



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναγνώριση Προτύπων με χρήση Μη Επανδρωμένου Αεροσκάφους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ

Νικόλαου Τζαγκαράκη

Επιβλέπουσα: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Αθήνα, Ιούνιος 2019



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναγνώριση Προτύπων με χρήση Μη Επανδρωμένου Αεροσκάφους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ

Νικόλαου Τζαγκαράκη

Επιβλέπουσα: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Ιουνίου 2019.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

Θεοδώρα Βαρβαρίγου

.....

Εμμανουήλ Βαρβαρίγος

.....

Συμεών Παπαβασιλείου

Αθήνα, Ιούνιος 2019

.....
Νικόλαος Τζαγκαράκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Νικόλαος Τζαγκαράκης, 2019.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Η ιλιγγιώδης εξέλιξη της τεχνολογίας έχει ως αποτέλεσμα, την εύκολη πρόσβαση σε τεχνολογίες που παλαιότερα δεν ήταν διαθέσιμες σε απλούς χρήστες, αλλά παρά μόνο σε κυβερνητικές οργανώσεις ή σε μεγάλες εταιρίες. Μια τέτοια περίπτωση είναι και τα μη επανδρωμένα αεροσκάφη (Unmanned Aerial Vehicles – UAVs), ή όπως ευρέως αποκαλούνται “drones”. Αρχικά η χρήση τους αφορούσε καθαρά στρατιωτικές εφαρμογές (αναγνωριστικούς, κατασκοπευτικούς ή και για μεταφορά βομβών) ή για ερασιτεχνικούς και ερευνητικούς σκοπούς, όμως με την πάροδο του χρόνου έχει αρχίσει να αξιοποιείται και η ειρηνική του πλευρά όπως μεταφορά φαρμάκων, βιολογικών υλικών για ιατρικές εξετάσεις και τροφίμων από και προς δύσβατες περιοχές. Λόγω της ραγδαίας εξέλιξης τους έχει μειωθεί αρκετά το κόστος παραγωγής τους, και αυτό έχει ως αποτέλεσμα να μπορούν να αποκτηθούν και από απλούς πολίτες.

Η χρήση τους μπορεί να αποδειχθεί αρκετά σημαντική, ιδιαίτερα στις περιπτώσεις όπου σκοπός της εφαρμογής τους είναι η αναγνώριση καταστάσεων που εμπλέκονται ανθρώπινες ζωές, στοχεύοντας στην αποτροπή δυσάρεστων γεγονότων. Παραδείγματος χάριν σε περιπτώσεις φυσικών καταστροφών, κρίνεται αναγκαίο να γνωρίζουμε άμεσα πόσοι άνθρωποι βρίσκονται στις συγκεκριμένες περιοχές, και να είναι εύκολη και γρήγορη η σάρωση μεγάλων εκτάσεων. Εδώ μπορούν εύκολα να μας βοηθήσουν τα μη επανδρωμένα αεροσκάφη, όπου σαρώνοντας περιοχές, και φωτογραφίζοντας αυτές, μπορεί εύκολα να αναγνωριστεί η κρισιμότητα της κατάστασης, με σκοπό την αποτελεσματικότερη αντιμετώπισης της.

Στην συγκεκριμένη εργασία αναπτύσσεται μια εφαρμογή για έξυπνες συσκευές με λειτουργικό Android, όπου αφορά μη επανδρωμένα αεροσκάφη της εταιρείας DJI, η οποία προσφέρει και το απαραίτητο SDK για την δημιουργία των εφαρμογών. Σκοπός της εφαρμογής είναι η σάρωση περιοχών που επιλέγονται μέσω των χαρτών της google, και η αναγνώριση ανθρώπων καθώς και διαφόρων αντικειμένων σε αυτές χρησιμοποιώντας το λογισμικό opencv.

Λέξεις Κλειδιά: αναγνώριση προτύπων, με επανδρωμένα αεροσκάφη, έξυπνες κινητές συσκευές, χάρτες google, Android studio, μηχανική μάθηση

Abstract

The turbulent development of technology has resulted in easy access to technologies that were previously unavailable to ordinary users, but only to government organizations or large corporations. Unmanned Aerial Vehicles (UAVs), or as commonly called "drones", are such a case. Initially, their use was purely military applications (identifying, spying or even transporting bombs) or for amateur and research purposes; however, over the years, its peaceful side has also begun to be exploited, such as transport of medicines, biological materials for medical examinations, and food from and to inaccessible areas. Due to their rapid development, their production costs have been reduced considerably, and this can also be achieved by ordinary citizens.

Their use may prove to be quite significant, especially when their purpose is to recognize situations involving human lives, aiming at preventing unpleasant events. For example, in the case of natural disasters, it is necessary to know directly how many people are in these areas and to scan large areas easily and quickly. Here, unmanned aircrafts can help, by scanning these areas, and photographing them, can easily recognize the criticality of the situation in order to tackle it more effectively.

In this thesis, an application for smart Android-powered devices is deployed, involving unmanned aircraft from DJI, which also provides the necessary SDK for creating applications. The purpose of the application is to scan areas selected through google maps, and to identify people and various objects in them.

Keywords: Pattern Recognition, Manned Aircraft, Smart Mobile Devices, google maps, Android studio, machine learning

Ευχαριστίες

Η διπλωματική εργασία αυτή αποτελεί το τελευταίο στάδιο της φοίτησής μου στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου και αποτελεί για εμένα το πρώτο έργο ακαδημαϊκής έρευνας.

Ευχαριστώ ιδιαίτερα την καθηγήτρια Θεοδώρα Βαρβαρίγου για την ευκαιρία που μου έδωσε να εμβαθύνω σε ζητήματα αναγνώρισης προτύπων με χρήση μη επανδρωμένων αεροσκαφών.

Επίσης, ευχαριστώ τον Τάσο Νικολακόπουλο για την βοήθεια και την καθοδήγησή του καθ' όλη τη διάρκεια της εκπόνησης της εργασίας. Τέλος, ευχαριστώ όλους όσους ήταν δίπλα μου κατά τα δημιουργικά αυτά χρόνια.

Πίνακας Περιεχομένων

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Πίνακας Περιεχομένων	10
Κατάλογος σχημάτων	12
1. Εισαγωγή	14
1.1 Unmanned Aerial Vehicles	14
1.2 OpenCV	16
1.3 Αντικείμενο της Διπλωματικής	17
2.Σχετικές Εφαρμογές	19
2.1 Real-time People and Vehicle Detection from UAV Imagery	19
2.2 Embedded Real-Time Object Detection for a UAV Warning System	20
2.3 A fast object detector based on high-order gradients and Gaussian process regression for UAV images	23
2.4 DroneSAR - Search & Rescue	24
2.5 Nightingale Security	25
3.Object detection	26
3.1 Machine learning	26
3.2 Μοντέλα αναγνώρισης αντικειμένων	27
3.2.1 R-CNN	28
3.2.2 Fast R-CNN	29
3.2.3 Faster R-CNN	30
3.2.4 Single Shot MultiBox Detector	30
3.2.5 YOLO	31
4.Σχετικά με την εφαρμογή	32
4.1 Λειτουργία εφαρμογής	33
4.2 Χρήσεις Εφαρμογής	37
5.Κώδικας και Γραφικό Περιβάλλον	40
5.1 Υλοποίηση Εφαρμογής	40
5.1.1 Android App	40
5.1.2 Receiver class	48
5.1.3 Object detection algorithm	48

5.2 Γραφικό περιβάλλον	51
5.2.1 Περιήγηση	51
5.2.2 Ανάλυση widgets	53
5.2.3 Ανάλυση ρυθμίσεων αποστολής	57
6.Μετρήσεις και αποτελέσματα	59
6.1 Μετρήσεις	59
6.1.1 Εντοπισμός ατόμων	59
6.1.2 Εντοπισμός οχημάτων	62
6.1.3 Χρόνος εκτέλεσης αλγορίθμου	64
6.1.4 Σχολιασμός αποτελεσμάτων	65
7.Συμπεράσματα και μελλοντικές επεκτάσεις	67
Βιβλιογραφία	69

Κατάλογος σχημάτων

1.1.1 Κατηγορίες μη επανδρωμένων	15
2.1.1 Αρχιτεκτονική συστήματος υποστήριξης αποφάσεων βασισμένου σε AUV για απόκριση έκτακτης ανάγκης	20
2.2.2 Σύγκριση ανάλυσης εικόνας εισόδου και ακρίβειας μοντέλου	21
2.2.3 Επίδραση ανάλυσης εικόνας στον ρυθμό επεξεργασίας	22
2.4.1 Εκτέλεση αποστολής σε Drone Sar	24
2.5.1 Σταθμός βάσης	25
3.2.1.1 Αρχιτεκτονική μοντέλου R-CNN	28
3.2.2.1 Αρχιτεκτονική μοντέλου Fast R-CNN	28
3.2.3.1 Αρχιτεκτονική μοντέλου Faster R-CNN	29
3.2.4.1 Αρχιτεκτονική μοντέλου SSD	30
3.2.5.1 Αρχιτεκτονική μοντέλου YOLO	30
4.1.1 Περιγραφή λειτουργίας εφαρμογής	33
4.1.2 Σύγκριση μοντέλων αναγνώρισης	34
4.1.3 Λειτουργία gimbal	35
4.1.4 Απόσταση κάμερας UAV από αντικείμενα	35
4.1.5 Εικόνα μετά την εκτέλεση του αλγορίθμου	36
5.1.1.1 Κώδικας ελέγχου δικαιωμάτων εφαρμογής	39
5.1.1.2 Κώδικας για εγγραφή εφαρμογής	40
5.1.1.3 Κώδικας για έλεγχο επιτυχούς εγγραφής	41
5.1.1.4 Κώδικας ελέγχου για συνδεδεμένο προϊόν	42
5.1.1.5 Κώδικας getProduct	42
5.1.1.6 Κώδικας μεθόδου για λήψη φωτογραφιών	43
5.1.1.7 Κώδικας getCamera	44
5.1.1.8 Κώδικας αρχικοποίησης mission builder	44
5.1.1.9 Κώδικας μεθόδου OnNewfile	46
5.1.1.10 Κώδικας για αποστολή φωτογραφιών σε υπολογιστή	46
5.1.2.1 Κώδικας για λήψη φωτογραφιών από υπολογιστή	47
5.1.3.1 Κώδικας για μέτρηση εντοπισμένων αντικειμένων	48
5.1.3.2 Έξοδος τερματικού μετά την εκτέλεση του αλγορίθμου	48

5.1.3.3 Κώδικας για αλλαγή τοποθεσίας αποθήκευσης εικόνας	49
5.1.3.4 Κώδικας για αλλαγή μεγέθους εικόνας	49
5.2.1.1 Λήψη δικαιωμάτων και εγγραφή	50
5.2.1.2 Σύνδεση προϊόντος	51
5.2.1.3 Κεντρικό περιβάλλον εφαρμογής	52
5.2.2.1 Pre Flight Status Widget	52
5.2.2.2 PreFlight CheckList Panel	53
5.2.2.3 Camera controls widget, exposure panel και settings advanced panel	55
5.2.2.4 Dashboard Widget	55
5.2.3.1 WayPoint configuration	56
6.1.1.1 Πίνακας μετρήσεων για εντοπισμό ατόμων	59
6.1.1.2 Διάγραμμα επιτυχούς εντοπισμού ατόμων	59
6.1.2.1 Πίνακας μετρήσεων για εντοπισμό οχημάτων	60
6.1.2.2 Διάγραμμα επιτυχούς εντοπισμού οχημάτων	61
6.1.3.1 Διάγραμμα χρόνου εκτέλεσης	62
6.1.3.2 Πίνακας σύγκρισης αποτελεσμάτων	63
6.1.4.1 Εντοπισμός ατόμων σε μεγάλη απόσταση	65
6.1.4.2 Μεγάλος αριθμός οχημάτων σε εικόνα	66

Κεφάλαιο 1

Εισαγωγή

1.1 Unmanned Aerial Vehicles

Με τον όρο UAV (Unmanned Aerial Vehicles) ή απλά drones, γίνεται αναφορά σε κάθε είδους ιπτάμενη μηχανή, η οποία πραγματοποιεί πτήσεις χωρίς την παρουσία ανθρώπων εντός αυτής. Το μη επανδρωμένο αεροσκάφος μπορεί είτε να ελέγχεται από κάποιον χειριστή που βρίσκεται στο έδαφος, να είναι δηλαδή ημι-αυτόνομο, είτε να ελέγχεται μέσω κάποιου λειτουργικού συστήματος, και συνεπώς να είναι πλήρως αυτόνομο. Η αυτόνομη πτήση προϋποθέτει την χρήση εξειδικευμένου λογισμικού το οποίο ελέγχει το σχέδιο πτήσης και σε συνεργασία με τους onboard αισθητήρες και το GPS καθοδηγεί το αεροσκάφος[1].

