. Συγκεκριμένα, το συγκεκριμένο κριτήριο αφορά τον τρόπο οργάνωσης της βάσης δεδομένων που χρησιμοποιείται στην οργάνωση των fingerprints. Η κατηγοριοποίηση αυτή αναφέρεται στο ποιος δημιουργεί τα διάφορα fingerprints για την κάθε θέση, αν δηλαδή είναι ένας ειδικός που γνωρίζει τον τρόπο που δουλεύει το συγκεκριμένο πρόγραμμα ή είναι ένας εθελοντής. Στη δεύτερη περίπτωση πιθανόν να μην εισάγει μεγάλο αριθμό δεδομένων. Συγκεκριμένα οι δύο κατηγορίες είναι: α) τα non-participatory και β) τα participatory συστήματα [1]. Τα non-participatory συστήματα είναι αυτά των οποίων η βάση δεδομένων είναι οργανωμένη από ειδικούς [7, 8]. Κάποιος δηλαδή ειδικός αναλαμβάνει τη διαδικασία να δημιουργήσει αρκετά fingerprints σε διάφορες θέσεις, δηλαδή ένα μεγάλο πλήθος fingerprints. Αντίθετα τα participatory συστήματα είναι αυτά που η εισαγωγή δεδομένων στη βάση γίνονται από πολλούς εθελοντές που εισάγουν λίγα fingerprints ο καθένας [9, 10, 11]. Το σύστημα που υλοποιήθηκε στην παρούσα διπλωματική ανήκει στα participatory συστήματα. Λόγω της

πίεσης χρόνου, τα fingerprints δημιουργήθηκαν από φοιτητές σε περιορισμένο αριθμό θέσεων, μόνο στο Πανεπιστήμιο Κύπρου.

Terminal-based/Network-based

Ένα βασικό μειονέκτημα των περισσότερων συστημάτων γεωτοποθέτησης είναι το γεγονός ότι βασίζονται στο διαδίκτυο, και έτσι η υπηρεσία μπορεί συνεχώς να γνωρίζει τη θέση ενός χρήστη ενώ τον εξυπηρετεί. Αντίθετα, η μεταβίβαση ολόκληρων των βάσεων δεδομένων γεωγραφικής θέσης στη συσκευή του χρήστη, για να αποφεύγεται η παραβίαση του απορρήτου, σημαίνει συνεχώς μεταφορά τεράστιου αριθμού δεδομένων μέσω ασυρμάτων συνδέσεων περιορισμένων πόρων. Για αυτό το λόγο, απαιτείται η κατηγοριοποίηση των συστημάτων ανάλογα με την προσέγγιση που χρησιμοποιεί το κάθε σύστημα. Τα συστήματα πλοήγησης εσωτερικών χωρών διαχωρίζονται δηλαδή σε κατηγορίες ανάλογα με την ύπαρξη ή όχι ιδιωτικότητα (privacy) του χρήστη στο σύστημα [1].

Η ιδιωτικότητα ενός συστήματος είναι αρκετά σημαντική για τους χρήστες του, αφού λόγω του ότι αποτελεί σύστημα γεωτοποθέτησης έχει τη δυνατότητα να γνωρίζει πού ακριβώς βρίσκονται οι χρήστες, κάτι το οποίο ανησυχεί ιδιαίτερα τους τελικούς χρήστες αφού παραβιάζει την ιδιωτικότητά τους. Αυτό αποτελεί ένα αρκετά μεγάλο πρόβλημα για τους χρήστες. Οι δύο κατηγορίες σχετικά με την ιδιωτικότητα είναι α) terminal-based και β) network-based. Τα terminal-based συστήματα [12] υπολογίζουν τη θέση του χρήστη πάνω στην προσωπική συσκευή του η οποία γνωρίζει τον αλγόριθμο γεωτοποθέτησης με αποτέλεσμα η εφαρμογή να μπορεί να τρέξει και σε offline mode. Η δεύτερη κατηγορία, η network-based, δεν προσφέρει καθόλου ιδιωτικότητα λόγω του ότι οι πληροφορίες που συλλέγονται από την προσωπική συσκευή του χρήστη στέλνονται διαδικτυακά σε κάποιο infrastructure [13, 14]. Εκεί γίνεται η επεξεργασία των δεδομένων του χρήστη και αποστέλλονται πίσω στη συσκευή του χρήστη. Στη συσκευή θα παρουσιαστεί η ακριβής θέση του χρήστη, δηλαδή θα γίνει η γεωτοποθέτησή του.

Το σύστημα που υλοποιήθηκε στην παρούσα διπλωματική είναι terminal-based αφού η βάση του συστήματος βρίσκεται στο κινητό του χρήστη, αλλά και ο αλγόριθμος για τη γεωτοποθέτηση βρίσκεται και πάλι στην προσωπική του συσκευή με αποτέλεσμα να μην εμπλέκεται κάποιο δίκτυο και να επηρεάζεται η ιδιωτικότητα του χρήστη.

2.3 Σύστημα YOLO

Η εφαρμογή, που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής, όπως προαναφέρθηκε, έχει ως κύριο σκοπό τη δημιουργία μιας εφαρμογής για γεωτοποθέτηση ενός χρήστη μέσα σε ένα κτήριο χωρίς τη χρήση υπηρεσιών GPS, Wi-Fi ή cellular. Για τη δημιουργία αυτής της εφαρμογής χρησιμοποιήθηκε η τεχνολογία μηχανικής όρασης, δηλαδή ο χρήστης μέσω της κάμεράς του βιντεογραφεί το χώρο που τον περιβάλλει και στη συνέχεια τα διάφορα αντικείμενα που έχουν βιντεογραφεί αναγνωρίζονται μέσω ενός συστήματος. Με αυτόν τον τρόπο το σύστημα μπορεί να εντοπίσει την ακριβή θέση του χρήστη.

Η συγκεκριμένη υλοποίηση απαιτούσε την αναγνώριση αντικειμένων σε πραγματικό χώρο για αυτό χρησιμοποιήθηκε το ήδη υλοποιημένο open source σύστημα του YOLO (=You only look once) και συγκεκριμένα το tiny version 3. Σημειώνεται ότι χρησιμοποιήθηκε το tiny έναντι του κανονικού μεγέθους YOLO λόγω του ότι η εφαρμογή είναι για προσωπική έξυπνη συσκευή όπως κινητό και το YOLO έχει αρκετά χαμηλή απόδοση. Απαιτεί, δηλαδή αρκετό χρόνο για αναγνώριση των αντικειμένων ενώ το tiny YOLO απαιτεί αρκετά λιγότερο αποθηκευτικό χώρο και επίσης το tiny-YOLO είναι πολύ γρηγορότερο αφού το tiny-yolo έχει μπορεί να αναγνωρίσει μέχρι και 244 frames το δευτερόλεπτο ενώ το κανονικό YOLO αναγνωρίζει μόλις 40 frames ανά δευτερόλεπτο κάτι που είναι αρκετά χρήσιμο όταν η υλοποίηση προϋποθέτει την αναγνώριση αντικειμένων σε πραγματικό χρόνο.

Μερικά παραδείγματα από αντικείμενα που αναγνωρίζει το σύστημα YOLO είναι έπιπλα όπως καρέκλες και τραπέζια, μέσα συγκοινωνίας όπως αυτοκίνητα και λεωφορεία, αστικός εξοπλισμός όπως φώτα τροχαίας κ.λπ. Το YOLO, λόγω του ότι είναι και αυτό ένα υπολογιστικό σύστημα, έπρεπε να μπορεί T) που προσδιορίζει το COCO, οι οποίες παρουσιάζονται πιο αναλυτικά στο Παράρτημα Α (Πίνακας Α1). Παρ' όλα αυτά με τη χρήση των κατάλληλων εργαλείων το YOLO μπορεί να εκπαιδευτεί και να αναγνωρίζει πολύ περισσότερα αντικείμενα από αυτά που αναγνωρίζει ως default, αλλά αυτή η διαδικασία προϋποθέτει αρκετά μεγάλη υπολογιστική δύναμη.

2.3.1 Τρόπος Λειτουργίας YOLO

Σύμφωνα με την υφιστάμενη βιβλιογραφία [15, 16, 17, 18] το YOLO αναγνωρίζει τα διαφορετικά αντικείμενα ακολουθώντας γενικά την εξής διαδικασία. Αρχικά, ως είσοδο λαμβάνει μια εικόνα ή ένα frame από κάποιο βίντεο, το επεξεργάζεται μέσω μιας διαδικασίας που περιγράφεται πιο κάτω, και στη συνέχεια επιστρέφει την ίδια εικόνα δημιουργώντας ένα ορθογώνιο με λεξάντα για κάθε ένα από τα αντικείμενα που αναγνώρισε.

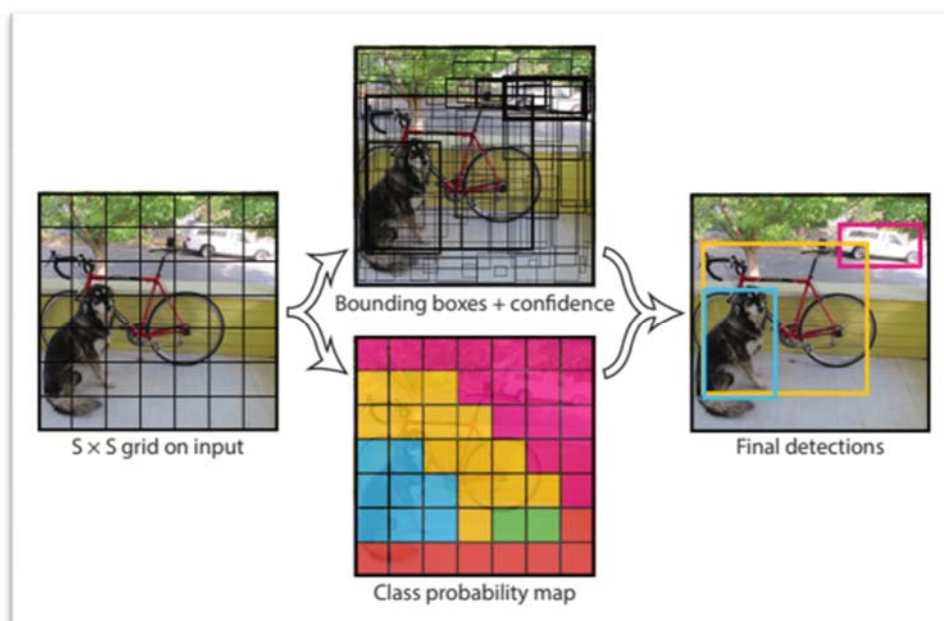
Το μοντέλο του YOLO λειτουργεί αρκετά γρήγορα λόγω του ότι εφαρμόζει το νευρωνικό δίκτυο ταυτόχρονα σε όλη την εικόνα. Άλλα παρόμοια συστήματα χωρίζουν την εικόνα σε διαφορετικές περιοχές και εφαρμόζουν το δίκτυο σε κάθε περιοχή ξεχωριστά με αποτέλεσμα να δαπανούν αρκετό χρόνο, πολύ περισσότερο από ότι απαιτείται από το YOLO [15]. Παρόλα αυτά, κατά την υλοποίηση του YOLO δόθηκε μεγαλύτερη σημασία στην επίτευξη μέγιστης δυνατής ταχύτητας παρά ακρίβεια στην αναγνώριση με αποτέλεσμα το YOLO να μην θεωρείται κατάλληλο για χρήση σε εξειδικευμένα μοντέλα στα οποία είναι δύσκολο να ληφθούν πολλά δεδομένα, δηλαδή μεγάλος αριθμός εικόνων των συγκεκριμένων αντικειμένων [16].

Ο τρόπος λειτουργίας του συστήματος YOLO στηρίζεται στη χρήση νευρωνικών δικτύων. Το νευρωνικό δίκτυο το οποίο χρησιμοποιείται από το YOLO για την αναγνώριση των διαφόρων αντικειμένων είναι το darknet. Το darknet είναι ένα open-source custom neural-network framework υλοποιημένο στην C και στην CUDA που επιτρέπει διάφορους υπολογισμούς στην CPU αλλά και στην GPU. Συγκεκριμένα για την αναγνώριση των διαφόρων αντικειμένων στο YOLO με τη χρήση CPU απαιτούνται 6-12 δευτερόλεπτα για κάθε εικόνα ενώ με GPU απαιτείται πολύ λιγότερος χρόνος [16].

Τα δίκτυα αυτά μπορούν να επεξεργαστούν εικόνες σαν οργανωμένους πίνακες με δεδομένα και μπορούν να αναγνωρίσουν μοτίβα με αυτά τα δεδομένα. Η αναγνώριση των διαφορετικών αντικειμένων πραγματοποιείται με την αναγνώριση μοτίβων, τα οποία ομοιάζουν με τα μοτίβα των προ-προσδιορισμένων (predefined) κλάσεων που ήδη γνωρίζει το σύστημα YOLO. Στη συνέχεια ανάλογα με την κλάση με την οποία τα μοτίβα έχουν τις περισσότερες ομοιότητες, το σύστημα θεωρεί ότι αυτή είναι η πιθανότερη ταύτιση. Η αναγνώριση των ομοιοτήτων γίνεται με ποσοστά. Αν δηλαδή η εικόνα περιλαμβάνει ένα αντικείμενο χ , και αφού το YOLO γνωρίζει ήδη 80 κλάσεις, τότε θα συγκρίνει το αντικείμενο χ με τα άλλα 80 αντικείμενα που ήδη γνωρίζει. Ανάλογα με το

αντικείμενο με το οποίο έχει το μεγαλύτερο ποσοστό ομοιότητας, θα δημιουργήσει ένα ορθογώνιο γύρω του με το όνομα της κλάσης και το ποσοστό ομοιότητας που υπολόγισε.

Γενικά, το YOLO λαμβάνει ως είσοδο μια εικόνα ή ένα frame από κάποιο βίντεο. Μετά την είσοδο της εικόνας στο σύστημα και την εφαρμογή σε αυτήν του νευρωνικού δικτύου γίνεται διαχωρισμός της από το δίκτυο σε πολλές περιοχές (Σχήμα 2.2). Στη συνέχεια γίνεται προσπάθεια από το σύστημα για κάθε περιοχή να προβλέψει ένα κουτί οριοθέτησης και την πιθανότητα να ταυτίζεται με κάποιο αντικείμενο, σύμφωνα με τη διαδικασία που ήδη αναφέραμε πιο πάνω. Στη συνέχεια, τα πλαίσια οριοθέτησης μαζί με την πιθανότητα που έχει υπολογιστεί σταθμίζονται από τις προβλεπόμενες πιθανότητες, οι οποίες είναι καθορισμένες μέσω του δικτύου. Από τις πιθανότητες αυτές, το σύστημα επιλέγει τις πιο μεγάλες τιμές και τις θεωρεί ως προβλέψεις. Τις τιμές αυτές θα εμφανίσει πάνω στην εικόνα, που δόθηκε για κάθε περιοχή, και θα τις παρουσιάζει στον χρήστη. Το δίκτυο που χρησιμοποιείται από το σύστημα είναι το darknet, το οποίο είναι ήδη εκπαιδευμένο μέσω του COCO Dataset.



Σχήμα 2.2: Διαδικασία υποδιαίρεσης εικόνας σε περιοχές και εντοπισμός γνωστών αντικειμένων [15]

2.3.2 Εκπαίδευση Συστήματος YOLO

Το YOLO είναι ένα σύστημα το οποίο είναι ήδη εκπαιδευμένο με τη χρήση των convolutional layers του ImageNet Dataset [15]. Η πλήρης εκπαίδευση του YOLO απαιτεί κατά μέσο όρο μία εβδομάδα και μπορεί να αποκτήσει ακρίβεια μέχρι και 88%. Για επίτευξη όμως αυτού του ποσοστού, όπως αναφέρεται πιο πάνω, απαιτείται μεγάλη υπολογιστική δύναμη.

Η επίτευξη μιας καλής εκπαίδευσης για ένα νέο αντικείμενο απαιτεί την εισαγωγή εικόνων με σήμανση των νέων αντικειμένων που αναμένεται πως θα μπορεί να αναγνωρίσει. Δηλαδή για να εκπαιδύσουμε το YOLO να αναγνωρίζει ένα αντικείμενο χ , τότε πρέπει να εισάγουμε για παράδειγμα 100 εικόνες του συγκεκριμένου αντικειμένου με 100 επιπλέον αρχεία κειμένου τα οποία προσδιορίζουν πού βρίσκεται το αντικείμενο χ σε κάθε εικόνα. Τα διάφορα αρχεία κειμένου τα οποία δίνονται μαζί με τις εικόνες ως είσοδο για την εκπαίδευση του συστήματος δημιουργούνται μηχανικά από τον χρήστη για κάθε εικόνα χρησιμοποιώντας διάφορα υλοποιημένα προγράμματα ανοιχτού κώδικα

Κατά τη δημιουργία αυτών των αρχείων κειμένου στη δική μας περίπτωση χρησιμοποιήσαμε το πρόγραμμα CVAT. Το CVAT είναι μια διαδικτυακή εφαρμογή ανοιχτού κώδικα από την Intel που έχει σκοπό τη σήμανση (annotation) εικόνων και βίντεο για χρήση σε αλγόριθμους υπολογιστικής όρασης. Το γραφικό περιβάλλον της εφαρμογής είναι βέλτιστο για εργασίες σήμανσης υπολογιστικής όρασης. Κατά την εκπαίδευση του YOLO ο χρήστης πολύ πιθανόν θα θέλει να το εκπαιδεύσει για πολλά αντικείμενα, αφού στην εκπαίδευση δεν μπορεί να προσθέσει εκ των υστέρων νέα αντικείμενα χωρίς να απαιτείται εκ νέου εκπαίδευση. Για αυτό συνήθως ο χρήστης επιλέγει να εισάγει πολλά αντικείμενα και αυτό σημαίνει πολλές εικόνες και πολλά αρχεία κειμένου που περιλαμβάνουν τις συντεταγμένες του ορθογωνίου στο οποίο βρίσκεται το κάθε αντικείμενο. Η εισαγωγή μεγάλου αριθμού αντικειμένων μπορεί να απαιτεί αρκετές ώρες μέχρι να δημιουργηθούν τα ανάλογα αρχεία κειμένου και να βρεθούν οι ανάλογες εικόνες. Γι' αυτό γίνεται χρήση του εργαλείου CVAT με το οποίο ο χρήστης μπορεί να εισάγει κάποιο βίντεο με αρκετά αντικείμενα.

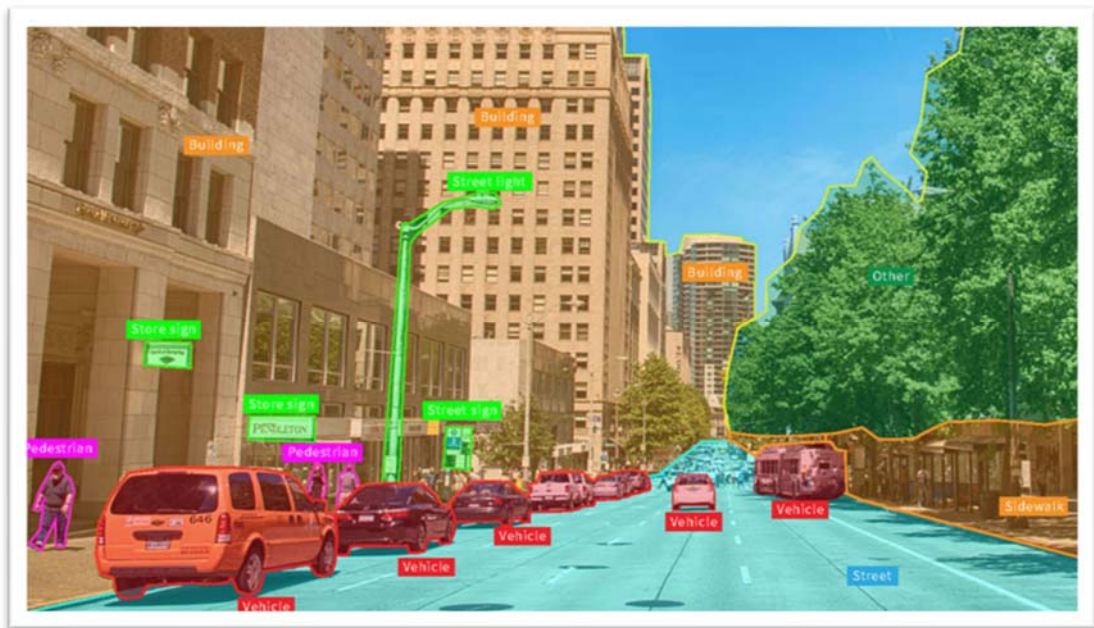
Το εργαλείο CVAT εμφανίζει στον χρήστη ένα frame στο οποίο επισημαίνει πού βρίσκεται το κάθε αντικείμενο. Λόγω της χρήσης βίντεο μπορεί ο χρήστης να αναφέρει πως οι επισημάνσεις είναι οι ίδιες για τα επόμενα χ frames και το πρόγραμμα CVAT θα

αφήσει τις επισημάνσεις ίδιες. Όταν ολοκληρωθεί η όλη διαδικασία το CVAT θα αποθηκεύσει στον υπολογιστή του χρήστη σαν εικόνα ξεχωριστό το κάθε frame του βίντεο που του δόθηκε αλλά και αρχείο κειμένου για την κάθε εικόνα, δηλαδή το κάθε frame από το βίντεο. Αυτό έχει ως αποτέλεσμα η όλη διαδικασία εισαγωγής επισημάνσεων σε εικόνες να γίνεται αρκετά γρήγορα και αποτελεσματικά αφού αντί ο χρήστης να πάρει το κάθε frame ξεχωριστά και να κάνει τις επισημάνσεις μπορεί να κάνει τις ίδιες επισημάνσεις για αρκετά frames του βίντεο με αποτέλεσμα η όλη διαδικασία να μπορεί να εκτελεστεί πολύ πιο γρήγορα.