Η ιδέα για τη χρήση μη επανδρωμένων αεροσκαφών ξεκίνησε πριν από τουλάχιστον πενήντα χρόνια. Αρχικά, οι σκοποί ήταν πολεμικοί και έκαναν την εμφάνιση τους στο Β παγκόσμιο πόλεμο. Το πρώτο μη επανδρωμένο που χρησιμοποιήθηκε ως όπλο ήταν το Γερμανικό FX-1400 ή “Fritz”, το οποίο αποτελούσε μια βόμβα που έπεφτε από ένα αεροπλάνο και ελεγχόταν απομακρυσμένα (ζύγιζε περίπου 1.5 τόνο). Μετά το τέλος του πολέμου σημειώθηκε κάποια εξέλιξη στην τεχνολογία τους, και εστίασαν περισσότερο στη στόχευση. Στην συνέχεια, αφού τελείωσε ο πόλεμος του Βιετνάμ, οι Ηνωμένες Πολιτείες της Αμερικής άρχισαν να δαπανούν αρκετά χρήματα για μικρά, μεγάλης εμβέλειας μη επανδρωμένα αεροσκάφη, όμως το πρόγραμμα σταμάτησε αρκετά γρήγορα επειδή ξέφευγε από τον προϋπολογισμό της κυβέρνησης. Στη συνέχεια, ακολούθησε το Ισραήλ, όπου ανέπτυξε το δικό του οπλισμένο drone το 1982, με ονομασία Pioneer, σημειώνοντας μεγάλη επιτυχία τη δεκαετία του 90. [2]

Με την πάροδο του χρόνου το ενδιαφέρον για τα μικρά μη επανδρωμένα αεροσκάφη έχει αυξηθεί δραματικά. Αρχικά, η κατασκευή των στρατιωτικών μη επανδρωμένων ζύγιζε αρκετά κιλά και η πτήση τους αφορούσε μεγάλα υψόμετρα. Έπειτα, επικεντρώθηκαν σε μικρότερα, τα οποία όμως πετούσαν χαμηλότερα. Όμως τα συγκεκριμένα ήταν πολύ ακριβά, με αποτέλεσμα να μη μπορούν να αποκτηθούν από πολίτες. Στη συνέχεια η βελτίωση της αυτονομίας της μπαταρίας καθώς και η μεγάλη μείωση του μεγέθους τους τα κατέστησε διαθέσιμα σε ερευνητές αλλά και σε άτομα που ήθελαν απλά να πειραματιστούν [3].

Υπάρχει μεγάλη ποικιλία από UAVs. Ο βασικός τρόπος με τον οποίο χωρίζονται σε κατηγορίες είναι το μέγεθος τους, καθώς και η ισχύ που

καταναλώνουν. Τα παραπάνω 2 χαρακτηριστικά είναι εκείνα, με βάση τα οποία επιλέγεται το κατάλληλο μη επανδρωμένο αεροσκάφος για την εκάστοτε εφαρμογή. Φυσικά υπάρχουν διάφορες κατηγοριοποιήσεις, παρακάτω παρουσιάζεται μια απλή διαχώριση μεταξύ 4 τύπων.

Large και Medium UAVs

Σε αυτές τις κατηγορίες περιλαμβάνονται αεροσκάφη κυρίως για κυβερνητικούς σκοπούς, όπως αυτά που χρησιμοποιεί η NASA. Το μεγάλο μειονέκτημα σε αυτές τις περιπτώσεις είναι το κόστος. Συνήθως απαιτείται διάδρομος απογείωσης καθώς και εξειδικευμένο προσωπικό για την πραγματοποίηση της συγκεκριμένης αποστολής.

Small, mini, micro, και nano UAVs

Αυτές τις κατηγορίες πλέον, ξεφεύγουν από τις κυβερνητικές οργανώσεις, και απευθύνονται σε ερευνητές, καθώς και σε οποιουσδήποτε άλλους χρήστες. Όσο αφορά το βάρος, ζυγίζουν λιγότερο από 20 κιλά, έχουν αυτονομία λίγων ωρών και φυσικά πολύ μικρότερη εμβέλεια από την πρώτη κατηγορία. Αυτά τα μεγέθη χωρίζονται σε 3 διαφορετικές κατηγορίες: (1) fixed-wing platforms, (2) rotor-based copter systems, και (3) nano-UAVs τελευταίας γενιάς. Τέλος, ακολουθεί μια εικόνα που συνοψίζει όλα τα παραπάνω [4].

Size	Characteristics	Payload size	Operational constraints	Example platforms
Large	Large operating range (~500 km); long flight time (up to 2 days); medium to high altitude (3–20 km)	~200 kg internally and ~900 kg in under-wing pods	High set-up and running costs; requires ground-station support, full aviation clearance, long runway for takeoff and landing, hangar for storage; altitude ceiling above commercial air traffic	NASA Ikhana 
Medium	Large operating range (~500 km); medium flight time (~10 hours); medium altitude (< 4 km)	~50 kg	Similar requirements to large UAVs but with reduced overall costs, reduced requirements for takeoff and landing, and easier control	NASA SIERRA 
Small and mini	Small operating range (< 10 km); low endurance (< 2 hours); low altitude (< 1 km)	Less than 30 kg (small); up to 5 kg (mini)	Line-of-sight flight only; largely fixed wing; simple launch gear and minimal landing/takeoff requirements; flown by flight planning software or by direct radio control	Quest UAV 
Micro and nano	Small operating range (< 10 km); very short flight time (< 1 hour); very low altitude (< 250 m)	Less than 5 kg	Hand-launched; line-of-sight flight only; soft landing place required; usually copter-type UAVs with rotor blade control; flown by flight planning software or by direct radio control	AR-Drone Parrot 

Σχήμα 1.1.1: Κατηγορίες μη επανδρωμένων

1.2 OpenCV

Ένα ευρέως γνωστό εργαλείο στην όραση υπολογιστών είναι το OpenCV. Το OpenCv (Open source computer vision), είναι μια βιβλιοθήκη όρασης υπολογιστών ανοικτού κώδικα, με βαρύτητα σε εφαρμογές πραγματικού χρόνου. Η πρώτη του εμφάνιση έγινε τον Ιανουάριου του 1999 και από τότε έχει χρησιμοποιηθεί σε πολλές εφαρμογές. Περιέχει περισσότερες από 500 συναρτήσεις που καλύπτουν διάφορες περιοχές της όρασης υπολογιστών. Εκτός τις παραπάνω συναρτήσεις, περιέχει και βιβλιοθήκες μηχανικής μάθησης γενικού σκοπού για ακόμα καλύτερα αποτελέσματα.

Όραση υπολογιστών (Computer Vision), είναι η επιστήμη όπου ένας υπολογιστής προγραμματίζεται ώστε να μπορεί να επεξεργαστεί και να κατανοεί εικόνες. Πλέον η βιβλιοθήκη του OpenCV περιέχει περισσότερους από 2500 αλγόριθμους, ενώ η χρησιμότητα του είναι εμφανέστατη αφού το 2012 είχε περισσότερες από 2.5 εκατομμύρια μεταφορτώσεις [5].

Η όραση υπολογιστών συναντάται συνεχώς σε διάφορες εφαρμογές, πολλές από τις οποίες ορισμένοι δεν φαντάζονται καν. Στη πλειονότητα βρίσκει εφαρμογή σε εικόνες και βίντεο στο διαδίκτυο. Επίσης, εμφανίζεται σε πλατφόρμες παιχνιδιών καθώς και σε εναέριες αλλά και επίγειες φωτογραφίες χαρτών, όπου απαιτείται βαθμονόμηση της κάμερας. Κάποιες, ακόμη, πιο εξειδικευμένες εφαρμογές είναι αυτές που αφορούν τον τομέα της παρακολούθησης και της χρήσης μη επανδρωμένων αεροσκαφών.

Ο τρόπος με τον οποίο ένας άνθρωπος αντιλαμβάνεται μια εικόνα και αναγνωρίζει αντικείμενα σε αυτή, είναι τελείως διαφορετικός με το πως την αντιλαμβάνεται ένας υπολογιστής. Όταν ο υπολογιστής διαβάζει μια εικόνα είτε από κάποια κάμερα, είτε από το σκληρό δίσκο, το μόνο που βλέπει είναι ένας πίνακας αριθμών. Ανάλογα την εφαρμογή, σε κάθε περίπτωση χρησιμοποιούνται διαφορετικές βιβλιοθήκες για την επίλυση του αντίστοιχου προβλήματος. Κλειδί για την εύκολη αντιμετώπιση, είναι η εκμετάλλευση όλων των διαθέσιμων δεδομένων, με σκοπό την αναγωγή του αρχικού προβλήματος σε άλλα μικρότερα και απλούστερα [6].

Όσο περνάει ο καιρός, με την εξέλιξη της τεχνολογίας, όλο και περισσότερες συσκευές εφοδιάζονται με κάμερες, από smartphones μέχρι και αυτοκίνητα. Στην περίπτωση των αυτοκινήτων τα περισσότερα μοντέλα περιλαμβάνουν βοηθό παρκαρίσματος και ειδοποιήσεις για επικίνδυνες καταστάσεις. Ακόμη και σε αυτή την περίπτωση αυτό επιτυγχάνεται με την λειτουργία της κάμερας και την άντληση δεδομένων από τις εικόνες. Προϋπόθεση αυτών είναι η ύπαρξη της αντίστοιχης

υπολογιστικής ισχύος. Ειδικότερα, στα smartphones λόγω των μεγάλων απαιτήσεων για υπολογιστική ισχύ, συνήθως γίνεται συμβιβασμός μεταξύ ποιότητας και χρόνου, ώστε να είναι δυνατή η εκτέλεση εφαρμογών σε πραγματικό χρόνο. Συνεπώς, είναι εμφανές ότι η ανάγκη για την επιλογή αποφάσεων που βασίζονται σε εικόνες είναι πολύ μεγάλη, και έτσι φαίνεται η σπουδαιότητα των βιβλιοθηκών του OpenCV. Εδώ πρέπει να σημειωθεί ότι από το 2010 το OpenCV υποστηρίζει και GPU acceleration για ακόμη μεγαλύτερη ταχύτητα [7].

Μια από τις συχνότερες εφαρμογές του OpenCV είναι η αναγνώριση αντικειμένων. Για το σκοπό αυτό εμπεριέχει στις βιβλιοθήκες του διάφορα εργαλεία σχετικά με αυτό.

1.3 Αντικείμενο της Διπλωματικής

Υπάρχουν αρκετές εφαρμογές που δέχονται ως είσοδο εικόνες και στη συνέχεια εντοπίζουν σε αυτές ανθρώπους και αντικείμενα. Συνήθως αυτό, από μόνο του, δεν είναι και τόσο λειτουργικό. Τις περισσότερες φορές, είναι αναγκαίο ο εντοπισμός να γίνεται σε μεγαλύτερες εκτάσεις, αλλά και σε πραγματικό χρόνο. Διαθέτοντας ως εξοπλισμό ένα μικρό μη επανδρωμένο αεροσκάφος, εφοδιασμένο με κάμερα είναι εφικτό ο χρήστης να φωτογραφίζει συγκεκριμένες περιοχές που επιθυμεί και στη συνέχεια τα δεδομένα αυτά να αποστέλλονται σε κάποιο υπολογιστή. Έτσι χρησιμοποιώντας το OpenCV μπορεί να ακολουθεί ο εντοπισμός αντικειμένων. Όλα τα παραπάνω είναι και το αντικείμενο της παρούσας διπλωματικής. Φυσικά υπάρχουν στο εμπόριο εφαρμογές που επιτελούν τις παραπάνω λειτουργίες, αλλά απευθύνονται κυρίως σε επαγγελματικούς σκοπούς. Στην συγκεκριμένη περίπτωση, η εφαρμογή απευθύνεται σε απλούς χρήστες, με περισσότερο user friendly λειτουργίες.

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής, διαθέτει κώδικα java, έχει υλοποιηθεί για συσκευές με λειτουργικό Android, και αφορά τα drones της εταιρείας DJI. Σκοπός είναι να παρέχει σε καθημερινούς χρήστες που διαθέτουν μη επανδρωμένα αεροσκάφη της εν λόγω εταιρείας την ευκαιρία να εκμεταλλευτούν τις δυνατότητες αναγνώρισης που μπορεί να τους παρέχει η κάμερα του UAV. Αναλυτικότερα, μπορούν να επιλεχθούν διάφορα σημεία στο χάρτη εντός της εμβέλειας του αεροσκάφους, να γίνει πτήση σε αυτά, και να ληφθούν φωτογραφίες ώστε να γίνει στη συνέχεια αναγνώριση προτύπων σε αυτές.

Η συγκεκριμένη εφαρμογή μπορεί να έχει πολλές χρήσεις. Στην πιο απλή της

εκδοχή μπορεί να χρησιμοποιηθεί για εντοπισμό οχημάτων και ανθρώπων σε μια περιοχή . Η χρήση της όμως μπορεί να αποδειχθεί ιδιαίτερα σημαντική όταν αυτή αφορά την προστασία ανθρώπινων ζώων. Συχνά, σε φυσικές καταστροφές αλλά και σε ατυχήματα, ζωτικής σημασίας είναι η γρήγορη αξιολόγηση της κατάστασης. Πιο συγκεκριμένα, όταν απαιτείται εκκένωση αλλά και διάσωση ατόμων απο περιοχές είναι πιο εύκολο αλλά και γρηγορότερο να σαρωθούν αυτές με τη χρήση ενός μη επανδρωμένου αεροσκάφους. Κατ' αυτό τον τρόπο, ο χρήστης ενημερώνεται για τον αριθμό των ατόμων καθώς και για την τοποθεσία τους. Η εφαρμογή, επίσης, μπορεί να χρησιμοποιηθεί για τον έλεγχο ασφαλείας περιουσιών, ορίζοντας ως σημείο ενδιαφέροντος στο χάρτη το αντίστοιχο σημείο. Επιπρόσθετα, η έρευνα σε δύσβατες περιοχές όπου η πρόσβαση από ανθρώπους θεωρείται δύσκολη μπορεί να επιτευχθεί χρησιμοποιώντας το drone. Σε περίπτωση εκτάκτων αναγκών μπορεί να πραγματοποιηθεί ακόμα και μεταφορά φαρμάκων. Γενικά, δεν υπάρχουν περιορισμοί και ο χρήστης μπορεί να τη χρησιμοποιήσει όπως εκείνος επιθυμεί, πάντα εντός των νομικών πλαισίων.