Ένα επίσης πλεονέκτημα της συγκεκριμένης τεχνολογίας είναι ο τρόπος δημιουργίας πλαισίων γύρω από τα αντικείμενα που αναγνωρίζει. Συνήθως οι εφαρμογές που χρησιμοποιούνται για τη διεκπεραίωση τέτοιους είδους δραστηριοτήτων απλά δημιουργούν ένα ορθογώνιο γύρω από το αντικείμενο που θέλουν να εκπαιδεύσουν για αναγνώριση. Το ορθογώνιο αυτό σε αρκετές περιπτώσεις μπορεί να περιλαμβάνει και άλλα αντικείμενα με αποτέλεσμα να δυσκολεύει αρκετά την εκπαίδευση ή και να οδηγεί την εκπαίδευση σε πλήρη αποτυχία αναγνωρίζοντας λάθος τα αντικείμενα. Αντίθετα το CVAT δίνει τη δυνατότητα δημιουργίας πολυγώνων σχημάτων (Σχήμα 2.3) ακριβώς γύρω από το επιθυμητό αντικείμενο με αποτέλεσμα την πιο ορθή εκπαίδευση του συστήματος και στη συνέχεια την πιο σωστή αναγνώριση των αντικειμένων. Πρέπει όμως να αναφερθεί πως ανεξαρτήτως του τρόπου που χρησιμοποιείται για την επισημάνση των αντικειμένων κατά το στάδιο της εκπαίδευσης, στη συνέχεια κατά την αναγνώριση των αντικειμένων μέσω του συστήματος YOLO, θα δημιουργείται πάντοτε ένα ορθογώνιο γύρω από το αντικείμενο που θα αναγνωριστεί.

Η εκπαίδευση του YOLO με νέα συλλογή δεδομένων είναι αρκετά χρονοβόρα, αφού, όπως αναφέρθηκε πιο πάνω, πρέπει να δίδεται ένας ικανοποιητικός αριθμός από εικόνες και αρχεία κειμένου. Επιπλέον, απαραίτητη προϋπόθεση για τη σωστή εκπαίδευση ενός νευρωνικού δικτύου με τη χρήση του συστήματος YOLO είναι το υπολογιστικό σύστημα, στο οποίο θα γίνει η εκπαίδευση, να υποστηρίζει μια GPU. Αφού δηλαδή, αναφερόμαστε σε εικόνες ως είσοδο και έξοδο του συστήματος είναι απαραίτητο το σύστημα να μπορεί τουλάχιστον να διαβάσει τις εικόνες. Σε διαφορετική περίπτωση δεν θα μπορεί να εκπαιδευτεί το σύστημα, γιατί πολύ απλά δεν θα του είναι δυνατό να διαβάσει τις εικόνες για να τις μάθει και να μπορεί να δημιουργήσει ένα νευρωνικό δίκτυο με τα απαιτούμενα δεδομένα. Η εκπαίδευση του συστήματος γίνεται αφού

εισαχθούν οι διάφορες εικόνες και τα διάφορα αρχεία κειμένου και ολοκληρωθεί η διαδικασία δημιουργίας ενός νέου νευρωνικού δικτύου με τις νέες κλάσεις (classes).



Σχήμα 2.3: Σήμανση αντικειμένων με CVAT

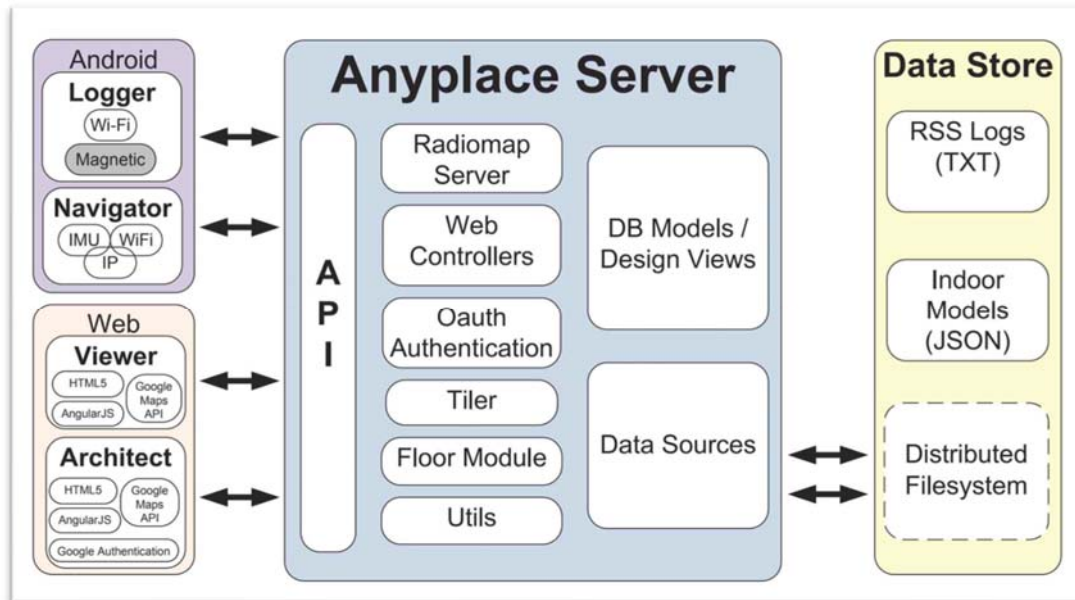
Εδώ θα πρέπει να σημειωθεί ότι λόγω του ότι εμπλέκονται νευρωνικά δίκτυα δεν είναι δυνατό να συνεχισθεί η εκπαίδευση του συστήματος μετά την ολοκλήρωσή της. Αν δηλαδή το σύστημα γνωρίζει 80 κλάσεις δεν μπορούμε να προσθέσουμε εικόνες για ακόμα ένα αντικείμενο και να αναμένουμε πως στο τέλος της εκπαίδευσης θα έχει τη δυνατότητα να αναγνωρίζει 81 αντικείμενα. Αντίθετα, αν προσθέσουμε εικόνες και σημάνσεις για ένα επιπλέον αντικείμενο, λόγω του ότι εμπλέκονται νευρωνικά δίκτυα, στο τέλος της εκπαίδευσης του συστήματος θα αναγνωρίζει μόνο το τελευταίο αντικείμενο για το οποίο έγινε η πιο πρόσφατη εκπαίδευση αφού κατά την εκπαίδευση αυτή θα δημιουργήσει εκ νέου ένα νευρωνικό δίκτυο [15].

2.4 Άλλα Συστήματα Πλοήγησης Εσωτερικού Χώρου

2.4.1 AnyPlace

Ένα παρόμοιο σύστημα γεωτοποθέτησης με αυτό της παρούσας διπλωματικής που αφορά εσωτερικούς χώρους είναι το AnyPlace. Το Anyplace όπως αναφέρουν οι Mreis et al. 2020 [19] εφαρμόζεται σε διάφορους τύπους εσωτερικών χώρων όπως πολυώροφα

κτήρια χωρίς τη χρήση της τεχνολογίας GPS. Το AnyPlace έχει υλοποιηθεί από ερευνητές και μαθητές του εργαστηρίου Data Management Systems Laboratory του Πανεπιστημίου Κύπρου. Είναι ένα open source σύστημα το οποίο μπορεί να εγκατασταθεί στο κινητό οποιουδήποτε χρήστη (Σχήμα 2.4).



Σχήμα 2.4: Αρχιτεκτονική συστήματος AnyPlace

Η πλατφόρμα του AnyPlace, σύμφωνα με τους μελετητές του συστήματος [19] αποτελείται από πέντε βασικά υποσυστήματα, συμπεριλαμβανομένου του Server, του Architect, του Viewer αλλά και μιας εφαρμογής η οποία τρέχει πάνω σε Android συσκευές και συμπεριφέρεται σαν Logger ή σαν Navigator. Τα σημαντικότερα υποσυστήματα του AnyPlace είναι το υποσύστημα του Navigator και του Logger [20] τα οποία περιγράφονται στη συνέχεια.

Τα υποσυστήματα του Navigator και του Logger χρησιμοποιούνται από τον τελικό χρήστη για την πλοήγησή του σε κάποιο χώρο από έναν εθελοντή, για να δημιουργήσει τα διάφορα fingerprints σε ένα συγκεκριμένο κτήριο/χώρο. Τα τρία άλλα συστήματα, δηλαδή ο server, ο architect και ο viewer χρησιμοποιούνται ως το backend του συστήματος και διαχειρίζονται τα βασικά δεδομένα για τη λειτουργία του συστήματος, όπως τα κτήρια και τα διάφορα fingerprints που δημιουργήθηκαν πριν από το Logger [19]. Το μόνο backend σύστημα, το οποίο χρησιμοποιείται από κάποιο χρήστη του συστήματος, είναι το Architect. Το σύστημα αυτό (Architect) χρησιμοποιείται για την

εισαγωγή των αρχιτεκτονικών σχεδίων κάποιου κτηρίου καθώς και μερικών πιθανών σημείων ενδιαφέροντος για το συγκεκριμένο κτήριο, όπως για παράδειγμα το πού βρίσκονται οι σκάλες ή οι έξοδοι κίνδυνου [21].

Η γενική ιδέα της αρχιτεκτονικής του συγκεκριμένου συστήματος, όπως περιγράφεται και από τους Konstantinides et al. 2018 [22], είναι η εξής: Αρχικά κάποιος εθελοντής χρησιμοποιεί την εφαρμογή ως Logger και «ηχογραφεί» τα Wi-Fi fingerprints τα οποία περιλαμβάνουν την ισχύ του σήματος Wi-Fi και συντεταγμένες x,y οι οποίες εξάγονται από την κάτοψη του κτηρίου. Το κάθε Wi-Fi fingerprint εισάγεται στον server μέσω του web API σε μορφή JSON [23].

Στη συνέχεια, ο τελικός χρήστης μέσω του Navigator μπορεί να χρησιμοποιήσει την εφαρμογή. Η εφαρμογή, ενώ μετακινείται στο κτήριο, τρέχει τους κατάλληλους αλγόριθμους στα fingerprints, τα οποία εισήγαγε πριν ο εθελοντής, που δρούσε σαν Logger. Έτσι στη συνέχεια ο Navigator μπορεί να εντοπίσει τη θέση κάποιου χρήστη αν του ζητηθεί, συγκρίνοντας το τρέχον fingerprint του χρήστη με τα fingerprint που δημιουργήθηκαν πριν από τον Logger και να εμφανίσει το κοντινότερο ως η τρέχουσα θέση του χρήστη πάνω στα αρχιτεκτονικά σχέδια του κτηρίου στο οποίο βρίσκεται [22]. Ο Navigator επιτρέπει στους χρήστες να βλέπουν με μεγάλη ακρίβεια την τρέχουσα θέση τους πάνω σε μία κάτοψη ενός ορόφου του συγκεκριμένου κτηρίου στο οποίο βρίσκονται. Επίσης, ο Navigator χρησιμοποιεί αισθητήρες του κινητού όπως η πυξίδα και το γυρόμετρο. Οι αισθητήρες αυτοί είναι ενσωματωμένοι στην υλοποίηση για να υποβοηθούν τα Wi-Fi σημεία.

Σημειώνεται επίσης ότι πριν δημιουργηθούν τα fingerprints για κάποιο κτήριο πρέπει πρώτα να γίνει η εισαγωγή στο σύστημα της κάτοψης του κτηρίου. Αυτό γίνεται με τη χρήση του υποσυστήματος Architect [19]. Το υποσύστημα αυτό είναι μια φιλική προς τον χρήστη εφαρμογή διαδικτύου (user friendly) με σκοπό τη διαχείριση των κτηρίων στο σύστημα. Η κύρια λειτουργία του συγκεκριμένου υποσυστήματος είναι η εισαγωγή των σχεδίων των κτηρίων. Αυτό γίνεται με το ανέβασμα των αρχιτεκτονικών σχεδίων κάθε κτηρίου πάνω από την αεροφωτογραφία του αντίστοιχου κτηρίου, όπως αυτή εμφανίζεται στο google maps. Στη συνέχεια, πάνω στα σχέδια, σημειώνονται πιθανές διαδρομές αλλά και τα διάφορα σημεία ενδιαφέροντος.

Με παρόμοιο τρόπο λειτουργεί και το υποσύστημα Viewer, το οποίο είναι επίσης μια εφαρμογή διαδικτύου. Η εφαρμογή αυτή σύμφωνα με τον Mpreis et al 2020 [19] χρησιμοποιείται ως μηχανή αναζήτησης των κτηρίων που βρίσκονται στο AnyPlace, αλλά και ως μηχανή πλοήγησης στα κτήρια που βρίσκονται ήδη στο AnyPlace. Η εφαρμογή και αυτή, όπως και η Architect, είναι διαδικτυακές εφαρμογές για την ευκολότερη τους χρήση.

Το υποσύστημα Server, σύμφωνα με τους Zeinalipour-Yazti et al. 2017 [24] είναι υλοποιημένο χρησιμοποιώντας MVC framework. Στο συγκεκριμένο υποσύστημα βρίσκονται αποθηκευμένα τα δεδομένα που προκύπτουν από τα υποσυστήματα Architect και Logger. Ο server περιλαμβάνει επίσης, το backend όλου του συστήματος, στο οποίο περιλαμβάνεται η μοντελοποίηση, το crowdsourcing αλλά και όλη η λειτουργικότητα του API. Ο server προσφέρει modules τα οποία υποβοηθά στο crowdsourcing, στη λειτουργικότητα, στο authentication του χρήστη αλλά και στην εμφάνιση των εικόνων.

2.4.2 MazeMap

Ένα άλλο παρόμοιο σύστημα με αυτό που υλοποιήθηκε στην παρούσα διπλωματική είναι το MazeMap [25, 26, 27]. Το MazeMap είναι ένα σύστημα εσωτερικής πλοήγησης που αναπτύχθηκε από το Norwegian University of Science and Technology (NTNU). Η κύρια λειτουργία του συγκεκριμένου συστήματος είναι η γεωτοποθέτηση του χρήστη μέσα σε έναν εσωτερικό χώρο και η πλοήγησή του μέσα σε αυτό. Συγκεκριμένα, εντοπίζοντας τη θέση στην οποία βρίσκεται, είναι δυνατό να εμφανίζεται και η διαδρομή από το σημείο στο οποίο βρίσκεται, στο σημείο που επιθυμεί να πάει. Η τεχνολογία που χρησιμοποιήθηκε για την υλοποίηση του συγκεκριμένου συστήματος είναι τα beacons. Ο κύριος σκοπός του MazeMap είναι η εξυπηρέτηση χρηστών, οι οποίοι προσπαθούν να περιηγηθούν σε κάποιο Πανεπιστήμιο, είναι δηλαδή κυρίως φοιτητές (Σχήμα 2.5). Το σύστημα αυτό, μπορεί επίσης να χρησιμοποιηθεί και σε άλλα δημόσια κτήρια όπως νοσοκομεία, εμπορικά κέντρα και χώρους εκδηλώσεων.



Σχήμα 2.5. Σύστημα MazeMap [26]

Το σύστημα MazeMap λειτουργεί ως εξής. Αρχικά εισάγονται στο σύστημα τα αρχιτεκτονικά σχέδια της περιοχής, που επιθυμεί ο ιδιοκτήτης του χώρου, μαζί με τα σημεία ενδιαφέροντος για το συγκεκριμένο σχέδιο. Στη συνέχεια, εισάγονται τα αρχιτεκτονικά σχέδια, είτε από τον ιδιοκτήτη χρησιμοποιώντας τη σχετική πλατφόρμα, είτε από κάποιο διαχειριστή του συστήματος. Όταν εισαχθεί ο χάρτης, τότε ο χρήστης μπορεί εύκολα να αναζητήσει κάποιο από τα σημεία ενδιαφέροντος στον χάρτη αλλά και να εντοπίσει την τρέχουσά του θέση πάνω στο αρχιτεκτονικό σχέδιο του κτηρίου [25].

Η τεχνολογία, η οποία χρησιμοποιείται για τη λειτουργία του συστήματος, είναι αυτή του Wi-Fi, των beacons και των γεωμαγνητικών πεδίων. Για κάθε κτήριο το σύστημα είναι διαφορετικό και για αυτό αξιολογείται η κάθε περίπτωση διαφορετικά και, ανάλογα με τα πλεονεκτήματα και τα μειονεκτήματα, χρησιμοποιείται η κατάλληλη τεχνολογία.

Μια επιπλέον λειτουργία του συστήματος MazeMap είναι η δημιουργία ενός χάρτη για το πού βρίσκεται ο κάθε χρήστης στο κτήριο, μέσω των διαφορετικών σημείων πρόσβασης του Wi-Fi. Αυτό έχει ως αποτέλεσμα να μπορεί να παρατηρηθεί πού βρίσκονται πολλά και πού λίγα άτομα μέσα σε ένα συγκεκριμένο κτήριο. Τέλος, σημειώνεται ότι το σύστημα MazeMap μπορεί να ενσωματωθεί και σε άλλα συστήματα [25].

Κεφάλαιο 3

Αρχιτεκτονική Συστήματος

3.1	Ροή Δεδομένων	20
3.2	Εισαγωγή Δεδομένων	22
3.3	Εμφάνιση Αποτελεσμάτων	24

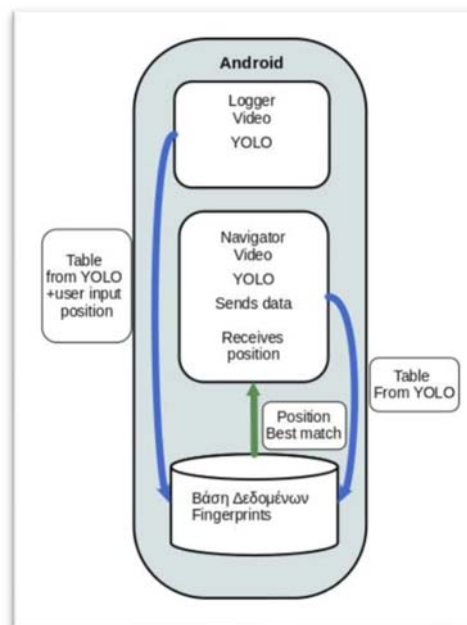
Σε αυτό το κεφάλαιο αναλύεται η αρχιτεκτονική της εφαρμογής που υλοποιήθηκε, η οποία αποτελείται από 3 βασικά υποσυστήματα: τον Logger, τον Navigator και τη βάση δεδομένων, στην οποία αποθηκεύονται τα διάφορα fingerprints. Στο συγκεκριμένο κεφάλαιο περιγράφεται επίσης η ροή της πληροφορίας από την καταγραφή από τον χρήστη, μέσω του υποσυστήματος Logger, μέχρι και την εύρεση της θέσης του χρήστη μέσω της χρήσης του Navigator. Στη συνέχεια, αναλύεται συνοπτικά η διαδικασία εισαγωγής πληροφοριών στη βάση δεδομένων, αλλά και η εμφάνιση αποτελεσμάτων από τη βάση δεδομένων στον τελικό χρήστη, που χρησιμοποιεί την εφαρμογή.

3.1 Ροή Δεδομένων

Η ροή των πληροφοριών από το Logger στη βάση και στη συνέχεια στο Navigator, είναι αρκετά απλή. Ο Logger δημιουργεί και εισάγει fingerprints στη βάση δεδομένων και στη συνέχεια ο Navigator, ανάλογα με τις πληροφορίες που συλλέγει από την κάμερα του χρήστη, ανακτά από τη βάση δεδομένων το πιο παρόμοιο fingerprint, του οποίου εμφανίζει τις συντεταγμένες στον χρήστη για σκοπούς γεωτοποθέτησής του.

Αρχικά, ένας χρήστης χρησιμοποιεί την εφαρμογή στο κινητό του ως Logger, δηλαδή κινείται σε διάφορους χώρους με σκοπό τη δημιουργία fingerprints, τα οποία θα χρησιμοποιηθούν στη συνέχεια από τον Navigator. Ο Logger θα χρησιμοποιήσει την κάμερα του κινητού με την οποία θα τραβήξει ένα βίντεο από τον περιβάλλοντα χώρο και θα στείλει κάποια στιγμιότυπα στο σύστημα YOLO το οποίο εκτελείται/ή είναι

Στη συνέχεια, ένας άλλος χρήστης, ή ο ίδιος χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή, η οποία όμως θα είναι αυτή τη φορά ρυθμισμένη ως Navigator. Ο Navigator και πάλι θα βιντεογραφήσει τον περιβάλλοντα χώρο και με τον ίδιο τρόπο όπως ο Logger, θα στείλει τα διάφορα στιγμιότυπα στο YOLO, το οποίο θα του επιστρέφει τα αντικείμενα που έχει αναγνωρίσει. Τα αντικείμενα αυτά τα εισάγει σε ένα πίνακα και χρησιμοποιώντας ένα αλγόριθμο, ο οποίος περιγράφεται σε ακόλουθα κεφάλαια, βρίσκει με ποιο fingerprint έχει τα περισσότερα κοινά στοιχεία. Στη συνέχεια επιστρέφει πίσω στην εφαρμογή τις συντεταγμένες του συγκεκριμένου fingerprint τις οποίες παρουσιάζει στον τελικό χρήστη που χρησιμοποιεί την εφαρμογή στην προσωπική έξυπνή του συσκευή. Η όλη διαδικασία είναι αρκετά απλή και εμφανίζεται στο ακόλουθο σχήμα 3.1.



21

3.2 Εισαγωγή Δεδομένων

Η εισαγωγή πληροφοριών στο συγκεκριμένο σύστημα εκτελείται από τον Logger, ο οποίος κατά την εκτέλεσή του ανακτά τις απαραίτητες πληροφορίες από τον χρήστη, δημιουργεί ένα fingerprint και το αποθηκεύει στη βάση δεδομένων, για να μπορεί να εκτελεστεί η αναζήτηση και η ανάκτηση ακολούθως από τον Navigator.