Κεφάλαιο 2

Σχετικές Εφαρμογές

Η χρήση της κάμερας των μη επανδρωμένων αεροσκαφών για την υλοποίηση διαφόρων εφαρμογών εντοπισμού έχει σημειώσει μεγάλη εξέλιξη τα τελευταία χρόνια. Κυκλοφορούν αρκετές επιλογές, άλλες δωρεάν και άλλες περισσότερο επαγγελματικές επί πληρωμή. Παρακάτω παρουσιάζονται υλοποιήσεις και των 2 κατηγοριών. Οι πρώτες 3 είναι ανοικτού κώδικα και συνεπώς διατίθενται δωρεάν.

2.1 Real-time People and Vehicle Detection from UAV Imagery

Παρακάτω παρουσιάζεται μια μέθοδος εντοπισμού οχημάτων και ανθρώπων χρησιμοποιώντας εικόνες από μη επανδρωμένο αεροσκάφος. Ο εντοπισμός οχημάτων βασίζεται σε πολλαπλά εκπαιδευμένα cascaded Haar classifiers, ενώ ακολουθεί επιβεβαίωση του εντοπισμού με τη χρήση θερμικών εικόνων. Χρησιμοποιώντας παρόμοια μοντέλα επιτυγχάνεται και ο αντιστοιχος εντοπισμός ανθρώπων. Το ποσοστό επιτυχούς ανίχνευσης ατόμων κυμαίνεται περίπου στο 70%, ενώ των οχημάτων στο 80%. Τέλος, σε κάθε πτήση το συνολικό ποσοστό είναι μεγαλύτερο και αγγίζει το 90%.

Στην προτεινόμενη υλοποίηση η δειγματοληψία γίνεται με 1 fps με σκοπό τη μεγιστοποίηση της ποιότητας της εικόνας, καθώς και την μη υπέρβαση του εύρους ζώνης μετάδοσης. Βασική διαφορά στον εντοπισμό ατόμων και οχημάτων είναι το μέγεθος τους. Είναι ευκολότερος ο εντοπισμός ενός μεγάλου οχήματος, σε σχέση με τα άτομα που συχνά λόγω χαμηλής αντίθεσης δεν ξεχωρίζονται εύκολα. Λύση σε αυτό το πρόβλημα έρχονται να δώσουν οι θερμικές εικόνες. Βελτιώνουν την αρχική κατάσταση, αλλά το πρόβλημα εξακολουθεί να είναι μια πρόκληση λόγω των καιρικών φαινομένων που επικρατούν σε κάθε περίπτωση και επηρεάζουν την θερμική εικόνα.

Εκτός την ανίχνευση, στο τέλος της επεξεργασίας, παρέχεται και ένας δείκτης εμπιστοσύνης για κάθε αντικείμενο ή άτομο. Η παραπάνω πληροφορία μπορεί να χρησιμοποιηθεί ως είσοδος σε κάποιο δευτερεύον σύστημα ελέγχου με σκοπό την αποτελεσματικότερη ταξινόμηση των αποτελεσμάτων αλλά και την περαιτέρω διερεύνηση τους. Αυξημένη ακρίβεια επιτυγχάνεται με τη χρήση πολλαπλών ταξινομητών και αισθητήρων.

Το σύστημα που χρησιμοποιείται για τη λήψη φωτογραφιών αποτελείται από ένα μικρό μη επανδρωμένο αεροσκάφος με σταθερά πτερύγια, εφοδιασμένο με οπτική και θερμική κάμερα. Είναι ρυθμισμένο να πετάει σε υψόμετρο περίπου 60 μέτρων, με την κάμερα στραμμένη προς το έδαφος υπό γωνία 45 μοιρών, δημιουργώντας έτσι απόσταση από το στόχο σε ευθεία γραμμή 180 μέτρα. Οι εικόνες αποστέλλονται στο έδαφος μέσω ραδιο ζεύκτη και η ανίχνευση πραγματοποιείται σε τετραπύρρηνο επεξεργαστή της Intel συχνότητας 2.33 GHz.

Οι ρουτίνες εντοπισμού χρησιμοποιούν ένα ζεύγος εικόνων, μια απλή οπτική εικόνα και μια θερμική. Σε πρώτο στάδιο οι 2 εικόνες επεξεργάζονται ανεξάρτητα και στη συνέχεια οι πληροφορίες που εξάγονται χρησιμοποιούνται συνδυαστικά για το τελικό αποτέλεσμα. Για τον εντοπισμό οχημάτων η οπτική εικόνα ελέγχεται με πολλά διαφορετικά cascaded Haar classifiers που έχουν εκπαιδευτεί για εντοπισμό οχημάτων σε διαφορετικούς προσανατολισμούς και φωτισμό. Αφού εντοπιστεί κάποιο όχημα με τα παραπάνω μοντέλα, το συγκεκριμένο σημείο της εικόνας ελέγχεται στην αντίστοιχη θερμική. Με παρόμοια μέθοδο γίνεται και ο εντοπισμός ατόμων. Ο δείκτης εμπιστοσύνης αυξάνεται περισσότερο αν το σημείο που τοποθετηθεί το περίγραμμα ταιριάζει με το Gaussian μοντέλο και υπάρχει μέσα στην αντίστοιχη περιοχή ανίχνευσης [8].

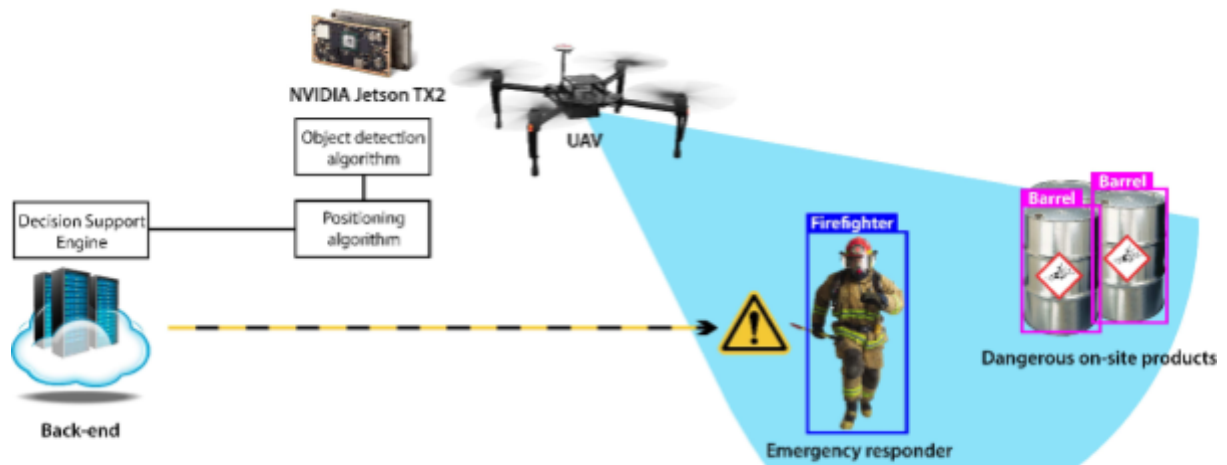
2.2 Embedded Real-Time Object Detection for a UAV Warning System

Σε αυτό το paper ο εντοπισμός αντικειμένων πραγματοποιείται με τη χρήση του αλγορίθμου YOLOv2. Ο αλγόριθμος εκτελείται σε GPU στοχεύοντας σε χαμηλές απαιτήσεις ισχύος ώστε να είναι δυνατή η εκτέλεση σε κινητές συσκευές. Προτείνονται αρκετές διαμορφώσεις που σχετίζονται με τον ρυθμό που συλλέγονται οι φωτογραφίες καθώς και την ακρίβεια της πρόβλεψης. Αφορά κυρίως έκτακτες καταστάσεις που λαμβάνουν χώρα σε πραγματικό χρόνο. Οι εναέριες εικόνες που προσφέρει το UAV μπορεί να υποδείξουν πιθανά σημεία εισόδου σε πυρκαγιές, να βοηθήσουν στον εντοπισμό αγνοούμενων σε έρευνες, και στη διάσωση τους. Ο σκοπός εδώ είναι ο εντοπισμός και η ειδοποίηση για κάποια έκτακτη κατάσταση. Επίσης όλη η εκτέλεση γίνεται onboard ώστε να αποφευχθεί ο κίνδυνος απώλειας δεδομένων σε μια ασύρματη σύνδεση. Έχει τοποθετηθεί μια GPU πάνω στο UAV, αφού πρώτα μελετήθηκαν διάφορες μεταβλητές όπως το βάρος και η ισχύς που χρειάζεται. Στη συγκεκριμένη δημοσίευση συγκρίνονται 3 αλγόριθμοι: YOLOv2, TinyYOLO και ACF. Οι παραπάνω προσφέρουν πολύ καλή αναλογία ακρίβειας και ταχύτητας.

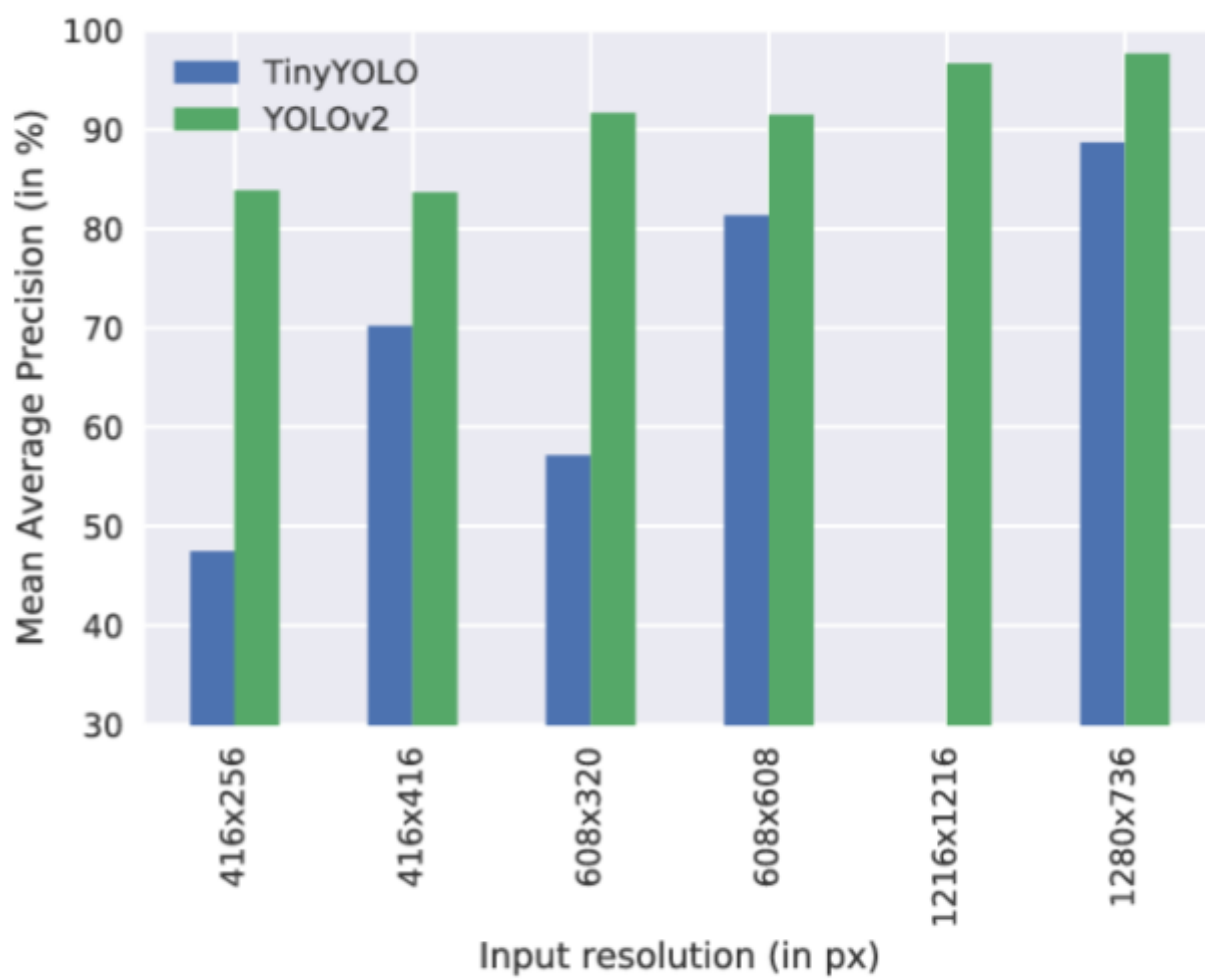
Βασικό πλεονέκτημα των μικρών μη επανδρωμένων αεροσκαφών είναι ότι μπορούν να φτάσουν σε περιοχές πολύ γρηγορότερα, και σε μέρη που ο άνθρωπος δεν μπορεί καν να πλησιάσει. Έτσι όταν το άτομο πλησιάσει διαθέτει πλέον πλήρη επίγνωση της κατάστασης. Αφού γίνει ο εντοπισμός των αντικειμένων, ακολουθεί μια αξιολόγηση κινδύνου. Πιο συγκεκριμένα, έχοντας εντοπιστεί άνθρωποι, οχήματα, ντεπόζιτα καυσίμων και άλλα επικίνδυνα αντικείμενα παρέχεται μια καλύτερη εικόνα για τους πιθανούς κινδύνους. Οι πληροφορίες αυτές δίνονται ως είσοδο σε μια μηχανή αποφάσεων. Σκοπός της αυτοματοποιημένης μηχανής αποφάσεων είναι η ταχύτερη απόφαση και η αποφυγή του ανθρώπινου λάθους. Παράγοντες όπως το άγχος, και τα μικρά περιθώρια αντίδρασης συμβάλλουν συχνά στη λανθασμένη αξιολόγηση της κατάστασης. Ακολουθεί μια εικόνα που περιγράφει μια τέτοια κατάσταση.