Η βάση που χρησιμοποιείται για την αποθήκευση των διαφόρων πληροφοριών είναι της μορφής SQLite, η οποία δημιουργείται και αποθηκεύεται μέσα στην ίδια εφαρμογή.

Η εισαγωγή πληροφοριών γίνεται από το υποσύστημα του Logger. Ο Logger, εκτελείται από κάποιο χρήστη ή από έναν εθελοντή ο οποίος θα δημιουργήσει τα διάφορα fingerprints για κάποιο χρήστη. Η εκτέλεση του Logger συνοπτικά είναι η εξής. Το υποσύστημα εκτελείται από ένα χρήστη ο οποίος με τη χρήση της κάμερας της συσκευής του κάνει καταγραφή του περιβάλλοντά του χώρου.

Μετά το τέλος της καταγραφής του χώρου και την παράλληλη εκτέλεση του συστήματος YOLO, γίνεται αναγνώριση σε πραγματικό χώρο των αντικείμενων που εμφανίζονται κατά την καταγραφή. Τα διάφορα αντικείμενα, τα οποία αναγνωρίστηκαν, εισάγονται σε ένα πίνακα. Όταν ολοκληρωθεί η καταγραφή του χώρου ο πίνακας αυτός στέλνεται στη βάση δεδομένων μαζί με τις συντεταγμένες του GPS της τρέχουσας θέσης για την εισαγωγή τους στη βάση δεδομένων. Η διαδικασία αυτή υλοποιείται όπως φαίνεται στο ακόλουθο σχήμα (Σχήμα 3.2).

```

public void addLocation(int[] objects, ArrayList<Double> Coordinates){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put("coordinateX",Coordinates.get(0));
    values.put("coordinateY",Coordinates.get(1));

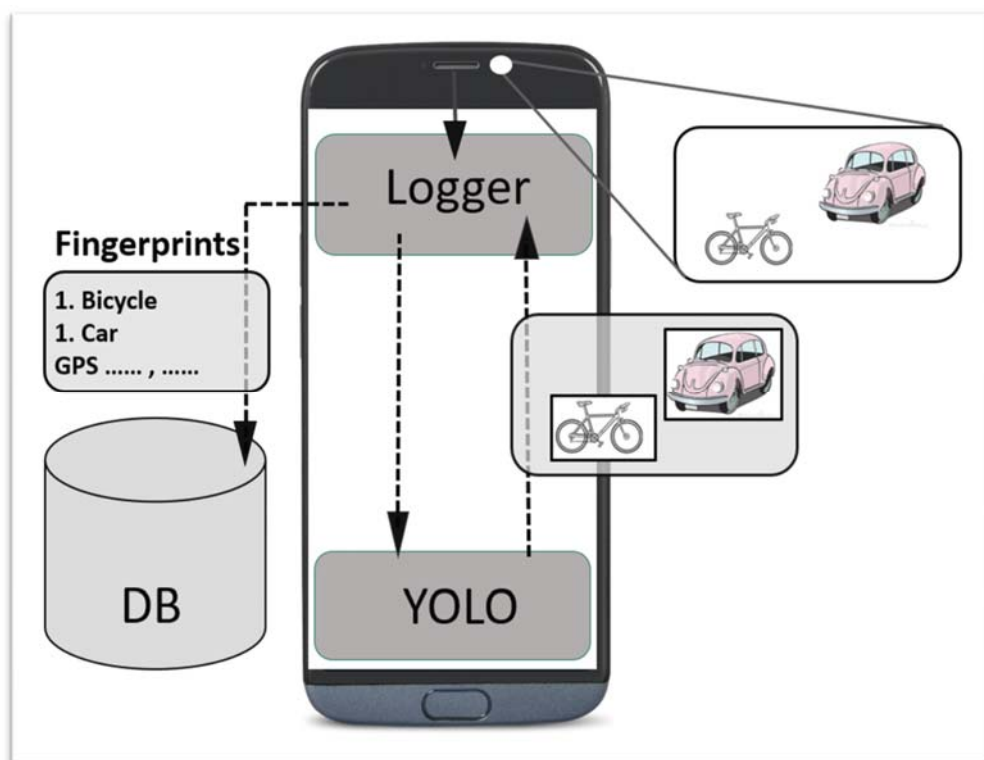
    for(int i=0;i<cocoNames.size()-1;i++)
        values.put(cocoNames.get(i),objects[i]);

    db.insert(TABLE_NAME, nullColumnHack: null,values);
    db.close();
}

```

Σχήμα 3.2: Υλοποίηση συνάρτησης εισαγωγής fingerprint σε βάση δεδομένων

Η κάθε εισαγωγή γραμμής στη βάση δεδομένων αντιπροσωπεύει ένα fingerprint, το οποίο χρησιμοποιείται στη συνέχεια από τον Navigator. Συγκεκριμένα, όπως αναλύθηκε στο προηγούμενο κεφάλαιο, το YOLO λαμβάνει εικόνες / στιγμιότυπα από τον περιβάλλοντα χώρο τις αναγνωρίζει και στη συνέχεια επιστρέφει στο σύστημα τα ονόματα των αντικειμένων που αναγνώρισε στη συγκεκριμένη εικόνα / στιγμιότυπο (Σχήμα 3.3).



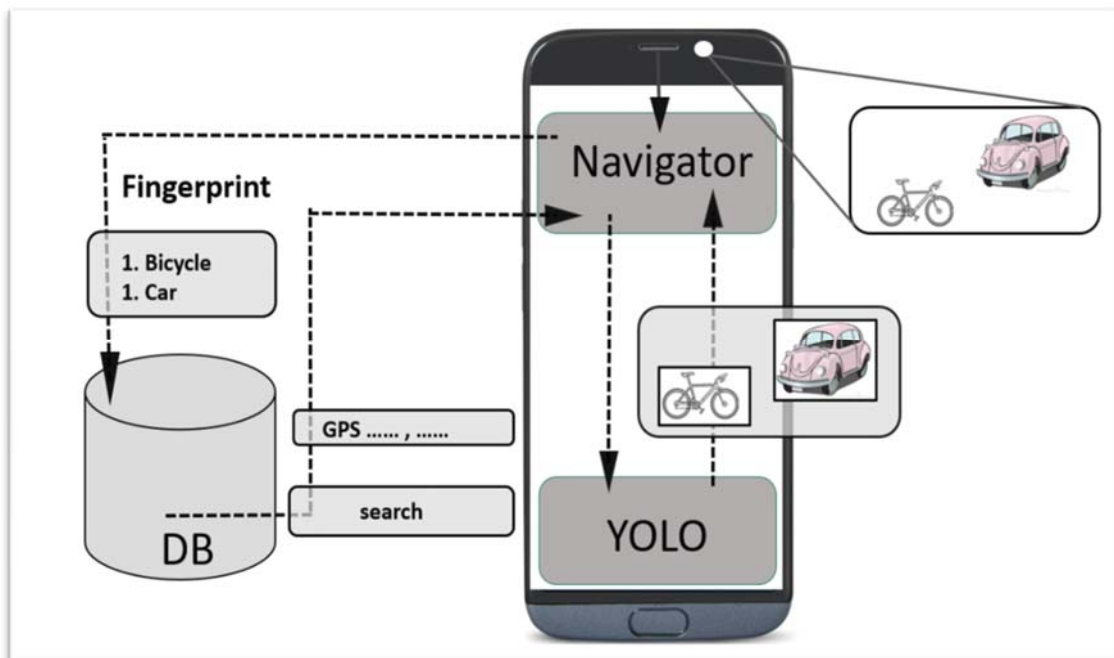
Σχήμα 3.3. Αρχιτεκτονική υποσυστήματος Logger

3.3 Εμφάνιση Αποτελεσμάτων

Η εμφάνιση αποτελεσμάτων γίνεται από τον Navigator, μετά τη δημιουργία ενός αριθμού από fingerprints. Ο Navigator μπορεί εκτελώντας τον αλγόριθμο του κοντινότερου γείτονα, να εντοπίζει ποιο fingerprint στη βάση ταιριάζει περισσότερο στις πληροφορίες που ανέκτησε από το χρήστη και να εμφανίζει τις συντεταγμένες που αντιστοιχούν στο συγκεκριμένο fingerprint στον χρήστη με σκοπό την γεωτοποθέτησή του.

Η εμφάνιση αποτελεσμάτων είναι μέρος της εκτέλεσης του Navigator. Ο Navigator, όπως και ο Logger έχουν πρωταρχική λειτουργία την καταγραφή του περιβάλλοντα χώρου. Ο χρήστης, οποίος εκτελεί το υποσύστημα θα καταγράψει τον περιβάλλοντά του χώρο, δηλαδή τι αντικείμενα βλέπει γύρω του. Στη συνέχεια με τη χρήση του συστήματος YOLO αναγνωρίζονται τα αντικείμενα που εμφανίζονται κατά τη διάρκεια καταγραφής του χώρου από το χρήστη. Τα αντικείμενα αυτά κατά το τέλος της καταγραφής αποθηκεύονται σε ένα πίνακα, ο οποίος θα χρησιμοποιηθεί από τον αλγόριθμο για την εύρεση της τρέχουσας τοποθεσίας του χρήστη.

Μετά τη λήψη στιγμιότυπων από τον περιβάλλοντα χώρο και την επεξεργασία τους μέσω των νευρωνικών δικτύων που παρέχονται από το YOLO, δημιουργείται μια προσωρινή καταγραφή. Η καταγραφή αυτή περιλαμβάνει είτε την τιμή 1 αν κάποιο αντικείμενο έχει αναγνωριστεί κατά την καταγραφή του χώρου, είτε την τιμή 0 αν δεν έχει εντοπιστεί κάποιο αντικείμενο στα στιγμιότυπα που κατέγραψε η κάμερα. Στη συνέχεια, εκτελώντας ένα αλγόριθμο ο οποίος εντοπίζει ποιο αποθηκευμένο fingerprint έχει τα περισσότερα κοινά στοιχεία και τη λιγότερη απόσταση, θεωρεί πως το πιθανότερο είναι πως ο χρήστης βρίσκεται στις συντεταγμένες του συγκεκριμένου fingerprint (Σχήμα 3.4). Τις συντεταγμένες αυτές τις εμφανίζει σε ένα νέο παράθυρο στον χρήστη με τις πιθανές του συντεταγμένες.



Σχήμα 3.4: Αρχιτεκτονική υποσυστήματος Navigator

Κεφάλαιο 4

Υλοποίηση Logger με Χρήση Μηχανικής Όραση

4.1	Περιγραφή Logger του AnyPlace	26
4.2	Πρώτη Εκτέλεση του Συστήματος	28
4.3	Γραφικό Περιβάλλον	29
4.4	Επεξεργασία Στιγμιότυπων με Χρήση YOLO	31
4.5	Εισαγωγή Δεδομένων σε Βάση	33
4.6	Υλοποίηση Υποσυστήματος Demo	35
4.7	Υλοποίηση Μεθόδου για μη Αναγνώριση Συγκεκριμένων Αντικειμένων	37

Ο Logger χρησιμοποιείται σε συστήματα γεωτοποθέτησης ενός χρήστη μέσα σε ένα εσωτερικό χώρο και δημιουργεί fingerprints για κάθε θέση στην οποία βρίσκεται ο χρήστης. Τα εν λόγω fingerprints καταγράφονται από το χρήστη. Για παράδειγμα στο σύστημα AnyPlace ο Logger καταγράφει σήματα ισχύος των διαφόρων σημείων πρόσβασης σε Wi-Fi και τα αποθηκεύει στη βάση δεδομένων. Κάθε εισαγωγή στη βάση δεδομένων θεωρείται ως ένα fingerprint. Στη δική μας περίπτωση ο Logger δημιουργεί ένα fingerprint το οποίο όμως περιλαμβάνει τα αντικείμενα που αναγνωρίζονται από τον περιβάλλοντα χώρο και τις συντεταγμένες GPS της συγκεκριμένης θέσης.

4.1 Περιγραφή Logger του AnyPlace

Ο Logger, είναι ένα υποσύστημα το οποίο χρησιμοποιείται από συστήματα γεωπλοήγησης. Ο Logger στα συστήματα αυτά, όπως και στο σύστημα της παρούσας πτυχιακής εργασίας είναι υπεύθυνος να καταγράψει διάφορα σημεία ενδιαφέροντος, είτε τα σήματα Wi-Fi, είτε στη δική μας περίπτωση τα αντικείμενα του περιβάλλοντα χώρου τα οποία εισάγονται σε κάποια βάση δεδομένων ώστε να χρησιμοποιηθούν αργότερα στο υποσύστημα του Navigator. Για την καλύτερη κατανόηση του υποσυστήματος Logger της παρούσας εργασίας θα επεξηγηθεί αρχικά πιο αναλυτικά ο Logger του ήδη

υλοποιημένου συστήματος AnyPlace. Αρχικά, είναι σημαντικό να αναφερθεί πως η κύρια ιδέα της γεωτοποθέτησης στο σύστημα AnyPlace είναι η εύρεση της θέσης του χρήστη με τη βοήθεια των σημάτων του Wi-Fi.

Το υποσύστημα Logger είναι μια android εφαρμογή που χρησιμοποιείται από εθελοντές οι οποίοι συλλέγουν δεδομένα. Οι εθελοντές επιλέγουν από την εφαρμογή σε ποιο κτήριο βρίσκονται και την ακριβή τους θέση για να δημιουργηθεί το ανάλογο fingerprint στη βάση [24]. Για την αποφυγή της δημιουργίας μεγάλου αριθμού fingerprints για μια συγκεκριμένη θέση στην οθόνη του εθελοντή, αμέσως μετά την επιλογή ενός κτηρίου εμφανίζεται ένας χάρτης θερμότητας (Σχήμα 4.1) στον οποίο υπάρχουν καταγραμμένα fingerprints. Έτσι ο εθελοντής διαπιστώνει σε ποιες περιοχές δεν υπάρχει μεγάλος αριθμός fingerprints και άρα απαιτείται η καταγραφή περισσότερων δεδομένων. Μετά τη συλλογή των δεδομένων που επιθυμεί ο εθελοντής, του παρέχεται η δυνατότητα να τα ανεβάσει στο cloud με μορφή JSON. Έτσι τα συγκεκριμένα δεδομένα που έχουν ανέβει από τον εθελοντή είναι πλέον διαθέσιμα σε άλλους χρήστες οι οποίοι θα μπορούν να βρουν τη θέση τους με περισσότερη ακρίβεια, μέσω του Navigator.



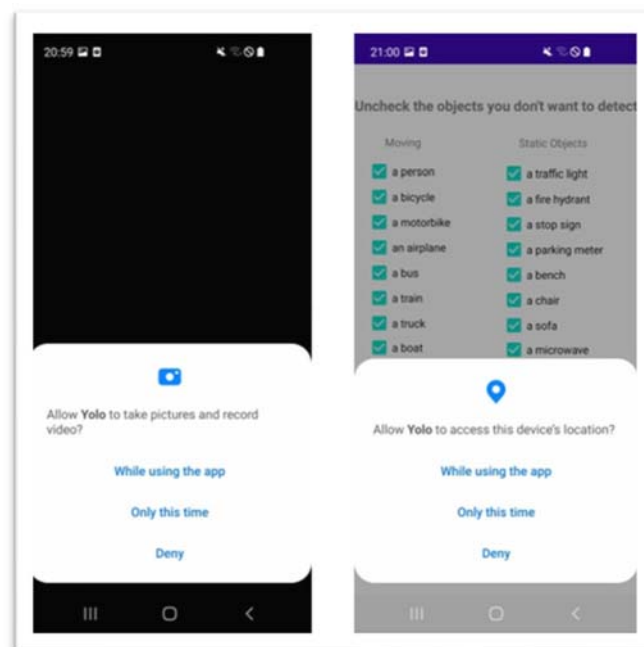
Σχήμα 4.1: Heatmap από κτήριο σε AnyPlace Logger

Τα διάφορα fingerprints που δημιουργούνται από το σύστημα του Logger στο AnyPlace περιλαμβάνουν την ένταση του σήματος Wi-Fi και τις συντεταγμένες χ , ψ . Οι συντεταγμένες αυτές, όπως αναφέρεται πιο πάνω, υπολογίζονται με βάση την επιλογή

του ακριβούς σημείου, σε ένα χάρτη του κτηρίου και του ορόφου, στο οποίο ο χρήστης βρίσκεται τη συγκεκριμένη στιγμή μετά την καταγραφή των δεδομένων αυτών.

4.2 Πρώτη Εκτέλεση του Συστήματος

Η εκτέλεση του Logger, αρχίζει από τη στιγμή που ο χρήστης ανοίγει την εφαρμογή και επιλέγει να την εκτελέσει ως Logger. Κατά την πρώτη εκτέλεση της εφαρμογής ζητείται από τον χρήστη η άδεια της χρήσης της κάμερας του κινητού, αφού η κύρια λειτουργία της εφαρμογής στηρίζεται στη χρήση της κάμερας της έξυπνης συσκευής. Αυτό φαίνεται πιο κάτω στο Σχήμα 4.2.



Σχήμα 4.2: Άδειες χρήσης κάμερας και τοποθεσίας συσκευής

Στη συνέχεια, με την έναρξη εκτέλεσης της εφαρμογής είναι απαραίτητη η μεταφορά των αρχείων YOLO πάνω στη συσκευή που αυτή εκτελείται. Το YOLO για να μπορεί να εκτελέσει τις διάφορες του λειτουργίες απαιτεί την πρόσβαση σε δύο αρχεία: στο νευρωνικό δίκτυο που χρησιμοποιείται ως default ή σε αυτό που προέκυψε από την εκπαίδευση κάποιας συλλογής δεδομένων (yolov3.weights). Επιπλέον, για τη λειτουργία του YOLO απαιτείται ένα ακόμα αρχείο (yolov3.cfg), το οποίο περιλαμβάνει τα διάφορα χαρακτηριστικά του νευρωνικού δικτύου που έχει αποθηκευτεί. Μερικά χαρακτηριστικά είναι για παράδειγμα ο αριθμός των αντικειμένων που μπορεί να αναγνωρίσει το

συγκεκριμένο μοντέλο καθώς και ο αριθμός των επιπέδων που περιλαμβάνει το συγκεκριμένο νευρωνικό δίκτυο. Είναι σημαντικό να σημειωθεί πως τα συγκεκριμένα αρχεία είναι ενσωματωμένα στον εσωτερικό αποθηκευτικό χώρο της εφαρμογής (assets folder). Δεν μπορούν όμως να διαβαστούν λόγω του τρόπου αποθήκευσής τους. Γι' αυτό κατά την πρώτη εκτέλεση της εφαρμογής σε οποιαδήποτε μορφή είτε ως Logger είτε ως Navigator, είναι απαραίτητο τα δύο αρχεία να αντιγραφούν και να μεταφερθούν στον αποθηκευτικό χώρο της συσκευής όπου είναι δυνατή και η πρόσβαση αλλά και η ανάγνωσή τους από το σύστημα για την εκτέλεση των απαραίτητων YOLO συναρτήσεων. Στη συνέχεια, αφού γίνει αποθήκευση των αρχείων στον αποθηκευτικό χώρο της προσωπικής συσκευής του τελικού χρήστη, η εφαρμογή αποθηκεύει τις απαραίτητες πληροφορίες από τα αρχεία σε μεταβλητές που θα χρησιμοποιηθούν αργότερα με τη βοήθεια συναρτήσεων που υλοποιήθηκαν από τη βιβλιοθήκη OpenCV.

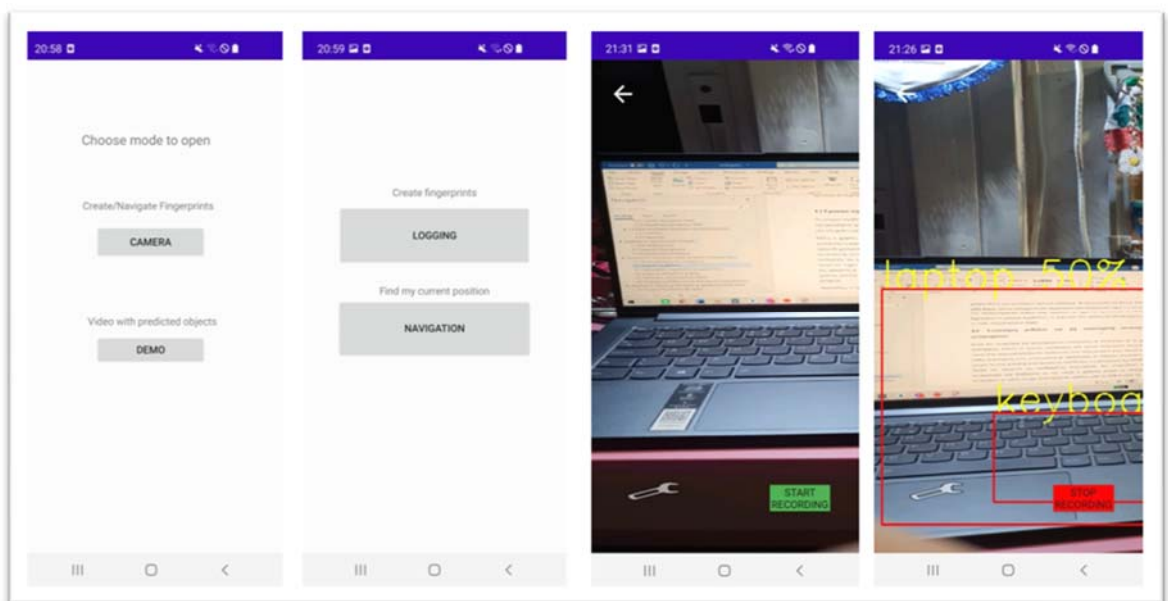
4.3 Γραφικό Περιβάλλον

Το γραφικό περιβάλλον της εφαρμογής είναι αρκετά κατανοητό, εύχρηστο και προσιτό για οποιοδήποτε χρήστη ανεξαρτήτως της εμπειρίας που έχει, είτε στον προγραμματισμό, είτε στη χρήση εφαρμογών σε έξυπνες συσκευές.

Μόλις ο χρήστης πατήσει το εικονίδιο για άνοιγμα της συγκεκριμένης εφαρμογής φορτώνεται η αρχική σελίδα της εφαρμογής. Τότε ο χρήστης καλείται να επιλέξει με ποιο τρόπο θα χρησιμοποιήσει την εφαρμογή. Συγκεκριμένα, είτε θα κάνει χρήση της κάμερας της συσκευής του, είτε για σκοπούς επίδειξης θα κάνει χρήση του βίντεο. Ο τρόπος αυτός λειτουργίας της εφαρμογής περιγράφεται στο υποκεφάλαιο 4.3. Το παρόν κεφάλαιο αφορά τον Logger και για αυτό θεωρούμε ότι ο χρήστης θα επιλέξει να χρησιμοποιήσει την εφαρμογή με χρήση της κάμερας. Στη συνέχεια, φορτώνεται μια άλλη οθόνη και ο χρήστης καλείται να επιλέξει αν επιθυμεί να εκτελέσει την εφαρμογή ως Logger ή ως Navigator.

Ακολούθως, η εφαρμογή φορτώνει το γραφικό περιβάλλον για τον Logger. Το γραφικό περιβάλλον υποβοηθά τον τελικό χρήστη να καταγράψει τον περιβάλλοντά του χώρο στην τρέχουσά του θέση. Στη συνέχεια ο χρήστης πατάει το κουμπί στην οθόνη του «Start Recording». Σημειώνεται ότι το χρώμα του εν λόγω κουμπιού μόλις πατηθεί από πράσινο γίνεται κόκκινο, έτσι ώστε ο χρήστης να μπορεί να γνωρίζει αν η εφαρμογή καταγράφει

και επεξεργάζεται τον περιβάλλοντά του χώρο βλέποντας μόνο το κουμπάκι. Στο κουμπί πλέον αναγράφεται η φράση «Stop Recording» και η εφαρμογή αρχίζει τη διαδικασία αναγνώρισης των διαφόρων αντικειμένων που παρατηρούνται (Σχήμα 4.3). Σε αυτό το σημείο να σημειωθεί πως η εφαρμογή δεν καταγράφει ούτε αποθηκεύει το χώρο στον οποίο φαίνεται ο χρήστης, ούτε κάνει αναγνώριση προσώπων ή άλλων προσωπικών στοιχείων για σκοπούς προστασίας των προσωπικών δεδομένων του χρήστη.



Σχήμα 4.3: Οθόνες για χρήση του Logger

Θα πρέπει επίσης να αναφερθεί πως η κάμερα που χρησιμοποιήθηκε στα πλαίσια της παρούσας διπλωματικής είναι αυτή της βιβλιοθήκης OpenCV. Η οθόνη της κάμερας αυτής ως default είναι οριζόντια (landscape). Λόγω όμως του γεγονότος ότι όλες οι οθόνες της εφαρμογής είναι κυρίως κάθετες (portrait) θεωρήθηκε απαραίτητο να βρεθεί τρόπος η κάμερα να εμφανίζεται ως κάθετη για να μην απαιτείται από το χρήστη να περιστρέφει το κινητό για χρήση της εφαρμογής. Η επίτευξη όμως του σκοπού αυτού, δηλαδή η μετατροπή της οθόνης σε κάθετη ήταν αρκετά δύσκολη. Ο τρόπος υλοποίησης που ακολουθήθηκε ήταν η αναδόμηση από την αρχή του κάθε pixel στην αντίστοιχη κάθετη του θέση. Λόγω του ότι αυτή η διαδικασία δεν υλοποιήθηκε από τη βιβλιοθήκη με προσεκτική παρατήρηση της κάμερας σε σχέση με την εικόνα που βλέπει κανείς στον πραγματικό κόσμο μπορεί να διαπιστωθεί ότι η εικόνα στην κάμερα «απλώνεται» (επιμηκύνεται) δηλαδή η κάμερα μεγαλώνει (διευρύνει) κάπως τα αντικείμενα σε πλάτος.

Ένα ακόμη στοιχείο που προσπαθήσαμε να υλοποιήσουμε για το γραφικό περιβάλλον του συστήματος ήταν η δημιουργία ενός άσπρου ορθογωνίου πλαισίου κειμένου πάνω στο οποίο θα αναγράφονται τα διάφορα αντικείμενα που αναγνώρισε το σύστημα. Η ανάγκη αυτή δημιουργήθηκε λόγω του ότι σε κάποιες περιπτώσεις στην κάμερα εμφανίζονταν πολλά αντικείμενα για αναγνώριση. Στις περιπτώσεις αυτές όταν γινόταν η αναγνώριση τα ορθογώνια με τις λεζάντες των διαφόρων αντικειμένων πιθανόν να εμφανίζονταν πολύ κοντά και να μην ήταν κατανοητό στο χρήστη ποια αντικείμενα έχουν αναγνωρισθεί. Η δημιουργία αυτού του άσπρου ορθογωνίου πλαισίου, παρόλο ότι έχει υλοποιηθεί, στο τέλος αποφασίστηκε να αφαιρεθεί γιατί κάλυπτε μεγάλο μέρος από την οθόνη του χρήστη και δυσκόλευε την αναγνώριση αντικειμένων. Για αυτό η προσπάθεια αυτή δεν περιλήφθηκε στο τελικό σύστημα της διπλωματικής.

4.4 Επεξεργασία Στιγμιότυπων με Χρήση YOLO

Στο σημείο αυτό σημειώνεται ότι η εφαρμογή δεν είναι δυνατό να λάβει κάθε frame κατά τη μετακίνηση του χρήστη στον χώρο. Συγκεκριμένα λαμβάνει κατά μέσο όρο 30 frames ανά δευτερόλεπτο. Αυτό είναι προκαθορισμένο στις ρυθμίσεις της κάμερας του κινητού καθώς και στις συναρτήσεις της βιβλιοθήκης OpenCV η οποία χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής. Με τη λήψη ενός frame ως είσοδο το πρόγραμμα εκτελεί μια σειρά εντολών παρόμοιων με αυτές που εκτελεί το YOLO όταν επεξεργάζεται μια εικόνα στον υπολογιστή. Η εκτέλεση αυτή έχει ήδη περιγράψει αναλυτικά στο κεφάλαιο 2.

Συγκεκριμένα, το πρόγραμμα (Σχήμα 4.4) αρχικά διασπά το frame που έλαβε ως είσοδο σε όσες περισσότερες περιοχές είναι δυνατόν, ανάλογα με το μέγεθος της εικόνας, όπως φαίνεται στο σχήμα 2.2. Ακολούθως, για κάθε περιοχή που διαχωρίστηκε χρησιμοποιώντας τις συναρτήσεις της βιβλιοθήκης OpenCV το πρόγραμμα προσπαθεί να αναγνωρίσει τη συγκεκριμένη περιοχή σαν ένα αντικείμενο. Αν η περιοχή δεν μπορεί να ταυτιστεί με κάποιο μοτίβο ενός αντικειμένου, που είναι καταχωρημένο στο νευρωνικό δίκτυο του προγράμματος, τότε απλά θα προχωρήσει στην επόμενη περιοχή και θα επαναλάβει την ίδια διαδικασία μέχρι να επεξεργαστεί όλες τις περιοχές. Αν όμως, σε κάποια περιοχή αναγνωρίσει μέσω των μοτίβων της κάποιο αντικείμενο, τότε θα δημιουργήσει ένα ορθογώνιο γύρω από το αντικείμενο αυτό, και από κάτω θα σημειώσει

το όνομά του. Ταυτόχρονα θα ορίσει την πιθανότητα απεικόνισης του συγκεκριμένου αντικειμένου.

```
for (int i = 0; i < result.size(); ++i) {
    Mat level = result.get(i);
    for (int j = 0; j < level.rows(); ++j) {
        Mat row = level.row(j);
        Mat scores = row.colRange(5, level.cols());
        Core.MinMaxLocResult mm = Core.minMaxLoc(scores);
        float confidence = (float)mm.maxVal;
        Point classIdPoint = mm.maxLoc;
        if (confidence > confThreshold)
        {
            int centerX = (int)(row.get( row: 0, col: 0)[0] * frame.cols());
            int centerY = (int)(row.get( row: 0, col: 1)[0] * frame.rows());
            int width = (int)(row.get( row: 0, col: 2)[0] * frame.cols());
            int height = (int)(row.get( row: 0, col: 3)[0] * frame.rows());
            int left = centerX - width / 2;
            int top = centerY - height / 2;
            clsIds.add((int)classIdPoint.x);
            confs.add((float)confidence);
            rects.add(new Rect(left, top, width, height));
        }
    }
}
```

Σχήμα 4.4: Κώδικας διαχωρισμού περιοχών σε frames

Όταν ολοκληρωθεί η διαδικασία αυτή για κάθε περιοχή της εικόνας που δόθηκε στο σύστημα σαν είσοδο, την επιστρέφει στη συνάρτηση που χειρίζεται το γραφικό περιβάλλον, και προβάλλει στον χρήστη το frame με τα ορθογώνια των αντικειμένων που αναγνώρισε (Σχήμα 4.5). Η όλη διαδικασία γίνεται αρκετά γρήγορα και έτσι δεν υπάρχει χρόνος αναμονής του frame, που μόλις κατέγραψε η κάμερα της συσκευής για το χρήστη. Το frame, δηλαδή, εμφανίζεται σχεδόν κατευθείαν. Μόλις καταγράψει η κάμερα μια περιοχή εμφανίζονται τα ορθογώνια με τα αναγνωρισμένα αντικείμενα. Η εκτέλεση επαναλαμβάνεται συνεχώς για όση διάρκεια επιθυμεί ο χρήστης.

```

float nmsThresh = 0.2f;
MatOfFloat confidences = new MatOfFloat(Converters.vector_float_to_Mat(confs));
Rect[] boxesArray = rects.toArray(new Rect[0]);
MatOfRect boxes = new MatOfRect(boxesArray);
MatOfInt indices = new MatOfInt();
Dnn.NMSBoxes(boxes, confidences, confThreshold, nmsThresh, indices);
// Draw result boxes:
int[] ind = indices.toArray();
for (int i = 0; i < ind.length; ++i) {
    int idx = ind[i];
    Rect box = boxesArray[idx];
    int idGuy = clsIds.get(idx);
    float conf = confs.get(idx);
    // List<String> coco = Arrays.asList("a person", "a bicycle", "a motorbike", "an airplane", "a bus", "a train", "a truck", "a boat", "a traffic light", "a
    int intConf = (int) (conf * 100);
    Improc.putText(frame, text: coco.get(idGuy) + " " + intConf + "%", box.tl(), Core.FONT_HERSHEY_SIMPLEX, fontScale: 2, new Scalar(255, 255, 0), thickness: 2);
    Improc.rectangle(frame, box.tl(), box.br(), new Scalar(255, 0, 0), thickness: 2);
    objects[idGuy] = objects[idGuy] + 1;
}

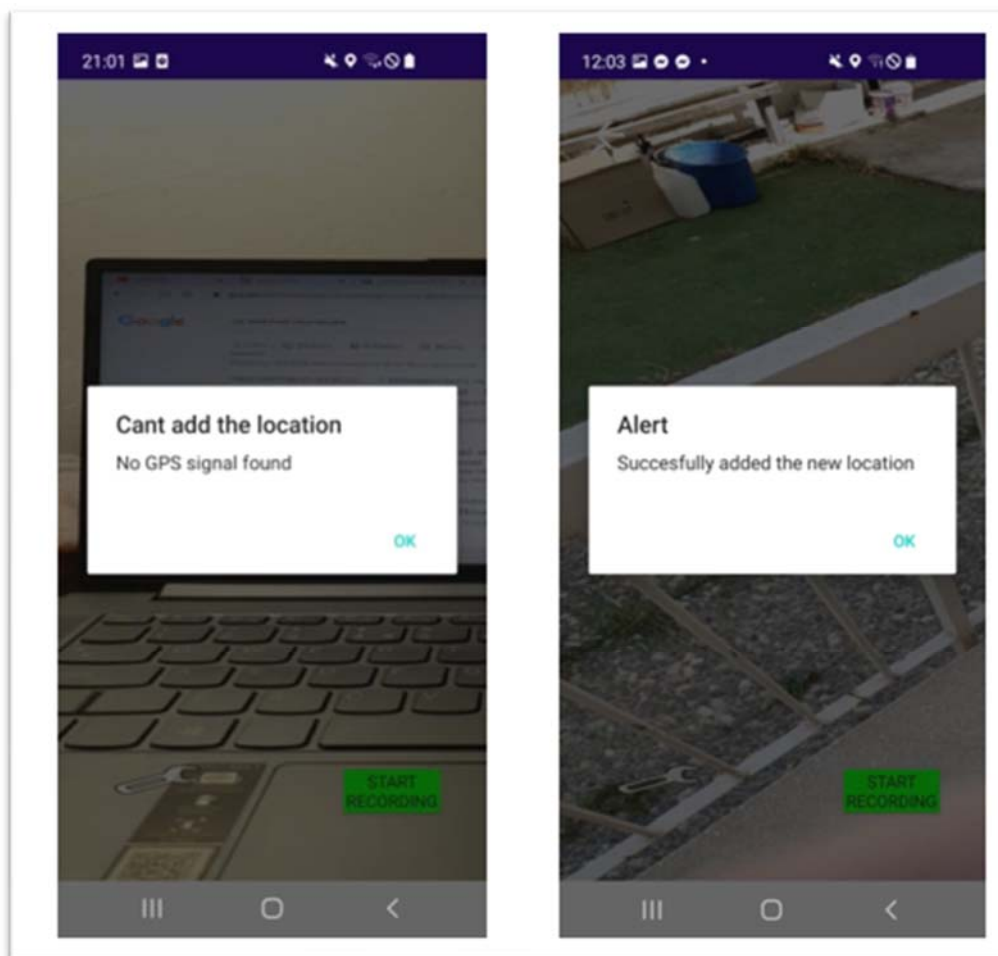
```

Σχήμα 4.5 : Κώδικας εντοπισμού ενός αντικειμένου σε μια περιοχή και δημιουργία ορθογωνίου πλαισίου γύρω από τη περιοχή αυτή

Όταν ο χρήστης θεωρήσει πως παρείχε αρκετές πληροφορίες στο σύστημα για να γίνει σωστή γεωτοποθέτησή του πατά το κουμπί «Stop Recording». Στην κάμερα δεν εμφανίζονται πλέον τα αναγνωρισμένα αντικείμενα αφού η εκτέλεση του YOLO σταματά. Καθ' όλη τη διάρκεια της εκτέλεσης του YOLO, δηλαδή κατά την καταγραφή των αντικειμένων, τα ονόματα των αντικειμένων αποθηκεύονται σε έναν πίνακα ο οποίος θα εισαχθεί αργότερα στη βάση δεδομένων. Με το πάτημα του κουμπιού «Stop Recording» το πρόγραμμα θα αρχίσει τη διαδικασία εισαγωγής δεδομένων στη βάση.

4.5 Εισαγωγή Δεδομένων σε Βάση

Αν η εφαρμογή εκτελείται πρώτη φορά στο κινητό τότε θα δημιουργηθεί η βάση δεδομένων. Η βάση δεδομένων περιλαμβάνει μόνο έναν πίνακα ο οποίος έχει ως στήλες τις συντεταγμένες του GPS και μια στήλη για κάθε αντικείμενο που αναγνωρίζεται από το YOLO. Ο συγκεκριμένος πίνακας δεν έχει κάποιο πρωτεύον κλειδί λόγω του ότι εμπλέκονται οι συντεταγμένες GPS οι οποίες έχουν ακρίβεια θέσης μερικά μέτρα. Για μια δυάδα συντεταγμένων πιθανόν να υπάρχουν δύο ως τρία fingerprints αντίστοιχα για τη συγκεκριμένη θέση. Πριν γίνει η εισαγωγή των δεδομένων στη βάση, εκτελείται μια συνάρτηση η οποία ανακτά τις τρέχουσες συντεταγμένες του χρήστη. Αν δεν βρεθούν συντεταγμένες για την συγκεκριμένη θέση τότε θα εμφανιστεί ένα μήνυμα πως δεν βρέθηκε σήμα GPS όπως φαίνεται στο Σχήμα 4.6. Αν όμως βρεθούν οι συντεταγμένες και θα εμφανιστεί στο χρήστη ένα μήνυμα επιβεβαίωσης ότι έγινε επιτυχής εισαγωγή της θέσης στη βάση δεδομένων.



Σχήμα 4.6: Μηνύματα αποτυχίας και επιτυχίας από τον Logger

Εδώ κρίνεται σκόπιμο να αναφερθεί ένα πρόβλημα που προκύπτει στο σύστημα λόγω της επεξεργασίας συνεχόμενων frames της κάμερας από το YOLO. Αν για παράδειγμα το σύστημα αναγνωρίσει ένα αυτοκίνητο, και το επόμενο frame περιλαμβάνει το ίδιο αυτοκίνητο, τότε η υλοποίηση του YOLO δεν του επιτρέπει εύκολα να αναγνωρίσει ότι το αυτοκίνητο που αναγνώρισε στο προηγούμενο frame είναι το ίδιο με το αυτοκίνητο στο τρέχον frame. Για αυτό κατά την καταμέτρηση αντικειμένων, όπως στο παράδειγμα των αυτοκινήτων που προαναφέρθηκε, το σύστημα μετρά περισσότερα αντικείμενα από όσα πραγματικά θα καταμετρούσε ένας πραγματικός άνθρωπος. Στην καταμέτρηση δηλαδή των αυτοκινήτων το σύστημα θα κατάγραφε δύο ενώ στην πραγματικότητα ήταν ένα και μετρήθηκε δύο φορές.

Για να αποφευχθούν περιπτώσεις λάθους ακολούθως από τον Navigator, στη βάση δεν εισάγεται το πλήθος αντικειμένων που βρέθηκε αλλά εισάγεται ο αριθμός 1 αν βρέθηκε

το συγκεκριμένο αντικείμενο κατά την καταγραφή του χώρου ή 0 αν δεν βρέθηκε το συγκεκριμένο αντικείμενο στον περιβάλλοντα χώρο.

4.6 Υλοποίηση Υποσυστήματος Demo

Κατά την έναρξη της υλοποίησης της συγκεκριμένης εφαρμογής κρίθηκε αναγκαία η δημιουργία ενός υποσυστήματος Demo. Αυτό έγινε γιατί το YOLO χρησιμοποιείται κυρίως σαν υπολογιστική εφαρμογή και ο επιθυμητός στόχος ήταν η υιοθέτηση της τεχνολογίας του YOLO σε κάποια κινητή συσκευή με τη μέγιστη δυνατή αποδοτικότητα. Το υποσύστημα αυτό λαμβάνει ως είσοδο ένα βίντεο και το αναπαράγει στον χρήστη. Αναγνωρίζει πάνω στα frames τα διάφορα αντικείμενα που θα αναγνώριζε αν το βίντεο ήταν ο περιβάλλοντας χώρος του χρήστη, αν δηλαδή χρησιμοποιούσε για παράδειγμα το υποσύστημα του Logger. Η υλοποίηση του υποσυστήματος Demo εκτελείται όπως περιγράφεται στη συνέχεια.

Αρχικά, ο χρήστης καλείται να επιλέξει αν θέλει να χρησιμοποιήσει την εφαρμογή με τη χρήση της κάμερας ή με τη χρήση βίντεο. Στη δεύτερη περίπτωση ο χρήστης θα επιλέξει τη ρύθμιση Demo. Στη συνέχεια, καλείται να επιλέξει ποιο βίντεο θέλει να χρησιμοποιήσει (Σχήμα 4.7). Υπάρχουν τρία βίντεο, τα οποία εισάγονται στο κινητό ως παραδείγματα εκτέλεσης του συγκεκριμένου υποσυστήματος. Μπορεί όμως ο χρήστης να προσθέσει κάποιο δικό του βίντεο. Η προϋπόθεση για την εισαγωγή ενός επιπλέον βίντεο είναι μόνο η απαίτηση αποθήκευσής του στον αποθηκευτικό χώρο της εφαρμογής στον υποφάκελο Movies. Μετά την επιλογή του βίντεο που επιθυμεί ο χρήστης, η εφαρμογή επεξεργάζεται διαδοχικά τα frames του βίντεο, σε αντίθεση με την κάμερα που όπως αναφέρθηκε πιο πάνω επεξεργάζεται μόνο 30 frames ανά δευτερόλεπτο.