Ένας άλλος σημαντικός παράγοντας είναι η ανάλυση της εικόνας. Αύξηση του μεγέθους ισοδυναμεί με αύξηση του χρόνου επεξεργασίας και της ακρίβειας. Το YOLOv2 χρησιμοποιεί ως προκαθορισμένο μέγεθος το 416x416. Με βάση τις μετρήσεις που έγιναν η ακρίβεια αυξάνεται οριακά μόνο όταν το πλάτος παραμείνει ίδιο ενώ το ύψος αυξηθεί ώστε να προκύψει τετράγωνο (83.67% @ 416x256),

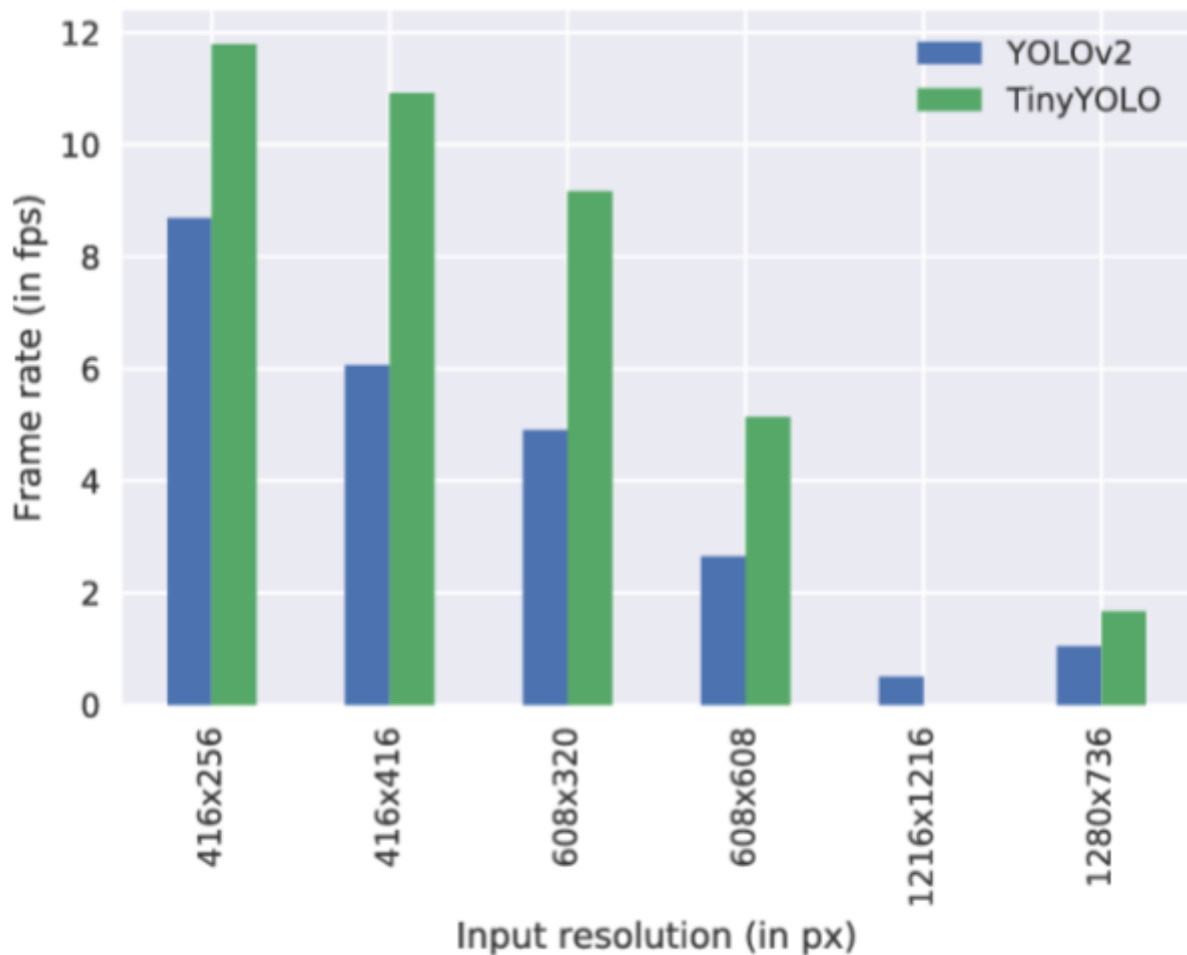
(97.67% @ 1216x1216). Όπως φαίνεται και στις μετρήσεις ενώ η μεταβολή της ακρίβειας δεν είναι τόσο μεγάλη, το fps παρουσιάζει μεγάλες διαφορές [9].



Σχήμα 2.2.1: Αρχιτεκτονική συστήματος υποστήριξης αποφάσεων βασισμένου σε UAV για απόκριση έκτακτης ανάγκης



Σχήμα 2.2.2: Σύγκριση ανάλυσης εικόνας εισόδου και ακρίβειας μοντέλου



Σχήμα 2.2.3: Επίδραση ανάλυσης εικόνας στον ρυθμό επεξεργασίας

2.3 A fast object detector based on high-order gradients and Gaussian process regression for UAV images

Αυτό που κάνει αυτή την εφαρμογή ιδιαίτερη είναι το ύψος του μη επανδρωμένου αεροσκάφους. Το υψόμετρο του AUV είναι περίπου 2 χιλιόμετρα σε αυτή την περίπτωση, κάτι που κάνει σχεδόν αδύνατο τον εντοπισμό μέσω της ανθρώπινης φιγούρας, αφού σε κάθε εικόνα τα pixels που αντιστοιχούν ανά άνθρωπο είναι πολύ λίγα. Η σκιές που δημιουργούνται όμως είναι πολύ πιο εύκολο να εντοπιστούν.

Αρχικά χρησιμοποιείται ένας αλγόριθμος που εντοπίζει πιθανούς στόχους χρησιμοποιώντας Maximally Stable Extremal Regions (MSER). Στη συνέχεια οι

στόχοι που εντοπίστηκαν δίνονται ως είσοδοι σε διάφορα φίλτρα ώστε να εντοπιστούν άνθρωποι και οι σκιές τους. Αυτό επιτυγχάνεται απορρίπτοντας περιπτώσεις όπου ο άνθρωπος είναι πολύ μεγάλος ή πολύ μικρός σε σχέση με μια μέση περίπτωση. Για την εύρεση των σκιών χρησιμοποιούνται διάφορα στοιχεία όπως τη γωνία λήψης, την ημέρα αλλά και την ώρα που τραβήχτηκε η εικόνα. Με αυτόν τον τρόπο απορρίπτονται στόχοι που επιλέχθηκαν λανθασμένα. Για να εντοπιστεί κάποιος άτομο επιτυχώς πρέπει να υπάρχει ζευγάρι ανθρώπινης φιγούρας και σκιάς που να ταιριάζουν. Ωστόσο, προκειμένου να βελτιωθεί περαιτέρω ο ρυθμός αναγνώρισης, τα μοτίβα αναγνώρισης, που προέρχονται από ιστορικά δεδομένα, λαμβάνονται επίσης υπόψη.

Για την αναγνώριση πυρκαγιών χρησιμοποιείται η θερμική κάμερα, αφού οι διαφορές θερμοκρασίας είναι μεγάλες κάνοντας έτσι τον εντοπισμό εύκολο. Και πάλι χρησιμοποιώντας τον MSER εξάγονται πιθανές εστίες πυρκαγιάς. Επιπροσθέτως για την επιβεβαίωση πιθανής πυρκαγιάς χρησιμοποιείται η αντίστοιχη εικόνα που λήφθηκε από την έγχρωμη κάμερα. Ακολουθεί χρωματικός έλεγχος, για να επιβεβαιωθεί αν οι πιθανές εστίες πυρκαγιάς εμπίπτουν στον αντίστοιχο χρωματισμό [10].

2.4 DroneSAR - Search & Rescue

Το DroneSAR είναι ένα πρόγραμμα που αναπτύχθηκε μετά από πολλούς μήνες έρευνας και εμπειρίας στη χρήση drones για αντιμετώπιση έκτακτης ανάγκης. Παρέχει τις σωστές πληροφορίες στους σωστούς ανθρώπους όταν ο χρόνος και οι πόροι είναι περιορισμένοι. Όπως και στην εφαρμογή που υλοποιείται στην παρούσα διπλωματική, έτσι και αυτή είναι συμβατή με drones της εταιρείας DJI. Η λειτουργικότητα του λογισμικού βασίζεται στις ανάγκες των πιλότων.

Παρέχονται μια σειρά από επιλογές αποστολής. Πιο συγκεκριμένα επιλέγοντας custom grid function, οι πιλότοι μπορούν να επιλέξουν το μέγεθος και το σχήμα της περιοχής που θα ήθελαν να ψάξουν. Επίσης υπάρχει Waypoint Mission. Σε αυτή την αποστολή επιλέγονται εύκολα σημεία πάνω στο χάρτη που πρέπει να ακολουθήσει το drone. Αγγίζοντας την οθόνη θα τοποθετηθεί ένα σημείο στο χάρτη, το οποίο μπορεί να ακολουθήσει οποιοσδήποτε αριθμός σημείων. Ακόμα όταν ο χρήστης γνωρίζει το γεωγραφικό μήκος και πλάτος του σημείου ενδιαφέροντος μπορεί να επιλέξει το Lat/Long Mission. Επίσης σε όλα τα είδη αποστολών παρέχονται στην οθόνη διάφορες πληροφορίες σχετικές με το αεροσκάφος.



Σχήμα 2.4.1: Εκτέλεση αποστολής σε DronSAR

Κατά την εκτέλεση της αποστολής παρέχονται πληροφορίες στον χρήστη μέσω της κινητής συσκευής που έχει εγκατασταθεί η εφαρμογή. Γίνεται ζωντανή μετάδοση βίντεο από το μέρος που πετάει το drone και παρέχεται η δυνατότητα κοινοποίησης τοποθεσίας. Υπάρχει δυνατότητα δωρεάν δοκιμής για 30 μέρες, ενώ στη συνέχεια κάθε μήνας χρεώνεται με 120 ευρώ [11].

2.5 Nightingale Security

Η Nightingale Security παρέχει ρομποτική αεροναυτική ασφάλεια για τις εταιρείες. Η υπηρεσία αποτελείται από αεροσκάφη, σταθμούς βάσης και ισχυρό λογισμικό ελέγχου αποστολής, παρέχοντας πλήρη αυτόνομη, φυσικής ασφάλεια σε πραγματικό χρόνο με κάμερες παρακολούθησης και αισθητήρες συλλογής δεδομένων. Οι σταθμοί βάσης τοποθετούνται σε στέγες και άλλες ασφαλείς τοποθεσίες γύρω από την εγκατάσταση. Είναι ο κόμβος επικοινωνίας του στόλου των μη επανδρωμένων με μεγάλη αντοχή.



Σχήμα 2.5.1: Σταθμός Βάσης

Σε περίπτωση εντοπισμού κάποιας απειλής, απογειώνεται ένα drone αυτόματα και πηγαίνει στο σημείο απειλής παρέχοντας ζωντανή εικόνα στην ομάδα ασφαλείας. Επίσης μπορούν να οριστούν περιπολίες για συγκεκριμένες ώρες της ημέρας. Το σύστημα παρέχει αυτόνομες απομακρυσμένες λειτουργίες (ARO), οπότε δεν απαιτείται ανθρώπινη παρέμβαση για να διατηρηθεί λειτουργικό [12].

Κεφάλαιο 3

Object detection

3.1 Machine learning

Οι υπολογιστές παρόλο που έχουν τεράστια υπολογιστή ισχύ αυτό που πολλές φορές τους κάνει να υστερούν είναι ότι δεν μπορούν να πάρουν αποφάσεις όπως θα έκανε ένας άνθρωπος. Αυτό είναι ένα πρόβλημα που έρχεται να αντιμετωπίσει η μηχανική μάθηση. Η μηχανική μάθηση είναι ο προγραμματισμός των υπολογιστών ώστε να “μαθαίνουν” από τα δεδομένα που δέχονται ως είσοδο, να μετατρέπεται δηλαδή η εμπειρία σε εξειδίκευση και γνώση. Σε έναν αλγόριθμο εκμάθησης τα δεδομένα εισόδου αντιπροσωπεύουν την εμπειρία, ενώ συνήθως η έξοδος που ισοδυναμεί με την εξειδίκευση δίνεται ως είσοδος σε κάποιο άλλο πρόγραμμα.

Ένα μεγάλο ερώτημα είναι σε ποιες περιπτώσεις χρειάζονται οι αλγόριθμοι μηχανικής μάθησης. Πολλές καθημερινές δραστηριότητες των ανθρώπων γίνονται κατά κάποιο τρόπο ασυνείδητα, με την έννοια ότι δεν χρειάζεται αρκετή σκέψη αλλά εκτελούνται μηχανικά. Μερικά παραδείγματα είναι η οδήγηση, η αναγνώριση φωνής και ο εντοπισμός αντικειμένων σε εικόνες. Σε όλα τα παραπάνω η μηχανική μάθηση δίνει πολύ καλά αποτελέσματα όταν χρησιμοποιούνται καλά εκπαιδευμένα μοντέλα. Όλες οι προηγούμενες εργασίες εκτελούνται από ανθρώπους, όμως μια άλλη μεγάλη κατηγορία που χρειάζεται μηχανική μάθηση είναι οι εργασίες που είναι αδύνατο ή μη αποδοτικό να εκτελούνται από ανθρώπους. Τέτοιες περιπτώσεις είναι ανάλυση μεγάλου όγκου και πολύπλοκων δεδομένων. Αναλυτικότερα: αστρονομικά δεδομένα, η πρόβλεψη του καιρού, μηχανές αναζήτησης στο διαδίκτυο είναι μόνο μερικές από τις υπεράριθμες εφαρμογές που συμβάλλει η μηχανική μάθηση. Ένα μεγάλο μειονέκτημα αρκετών προγραμματιστικών εργαλείων είναι ότι μένουν αμετάβλητα, ενώ τα δεδομένα που επεξεργάζονται μεταβάλλονται στο χρόνο. Τα μοντέλα μηχανικής μάθησης προσαρμόζονται συνεχώς σε όλες τις αλλαγές μέσω των δεδομένων εισόδου που επεξεργάζονται [13].

3.2 Μοντέλα αναγνώρισης αντικειμένων

Έχουν δημιουργηθεί αρκετά μοντέλα για την αναγνώριση αντικειμένων. Ορισμένα από αυτά εστιάζουν στην ακρίβεια των αποτελεσμάτων, ενώ άλλα στην ταχύτητα ώστε να είναι δυνατός ο εντοπισμός σε πραγματικό χρόνο. Οι μέθοδοι ανίχνευσης αντικειμένων εν γένει εμπίπτουν είτε σε προσεγγίσεις που βασίζονται στη μηχανική μάθηση είτε σε προσεγγίσεις βασισμένες στη βαθιά εκμάθηση. Για τις προσεγγίσεις μηχανικής μάθησης, καθίσταται απαραίτητο να οριστούν πρώτα τα χαρακτηριστικά χρησιμοποιώντας μία από τις παρακάτω μεθόδους και στη συνέχεια να χρησιμοποιηθεί μια τεχνική όπως η μηχανή φορέα υποστήριξης (SVM) για την ταξινόμηση. Από την άλλη πλευρά, τεχνικές βαθιάς εκμάθησης που είναι σε θέση να κάνουν ανίχνευση αντικειμένων από άκρο σε άκρο χωρίς να καθορίζουν συγκεκριμένα χαρακτηριστικά και συνήθως βασίζονται σε συνελκτικά νευρωνικά δίκτυα (CNN) [14].

- **Machine Learning approaches:**
 - Viola–Jones object detection framework based on Haar features
 - Scale-invariant feature transform (SIFT)
 - Histogram of oriented gradients (HOG) features
- **Deep Learning approaches:**

- Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN)
- Single Shot MultiBox Detector (SSD)
- You Only Look Once (YOLO)

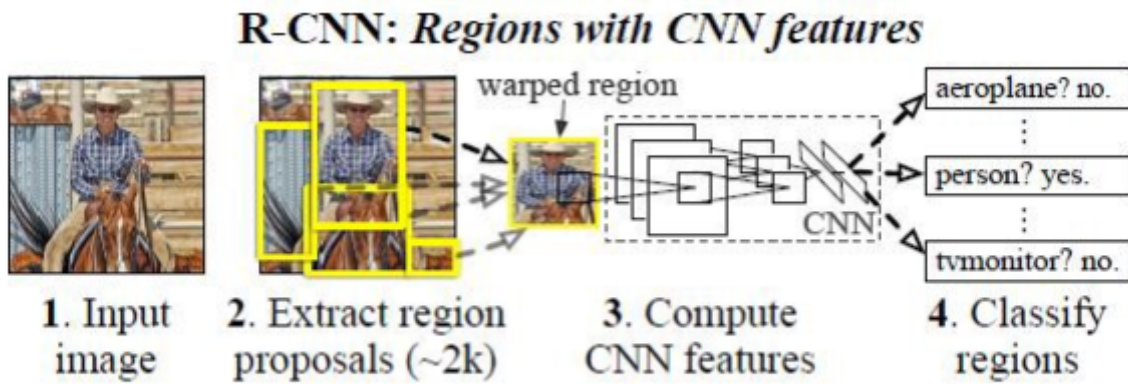
Παρακάτω ακολουθεί μια παρουσίαση των Deep learning προσεγγίσεων.

3.2.1 R-CNN

Τα συνελικτικά νευρωνικά δίκτυα ενώ γνώρισαν μεγάλη ανάπτυξη το 1990 ξεχάστηκαν λόγω της άνθισης που γνώρισαν οι μηχανές διανυσμάτων. Το 2012 ξανά εμφανίστηκαν παρουσιάζοντας μεγάλη ακρίβεια στην ταξινόμηση εικόνων. Η επιτυχία τους προήλθε από την εκπαίδευση νευρωνικού δικτύου με 1.2 εκατομμύρια εικόνες.

Παρατηρήθηκε ότι με τη χρήση νευρωνικού δικτύου μπορούμε να έχουμε αρκετά καλύτερη απόδοση από ότι με συστήματα που βασίζονται σε HOG χαρακτηριστικά. Για να επιτευχθεί αυτό απαιτούνται 2 πράγματα: εντοπισμός των αντικειμένων με χρήση deep networks και η εκπαίδευση του μοντέλου με μια μικρή μόνο ποσότητα δεδομένων σχετικά με την ανίχνευση. Σε αντίθεση με την ταξινόμηση εικόνων, η ανίχνευση απαιτεί εντοπισμό (πιθανών πολλών) αντικειμένων μέσα σε μια εικόνα.

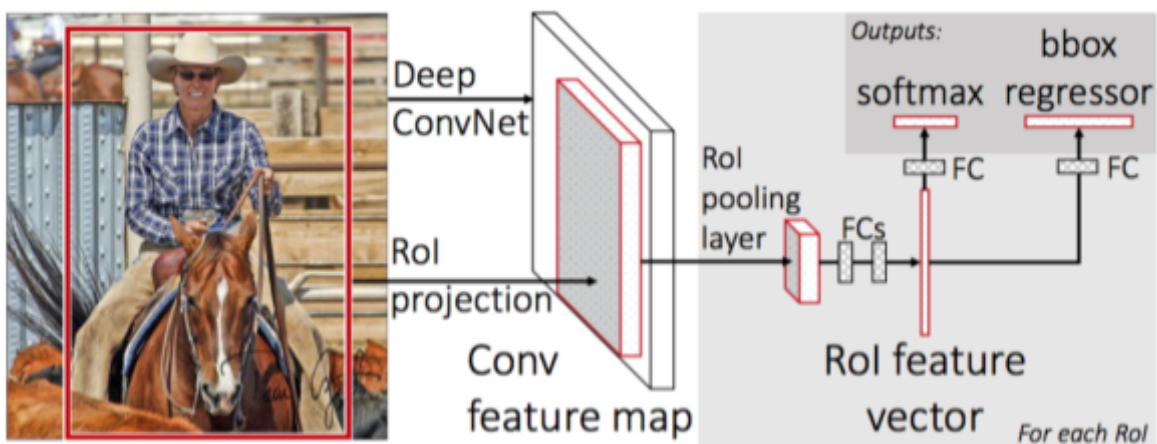
Για να διατηρήσουν υψηλή χωρική ανάλυση, τα CNN τυπικά έχουν μόνο δύο συνελικτικά και συγκεντρωτικά στρώματα. Ωστόσο στο R-CNN δίκτυο τα στρώματα είναι περισσότερα. Έχει 5 συνελικτικά στρώματα, πολύ μεγάλα receptive fields (195×195 pixels) και strides (32×32 pixels) στην εικόνα εισόδου. Στο test-time δημιουργούνται περίπου 2000 ανεξάρτητες περιοχές ενδιαφέροντος, εξάγει ένα χαρακτηριστικό σταθερού μήκους διάνυσμα από κάθε περιοχή χρησιμοποιώντας ένα CNN και στη συνέχεια ταξινομεί κάθε περιοχή με γραμμικά SVM για συγκεκριμένες κατηγορίες. Δεδομένου ότι το σύστημα συνδυάζει τις προτάσεις περιοχής(region) με το CNN, προκύπτει και το όνομα R-CNN: Regions with CNN features. Το σύστημα έχει απόδοση 53.3% mAP. Επιτυγχάνει μικρότερο χρόνο εκτέλεσης και μεγαλύτερη ακρίβεια σε σχέση με τις μεθόδους κυλιόμενου παραθύρου. Όμως επειδή κάθε μία από τις 2000 προτεινόμενες περιοχές ενδιαφέροντος επεξεργάζεται ξεχωριστά, η μέθοδος συνολικά παραμένει αργή τόσο στην εκπαίδευση όσο και στην εξαγωγή αποτελέσματος [15].



Σχήμα 3.2.1.1: Αρχιτεκτονική μοντέλου R-CNN

3.2.2 Fast R-CNN

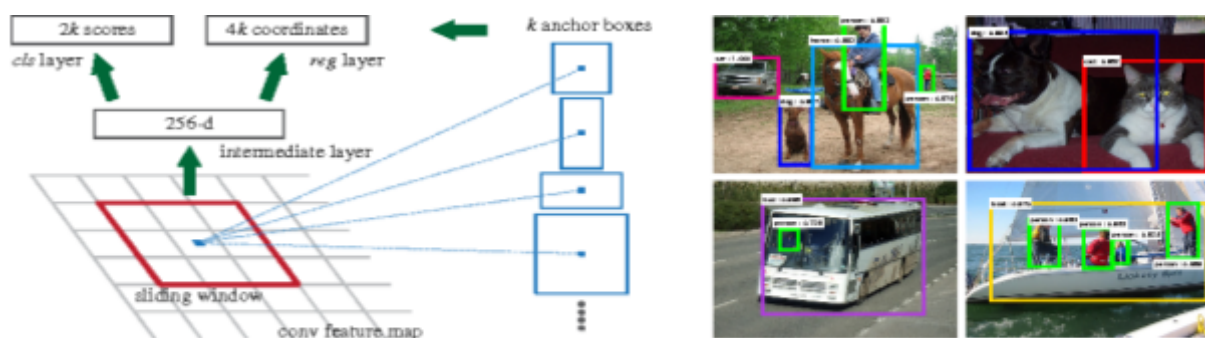
Λύση στο πρόβλημα της πολύ αργής ταχύτητας ήρθε να δώσει το Fast R-CNN. Το R-CNN ήταν πολύ αργό επειδή δεν μοιραζόταν τους υπολογισμούς και έτρεχε για κάθε μια από τις προτάσεις ξεχωριστά. Τα Spatial pyramid pooling networks (SPPnets) προτάθηκαν για την επιτάχυνση του R-CNN με την κατανομή των υπολογισμών. Η Μέθοδος SPPnet υπολογίζει ένα συνελκτικό χάρτη για ολόκληρη την εικόνα εισόδου και στη συνέχεια ταξινομεί κάθε πρόταση αντικειμένου χρησιμοποιώντας ένα διάνυσμα που εξάγεται από τον κοινόχρηστο χάρτη. Με αυτόν τον τρόπο η ταχύτητα του R-CNN στο test time αυξάνεται από 10 έως 100 φορές. Ο χρόνος εκπαίδευσης μειώνεται επίσης κατά 3 φορές λόγω της ταχύτερης εξαγωγής χαρακτηριστικών προτάσεων. Το σύστημα έχει απόδοση 66% mAP [16].



Σχήμα 3.2.2.1: Αρχιτεκτονική μοντέλου Fast R-CNN

3.2.3 Faster R-CNN

Μια αλγοριθμική αλλαγή, υπολογισμός προτάσεων με μια βαθύ δίκτυο οδηγεί σε μια κομψή και αποτελεσματική λύση, όπου ο υπολογισμός της πρότασης είναι σχεδόν μηδαμινός. Έχει παρατηρηθεί ότι οι χάρτες συνελκτικών χαρακτηριστικών που χρησιμοποιούνται από τους ανιχνευτές με βάση την περιοχή, όπως το Fast R-CNN, μπορούν επίσης να χρησιμοποιηθούν για τη δημιουργία προτάσεων περιοχής. Προσθέτοντας δύο επιπλέον επίπεδα στρώσεων: ένα που κωδικοποιεί κάθε θέση χάρτη μετατροπής σε ένα σύντομο διάνυσμα χαρακτηριστικών, και ένα δεύτερο που σε κάθε θέση του συνελκτικού χάρτη εξάγει ένα σκορ αντικειμενικότητας. Το σύστημα έχει απόδοση 70.4% mAP [17].



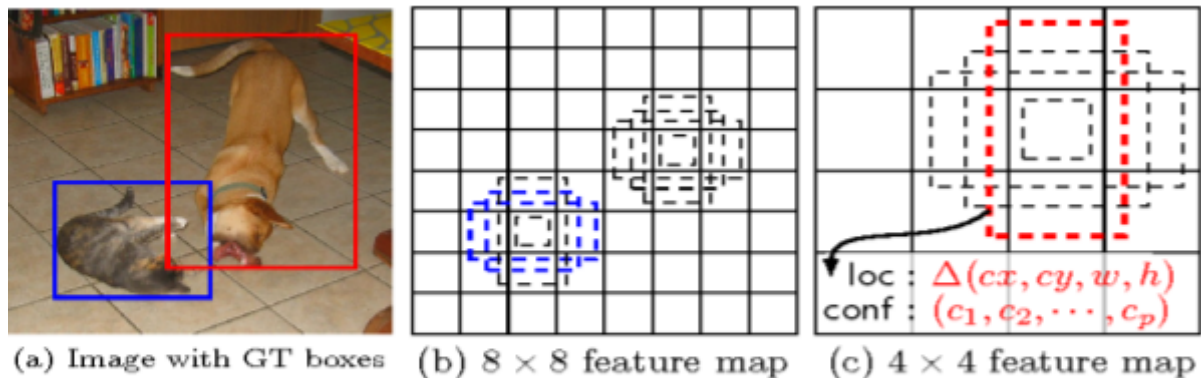
Σχήμα 3.2.3.1: Αρχιτεκτονική μοντέλου Faster R-CNN

3.2.4 Single Shot MultiBox Detector

Το SSD δημιουργήθηκε με σκοπό να μειώσει το χρόνο υπολογισμού κάνοντας έτσι δυνατό τον εντοπισμό αντικειμένων σε πραγματικό χρόνο. Το πρόβλημα είναι το μεγάλο και περίπλοκο pipeline που διαθέτουν τα περισσότερα μοντέλα. Ακόμα και στο βελτιωμένο Faster-RCNN η ταχύτητα ήταν μόλις 7 fps. Σε προσπάθειες να μειωθεί ο χρόνος στο pipeline, το κόστος μείωσης της ακρίβειας ήταν πολύ μεγάλο.