Σχήμα 4.7: Επιλογή βίντεο που τύχει επεξεργασίας

Το σύστημα λαμβάνοντας ένα frame το διασπά σε περιοχές, όπως αναφέρθηκε στην επεξήγηση της επεξεργασίας των frames, και προσπαθεί να διαπιστώσει αν το μοτίβο της συγκεκριμένης περιοχής ταιριάζει με κάποιο αντικείμενο που αναγνωρίζει σύμφωνα με το νευρωνικό του δίκτυο. Αν το μοτίβο ταιριάζει, τότε το σύστημα θα δημιουργήσει ένα ορθογώνιο γύρω από το αναγνωρισμένο αντικείμενο μαζί με μια λεζάντα του ονόματος του αντικειμένου, και θα σημειώσει ένα ποσοστό το οποίο αντιπροσωπεύει την πιθανότητα το συγκεκριμένο αντικείμενο, που αναγνώρισε να είναι όντως το ορθό αντικείμενο. Αφού ολοκληρωθεί η διαδικασία για το συγκεκριμένο frame, τότε αυτό εμφανίζεται στον χρήστη (Σχήμα 4.8). Κατά την υλοποίηση του συγκεκριμένου υποσυστήματος, δυστυχώς δεν κατέστη δυνατή, μετά την επεξεργασία, η αναδόμηση του βίντεο με τα επεξεργασμένα frames. Για αυτό κατά την εμφάνιση του αποτελέσματος, εμφανίζεται το κάθε frame σαν εικόνα. Ο τελικός όμως χρήστης, λόγω της μεγάλης σχετικά ταχύτητας λειτουργίας του υποσυστήματος, μπορεί να πιστεύει ότι βλέπει βίντεο. Επιπλέον σημειώνεται ότι παρόλο ότι τα frames εμφανίζονται σαν εικόνες αντί σαν βίντεο, υλοποιήθηκε η λειτουργία της εκτέλεσης/παύσης του βίντεο. Με το πάτημα

δηλαδή του κουμπιού γίνεται η παύση της εμφάνισης εικόνων, και με το εκ νέου πάτημα του κουμπιού το βίντεο συνεχίζει να εκτελείται από το σημείο της παύσης.



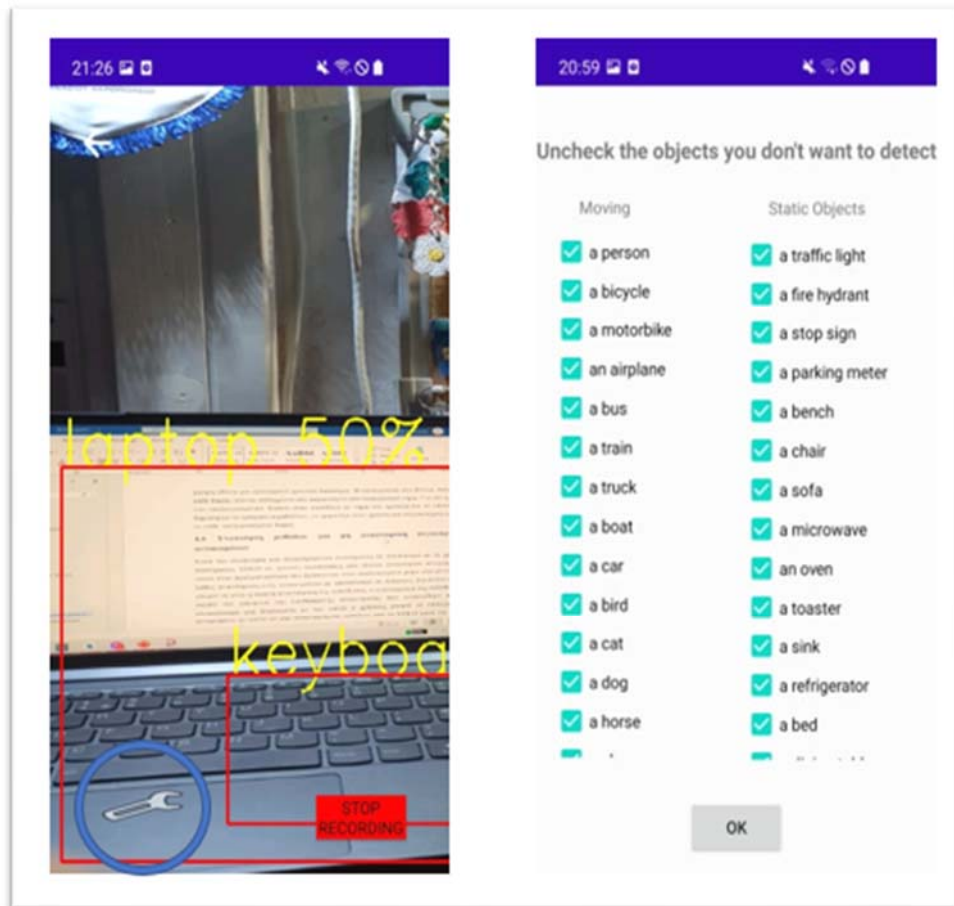
Σχήμα 4.8 : Στιγμιότυπο εμφάνισης στην οθόνη επεξεργασμένου βίντεο

Επιπλέον, λόγω του ότι η επεξεργασία ενός frame με τη χρήση του YOLO είναι αρκετά χρονοβόρα, αφού πρέπει πρώτα να γίνει επεξεργασία όλων των frames και στη συνέχεια να γίνει φόρτωση του γραφικού περιβάλλοντος, ο χρήστης αναγκάζεται να έχει μια μαύρη οθόνη για εκτεταμένο χρονικό διάστημα. Η επεξεργασία του βίντεο, δηλαδή του κάθε frame, γίνεται ασύγχρονα στο παρασκήνιο από διαφορετικό νήμα. Για την εμφάνιση των επεξεργασμένων frames είναι υπεύθυνο το νήμα του προσκηνίου, το οποίο αρχικά δημιουργεί το γραφικό περιβάλλον, το εμφανίζει στον χρήστη και στη συνέχεια εμφανίζει το κάθε επεξεργασμένο frame.

4.7 Υλοποίηση Μεθόδου για μη Αναγνώριση Συγκεκριμένων Αντικειμένων

Κατά την υλοποίηση του συγκεκριμένου συστήματος, σε συνδυασμό με τη χρήση του συστήματος YOLO, σε αρκετές περιπτώσεις είτε γίνεται αναγνώριση αντικειμένων τα οποία στην πραγματικότητα δεν βρίσκονται στον συγκεκριμένο χώρο είτε γίνεται συχνά λάθος αναγνώριση ενός αντικειμένου, με αποτέλεσμα σε διάφορες περιπτώσεις να μην μπορεί να γίνει η σωστή αναγνώριση της θέσης ή η καταγραφή της. Με σκοπό την αποφυγή της λανθασμένης αναγνώρισης που αναφέρθηκε πιο πάνω, υλοποιήσαμε μια

διαδικασία με την οποία ο χρήστης μπορεί να επιλέξει κάποιο αντικείμενο το οποίο να μην αναγνωρίζεται καθόλου από το YOLO κατά την εκτέλεση του προγράμματος (Σχήμα 4.9).



Σχήμα 4.9 : Οθόνη μη αναγνώρισης αντικειμένων

Σημειώνεται ότι στο γραφικό περιβάλλον του Logger υπάρχει ένα εικονίδιο σουηδικού κλειδιού, που παρουσιάζεται στο κάτω αριστερά μέρος της οθόνης, το οποίο αν πατηθεί από τον χρήστη, θα εμφανιστεί μια νέα οθόνη. Στην οθόνη αυτή υπάρχουν δύο κατάλογοι αντικειμένων: τα στατικά αντικείμενα, δηλαδή αυτά που όταν εντοπιστούν το πιο πιθανό δεν θα αλλάξουν θέση όσες φορές και να επισκεφτεί κάποιος τον συγκεκριμένο χώρο και τα αντικείμενα που κινούνται, δηλαδή τα αντικείμενα τα οποία δεν μένουν σταθερά σε μία θέση και την επόμενη φορά που κάποιος θα επισκεφτεί τον συγκεκριμένο χώρο, τα συγκεκριμένα αντικείμενα ή θα έχουν μετακινηθεί σε άλλη θέση είτε μπορεί να μην υπάρχουν καν στον περιβάλλοντα χώρο.

Επιπλέον αναφέρεται ότι οι κατάλογοι των αντικειμένων περιλαμβάνουν τα ονόματα των αντικειμένων που αναγνωρίζει το YOLO και εμπίπτουν στις συγκεκριμένες κατηγορίες. Δίπλα στο όνομα του κάθε αντικειμένου υπάρχουν πλαίσια ελέγχου (checkboxes) τα οποία όπως θα παρατηρήσει ο χρήστης είναι όλα επιλεγμένα, γιατί ως default το σύστημα αναγνωρίζει και εμφανίζει όλα τα αντικείμενα που αναγνωρίζει το YOLO. Αν ο χρήστης αναρρέσει την επιλογή (un-checks the checkbox) του πλαισίου ελέγχου και πατήσει το κουμπί «συνέχισε» (“continue”) τότε η εφαρμογή θα επιστρέψει στην οθόνη του Logger και όταν ο χρήστης ξαναπατήσει το κουμπί «Start Recording» τότε κατά την εκτέλεση της εφαρμογής δεν θα εμφανίζει τα αντικείμενα τα οποία δεν είναι πλέον επιλεγμένα στην οθόνη.

Κεφάλαιο 5

Υλοποίηση Navigator με Χρήση Μηχανική Όραση

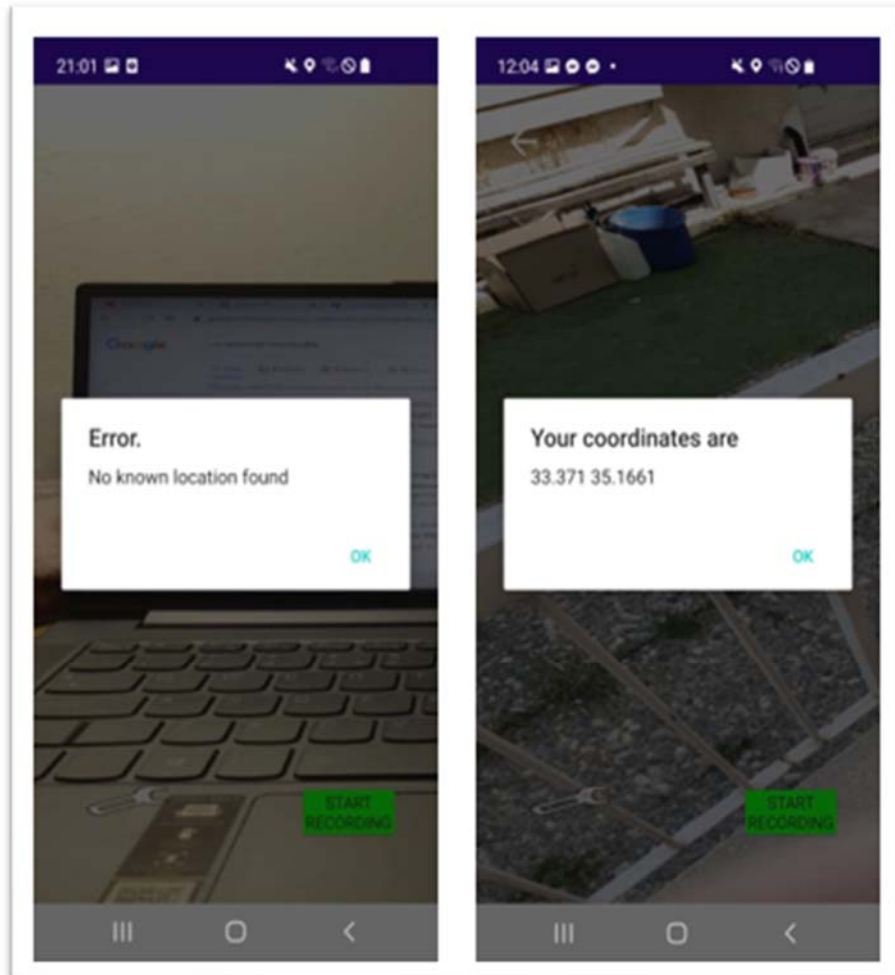
5.1	Γραφικό Περιβάλλον Navigator	40
5.2	Τρόπος Εκτέλεσης	42
5.3	Περιγραφή Αλγορίθμου για Εύρεση Fingerprint	43

Ο Navigator χρησιμοποιείται ως σύστημα για την εύρεση της τρέχουσας θέσης του τελικού χρήστη με τη βοήθεια των πληροφοριών που είχαν ήδη εισαχθεί στον Logger κατά την καταγραφή των διαφόρων θέσεων. Ένα παράδειγμα υλοποιημένου Navigator αποτελεί αυτό του AnyPlace. Στην περίπτωση αυτή ο Navigator λαμβάνει από τον χρήστη τα διάφορα σήματα Wi-Fi της τρέχουσας θέσης και τα συγκρίνει με αυτά που βρίσκονται στη βάση δεδομένων. Στη συνέχεια επιλέγει αυτό που ταιριάζει περισσότερο, επιστρέφοντας τη συγκεκριμένη θέση του χρήστη σαν σήμανση πάνω σε ένα χάρτη του κτηρίου μέσα στο οποίο βρίσκεται. Ο Navigator της παρούσας διπλωματικής υλοποιήθηκε κάπως διαφορετικά. Η υλοποίηση αυτή θα επεξηγηθεί αναλυτικά στη συνέχεια.

5.1 Γραφικό Περιβάλλον Navigator

Καταρχήν σημειώνεται ότι το γραφικό περιβάλλον του Navigator, είναι το ίδιο με αυτό του Logger. Η κυρίως δηλαδή οθόνη είναι η κάμερα του κινητού του χρήστη που δείχνει τον περιβάλλοντα χώρο του χρήστη. Στην κάτω αριστερά γωνία, όπως προαναφέρθηκε, υπάρχει το εικονίδιο ενός σουηδικού εργαλείου το οποίο δίνει τη δυνατότητα στον χρήστη να επιλέξει μερικά αντικείμενα τα οποία δεν θα αναγνωρίσει κατά την καταγραφή του χώρου. Η διαδικασία αυτή περιγράφεται πιο αναλυτικά στο υποκεφάλαιο 4.6. Επιπλέον, στο κάτω δεξιά μέρος της οθόνης υπάρχει το πράσινο κουμπί «Start Recording». Μόλις ο χρήστης αρχίσει την καταγραφή της οθόνης και πατήσει το κουμπί αυτό γίνεται κόκκινο και πάνω του αναγράφεται η φράση «Stop Recording». Όταν

πατηθεί ξανά το κουμπί θα εμφανιστεί ένα pop-up παράθυρο στο οποίο εμφανίζονται οι πιθανές συντεταγμένες που αντιστοιχούν στη συγκεκριμένη θέση του χρήστη που κατέγραψε τον περιβάλλοντά του χώρο (Σχήμα 5.1).



Σχήμα 5.1: Μηνύματα αποτυχίας και επιτυχίας Navigator

Το γραφικό περιβάλλον του Navigator έχει το ίδιο πρόβλημα με την κάμερά του όπως και το γραφικό περιβάλλον του Logger. Το πρόβλημα αυτό αναλύθηκε λεπτομερώς στο υποκεφάλαιο 4.2. Αν ο χρήστης παρατηρήσει προσεκτικά την εικόνα που εμφανίζεται στην οθόνη μπορεί να καταλάβει πως μερικά μεγάλα αντικείμενα τα οποία είναι αρκετά κοντά στην κάμερα εμφανίζονται τραβηγμένα κατά πλάτος σε σχέση με την πραγματική τους εικόνα. Αυτή η παραμόρφωση, όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, οφείλεται στο ότι η κάμερα της βιβλιοθήκης OpenCV βρίσκεται σε landscape μορφή και για αυτό γίνεται προγραμματιστική αναδόμηση των διάφορων pixels στις αντίστοιχες θέσεις στην portrait μορφή της κάμερας.

5.2 Τρόπος Εκτέλεσης

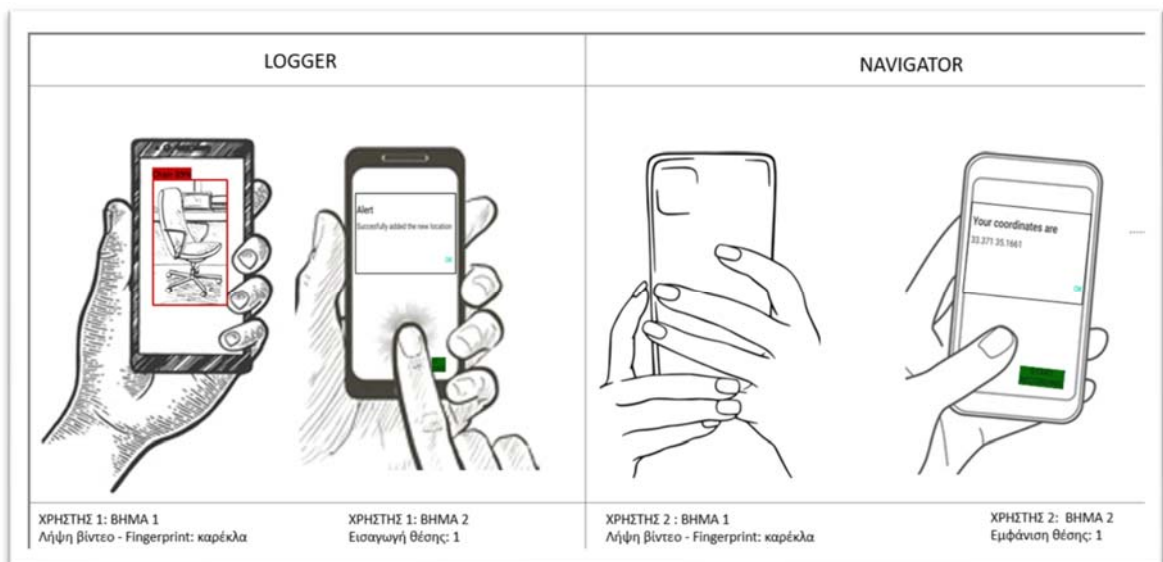
Ο τρόπος εκτέλεσης του Navigator, είναι αρκετά παρόμοιος με αυτόν του Logger. Το γραφικό περιβάλλον των δύο υποσυστημάτων είναι το ίδιο. Τα δύο υποσυστήματα διαφέρουν όμως στον τρόπο καταγραφής του περιβάλλοντα χώρου του χρήστη. Συγκεκριμένα το pop-up παράθυρο που εμφανίζεται στο Logger, απλά ενημερώνει τον χρήστη ότι δημιουργήθηκε με επιτυχία το fingerprint ενώ το παράθυρο αυτό στον Navigator καταγράφει την ολοκλήρωση της καταγραφής του περιβάλλοντα χώρου. Στο εν λόγω pop-up παράθυρο στην περίπτωση του Navigator εμφανίζονται είτε οι συντεταγμένες που βρέθηκαν ότι αντιστοιχούν στη συγκεκριμένη θέση, είτε στο παράθυρο αναγράφεται πως δεν βρέθηκαν κάποιες συντεταγμένες που αντιστοιχούν με τα αντικείμενα που μόλις αναγνωρίστηκαν από τον περιβάλλοντα χώρο του χρήστη.

Η διαδικασία όπως φαίνεται και στο σχήμα 5.2 αρχίζει ως εξής. Μόλις ο χρήστης επιλέξει να χρησιμοποιήσει για πρώτη φορά το υποσύστημα Navigator, τότε με την ενεργοποίηση της εφαρμογής γίνεται αποθήκευση των αρχείων της εφαρμογής πάνω στη συσκευή του χρήστη. Με τον τρόπο αυτό μπορεί να γίνει με επιτυχία η ανάγνωση των απαραίτητων δεδομένων. Τα απαραίτητα αυτά δεδομένα αποθηκεύονται σε μεταβλητές με τη χρήση διαφόρων συναρτήσεων. Για παράδειγμα αποθηκεύεται το νευρωνικό δίκτυο σε μια μεταβλητή για να μπορεί να χρησιμοποιηθεί στη συνέχεια από την εφαρμογή για την αναγνώριση των διαφόρων αντικειμένων. Μετά την αποθήκευση των αντικειμένων, η οποία εκτελείται αρκετά γρήγορα χωρίς ο χρήστης να την αντιλαμβάνεται ζητείται από τον χρήστη το δικαίωμα να χρησιμοποιήσει την κάμερα και το σύστημα GPS της συσκευής του.

Στη συνέχεια, αφού γίνει αποθήκευση των αρχείων και ο χρήστης ζητήσει τα απαραίτητα δικαιώματα, φορτώνεται το γραφικό περιβάλλον, όπως έχει ήδη περιγράψει στο προηγούμενο υποκεφάλαιο 5.1. Μετά δηλαδή το πάτημα του κουμπιού «Start Recording» από τον χρήστη το σύστημα αρχίζει την επεξεργασία των frames (πλαισίων) από την κάμερα. Το YOLO παίρνει ένα-ένα τα frames και τα διασπά σε διάφορες περιοχές και στη συνέχεια ελέγχει αν στην κάθε περιοχή υπάρχει μοτίβο στο οποίο να μπορεί να προσδιορίσει κάποιο αντικείμενο που εμφανίζεται. Αν όντως αναγνωρίσει κάποιο αντικείμενο τότε δημιουργεί ένα ορθογώνιο πλαίσιο γύρω από την περιοχή με το αντικείμενο μαζί με μια λεζάντα στην οποία αναγράφεται το όνομα του αντικειμένου και

ένα ποσοστό. Το ποσοστό αυτό παρουσιάζει την πιθανότητα το αντικείμενο το οποίο μόλις αναγνωρίστηκε να είναι όντως το αντικείμενο που αναγνώρισε.

Με το τέλος της επεξεργασίας κάποιου frame αποθηκεύονται σε ένα πίνακα τα ονόματα των αντικειμένων τα οποία αναγνωρίστηκαν στο συγκεκριμένο frame. Όταν ο χρήστης πατήσει το κουμπί «Stop Recording» τα διάφορα frames δεν τυγχάνουν επεξεργασίας από το YOLO και δημιουργείται ένας πίνακας ο οποίος περιλαμβάνει δύο τιμές. Συγκεκριμένα εμφανίζεται η τιμή 1 για τα κάθε αντικείμενο που αναγνωρίστηκε στον περιβάλλοντα χώρο και 0 για κάθε αντικείμενο που δεν εντοπίστηκε. Αυτός ο πίνακας χρησιμοποιείται στη συνέχεια από τον αλγόριθμο ο οποίος περιγράφεται αμέσως μετά στο υποκεφάλαιο 5.3. Ο αλγόριθμος θα πάρει ως είσοδο τον συγκεκριμένο πίνακα και θα προσπαθήσει να βρει τη θέση που βρίσκεται εκείνη τη στιγμή.