Είναι το πρώτο μοντέλο που δεν επαναπροσδιορίζει τα pixels ή τα χαρακτηριστικά για να τοποθετήσει τα πλαίσια, ενώ πετυχαίνει πάρα πολύ μεγάλα ποσοστά ακρίβειας. Αυτό επιτεύχθηκε περιορίζοντας αισθητά τις προτάσεις που γίνονται για πλαίσια, και την επαναδειγματοληψία των pixels. Αναλυτικότερα, προστέθηκε ένα φίλτρο το οποίο προβλέπει τις κατηγορίες των αντικειμένων, αντισταθμίσεις στα πλαίσια οριοθέτησης χρησιμοποιώντας ξεχωριστά προγνωστικά φίλτρα για διαφορετικές κατηγορίες πρόβλεψης. Έτσι το δίκτυο πραγματοποιεί

αναγνώριση σε διαφορετικές κλίμακες. Με αυτές τις τροποποιήσεις ειδικά χρησιμοποιώντας πολλαπλά layers για πρόβλεψη σε διαφορετικές κλίμακες μπορούμε να επιτύχουμε υψηλή ακρίβεια χρησιμοποιώντας σχετικά χαμηλή ανάλυση, αυξάνοντας περαιτέρω την ταχύτητα ανίχνευσης [18].

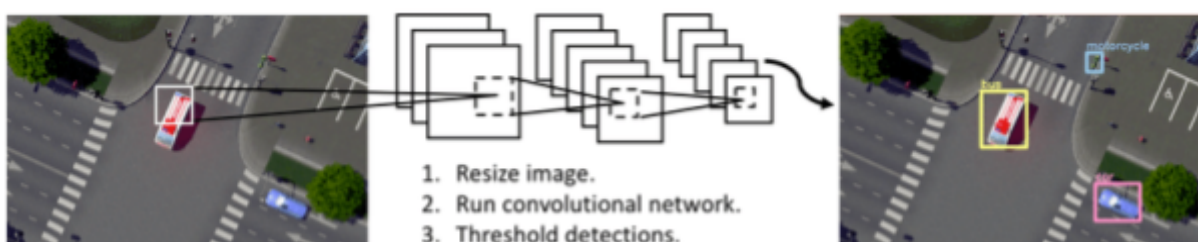


Σχήμα 3.2.4.1: Αρχιτεκτονική μοντέλου SSD

3.2.5 YOLO

Το μοντέλο αναγνώρισης YOLO ή You Only Look Once είναι ένα από τα γρηγορότερα που κυκλοφορούν προσφέροντας ταυτόχρονα υψηλή ακρίβεια και μεγάλη ταχύτητα. Σχεδόν όλα τα άλλα μοντέλα χρησιμοποιούν classifiers. Αντίθετα το YOLO σπάει την εικόνα σε μικρά κουτιά και ελέγχει ξεχωριστά τις πιθανότητες για κάθε κλάση. Για την ανίχνευση χρησιμοποιείται ένα μόνο νευρωνικό δίκτυο που επεξεργάζεται κατευθείαν την εικόνα και τοποθετεί κουτιά και την αντίστοιχη πιθανότητα για κάθε περίπτωση.

Μεγάλη διαφοροποίηση υπάρχει στον τρόπο που γίνεται η αναγνώριση. Σε άλλα δίκτυα, πρώτα χρησιμοποιούνται μέθοδοι για τη δημιουργία των κουτιών και μετά σε αυτές τις περιοχές χρησιμοποιούνται classifiers. Η λειτουργία του είναι αρκετά απλή: Αρχικά αλλάζει το μέγεθος της εικόνας και έπειτα το δίκτυο που χρησιμοποιείται προβλέπει ταυτόχρονα πολλά boxes και τις πιθανότητες κάθε κλάσης για αυτά.



Σχήμα 3.2.5.1: Αρχιτεκτονική μοντέλου YOLO

Έχει αρκετά πλεονεκτήματα σε σύγκριση με διάφορα άλλα μοντέλα. Είναι

αρκετά πιο γρήγορο, αφού αντιμετωπίζει την αναγνώριση παλινδρομικά και έχει απλό pipeline. Το δίκτυο απλά δέχεται την εικόνα και κάνει προβλέψεις. Η απλή έκδοση έχει ταχύτητα 45 fps (σε TITAN X GPU) ενώ η fast έκδοση 150 fps. Αυτό συνεπάγεται ότι μπορεί να χρησιμοποιηθεί και σε βίντεο, συνεπώς σε Real Time με latency λιγότερο από 25 ms. Επιτυγχάνει περισσότερο από δύο φορές τη μέση ακρίβεια από άλλα συστήματα. Αντίθετα με sliding window και region proposal-based τεχνικές, το YOLO επεξεργάζεται ολόκληρη την εικόνα κατά την εκπαίδευση και έτσι κωδικοποιεί έμμεσα πληροφορίες σχετικά με τις τάξεις καθώς και την εμφάνισή τους. Το Fast R-CNN, ένα από τα πιο γνωστά μοντέλα αναγνώρισης σφάλει πολύ σε background λάθος αναγνωρίσεις, επειδή δεν βλέπει την εικόνα συνολικά. Επιπροσθέτως το YOLO μαθαίνει γενικευμένες παραστάσεις αντικειμένων. Αυτό του δίνει τη δυνατότητα όταν εκπαιδεύεται σε φυσικές εικόνες να ξεπερνά ακόμα και κορυφαία μοντέλα όπως το DPM και R-CNN. Το μόνο σημείο που υστερεί είναι τα μικρά αντικείμενα. Το σύστημα διαιρεί την εικόνα $S \times S$ πλέγμα. Αν το κέντρο του αντικειμένου εμπεριέχεται σε κάποιο πλαίσιο, τότε αυτό το πλαίσιο είναι υπεύθυνο για την αναγνώριση [19].

Κεφάλαιο 4

Σχετικά με την εφαρμογή

Όπως αναλύθηκε παραπάνω οι διαδικασίες εντοπισμού με χρήση μη επανδρωμένων αεροσκαφών μπορούν να έχουν διάφορες προσεγγίσεις. Η παρούσα διπλωματική έχει σχεδιαστεί ώστε να μπορεί να εντοπίσει διάφορα αντικείμενα, εστιάζοντας περισσότερο στον εντοπισμό ανθρώπινων οντοτήτων και οχημάτων. Δεν υπάρχουν περιορισμοί όσο αφορά τη χρήση της εφαρμογής. Αυτό δίνει στο χρήστη την ευχέρεια να εκμεταλλευτεί τις δυνατότητες που παρέχονται όπως εκείνος επιθυμεί. Καλύπτει διάφορες περιπτώσεις προβλημάτων. Για το λόγο αυτό σε κάθε αποστολή οι περισσότερες ρυθμίσεις σχετικά με το ύψος, την ταχύτητα και την γωνία της κάμερας μπορούν να μεταβληθούν. Οι χρήστες μπορούν να ανήκουν σε διάφορες κατηγορίες. Πιο συγκεκριμένα οι σκοποί τους μπορούν να είναι εμπορικοί, ερευνητικοί ή ακόμη και απλή ενασχόληση χωρίς κάποιο συγκεκριμένο όφελος. Δεν απαιτούνται ειδικές γνώσεις για τον χειρισμό της εφαρμογής. Έχει σχεδιαστεί με τέτοιο τρόπο ώστε ο καθένας να είναι σε θέση να μπορεί να την διαχειριστεί. Ακόμη και στην περίπτωση που απαιτούνται κάποιες ρυθμίσεις υπάρχει καθοδήγηση και σχετικά μηνύματα εντός του γραφικού περιβάλλοντος. Πιο χρήσιμη από όλες τις προσεγγίσεις είναι η περίπτωση που η εφαρμογή χρησιμοποιείται ώστε να προσφέρει βοήθεια σε ατυχήματα και γενικά έκτακτες ανάγκες. Σε τέτοια

προβλήματα η αντιμετώπιση είναι κρίσιμη και απαιτεί γρήγορες και ακριβείς κινήσεις ώστε να αποφεύγονται ανεπιθύμητα αποτελέσματα.

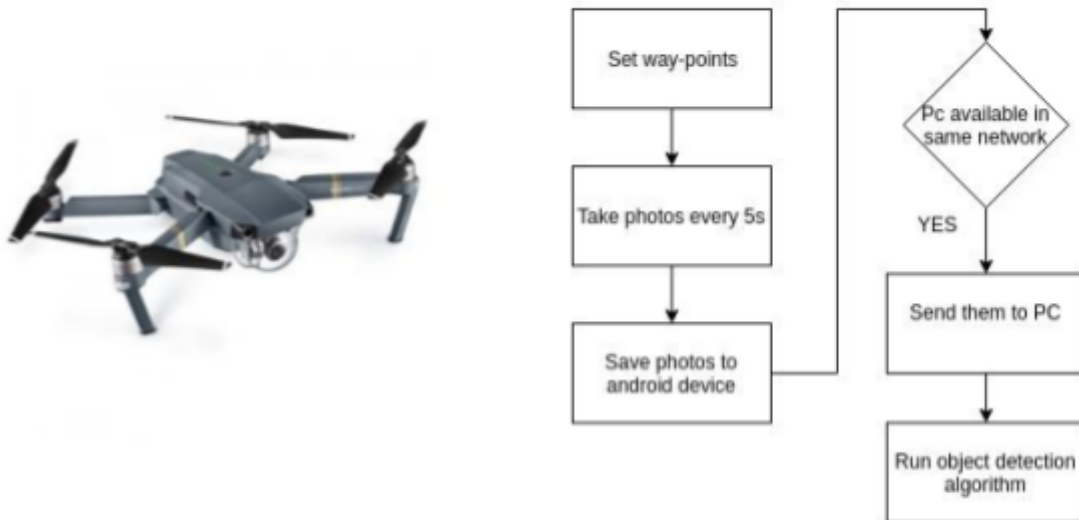
4.1 Λειτουργία εφαρμογής

Για τη χρήση της εφαρμογής απαιτείται ένα μη επανδρωμένο αεροσκάφος της εταιρείας DJI, μια Android συσκευή, και ένας υπολογιστής (Laptop ή Desktop). Σε περίπτωση που δεν διατίθεται υπολογιστής κατά τη διάρκεια της αποστολής, μπορεί να εκτελεστεί κανονικά με τη διαφορά ότι θα πραγματοποιηθεί μόνο η λήψη των φωτογραφιών οι οποίες θα αποθηκευτούν στη συσκευή αλλά και στο drone. Στη συνέχεια θα πρέπει να αποσταλούν χειροκίνητα στον υπολογιστή ώστε να εκτελεστεί ο κώδικας που αφορά τον εντοπισμό.

Οι περισσότερες αντίστοιχες εφαρμογές που κυκλοφορούν αφορούν κυρίως εμπορική χρήση. Η κυριότερη διαφορά της συγκεκριμένης υλοποίησης είναι ότι μπορεί να χρησιμοποιηθεί από τον καθένα που διαθέτει τον εξοπλισμό που περιεγράφηκε παραπάνω. Είναι φιλική προς το χρήστη χωρίς να απαιτούνται εξεζητημένες ρυθμίσεις. Η πλειοψηφία των λειτουργιών που συναντά κανείς στο γραφικό περιβάλλον της εφαρμογής είναι αυτοματοποιημένες. Αυτός είναι και ο σκοπός της, να μπορεί να εξυπηρετήσει ανάγκες μεγάλης ποικιλίας χρηστών και προβλημάτων χωρίς να προϋποθέτει την ύπαρξη γνώσεων σχετικά με τις ενέργειες που εκτελούνται στο υπόβαθρο.

Για να μπορεί όμως να εξυπηρετήσει ακόμη και τους πιο απαιτητικούς χρήστες, εκτός από προεπιλεγμένες ρυθμίσεις, παρέχεται η δυνατότητα μεταβολής αρκετών παραγόντων που σχετίζονται με την αποστολή που πρόκειται να εκτελεστεί. Ορισμένες από αυτές είναι το ύψος, η κλίση του gimbal, ο προσανατολισμός του αεροσκάφους κατά την εκτέλεση της πτήσης και η ενέργεια που θα ακολουθήσει μετά την περάτωση της αποστολής.