Σχήμα 5.2: Τρόπος Εκτέλεσης του συστήματος

5.3 Περιγραφή Αλγορίθμου για Εύρεση Fingerprint

Ο αλγόριθμος (Σχήμα 5.3) που εκτελείται από τον Navigator, για εύρεση της πιθανής θέσης ανάλογα με την είσοδο που λαμβάνει από τον χρήστη είναι ο εξής. Αρχικά, ο χρήστης ανοίγοντας την εφαρμογή και παρέχοντας τα απαραίτητα δικαιώματα, καταγράφει τον περιβάλλοντά του χώρο στον οποίο δημιουργεί ένα προσωρινό fingerprint που περιλαμβάνει μόνο τα αντικείμενα που εντόπισε το YOLO στον αυτό χώρο. Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, το σύστημα YOLO εμφανίζει ένα πρόβλημα ως προς την αναγνώριση επαναλαμβανόμενων αντικειμένων. Λόγω του

τρόπου αναγνώρισης των αντικειμένων frame by frame αν το σύστημα αναγνωρίσει σε ένα frame ένα αντικείμενο και στο επόμενο frame εμφανιστεί και πάλι το ίδιο αντικείμενο, δεν μπορεί να καταλάβει πως και οι δύο εικόνες αφορούν το ίδιο αντικείμενο. Για αυτό στο fingerprint καταγράφεται ότι αναγνώρισε 2 αντί 1 αντικείμενο που υφίσταται στην πραγματικότητα. Λόγω του συγκεκριμένου προβλήματος για την εύρεση των fingerprints που αντιστοιχούν σε κάποια θέση δεν χρησιμοποιείται κάποιος αλγόριθμος ο οποίος ψάχνει να εντοπίσει το ακριβώς όμοιο του fingerprint που μόλις έχει δοθεί για είσοδο. Αντίθετα εντοπίζει το fingerprint το οποίο είναι το πιο γειτονικό με αυτό που δόθηκε από τον Navigator για είσοδο. Η ανάκτηση των διαφόρων fingerprints από τη βάση δεδομένων γίνεται με τον αλγόριθμο που περιγράφεται στη συνέχεια. Αρχικά, ένα fingerprint περιλαμβάνει τις συντεταγμένες GPS και την τιμή 1 για τα αντικείμενα που αναγνωρίστηκαν στον περιβάλλοντά του χώρου και 0 για τα αντικείμενα που, είτε δεν βρέθηκαν στον χώρο, είτε δεν μπόρεσαν να αναγνωριστούν.

Με την ολοκλήρωση της εκτέλεσης του Navigator, δηλαδή με το άνοιγμα της εφαρμογής από το χρήστη και το πάτημα του κουμπιού «Start Recording» καταγράφεται ο περιβάλλοντας χώρος του χρήστη. Στη συνέχεια ο χρήστης ξαναπατάει το κουμπί για να γίνει «Stop Recording», και ο Navigator δημιουργεί ένα πίνακα ο οποίος περιλαμβάνει την τιμή 1 για τα αντικείμενα που αναγνώρισε στον περιβάλλοντα χώρο του χρήστη. Τον συγκεκριμένο πίνακα για ευκολία κατανόησης θα τον ονομάσουμε πίνακα X. Ο στόχος του αλγορίθμου είναι η εύρεση του πιο κοντινού γειτονικού fingerprint. Ο αλγόριθμος εξετάζει όλα τα fingerprints και σε κάθε επανάληψη επεξεργάζεται ένα fingerprint το οποίο θεωρούμε ως Ψ. Λαμβάνοντας το Ψ μετρά πόσα διαφορετικά αντικείμενα αναφέρονται στο συγκεκριμένο fingerprint και ακολούθως υπολογίζει το άθροισμα των fingerprints ως εξής :

$$\frac{1}{count} * \sum |X[l] - \Psi[l]|$$

Στην εξίσωση αυτή το count εκφράζει τον αριθμό των διαφορετικών αντικειμένων που αναγνωρίστηκαν για το συγκεκριμένο fingerprint, δηλαδή τον αριθμό των πεδίων του fingerprint που δεν είναι 0. Η συγκεκριμένη διαδικασία γίνεται για όλα τα αποθηκευμένα fingerprints στη βάση δεδομένων και εντοπίζεται το fingerprint το οποίο έχει το μικρότερο άθροισμα. Οι συντεταγμένες του συγκεκριμένου fingerprint θεωρούνται ότι αντιστοιχούν στην τρέχουσα θέση του χρήστη.

Με την ολοκλήρωση της διαδικασίας εμφανίζεται ένα pop-up window στον χρήστη στο οποίο εμφανίζονται είτε οι συντεταγμένες που του αντιστοιχούν είτε αναγράφεται ότι δεν βρέθηκε κάποιο fingerprint αρκετά κοντά στο fingerprint που του δόθηκε για να βρεθεί η θέση του. Στη δεύτερη δηλαδή περίπτωση στο pop-up window αναγράφεται πως δεν βρέθηκε η θέση του χρήστη. Τότε ο χρήστης έχει την επιλογή να καταγράψει εκ νέου τον περιβάλλοντά του χώρο. Αν δηλαδή έγινε κάποιο λάθος κατά την πρώτη καταγραφή του χώρου να γίνει ξανά καταγραφή πιθανώς πιο σωστά.

```
for(int j=0;j<data.size();j++) {  
  
    count = 0; sum =0;  
    int[] temp = data.get(j).getObjects();//each fingerprint from database  
  
    for (int k=0;k<cocoNames.size();k++)  
        if(seeing[k]!=0) count++;  
    if (count == 0 ) return null;  
    for(int h=0;h<cocoNames.size();h++)  
        if(seeing[h]!=0 || temp[h]!=0)  
            sum = sum + (1/count) * Math.abs(seeing[h]-temp[h]);  
    if (sum<min){  
        flag = 1;  
        min = sum;  
        minNo=j;  
    }  
}  
  
if (flag == 1){  
    ArrayList<Double> location = new ArrayList<>();  
    location.add(data.get(minNo).getCoordinateX());  
    location.add(data.get(minNo).getCoordinateY());  
    return location;}  
else  
    return null;  
}
```

Σχήμα 5.3: Αλγόριθμος εύρεσης fingerprint

Σημειώνεται ότι, στα πλαίσια της διερεύνησης αυτής, έγινε προσπάθεια δημιουργίας ενός αλγορίθμου ο οποίος να αξιοποιεί ορθά το πλήθος (αριθμό) αντικειμένων και να εντοπίζει τη θέση του χρήστη και το fingerprint του, περιλαμβάνοντας τον αριθμό των φορών που

είδε το αντικείμενο σωστά ή λάθος. Αυτή η προσπάθεια δυστυχώς απέτυχε, λόγω του ότι στον αλγόριθμο που υλοποιήσαμε δεν δινόταν κάποιο βάρος στα όμοια αντικείμενα που εντοπίζονταν και στο ότι ο αριθμός των αντικειμένων εξαρτάτο από το χρόνο της καταγραφής του βίντεο.

Ένα παράδειγμα εκτέλεσης του αλγορίθμου είναι το ακόλουθο. Θεωρούμε ότι έχουμε τα ακόλουθα fingerprints στη βάση όπως φαίνεται στο σχήμα 5.4. Για να είναι πιο κατανοητό το παράδειγμα αντικαθιστούμε τις συντεταγμένες GPS με μικρότερους αριθμούς. Αφού εκτελεστεί ο Navigator, και ολοκληρωθεί η καταγραφή στον περιβάλλοντα χώρο έχει γίνει αναγνώριση ενός ατόμου και ενός ποδηλάτου. Η εκτέλεση του αλγορίθμου, όπως περιγράφεται πιο πάνω, έχει τα ακόλουθα αποτελέσματα. Αρχικά, ο αλγόριθμος θα εφαρμόσει όλα τα fingerprints που βρίσκονται αποθηκευμένα στη βάση και θα υπολογίσει το άθροισμα του κάθε fingerprint με τις αντίστοιχες τιμές του πίνακα που δημιουργήθηκε από τον Navigator. Η εκτέλεση του αθροίσματος για κάθε fingerprint θα έχει τα ακόλουθα αποτελέσματα:

1° Fingerprint (coordinate 7.0) : $\frac{1}{2} * |1 - 1| + \frac{1}{2} * |1 - 0| = 1/2$

2° Fingerprint (coordinate 8.0) : $\frac{1}{2} * |1 - 1| + \frac{1}{2} * |1 - 0| + \frac{1}{2} * |0 - 1| = 1$

3° Fingerprint (coordinate 9.0) : $\frac{1}{2} * |1 - 0| + \frac{1}{2} * |1 - 0| + \frac{1}{2} * |0 - 1| = 1.5$

Ο αλγόριθμος μετά την επιλογή του μικρότερου αθροίσματος θα υποδείξει ότι η τοποθεσία του χρήστη ταυτίζεται με το 1° fingerprint. Άρα θα εμφανίσει πως η τοποθεσία του χρήστη είναι η (7.0, 7.0).

coordinateX	coordinateY	person	bicycle	car	motorbike	aeroplane	bus	train	truc
7.0	7.0	1	0	0	0	0	0	0	
8.0	8.0	1	0	0	0	0	0	0	1
9.0	9.0	0	0	1	0	0	0	0	0
Αποτέλεσμα από Navigator		1	1	0	0	0	0	0	0

Σχήμα 5.4: Παράδειγμα Εισαγωγής Fingerprint και παράδειγμα από Navigator

Το αποτέλεσμα αυτό αποδεικνύεται και με ένα διαφορετικό τρόπο. Βλέποντας το fingerprint παρατηρούμε ότι ο πίνακας που δημιουργήθηκε από τον Navigator έχει ένα κοινό στοιχείο με το 1° fingerprint. Το ίδιο φαίνεται να συμβαίνει και με το 2° fingerprint.

Στην περίπτωση όμως του 2^ο fingerprint έχει ως κοινό στοιχείο τον άνθρωπο. Το ποδήλατο όμως δεν υπάρχει στο 2^ο fingerprint. Ταυτόχρονα το 2^ο fingerprint έχει αναγνωρίσει και το τραίνο, το οποίο δεν περιλαμβάνεται στον πίνακα του Navigator με αποτέλεσμα το fingerprint αυτό να μην είναι αρκετά κοντά στον πίνακα του Navigator. Τέλος, ο πίνακας δεν έχει κανένα κοινό στοιχείο με το 3^ο fingerprint, κάνοντας το άθροισμα αυτό το μεγαλύτερο από τα τρία fingerprints. Σύμφωνα με τα πιο πάνω το πιο κοντινό fingerprint, δηλαδή το fingerprint με το οποίο ο πίνακας έχει τα περισσότερα κοινά στοιχεία, να είναι το 1^ο. Αυτό έχει ήδη αποδειχθεί πιο πάνω με τη χρήση του αλγόριθμου του Navigator. Η διαδικασία δηλαδή αυτή που μόλις περιγράφηκε αποδεικνύει την ορθότητα του αλγορίθμου που υλοποιήθηκε.

Για παράδειγμα, αν κάποιο fingerprint καταγραφόταν αρκετή ώρα, ο αριθμός των αντικειμένων που καταγράφονταν ήταν πιο αυξημένος από τον πραγματικό, ενώ αν ο χρόνος καταγραφής ήταν σύντομος τότε είχε περισσότερες πιθανότητες ο αριθμός να ήταν ο πραγματικός. Στη δεύτερη όμως περίπτωση μπορούσε να μην γίνει αναγνώριση ορισμένων αντικειμένων.

Για αυτό το λόγο θεωρήσαμε ότι θα ήταν πιο σωστό να θέτουμε απλά τον αριθμό 1 στα αντικείμενα που αναγνώρισε το σύστημα 0 σε αυτά που δεν αναγνώρισε αντίστοιχα, και έτσι ο αλγόριθμος να βρίσκει ορθά τη θέση σε περισσότερες περιπτώσεις. Θέτοντας δηλαδή μόνο τους αριθμούς 1 και 0 διευκολύνεται η διαδικασία αφού μπαίνει κάποιο είδους βάρος στα αντικείμενα που αναγνώρισε.

Κεφάλαιο 6

Πειραματική Αξιολόγηση του Συστήματος

6.1	Μέθοδος Αξιολόγησης	48
6.2	Περιγραφή Πειράματος 1: Με Χρήση Τυχαίων Αντικειμένων	49
6.3	Περιγραφή Πειράματος 2: Με Χρήση Φωτογραφιών	53
6.4	Περιγραφή Πειράματος 3: Με Συνδυασμό Αντικειμένων	54
6.5	Συμπεράσματα από την Πειραματική Εφαρμογή	57

Στο συγκεκριμένο κεφάλαιο παρουσιάζεται ο τρόπος αξιολόγησης του συστήματος, για να διαπιστωθεί αν όντως ο τρόπος υλοποίησης και εκτέλεσής του μπορεί να στηρίξει τον αρχικό στόχο που θέσαμε, αν δηλαδή η υλοποιημένη εφαρμογή έχει τη δυνατότητα γεωτοποθέτησης κάποιου χρήστη μέσω του συνδυασμού των δύο υποσυστημάτων Logger και Navigator. Αρχικά, περιγράφεται η μέθοδος που χρησιμοποιήσαμε για την αξιολόγηση, ακολούθως περιγράφεται πιο αναλυτικά η διαδικασία εκτέλεσης των πειραμάτων και τέλος παρουσιάζονται τα συμπεράσματα που απορρέουν από αυτά. Όπως θα αναφερθεί και στη συνέχεια στα πλαίσια της πειραματικής εφαρμογής δεν χρησιμοποιήθηκε το σύστημα GPS λόγω μη ακρίβειας του συστήματος GPS σε εφαρμογές που αφορούν μικρές αποστάσεις, και μη λειτουργίας του σε εσωτερικούς χώρους. Στην πειραματική ανάπτυξη χρησιμοποιήθηκε ένας διαφορετικός αριθμός για τον καθορισμό κάθε θέσης, και με τον τρόπο αυτό ελέγχθηκε η εφαρμογή και η ορθότητα του συστήματος χωρίς τη χρήση υποστηρικτικών συστημάτων.

6.1 Μέθοδος Αξιολόγησης

Τα πειράματα για έλεγχο της ορθότητας και σωστής λειτουργίας του συστήματος έγιναν σε περιορισμένο χώρο όπου οι συντεταγμένες με τη μορφή GPS είχαν διαφορά μόνο στα τρία τελευταία δεκαδικά (από τα 6 δεκαδικά) με αποτέλεσμα να μην είναι εύκολη η διαδικασία διαπίστωσης αν όντως το πρόγραμμα δίνει ορθά αποτελέσματα με τη χρήση

συντεταγμένων με μορφή GPS. Αφού όμως η ιδανική λύση για το συγκεκριμένο σύστημα είναι η μη χρήση συντεταγμένων GPS για τη δημιουργία των διαφόρων fingerprints προχωρήσαμε στην αλλαγή του προγράμματος ώστε να μην χρησιμοποιεί συντεταγμένες με τη μορφή GPS. Για τον ορισμό της θέσης αποφασίσαμε να χρησιμοποιήσουμε ένα αριθμό ο οποίος θα αντιπροσώπευε τη θέση στο κτήριο. Η κάθε θέση σημειώθηκε στην κάτοψη του κτηρίου. Στην εφαρμογή δημιουργήσαμε ένα πλαίσιο εισόδου κειμένου από τον χρήστη όπου ονομάζει τις διάφορες θέσεις. Η εφαρμογή κατά την εκτέλεση του Logger δεν λαμβάνει τις συντεταγμένες της θέσης από τη συσκευή αλλά ο χρήστης εισάγει ένα αριθμό ο οποίος αντιπροσωπεύει τη συγκεκριμένη θέση.

Επιπλέον σημειώνεται ότι, λόγω του ότι στα πλαίσια της παρούσας διπλωματικής εργασίας δεν επιτεύχθηκε η πλήρης εκπαίδευση του YOLO για επιπλέον αντικείμενα, τα πειράματα έπρεπε να γίνουν μέσα στα πλαίσια λειτουργίας του YOLO, με τη χρήση δηλαδή αντικειμένων που περιλαμβάνονται στο COCO Dataset τα οποία παρουσιάζονται αναλυτικά στο Παράρτημα Α.

Τα πειράματα επιλέχθηκε να γίνουν σε χώρο κατοικίας όπου ήταν διαθέσιμος μεγάλος αριθμός αντικειμένων που βρίσκονται διαθέσιμα στη βάση δεδομένων του COCO Dataset που χρησιμοποιεί το YOLO.

Ο τρόπος εκτέλεσης των πειραμάτων περιλάμβανε την τοποθέτηση αντικειμένων σε διάφορα σημεία της κατοικίας, τα οποία είχαν ελεγχθεί προκαταρκτικά ότι τις περισσότερες φορές αναγνωρίζονταν ορθά από το YOLO.

6.2 Περιγραφή Πειράματος 1: Με Χρήση Τυχαίων Αντικειμένων

Αρχικά, εντοπίστηκαν στο χώρο που επιλέχθηκε για πειραματική εφαρμογή του συστήματος διάφορα υφιστάμενα αντικείμενα του χώρου όπως για παράδειγμα μία γλάστρα με λουλούδια, ένα αυτοκίνητο κλπ. και έπειτα τοποθετήθηκαν στο χώρο αυτό νέα αντικείμενα όπως ένα ποδήλατο, ένα ψαλίδι, ένα πληκτρολόγιο ώστε να δημιουργηθούν οι κατάλληλες θέσεις για τα fingerprints. Ακολούθως σχεδιάστηκε ένα απλό διάγραμμα της κάτοψης του χώρου με τις θέσεις των fingerprints και οι θέσεις αυτές αριθμήθηκαν από το 1 μέχρι το 20 (Πίνακας 6.1).

Πίνακας 6.1: Καταγραφή των fingerprints του Πειράματος 1

<i>Position</i>	<i>Sum of Objects</i>	<i>Object 1</i>	<i>Object 2</i>	<i>Object 3</i>	<i>Object 4</i>	<i>Actual Objects</i>
1	4	Tvmonitor	Laptop	Mouse	Keyboard	Correct
2	2	Toilet	Sink			Correct
3	1	Scissors				Correct
4	1	Skateboard				Correct
5	1	Teddy_bear				Correct
6	2	Bicycle	Car			Correct
7	2	Bicycle	Car			Correct
8	1	Car				Correct
9	1	Car				Correct
10	1	Car				Car and Chair
11	1	Chair				Correct
20	1	Chair				Correct
12	1	Car				Correct
13	0	Nothing				Suitcase
13	0	Nothing				Suitcase
14	1	Pottedplant				Correct
15	1	Sportsball				Orange
16	1	Sportsball				Apple
17	1	Keyboard				Correct
18	1	Apple				Correct
20	2	Microwave	Cat			Only Cat

Με την εφαρμογή στο κινητό επιλέχθηκε το Logger και έγινε καταγραφή με βίντεο των αντικειμένων στις θέσεις που βρίσκονταν και στη συνέχεια εισαγωγή των fingerprints στην εφαρμογή. Για επιβεβαίωση της ορθότητας της καταγραφής έγινε ανάγνωση της βάσης δεδομένων από το κινητό και μετατροπή της σε μορφή CSV. Στη μορφή αυτή διαβάστηκε σε φύλλο εργασίας το οποίο χρησιμοποιήθηκε για την ανάλυση των αποτελεσμάτων που βρίσκονται στο Πίνακα 6.1.

Μέσω της πειραματικής αυτής εφαρμογής, διαπιστώθηκε γενικά η ορθότητα της καταγραφής με τον εντοπισμό όμως μερικών διαφορών σε ορισμένα σημεία. Για παράδειγμα στο σημείο 13 η φωτογραφία δεν είχε μεγάλη αντίθεση χρωμάτων και για αυτό η βαλίτσα (suitcase) που είχε τοποθετηθεί στο σημείο αυτό δεν αναγνωρίστηκε. Το εν λόγω σημείο καταγράφηκε δύο φορές αφού την πρώτη φορά διαπιστώθηκε η μη ανίχνευσή του. Το ίδιο όμως διαπιστώθηκε και κατά τη δεύτερη προσπάθεια ανίχνευσής του. Κατά τη διαδικασία έγινε αποθήκευση όλων των fingerprints έστω και αν έχουν το ίδιο όνομα θέσης. Αυτό δεν αποτελεί λάθος του συστήματος αφού για μια θέση μπορεί

να γίνουν δύο καταγραφές από δύο διαφορετικές γωνίες. Για παράδειγμα στο σημείο 10 η καρέκλα βρισκόταν στο βάθος μετά το αυτοκίνητο και δεν την αναγνώρισε το YOLO.

Στο ακόλουθο σχήμα φαίνεται τμήμα από ένα από τα fingerprints, τα οποία δημιουργήθηκαν κατά την εκτέλεση των πειραμάτων στη βάση δεδομένων (Σχήμα 6.1).

coordinateX	coordinateY	person	bicycle	car	motorbike	aeroplane	bus	train	truck	boat	traffic_light	fire_hydrant	stop_sign
6.0	6.0	0	1	1	0	0	0	0	0	0	0	0	0

Σχήμα 6.1: Fingerprint που παράχθηκε κατά την εκτέλεση των πειραμάτων

Στη συνέχεια μετά τη δημιουργία ενός αρκετά ικανοποιητικού αριθμού fingerprints, προχωρήσαμε στον έλεγχο του αλγορίθμου για τον Navigator. Για να ελέγξουμε το υποσύστημα του Navigator, ενεργοποιήσαμε την εφαρμογή στο κινητό, επιλέξαμε Navigator και αρχίσαμε τη λήψη βίντεο στις διάφορες θέσεις μέχρι να επιτευχθεί η αναγνώριση κάποιων αντικειμένων. Σημειώνεται ότι ο χρόνος για το βίντεο δεν είναι σημαντικός επειδή, όπως αναφέρθηκε πιο πάνω, η καταγραφή κάθε αντικειμένου στον Navigator γίνεται μια μόνο φορά κατά τη διάρκεια του βίντεο.