Περίληπτικά αυτό που κάνει η εφαρμογή φαίνεται στο παρακάτω διάγραμμα:



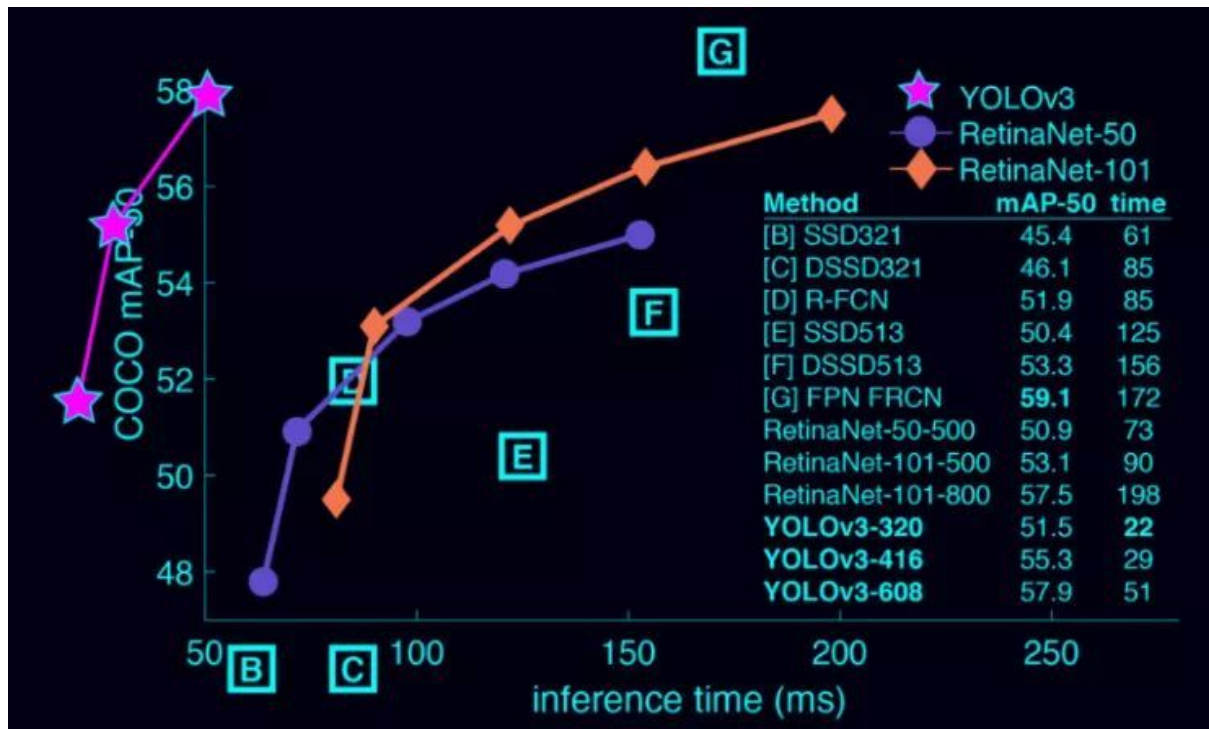
Σχήμα 4.1.1: Περιγραφή λειτουργίας εφαρμογής

Αρχικά, ορίζονται στον χάρτη τα σημεία που το μη επανδρωμένο θα επισκεφτεί. Αφού απογειωθεί, αν είναι ενεργοποιημένη η αυτόματη λήψη γίνονται φωτογραφίες κάθε 5 δευτερόλεπτα, και αποθηκεύονται στην Android συσκευή που εκτελείται η εφαρμογή. Αν υπάρχει διαθέσιμος υπολογιστής του οποίου η IP έχει δοθεί πριν την έναρξη της αποστολής, αποστέλλονται εκεί οι φωτογραφίες και τρέχει ο αλγόριθμος εντοπισμού αντικειμένων.

Ο αλγόριθμος εντοπισμού που εκτελείται είναι το YOLOv3. Το εκπαιδευμένο μοντέλο που χρησιμοποιείται μπορεί να εντοπίσει 80 διαφορετικά αντικείμενα. Για τους σκοπούς που εξυπηρετεί η συγκεκριμένη υλοποίηση ορισμένα από τα αντικείμενα δεν έχουν ιδιαίτερο ενδιαφέρον και αν εντοπιστεί κάποιο από αυτά δεν σχεδιάζεται στην τελική εικόνα. Γίνεται χρήση μόνο των 24 πρώτων αντικειμένων που περιλαμβάνουν ανθρώπους, ζώα και οχήματα. Όταν εντοπιστεί ένα από αυτά, δημιουργείται ένα περίγραμμα που το περιλαμβάνει και σημειώνεται το όνομα και ο βαθμός εμπιστοσύνης.

Υπάρχουν πάρα πολλά pre-trained μοντέλα σχετικά με τον εντοπισμό αντικειμένων. Αυτό που κάνει το YOLO να ξεχωρίζει και οδήγησε στην επιλογή του είναι η ταχύτητα που έχει σχετικά με όλα τα υπόλοιπα. Εξίσου βασικό χαρακτηριστικό είναι ότι παρέχει τη δυνατότητα στον χρήστη να επιλέξει την αναλογία χρόνου εκτέλεσης και ακρίβειας. Αυτό επιτυγχάνεται με την αλλαγή μεγέθους της εικόνας, χωρίς να απαιτείται επανεκπαίδευση του μοντέλου. Σε μεγαλύτερα μεγέθη εικόνας μπορούν να εντοπιστούν και μικρότερα αντικείμενα, το οποίο όμως οδηγεί σε μεγαλύτερο χρόνο επεξεργασίας. Η επιλογή αυτή είναι ένα

ακόμα χαρακτηριστικό που συμβάλλει στο να μη δημιουργούνται περιορισμοί και ο χρήστης να προσαρμόζει την εφαρμογή στις ανάγκες του.



Σχήμα 4.1.2: Σύγκριση μοντέλων αναγνώρισης

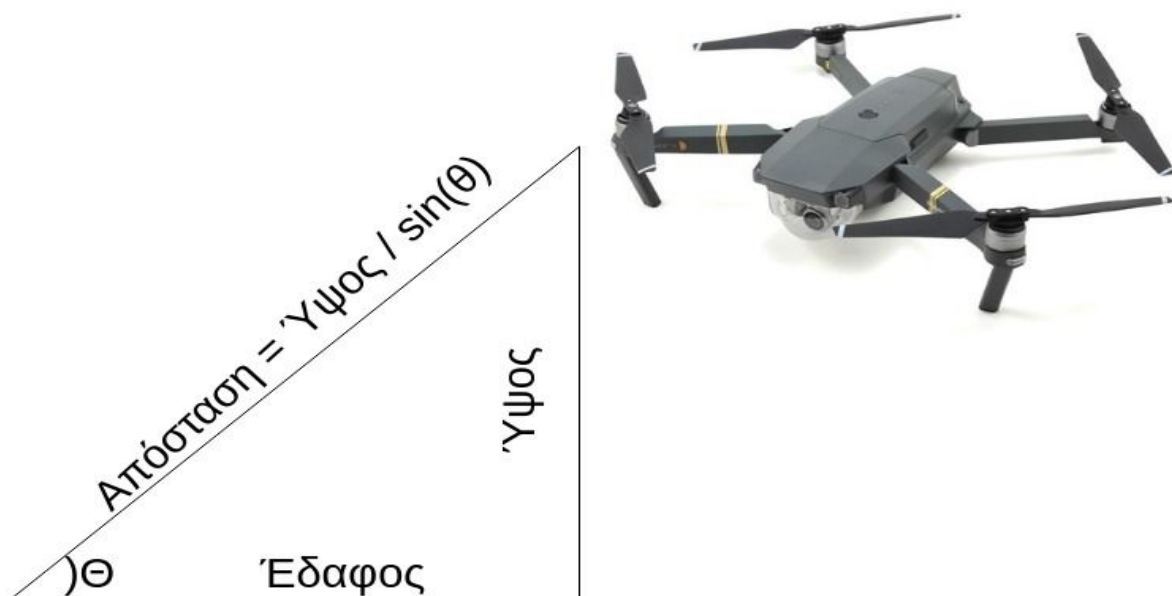
Στο παραπάνω διάγραμμα γίνεται σύγκριση μεταξύ του YOLO και διαφόρων άλλων μοντέλων εντοπισμού. Η ουσιαστικότερη διαφορά του με τα υπόλοιπα μοντέλα είναι ο τρόπος που κάνει τον εντοπισμό. Τα περισσότερα χρησιμοποιούν classifiers ή localizers σε πολλά διαφορετικά σημεία της εικόνας, με τις περιοχές που εμφανίζουν μεγάλη πιθανότητα να είναι και τα σημεία ενδιαφέροντος. Αντίθετα το YOLO χρησιμοποιεί ένα ενιαίο νευρωνικό δίκτυο σε ολόκληρη την εικόνα. Αυτό το δίκτυο χωρίζει την εικόνα σε κομμάτια και βρίσκει τις αντίστοιχες πιθανότητες για κάθε σημείο. Αυτό το κάνει σχεδόν χίλιες φορές γρηγορότερο συγκριτικά με το Fast R-CNN [20].

Για τον βέλτιστο εντοπισμό αντικειμένων πολύ σημαντική είναι η κλίση που έχει το gimbal της κάμερας. Σε περιπτώσεις που επιλέγεται χαμηλό υψόμετρο πτήσης του αεροσκάφους και η αντίστοιχη κλίση πρέπει να είναι εξίσου μικρή. Σε αντίθετη περίπτωση ο εντοπισμός συνήθως δεν θα είναι εφικτός, αφού στις φωτογραφίες οι οντότητες των αντικειμένων δεν θα φαίνονται ολοκληρωμένα. Πιο συγκεκριμένα είναι αδύνατος ο εντοπισμός ανθρώπων χρησιμοποιώντας μόνο το κεφάλι τους ή οχημάτων μέσω μόνο της οροφής. Σε αυτή την υλοποίηση οι 2 από τους 3 άξονες περιστροφής του gimbal διατηρούνται σταθεροί και μεταβάλλεται μόνο ο gimbal pitch axis.



Σχήμα 4.1.3: Λειτουργία gimbal

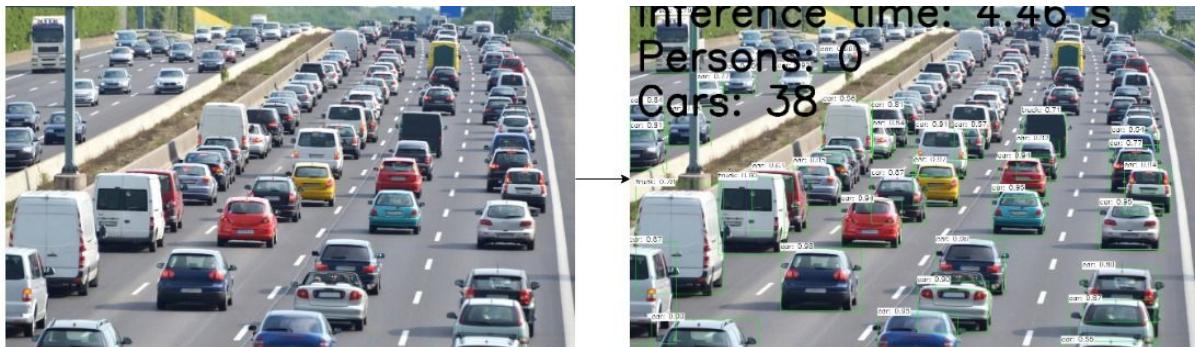
Οι προτεινόμενες τιμές είναι από -30 έως και -60 μοίρες ανάλογα το υψόμετρο. Μαθηματικά, η απόσταση σε ευθεία γραμμή μεταξύ κάμερας και εδάφους είναι: **Ύψος/sin(-pitch)**. Παράδειγμα σε περίπτωση που το ύψος είναι 10 μέτρα και η κλίση -45 μοίρες, η απόσταση θα είναι περίπου ίσο με 14 μέτρα. Αν όμως η κλίση ήταν -60 μοίρες η αντίστοιχη απόσταση υπολογίζεται στα 11.5 μέτρα. Όσο η απόσταση πλησιάζει την τιμή του ύψους τόσο πιο δύσκολος θα είναι και ο εντοπισμός αφού η εικόνες θα περιλαμβάνουν μόνο την πάνω όψη των αντικειμένων. Στις -90 μοίρες η κάμερα είναι τελείως κάθετη προς το έδαφος.



Σχήμα 4.1.4: Απόσταση κάμερας UAV από αντικείμενα

Πρέπει επίσης να σημειωθεί ότι όσο η γωνία μεγαλώνει αυξάνεται και το έδαφος που καλύπτει η κάμερα. Χρειάζεται προσοχή όμως ώστε να βρισκόμαστε εντός των ορίων στα οποία είναι εφικτή η αναγνώριση των αντικειμένων.

Εάν υπάρχει διαθέσιμος υπολογιστής κάθε εικόνα αποστέλλεται μέσω Server Socket που δημιουργείται σε Java. Αφού ληφθεί εκτελείται ο αλγόριθμος αναγνώρισης γραμμένος σε python. Τα αποτελέσματα για κάθε εικόνα είναι όπως το παράδειγμα που ακολουθεί.



Σχήμα 4.1.5: Εικόνα μετά την εκτέλεση του αλγορίθμου

Όπως φαίνεται έχουν εντοπιστεί 36 αυτοκίνητα. Περιγράμματα έχουν τοποθετηθεί και στις 36 περιπτώσεις. Επειδή σε αρκετές περιπτώσεις θα υπάρχουν πολλά αντικείμενα στον ίδιο χώρο, οι γραμμές το περιγράμματος είναι λεπτές ώστε να μην υπάρχει επικάλυψη. Λόγω της πολύ καλής ανάλυσης των εικόνων λεπτομέρειες φαίνονται κάνοντας ζουμ στην εικόνα. Επίσης σε κάθε εικόνα, στο πάνω αριστερά μέρος αναγράφονται ο αριθμός των ατόμων και οχημάτων που εντοπίστηκαν.

4.2 Χρήσεις Εφαρμογής

Όπως αναφέρθηκε και στις προηγούμενες ενότητες, η κυριότερη διαφοροποίηση της εφαρμογής είναι ότι καλύπτει πολλές περιοχές προβλημάτων. Παρόλα αυτά ακολουθεί μια συνοπτική παρουσίαση μερικών βασικών περιοχών που καλύπτει. Γενικότερα τα προβλήματα αυτά σχετίζονται με καταστάσεις έκτακτης ανάγκης, θέματα ασφαλείας, εντοπισμού, αλλά και απλά καθημερινά προβλήματα..