Σε κάθε τοποθεσία γινόταν καταγραφή της θέσης που εντόπιζε ο Navigator, αλλά και της πραγματικής θέσης. Για βοήθεια και μεγαλύτερη αποτελεσματικότητα στον έλεγχο, προσθέσαμε στο πρόγραμμα έναν κώδικα ώστε με την αναγνώριση της θέσης να δείχνει επιπλέον ποια αντικείμενα κατέγραψε. Με αυτήν την μέθοδο μπορούσε να ελεγχθεί άμεσα η ορθότητα λειτουργίας του αλγορίθμου.

Για εξαγωγή συμπερασμάτων για την αποτελεσματικότητα της εφαρμογής, το πείραμα επαναλήφθηκε αρκετές φορές τις οποίες δεν καταγράψαμε. Μετά την απόκτηση επαρκούς εμπειρίας έγινε καταγραφή της τελευταίας φορές διεξαγωγής του πειράματος. Έγινε διέλευση δύο φορές από κάθε θέση για επαλήθευση εφαρμόζοντας τον Navigator μια ή δύο φορές. Τα αποτελέσματα φαίνονται στον πίνακα 6.2.

Πίνακας 6.2: Εύρεση θέσης με το Navigator στο Πείραμα 1

<i>Object 1</i>	<i>Object 2</i>	<i>Object 3</i>	<i>Object 4</i>	<i>Sum of Objects</i>	<i>Actual Position</i>	<i>Recognized Position</i>
TVmonitor	Laptop	Mouse	Keyboard	4	1	1
TVmonitor	Laptop	Mouse	Keyboard	4	1	1
TVmonitor	Laptop	Mouse	Keyboard	4	1	1
Toilet	Sink			2	2	2
Toilet	Sink			2	2	2
Scissors				1	3	3
Scissors				1	3	3
Skateboard				1	4	4
Skateboard				1	4	1
Skateboard				1	4	4
Teddy_bear				1	5	5
Teddy_bear				1	5	5
Bicycle	Car			2	6	1
Bicycle	Car			2	6	6
Bicycle	Car			2	7	1
Bicycle	Car			2	7	8
Car				1	8	8
Car				1	8	8
Car				1	8	8
Car				1	9	8
Car				1	9	8
Car				1	10	8
Car				1	10	8
Chair				1	11	11
Chair				1	11	11
Chair				1	20	13
Car				1	12	8
Car				1	12	8
Nothing				0	13	13
Nothing				0	13	13
Pottedplant				1	14	14
Pottedplant				1	14	14
Pottedplant				1	14	14
Sportsball				1	15	15
Sportsball				1	15	15
Apple				1	18	18
Apple				1	18	18
Keyboard				1	17	17
Keyboard				1	17	17

6.3 Περιγραφή Πειράματος 2: Με Χρήση Φωτογραφιών

Στη συνέχεια και αφού αποκτήθηκε αρκετή εμπειρία από το Πείραμα 1 και μετά τη διόρθωση κάποιων λαθών (bugs) στο πρόγραμμα, που εντοπίστηκαν στη διαδικασία αυτή, προχωρήσαμε στην εκτέλεση των Πειραμάτων 2 και 3.

Στο Πείραμα 2, αποφασίστηκε να χρησιμοποιηθούν φωτογραφίες, οι οποίες επικολλήθηκαν σε θύρες και διάφορα αντικείμενα που θα ήταν εύκολο να αναγνωριστούν. Η επιλογή χρήσης και φωτογραφιών έγινε γιατί αρκετά αντικείμενα που αναγνωρίζει το σύστημα YOLO δεν είναι εύκολο να βρεθούν σε χώρο οικίας άρα χωρίς τη χρήση των φωτογραφιών θα είχαμε πολύ λίγα αντικείμενα για τη διεκπεραίωση των πειραμάτων. Επιπλέον με τον τρόπο αυτό ελέγχθηκε η ορθότητα του συστήματος στην αναγνώριση φωτογραφιών. Χρησιμοποιώντας τον Logger καταγράφηκαν τα αντικείμενα συμπεριλαμβανομένων και των φωτογραφιών από το 1 μέχρι το 10 (Πίνακας 6.3)

Πίνακας 6.3: Καταγραφή των fingerprints του Πειράματος 2

<i>Position</i>	<i>Sum of Objects</i>	<i>Object 1</i>	<i>Object 2</i>	<i>Actual Objects</i>
1	1	Teddy bear real		Correct
2	1	fork real		Correct
3	1	elephant picture		Correct
4	1	zebra picture		Correct
5	2	orange real	book real	Orange on a table
6	1	cat picture		Correct
7	1	person in picture		Person and train in picture
8	1	cup real		Correct
9	1	car picture		Correct
10	1	mouse real		Correct

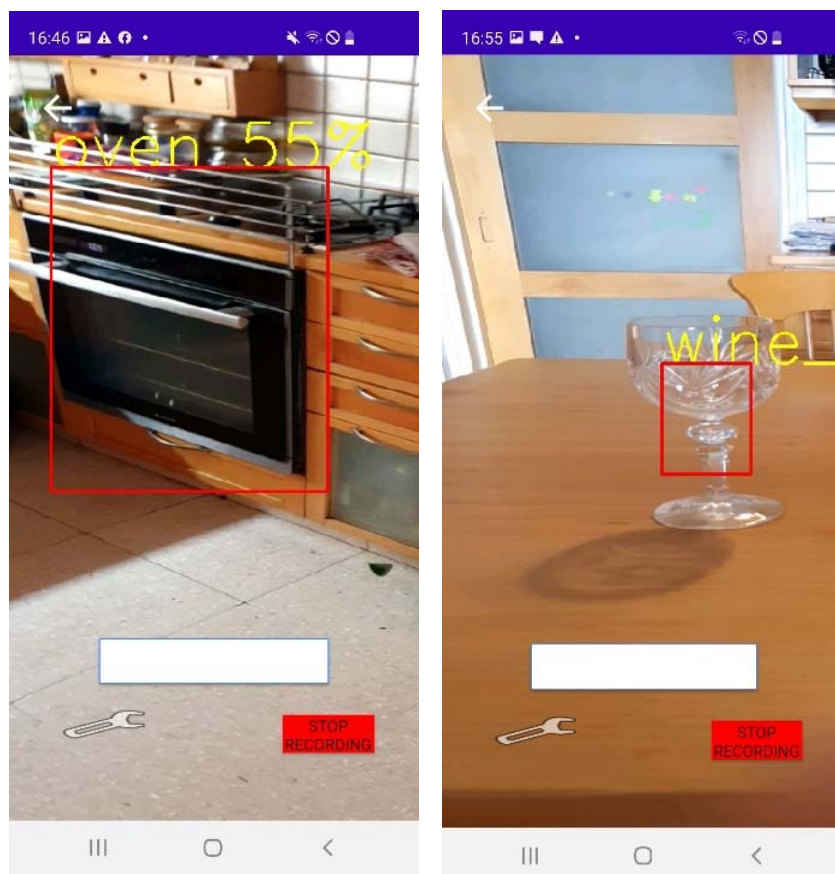
Στο Πείραμα 2, όπως φαίνεται και από τον πίνακα 6.4, η αναγνώριση θέσης ήταν 100%. Η αναγνώριση δηλαδή φωτογραφιών φαίνεται ότι γίνεται με πολύ μεγάλη επιτυχία.

Πίνακας 6.4: Εύρεση θέσης με το Navigator στο Πείραμα 2

<i>Object 2</i>	<i>Object 1</i>	<i>Sum of Objects</i>	<i>Actual Position</i>	<i>Recognized Position</i>
	Teddy bear real	1	1	1
	Teddy bear real	1	1	1
	Teddy bear real	1	1	1
	fork real	1	2	2
	elephant picture	1	3	3
	elephant picture	1	3	3
	elephant picture	1	3	3
	zebra picture	1	4	4
	zebra picture	1	4	4
	zebra picture	1	4	4
book real	orange real	2	5	5
book real	orange real	2	5	5
book real	orange real	2	5	5
	cat picture	1	6	6
	cat picture	1	6	6
	cat picture	1	6	6
	person in picture	1	7	7
	person in picture	1	7	7
	cup real	1	8	8
	cup real	1	8	8
	cup real	1	8	8
	car picture	1	9	9
	car picture	1	9	9
	car picture	1	9	9
	mouse real	1	10	10
	mouse real	1	10	10
	mouse real	1	10	10

6.4 Περιγραφή Πειράματος 3: Με Συνδυασμό Αντικειμένων

Στο σενάριο 3 αποφασίστηκε να χρησιμοποιηθούν αντικείμενα και συνδυασμός αντικειμένων. Χρησιμοποιώντας τον Logger καταγράφηκαν τα αντικείμενα από το 20 μέχρι το 28 ενώ συνέχισαν να υπάρχουν στο σύστημα τα fingerprints από το Πείραμα 1 (Πίνακας 6.3). Τα αντικείμενα αυτά βιντεογραφήθηκαν από κοντά και είχαν πολύ καλή επαναληψιμότητα στην ταυτοποίηση (Σχήμα 6.2).



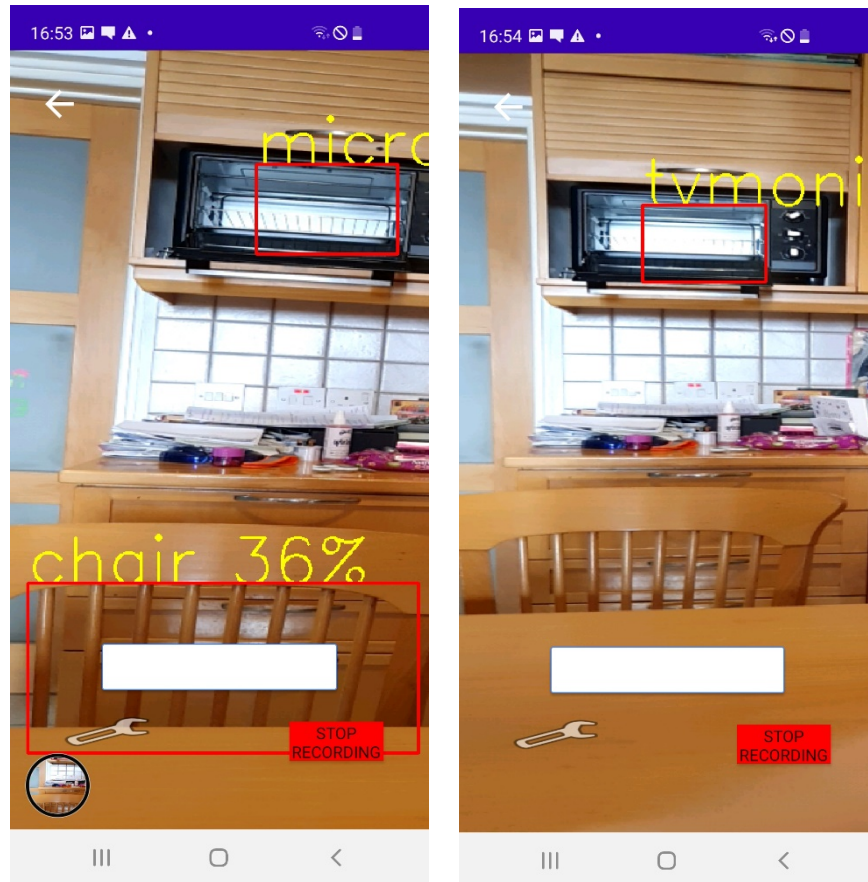
Σχήμα 6.2 Παράδειγμα από fingerprints με καλή επαναληψιμότητα (Fingerprint 28, 22)

Πίνακας 6.5: Καταγραφή των fingerprints του Πειράματος 3

<i>Position</i>	<i>Sum of Objects</i>	<i>Object 1</i>	<i>Object 2</i>	<i>Actual Objects</i>
20	1	bottle		Correct
21	1	wine glass		Correct
22	2	microwave	Chair	Correct
23	2	refrigerator	Oven	Correct
24	1	oven		Correct
25	1	bowl		Correct
26	1	sink		Correct
27	2	sofa	Chair	Correct
28	2	sofa	Chair	Correct

Στη θέση 22 υπήρχε μία καρέκλα και ένα φουρνάκι. Στην αναγνώριση το σύστημα ανίχνευε τρία αντικείμενα αφού σε κάποιες περιπτώσεις το φουρνάκι αναγνωριζόταν ως

microwave και σε άλλες ως tvmonitor (Σχήμα 6.3). Στην πραγματικότητα όμως στο fingerprint βρίσκονταν μόνο δύο αντικείμενα καρέκλα και φουρνάκι. Γινόταν όμως και πάλι αναγνώριση της ορθής θέσης αφού το αποτέλεσμα του αλγορίθμου εύρεσης ορθής θέσης ήταν 1/3, όπως φαίνεται στον Πίνακα 6.6 στη στήλη «Result», ενώ σε οποιαδήποτε άλλη θέση ήταν μεγαλύτερο (βλέπε Σχήμα 5.3).



Σχήμα 6.3 Παράδειγμα από Fingerprint με όχι καλή επαναληψιμότητα (Fingerprint 22)

Στη θέση 28 υπήρχαν δύο αντικείμενα και το σύστημα ανίχνευε μόνο ένα οπότε το αποτέλεσμα του αλγορίθμου ήταν 1. Σε οποιαδήποτε θέση που το αντικείμενο που ανιχνευόταν δεν ταίριαζε με το αντικείμενο της θέσης το αποτέλεσμα ήταν 2. Για αυτό στις περιπτώσεις που ανιχνεύεται το ένα από τα δύο αντικείμενα δίνεται η σωστή ταυτοποίηση θέσης.

Πίνακας 6.6: Εύρεση θέσης με το Navigator στο Πείραμα 3

Objects in fingerprint					Objects recognized						
Object 2	Object 1	Sum	Actual position	Recognized position	Object 1	Object 2	Object 3	Sum objects recognized	Match	Do not match	Result
	bottle	1	20	20	bottle			1	1	0	0.00
	bottle	1	20	20	bottle			1	1	0	0.00
	bottle	1	20	20	bottle	chair		2	1	1	1.00
	wine glass	1	21	21	wine glass			1	1	0	0.00
	wine glass	1	21	21	wine glass			1	1	0	0.00
	wine glass	1	21	21	wine glass			1	1	0	0.00
chair	microwave	2	22	22	tv monitor	chair	Microwave	3	2	1	0.33
chair	microwave	2	22	22	tv monitor	chair	Microwave	3	2	1	0.33
chair	microwave	2	22	22	tv monitor	chair	Microwave	3	2	1	0.33
oven	refrigerator	2	23	23	chair	oven	Refrigerator	3	2	1	0.33
oven	refrigerator	2	23	23	chair	oven	Refrigerator	3	2	1	0.33
oven	refrigerator	2	23	23	chair	oven	Refrigerator	3	2	1	0.33
	oven	1	24	24	oven			1	1	0	0.00
	oven	1	24	24	oven			1	1	0	0.00
	oven	1	24	24	oven			1	1	0	0.00
	bowl	1	25	25	bowl			1	1	0	0.00
	bowl	1	25	25	bowl			1	1	0	0.00
	bowl	1	25	25	bowl			1	1	0	0.00
	sink	1	26	26	sink			1	1	0	0.00
	sink	1	26	26	sink			1	1	0	0.00
	sink	1	26	26	sink			1	1	0	0.00
chair	sofa	2	27	27	chair	sofa		2	2	0	0.00
chair	sofa	2	27	27	chair	sofa		2	2	0	0.00
chair	sofa	2	27	27	chair	sofa		2	2	0	0.00
chair	sofa	2	28	28	sofa			1	1	1	1.00
chair	sofa	2	28	28	sofa			1	1	1	1.00

Σε περίπτωση που δεν ανιχνευόταν αντικείμενο γύριζε στο χρήστη μήνυμα ότι δεν εντοπίστηκε αντικείμενο και δεν προχωρούσε η εκτέλεση του αλγορίθμου.

6.5 Συμπεράσματα από την Πειραματική Εφαρμογή

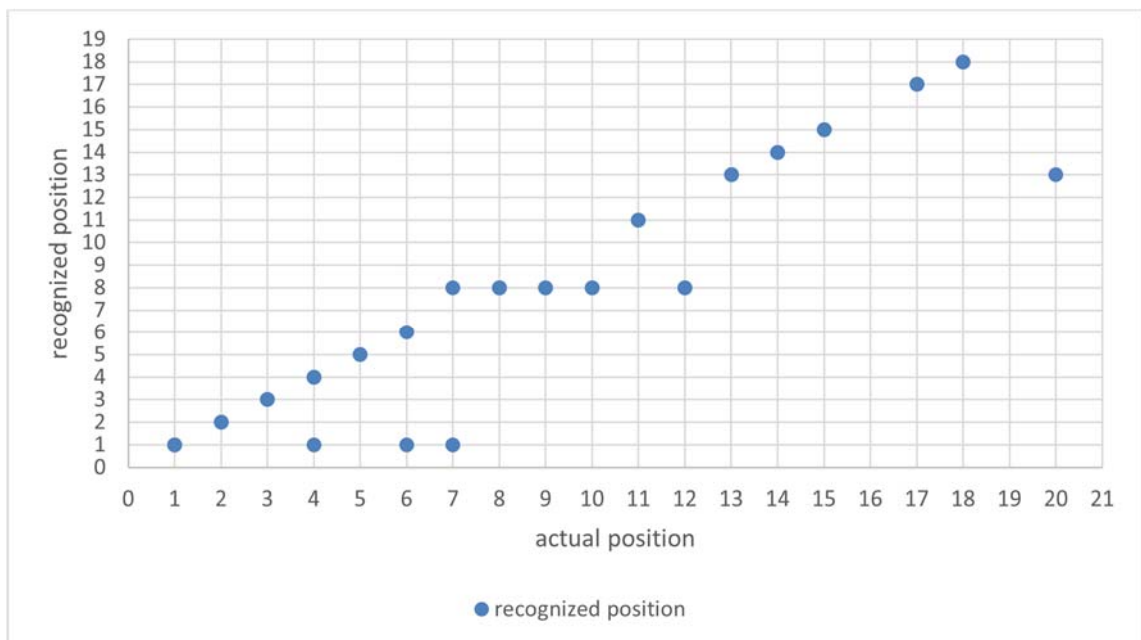
Μετά την ολοκλήρωση των πειραμάτων και την ανάλυση των πληροφοριών προέκυψαν κάποια ενδιαφέροντα συμπεράσματα για τον τρόπο λειτουργίας της εφαρμογής. Καταρχήν διαπιστώθηκε η ορθότητα της εφαρμογής, αφού με τον συνδυασμό της χρήσης Logger και Navigator προκύπτουν τις περισσότερες περιπτώσεις τα σωστά αποτελέσματα, δηλαδή ο Navigator εμφανίζει στον χρήστη τη σωστή του θέση.

Επιπλέον, από την εκτέλεση των πειραμάτων προκύπτει πως υπάρχει αρκετό περιθώριο βελτίωσης της εφαρμογής ειδικά όσον αφορά περιπτώσεις στις οποίες σε διαφορετικές

θέσεις υπάρχουν τα ίδια αντικείμενα αλλά σε διαφορετικό πλήθος. Αν δηλαδή σε μια θέση βρίσκονται δύο αυτοκίνητα και σε μια άλλη θέση βρίσκεται ένα αυτοκίνητο, η συγκεκριμένη υλοποίηση δεν θα αναγνωρίσει σωστά τη θέση στην οποία βρισκόταν ο χρήστης. Θα εμφανίσει δηλαδή πως βρίσκεται στη θέση η οποία πρώτη εισάχθηκε από τον Logger στη βάση δεδομένων, η οποία αντιστοιχεί με τη θέση αναγνώρισης του πρώτου αυτοκινήτου που εισάχθηκε από τον Logger στη βάση δεδομένων.

Η γραφική παράσταση του σχήματος 6.2. στον άξονα x παρουσιάζει την πραγματική θέση που βρισκόταν ο χρήστης και αναμέναμε να εμφανίσει ο Navigator ενώ στον άξονα y φαίνεται η θέση που εμφάνισε ο Navigator στον χρήστη.

Μια σημαντική παρατήρηση που έγινε από τη μελέτη της γραφικής παράστασης (Σχήμα 6.4) αφορά τα fingerprints 8, 9 και 10 από το πίνακα 6.2. Τα εν λόγω fingerprints έχουν ως καταγραφή μόνο ένα αντικείμενο, ένα αυτοκίνητο. Για αυτό και το σύστημα αναγνώριζε ότι και τα τρία fingerprints βρίσκονταν στη θέση 8, την πρώτη δηλαδή θέση με αυτοκίνητο. Από αυτό φαίνεται πως η εφαρμογή όπως έχει κτιστεί δεν μπορεί να εφαρμοστεί για κτίρια με επαναλαμβανόμενα στοιχεία αλλά είναι ιδανική σε κτήρια με διαφορετικά στοιχεία σε διάφορα σημεία του κτηρίου.



Σχήμα 6.4: Γραφική παράσταση από πειράματα όπου φαίνεται η συσχέτιση της πραγματικής θέσης των αντικειμένων και της θέσης αναγνώρισης από την εφαρμογή

Μια ακόμη παρατήρηση που προκύπτει από τα πειράματα είναι πως στο YOLO, ορισμένες φορές γίνονται περισσότερα λάθη στην αναγνώριση αντικειμένων από ότι γίνονται από το ανθρώπινο μάτι, για αυτό χρειάζεται προσοχή στον φωτισμό και την γωνία από την οποία θα βγει το βίντεο. Εδώ να σημειωθεί πως η χρήση βίντεο και όχι φωτογραφίας αποτελεί πλεονέκτημα αφού έτσι ο χρήστης έχει τη δυνατότητα να αλλάζει γωνία και θέση μέχρι η εφαρμογή να αναγνωρίσει κάποια αντικείμενα. Η λάθος αναγνώριση αντικειμένων έχει ως αποτέλεσμα είτε τη δημιουργία λανθασμένων fingerprints είτε τη μη αναγνώριση της σωστής θέσης του χρήστη.

Σε θέσεις που υπάρχουν περισσότερα από ένα αντικείμενα, και γίνεται ανίχνευση μερικών από αυτών των αντικειμένων, ο αλγόριθμός δίνει πολύ καλύτερα αποτελέσματα παρά στην περίπτωση που ανιχνεύεται ένα αντικείμενο και δεν είναι το σωστό. Η συλλογή περισσότερων αντικειμένων σε ένα fingerprint δεν χρειάζεται ειδική διαμόρφωση του χώρου αλλά δημιουργεί πρακτικό πρόβλημα επειδή ο χρήστης που χρησιμοποιεί το Navigation μπορεί να βρίσκεται σε οποιαδήποτε θέση του κτηρίου και να τραβά βίντεο από οποιαδήποτε γωνία. Για αυτό δημιουργούνται περισσότερες απαιτήσεις από τον Logger και απαιτούνται περισσότερα fingerprints.

Κεφάλαιο 7

Συμπεράσματα και Μελλοντικές Επεκτάσεις

7.1	Συμπεράσματα	60
7.2	Μελλοντικές Επεκτάσεις	61

7.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία αναπτύξαμε ένα σύστημα γεωτοποθέτησης με χρήση της τεχνολογίας Μηχανικής Όρασης. Προσπαθήσαμε να δώσουμε λύση σε ένα πρόβλημα που υπάρχει έντονα σήμερα, δηλαδή τον εντοπισμό της θέσης ενός χρήστη στο εσωτερικό ενός κτηρίου, αφού ο άνθρωπος σήμερα αναλώνει μεγαλύτερο ποσοστό του χρόνου του σε εσωτερικούς παρά εξωτερικούς χώρου. Ενώ δηλαδή έχουν γίνει πολλές μελέτες και έχουν βρεθεί λύσεις για τον εντοπισμό της θέσης ενός χρήστη σε εξωτερικό χώρο (GPS κλπ.), το θέμα αναζήτησης θέσεων σε εσωτερικά περιβάλλοντα παραμένει λιγότερο μελετημένο.

Η εφαρμογή αυτή κατά την έναρξη της υλοποίησης έθεσε ως πρωταρχικό ερώτημα τη δυνατότητα ή μη γεωτοποθέτησης μέσα σε εσωτερικό χώρο, χωρίς τη χρήση υφιστάμενων συστημάτων, αλλά απλά εντοπίζοντας και αναγνωρίζοντας διάφορα αντικείμενα που βρίσκονται στον υπό μελέτη χώρο.

Με την υλοποίηση της εφαρμογής επαληθεύεται η υπόθεση αυτή. Όπως αναφέρεται εκτενώς στα προηγούμενα κεφάλαια, καταφέραμε με την υλοποίηση των δύο υποσυστημάτων Logger και Navigator και σε συνδυασμό με τη χρήση του συστήματος YOLO, το οποίο χρησιμοποιείται για την αναγνώριση των διαφόρων αντικειμένων, να αναπτύξουμε ένα σύστημα αναγνώρισης της θέσης που βρίσκεται ο χρήστης, επιστρέφοντάς του τις συντεταγμένες GPS της τρέχουσάς του θέσης.

Κατά την πειραματική εφαρμογή του συστήματος εξήχθησαν ενδιαφέροντα συμπεράσματα και διαφάνηκε ότι η εν λόγω εφαρμογή μπορεί να υλοποιηθεί χωρίς

κανένα υποστηρικτικό σύστημα, στοιχείο πολύ σημαντικό για το θέμα πλοήγησης σε εσωτερικούς χώρους και γενικότερα σε χώρους που δεν υπάρχουν διαθέσιμα άλλα συστήματα γεωτοποθέτησης (GPS, Wi-Fi κλπ.).

Παρόλα τα προβλήματα που επισημάνθηκαν και αναλύθηκαν πιο πάνω (δυσκολία εντοπισμού όμοιων αντικειμένων, προβλήματα στο φωτισμό κάποιων αντικειμένων καθώς και στη γωνία λήψης εικόνων κλπ.) η αποτελεσματικότητα της εφαρμογής κρίνεται πολύ ικανοποιητική στην παρούσα αυτή φάση και μπορεί σίγουρα να αποτελέσει βάση για περαιτέρω διερεύνηση με επίλυση των προβλημάτων αυτών.

Με την ολοκλήρωση της παρούσας διπλωματικής μπορούμε να συμπεράνουμε ότι ανοίγεται ένα πεδίο έρευνας όσον αφορά τη γεωτοποθέτηση σε εσωτερικούς χώρους, η οποία μπορεί να οδηγήσει μελλοντικά σε πολύ ενδιαφέρουσες και εύκολες σε χρήση εφαρμογές με τη βελτίωση ίσως και περαιτέρω ανάπτυξη του συστήματος που ήδη περιγράφηκε.

7.2 Μελλοντικές Επεκτάσεις

Το σύστημα το οποίο υλοποιήσαμε, όπως ήδη αναφέρθηκε, έφτασε σε ένα αρκετά ικανοποιητικό σημείο επεξεργασίας και απόδοσης αλλά έχει αρκετά μεγάλο περιθώριο για βελτίωση και περαιτέρω ανάπτυξη.

Ένα πρόβλημα της συγκεκριμένης εφαρμογής είναι πως ενώ βασικός στόχος της είναι η γεωτοποθέτηση σε εσωτερικούς χώρους, κατά την πρώτη υλοποίησή της χρησιμοποιήσαμε την τεχνολογία του GPS. Στη συνέχεια όμως κατά την πειραματική εφαρμογή το βελτιώσαμε ώστε ο χρήστης να εισάγει μόνο την ονομασία της θέσης του.

Μια μελλοντική επέκταση / αναβάθμιση του συστήματος μπορεί να επιτευχθεί κατά τη διαδικασία του Logging. Αντί να λαμβάνονται από τον χρήστη οι συντεταγμένες του και το fingerprint να εισάγεται στη βάση δεδομένων, μπορεί να εμφανίζεται στην οθόνη ένα διάγραμμα της κάτοψης του κτηρίου στο οποίο βρίσκεται ο χρήστης. Ο χρήστης θα μπορεί έτσι να επιλέγει στην κάτοψη αυτή τη θέση στην οποία βρίσκεται, και εκείνη να εισάγεται στη βάση δεδομένων ως η θέση του. Ακολούθως, κατά τη χρήση του Navigator, αντί να εμφανίζονται στο χρήστη οι συντεταγμένες της θέσης του να εμφανίζεται η κάτοψη του

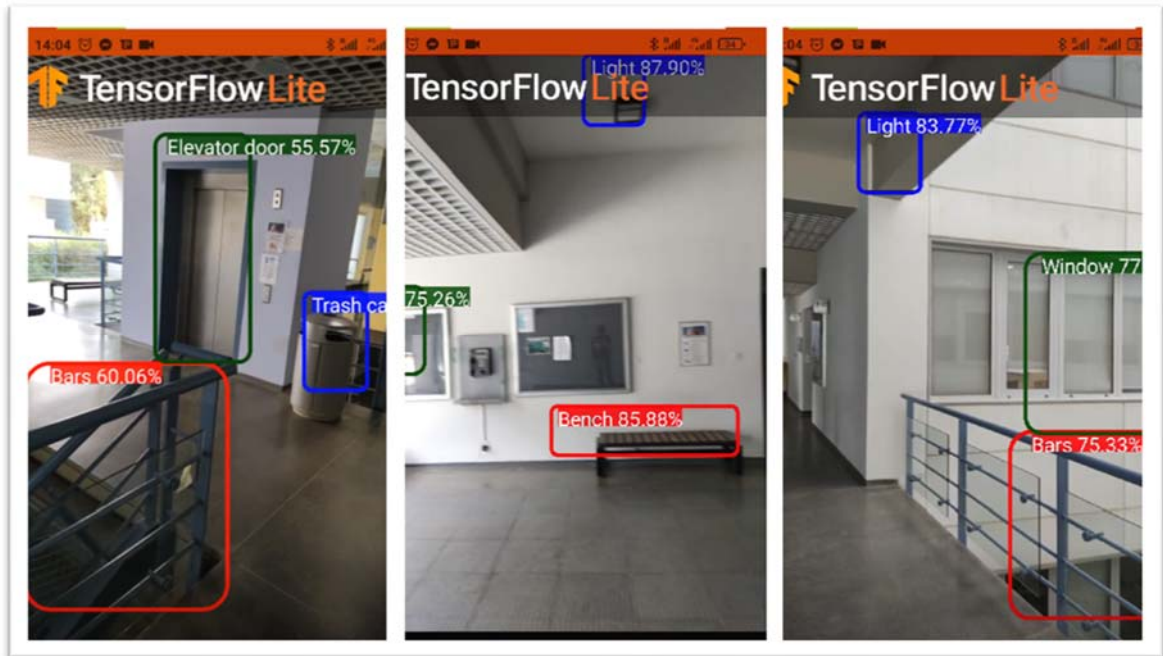
κτηρίου με σημειωμένη τη θέση στην οποία βρίσκεται, όπως είχε αρχικά επιλεγεί από τον χρήστη που δημιούργησε το αντίστοιχο fingerprint.

Ένα επίσης μεγάλο πρόβλημα της συγκεκριμένης εφαρμογής, που ήδη αναφέρθηκε, είναι η αναγνώριση των αντικειμένων όταν αυτά εμφανίζονται για δεύτερη φορά. Συγκεκριμένα το YOLO δεν μπορεί να καταλάβει πότε (τη χρονική στιγμή) αναγνώρισε ένα αντικείμενο, με αποτέλεσμα σε αρκετές περιπτώσεις να θεωρεί ως διαφορετικό αντικείμενο το ίδιο αντικείμενο, αυξάνοντας τον αριθμό των φορών που εντόπισε ένα αντικείμενο. Έτσι στις περιπτώσεις κατά τις οποίες η μόνη διαφορά δύο θέσεων είναι το πλήθος των αντικειμένων, η συγκεκριμένη υλοποίηση δεν μπορεί να αναγνωρίσει ορθά σε ποια θέση βρίσκεται ο χρήστης.

Ένα ακόμη πρόβλημα σχετίζεται με την αναγνώριση από το YOLO παρόμοιων αντικειμένων. Λόγω της εκπαίδευσης του YOLO με το COCO Dataset αυτό αναγνωρίζει αρκετά αντικείμενα τα οποία είναι παρόμοια μεταξύ τους. Σε αρκετές όμως περιπτώσεις μπορεί να προκύψει λάθος αναγνώριση για παρόμοια αντικείμενα, όπως αναγνώριση μιας μπουκάλας ως ποτήρι ή ενός ποδηλάτου ως μοτοσικλέτα. Αυτό οφείλεται στην ανάγνωση των αντικειμένων από μια συσκευή η οποία φαίνεται λιγότερη αποτελεσματική σε σχέση με το ανθρώπινο μάτι. Επιπλέον, σε κάποιες περιπτώσεις μπορεί η εφαρμογή να εντοπίσει αντικείμενα τα οποία στην πραγματικότητα δεν υπάρχουν στη δεδομένη θέση. Μπορεί για παράδειγμα, η εφαρμογή να αναγνωρίσει σε ένα χώρο παγκάκια λόγω της σκίασης του χώρου χωρίς στην πραγματικότητα να υφίστανται οι κατασκευές αυτές στη δεδομένη θέση. Για αυτό σε μια σημαντική μελλοντική επέκταση / βελτίωση του συστήματος θα ήταν σκόπιμο κατά τη διαδικασία καταγραφής του περιβάλλοντα χώρου, είτε στη φάση του Logging είτε στη φάση του Navigation, μετά το τέλος της καταγραφής να εμφανίζεται μια λίστα από τα αντικείμενα τα οποία αναγνώρισε το σύστημα και να μπορεί ο χρήστης είτε να διορθώσει κάποια αντικείμενα που αναγνώρισε λανθασμένα, είτε να διαγράψει κάποια αντικείμενα. Η διόρθωση των αναγνωρισμένων αντικειμένων μπορεί να πραγματοποιηθεί κατά την εκτέλεση του αλγορίθμου για εύρεση της ακριβούς θέσης του χρήστη.

Επιπλέον, στα πλαίσια αυτής της διπλωματικής εργασίας δεν μπόρεσε να πραγματοποιηθεί με επιτυχία η εκπαίδευση του συστήματος στο περιβάλλον του πανεπιστημίου κυρίως λόγω έλλειψης υλικής υποδομής, χρόνου και υπολογιστικής δύναμης στο συγκεκριμένο χρονικό διάστημα. Για αυτό δεν συμπεριλήφθηκε στη

συγκεκριμένη διπλωματική η εν λόγω εκπαίδευση. Παρ' όλα αυτά κατά την υλοποίηση διαπιστώσαμε πως με την κατάλληλη υλική υποδομή η εκπαίδευση στο χώρο του πανεπιστημίου είναι δυνατή όπως φαίνεται και στο σχήμα 7.1



Σχήμα 7.1 Αποτελέσματα εκπαίδευσης στο χώρο του πανεπιστημίου για αναγνώριση αντικειμένων

Τέλος, ένα πρόβλημα ενός συστήματος που στηρίζεται στη χρήση Μηχανικής Όρασης είναι η αποτυχία διαχωρισμού διαφορετικών θέσεων όταν αυτές έχουν τα ίδια ακριβώς αντικείμενα. Αυτό οφείλεται στον τρόπο λειτουργίας του συστήματος αφού αυτός στηρίζεται στην αναγνώριση αντικειμένων του περιβάλλοντα χώρου. Αν δύο θέσεις έχουν ακριβώς τα ίδια αντικείμενα τότε το σύστημα δεν μπορεί να τις ξεχωρίσει σαν δύο διαφορετικές περιοχές, και εμφανίζει στον χρήστη ως λανθασμένη θέση. Για παράδειγμα στο κτήριο του Τμήματος Πληροφορικής οι δύο ανελκυστήρες του 1^{ου} ορόφου και ο περιβάλλοντάς τους χώρος είναι πανομοιότυποι. Αν δηλαδή καταγραφεί το περιβάλλον των δύο ανελκυστήρων το σύστημα θα εντοπίσει και στις δύο περιπτώσεις ένα ανελκυστήρα, σκάλες και ένα κάδο απορριμμάτων με αποτέλεσμα να μην καταφέρει να ξεχωρίσει τις δύο περιοχές. Το συγκεκριμένο πρόβλημα, θα μπορούσε να επιλυθεί χρησιμοποιώντας μόνο θέσεις οι οποίες έχουν κάτι διαφορετικό στην κάθε περίπτωση. Μια άλλη λύση για μείωση των περιπτώσεων λάθους είναι κατά την εκτέλεση του Navigator μέσω της χρήσης κάποιας λειτουργίας του κινητού να εντοπίζεται σε ποιο

κτήριο βρίσκεται ο χρήστης τη συγκεκριμένη στιγμή, και ο αλγόριθμος να εκτελείται μόνο για fingerprints τα οποία καταγράφηκαν στο συγκεκριμένο κτήριο. Προτείνεται δηλαδή να γίνεται ταξινόμηση των fingerprints ανά κτήριο, έτσι ώστε ο αλγόριθμος να εκτελείται στα fingerprints του ιδίου του κτηρίου και όχι σε όλα τα fingerprints που είναι καταχωρημένα στη βάση δεδομένων.

Βιβλιογραφία

1. D. Zeinalipour-Yazti, C. Laoudias, K. Georgiou, and G. Chatzimilioudis, “Internet-based indoor navigation services,” *IEEE Internet Computing*, vol. 21, no. 4, pp. 54–63, 2017.
2. Aislelabs, “The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs” Available at: <http://goo.gl/eCsUp9>, May 4, 2015
3. B. Ferris, D. Fox and N. Lawrence, “WiFi-SLAM using Gaussian process latent variable model”, in *IJCAI* 2007.
4. C.-L. Li, C. Laoudias, G. Larkou, Y.-K. Tsai, D. Zeinalipour-Yazti, and C.G. Panayiotou, “Indoor geolocation on multi-sensor smartphones”, In *ACM MobiSys* 2013.
5. Z. Yang, C. Wu, Z. Xinglin, X. Wang, and Y. Liu, “Mobility increases localizability: A survey on wireless indoor localization using inertial sensors,” In *ACM Computing Surveys*, vol. 47, no. 3, pp. 134, 2015.
6. P. Bahl, and V.N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system” In *IEEE INFOCOM* 2000.
7. D. Han, S.H. Jung, M. Lee and G. Yoon, “Building a practical Wi-Fi-based indoor navigation system”, In *IEEE Perv. Comp.*, vol. 13, no. 2, pp. 72-79, 2014.
8. J. Ledlie, et. al., “Mol’ e: a scalable, user-generated WiFi positioning engine”, *Journal of Loc. Based Serv.*, vol. 6, no. 2, pp. 55-80, 2012.
9. G. Chatzimilioudis, A. Konstantinidis, C. Laoudias and D. ZeinalipourYazti, “Crowdsourcing with smartphones”, In *IEEE Internet Computing*, vol. 16, no. 5, pp. 36-44, 2012.
10. S. He, S.-H.G. Chan, “Wi-Fi Fingerprint-based Indoor Positioning: Recent Advances and Comparisons”, *IEEE Communications Surveys & Tutorials*, 2015
11. A. LaMarca, et. al., *Place Lab: Device positioning using radio beacons in the wild*, In *Pervasive Computing*, Springer, pp. 116-133, 2005.

12. C. Laoudias, G. Constantinou, M. Constantinides, S. Nicolaou, D. Zeinalipour-Yazti and C.G. Panayiotou, "The Airplace Indoor Positioning Platform for Android Smartphones", In IEEE MDM 2012.
13. R. Hansen, B. Thomsen, L.L. Thomsen and F.S. Adamsen, "Smart CampusAAU: an open platform enabling indoor positioning and navigation, In IEEE MDM 2013
14. A. Konstantinidis, G. Chatzimilioudis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, Y. Theodoridis, "Privacy-Preserving Indoor Localization on Smartphones", In IEEE Trans. Know. Dat. Engr., Vol. 27, Iss. 11, pp. 3042-3055, 2015
15. J. Redmon and A. Farhadi "YOLOv3: An Incremental Improvement", University of Washington, online 2020 at <https://pjreddie.com/yolo/>.
16. "YOLOv3: Real-Time Object Detection Algorithm" <https://viso.ai/deep-learning/yolov3-overview/>.
17. C. Kumar B., R. Punitha and Mohana, "YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications," *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1316-1321, Doi: 10.1109/ICSSIT48917.2020.9214094, 2020.
18. Sujin Luo, Chenyu Xu, and Hongxin Li. "An Application of Object Detection Based on YOLOv3 in Traffic", In *Proceedings of the 2019 International Conference on Image, Video and Signal Processing (IVSP 2019)*. Association for Computing Machinery, New York, NY, USA, 68–72. DOI: <https://doi.org/10.1145/3317640.3317657>, 2019.
19. P. Mpeis, T. Roussel, M. Kumar, C. Costa, C. Laoudias, D. Capot-Ray, D. Zeinalipour-Yazti, "The Anyplace 4.0 IoT Localization Architecture" *Proceedings of the 21st IEEE International Conference on Mobile Data Management (MDM '20)*, IEEE Computer Society, ISBN: pp. 8, June 30 - July 3, 2020, Versailles, France, 2020.
20. D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proceedings of the 14th international conference on information processing in sensor networks*. ACM, 2015, pp. 178–189.

21. C.S. Jensen, H. Lu, B. Yang, "Indoor - A New Data Management Frontier," in IEEE Data Eng. Bull vol. 33, iss. 2, pp. 12–17, 2010.
22. A. Konstantinidis, P. Irakleous, Z. Georgiou, D. Zeinalipour-Yazti, and P. K. Chrysanthis, "Iot data prefetching in indoor navigation SOAs," ACM Transactions on Internet Technology (TOIT), vol. 19, no. 1, pp. 1–21, 2018.
23. P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless LANs". In Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2 (2002), 720–724, 2002
24. D. Zeinalipour-Yazti and C. Laoudias, "The anatomy of the anyplace indoor navigation service," SIGSPATIAL Special, vol. 9, no. 2, pp. 3– 10, 2017.
25. "MazeMap Homepage" <https://www.mazemap.com/>
26. "MazeMap: Intreractive Indoor Maps for Large Establishments" <https://api.cioapplications.com/vendor/mazemap-interactive-indoor-maps-for-large-establishments-cid-2370-mid-129.html>
27. G. Biczók, S. Díez Martínez, T. Jelle and J. Krogstie, "Navigating MazeMap: Indoor human mobility, spatio-logical ties and future potential," *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, 2014, pp. 266-271, doi: 10.1109/PerComW.2014.6815215.

ΠΑΡΑΡΤΗΜΑ Α

Πίνακας Α1

Κατάλογος από αντικείμενα τα οποία συμπεριλαμβάνονται στο COCO Dataset

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

Σημείωση: Ομαδοποίηση αντικειμένων σε αυτά που α) συσχετίζονται με το δρόμο (μέσα συγκοινωνίας και αστικός εξοπλισμός), β) ζώα, γ) αντικείμενα (συμπεριλαμβανομένων αντικειμένων αθλητικού εξοπλισμού), δ) οικιακά σκεύη (ποτήρια, πιάτα κλπ), ε) φρούτα και τέλος στ) άλλα αντικείμενα της οικίας (επίπλωση και μικρότερα αντικείμενα)