Έρευνα Αγνοουμένων

Μια αρκετά συνηθισμένη και ταυτόχρονα πολύ χρήσιμη εφαρμογή είναι ο εντοπισμός ανθρώπων σε περίπτωση που αγνοούνται. Είναι αρκετές οι περιπτώσεις που ενώ είναι γνωστή η περιοχή που εθεάθη κάποιο άτομο τελευταία φορά, η έκταση βρίσκεται αποκεντρωμένη, ενώ το μέγεθος της είναι αρκετά μεγάλο. Όμως το προσωπικό που ασχολείται με την έρευνα δεν είναι επαρκές, με αποτέλεσμα να απαιτείται αρκετός κρίσιμος χρόνος ώστε να διερευνηθούν όλα τα σημεία. Πιο συγκεκριμένα όταν πρέπει να επισκεφθούν μεγάλες μη κατοικήσιμες περιοχές, το μη

επανδρωμένο αεροσκάφος μπορεί να τις σκανάρει εύκολα και γρήγορα. Με αυτό τον τρόπο η έρευνα είναι αρκετά αποτελεσματικότερη. Δε δαπανάται άσκοπα ανθρώπινο δυναμικό, αλλά ερευνούν μόνο μέρη που το μη επανδρωμένο επισήμανε ως σημεία ενδιαφέροντος μέσω του γεωγραφικού μήκους και πλάτους των φωτογραφιών στις οποίες εντοπίστηκαν ανθρώπινες οντότητες. Έτσι χρησιμοποιώντας ένα ή και περισσότερα drones δημιουργείται μια πρώτη εικόνα για τις ανθρώπινες οντότητες που υπάρχουν σε αυτή την περιοχή. Ακόμη και αν δεν εντοπιστεί το άτομο σε αυτές τις φωτογραφίες, μπορούν να υπάρξουν κάποια άλλα στοιχεία, όπως το μεταφορικό μέσο που χρησιμοποίησε καθώς και πληροφορίες σχετικά με την προσβασιμότητα σε διάφορα σημεία της περιοχής.

Καταμέτρηση και εκκένωση

Συχνά σε διάφορα ατυχήματα καθώς και φυσικές καταστροφές κρίνεται αναγκαία η εκκένωση ανοιχτών χώρων. Για να πραγματοποιηθεί με ασφάλεια και ταχύτητα είναι πολύ σημαντικό να γνωρίζουμε εκ των προτέρων τον αριθμό των ατόμων, έστω και κατά προσέγγιση. Αυτό διευκολύνει την επιχείρηση της εκκένωσης. Η εφαρμογή που έχει υλοποιηθεί μπορεί να βοηθήσει και εδώ. Τοποθετώντας περιμετρικά waypoints γύρω από την περιοχή ενδιαφέροντος και ορίζοντας το κατάλληλο ύψος μπορεί να πραγματοποιηθεί μια καταμέτρηση με αρκετά μεγάλη ακρίβεια. Επίσης χρησιμοποιώντας ως ταχύτητα πτήσης low τα μη επανδρωμένα αεροσκάφη μπορούν να καθοδηγήσουν τον κόσμο προς τα σημεία εξόδου. Έτσι πραγματοποιείται ταυτόχρονα η χρονοβόρα διαδικασία της καταμέτρησης αυτοματοποιημένα, και η καθοδήγηση των ατόμων με ελάχιστη ανθρώπινη παρέμβαση πιθανώς αποτελεσματικότερα, και εμφανώς γρηγορότερα.

Ασφάλεια προσωπικών περιουσιών

Με την εγκληματικότητα να βρίσκεται στα ύψη αρκετοί είναι εκείνοι που θέλουν να βεβαιώνονται ανά τακτά χρονικά διαστήματα ότι οι περιουσίες τους δεν αντιμετωπίζουν κίνδυνο κλοπής. Επίσης πολύ σημαντικό είναι να εξασφαλίζεται και η προσωπική τους ασφάλεια. Η εφαρμογή είναι σε θέση να εντοπίζει και ανθρώπους και αυτοκίνητα. Συνεπώς μπορεί να ανιχνευθεί κάποια ύποπτη κίνηση όπως άνθρωποι σε μέρη που δεν πρέπει αλλά και ύποπτα οχήματα. Συνεπώς υπάρχουν αρκετές προοπτικές για χρήση από εταιρείες security και την αστυνομία. Με προγραμματισμένες περιπολίες ανά τακτά χρονικά διαστήματα σε κατοικημένες περιοχές, αλλά ακόμα πιο εύκολα σε βιομηχανικές εγκαταστάσεις είναι εφικτό να υπάρχει μια καλή εικόνα για την κίνηση που υπάρχει, κυρίως σε βραδινές ώρες. Αργά το βράδυ σε εργοστάσια και άλλες παρόμοιες μεγάλες εκτάσεις, οποιοσδήποτε εντοπισμός κίνησης μπορεί να θεωρηθεί ύποπτος και απαιτείται περαιτέρω διερεύνηση. Τελικά, με μόνο 1-2 άτομα για το χειρισμό της εφαρμογής μπορεί να συντονιστεί μια οργανωμένη περιπολία με συνεχές ενημέρωση στην ομάδα φύλαξης του εκάστοτε κτιρίου.

Κλοπή και εγκατάλειψη οχημάτων

Πολύ συχνά σε κλοπές αυτοκινήτων που χρησιμοποιούνται για ληστείες και άλλες εγκληματικές ενέργειες συνηθίζεται να ακολουθεί η εγκατάλειψη τους σε απομακρυσμένες περιοχές. Συνεπώς εδώ παρατηρείται και μια περισσότερο επαγγελματική χρήση της εφαρμογής που θα μπορούσε να συμβάλλει στον εντοπισμό εγκαταλελειμμένων οχημάτων. Αφορά κυρίως περιπτώσεις που η εγκατάλειψη γίνεται σε απομονωμένες περιοχές, κάτι που περιορίζει την πιθανότητα να υπάρχουν αρκετά άλλα οχήματα εκεί. Με αυτόν τον τρόπο εντοπισμός κάποιου οχήματος το κάνει αυτομάτως υποψήφιο προς αναζήτηση. Η παραπάνω χρήση θα μπορούσε να εξειδικευτεί ακόμα περισσότερο, αν το μοντέλο μπορούσε να εντοπίζει αυτοκίνητα συγκεκριμένου χρώματος, κάτι που θα περιόριζε αρκετά την αναζήτηση.

Εντοπισμός και παρακολούθηση ζώων

Αυτή η περίπτωση είναι παρόμοια με τη πρώτη με τη διαφορά ότι αφορά τα ζώα. Πιο συγκεκριμένα όταν αναζητούνται κοπάδια ζώων είτε για κυνήγι, είτε για σαφάρι, η αναζήτηση μπορεί να πραγματοποιείται πιο οργανωμένα και όχι στα τυφλά όπως συνηθίζεται. Αυτό συνεπάγεται με εξοικονόμηση χρόνου και καυσίμων, σκεπτόμενοι τις μεγάλες εκτάσεις που διερευνώνται σε κάθε περίπτωση. Παρομοίως μπορεί να χρησιμοποιηθεί για τον εντοπισμό συγκεκριμένων ζώων που βρίσκονται ελεύθερα στη φύση, κυρίως για ερευνητικούς σκοπούς.

Ενημέρωση σχετικά με την κίνηση σε αυτοκινητόδρομους

Ανάλογα με την ώρα της ημέρας σε πολλούς αυτοκινητοδρόμους η κίνηση είναι απελπιστική. Τα αυτοκίνητα λόγω του μεγέθους τους εντοπίζονται αρκετά εύκολα και συνήθως με μεγάλο δείκτη εμπιστοσύνης. Άρα μια ακόμα δυνατότητα που προσφέρεται μέσω της εφαρμογής είναι η καταμέτρηση των αυτοκινήτων σε δρόμους που συχνά επικρατεί κυκλοφοριακό πρόβλημα. Έτσι αυτό μπορεί να αξιοποιηθεί σε συνδυασμό με μια πλατφόρμα ενημέρωσης που θα πληροφορεί τους χρήστες για την κατάσταση που επικρατεί σε διάφορους δρόμους ώστε να είναι σε θέση να επιλέξουν εκείνο με τη λιγότερη κίνηση.

Αυτές είναι ορισμένες από τις περιπτώσεις που η εφαρμογή μπορεί να συμβάλλει αρκετά σε ανάγκες προσωπικού επιπέδου καθώς και να παρέχει βοήθεια και λύση σε κοινωνικά προβλήματα.

Κεφάλαιο 5

Κώδικας και Γραφικό Περιβάλλον

5.1 Υλοποίηση Εφαρμογής

5.1.1 Android App

Ο κώδικας της εφαρμογής είναι γραμμένος σε Java και για την υλοποίηση του χρησιμοποιήθηκε το Android studio. Για τις περισσότερες λειτουργίες της εφαρμογής χρησιμοποιείται το Mobile SDK που παρέχει η εταιρεία DJI. Η εφαρμογή αποτελείται από 3 διαφορετικά Activities. Αρχικά στο πρώτο γίνεται έλεγχος σχετικά με τα δικαιώματα που απαιτούνται, και αν αυτά έχουν δοθεί ακολουθεί η εγγραφή. Αφού ολοκληρωθεί η εγγραφή, το συγκεκριμένο activity τερματίζεται και γίνεται η εκκίνηση του επόμενου. Το πρώτο δεν περιλαμβάνει αλληλεπίδραση με το χρήστη, παρά μόνο κατά το request των δικαιωμάτων.

```
/**
 * Checks if there is any missing permissions, and
 * requests runtime permission if needed.
 */
private void checkAndRequestPermissions() {
    // Check for permissions
    for (String eachPermission : REQUIRED_PERMISSION_LIST) {
        if (ContextCompat.checkSelfPermission(context, this, eachPermission)
            != PackageManager.PERMISSION_GRANTED) {
            missingPermission.add(eachPermission);
        }
    }
    // Request for missing permissions
    if (!missingPermission.isEmpty() && Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        ActivityCompat.requestPermissions(activity, this,
            missingPermission.toArray(new String[missingPermission.size()]),
            REQUEST_PERMISSION_CODE);
    }
}
```

Σχήμα 5.1.1.1: Κώδικας ελέγχου δικαιωμάτων εφαρμογής

```

private void startSDKRegistration() {
    if (isRegistrationInProgress.compareAndSet( expect: false, update: true)) {
        AsyncTask.execute(() → {
            showToast( string: "registering, pls wait...");
            DJISDKManager.getInstance().registerApp(getApplicationContext(),
                new DJISDKManager.SDKManagerCallback() {
                    @Override
                    public void onRegister(DJIErrors djiError) {
                        if (djiError == DJISDKError.REGISTRATION_SUCCESS) {
                            DJILog.e( s: "App registration", DJISDKError.
                                REGISTRATION_SUCCESS.getDescription());
                            showToast( string: "Register Success");
                        } else {
                            showToast( string: "Register sdk fails, check network is available");
                        }
                        Log.v(TAG, djiError.getDescription());
                    }
                }
            );
        });
    }
}

```

Σχήμα 5.1.1.2: Κώδικας για εγγραφή εφαρμογής

Για την εγγραφή χρησιμοποιείται η μέθοδος `registerApp` από το πακέτο `dji.sdk.sdkmanager`. Η εγγραφή πραγματοποιείται μόνο στην αρχικοποίηση της εφαρμογής, δηλαδή στην πρώτη εκτέλεση μετά την εγκατάσταση. Πραγματοποιείται σύνδεση σε ένα server της DJI για να επαληθευτεί το Application Key. Το παραπάνω key έχει δημιουργηθεί μέσω του Developer Center της DJI. Εκτός από αυτό το key χρησιμοποιούνται κι άλλες πληροφορίες σχετικά με τη συσκευή, όπως Device UUID, όνομα και έκδοση πλατφόρμας, μοντέλο της συσκευής και άλλα παρόμοια χαρακτηριστικά. Ακόμα χρησιμοποιείται το callback `SDKManagerCallback` που περιέχει τις μεθόδους **`onRegister`**, **`onProductConnect`**, **`onProductDisconnect`**, **`onComponentChange`**. Η πρώτη ενημερώνει αν η εγγραφή πραγματοποιήθηκε επιτυχώς ή όχι και τις ενέργειες που ακολουθούν. Οι υπόλοιπες 3 εκτελούνται σε περίπτωση που το προϊόν συνδεθεί, αποσυνδεθεί ή γίνει κάποια αλλαγή. Ο χρήστης ενημερώνεται για κάθε μια από αυτές τις αλλαγές με μήνυμα στην οθόνη.

```

public void onRegister(DJIErrors djiError) {
    if (djiError == DJISDKError.REGISTRATION_SUCCESS) {
        DJILog.e( s: "App registration", DJISDKError.REGISTRATION_SUCCESS.getDescription());
        showToast( string: "Register Success");
    } else {
        showToast( string: "Register sdk fails, check network is available");
    }
    Log.v(TAG, djiError.getDescription());
}

@Override
public void onProductDisconnect() {
    Log.d(TAG, msg: "onProductDisconnect");
    showToast( string: "Product Disconnected");
}

@Override
public void onProductConnect(BaseProduct baseProduct) {
    Log.d(TAG, String.format("onProductConnect newProduct:%s", baseProduct));
    showToast( string: "Product Connected");
}

```

Σχήμα 5.1.1.3: Κώδικας για έλεγχο επιτυχούς εγγραφής

Στο δεύτερο Activity εμφανίζεται το Status της εφαρμογής, δηλαδή αν έχει συνδεθεί κάποια συσκευή ή όχι. Για να μπορέσει ο χρήστης να μεταβεί στο 3ο Activity που είναι και το λειτουργικό πρέπει κάποια συσκευή να είναι συνδεδεμένη. Ο έλεγχος σύνδεσης πραγματοποιείται με έναν BroadcastReceiver ο οποίος αρχικοποιείται όταν ξεκινήσει η διεργασία. Μόλις εντοπιστεί κάποια αλλαγή στις συνδεδεμένες συσκευές, εκτελείται μια συνάρτηση που ελέγχει αν είναι συνδεδεμένο κάποιο μη επανδρωμένο αεροσκάφος. Αν ο δείκτης που επιστρέφεται δεν είναι NULL ενημερώνεται το status, και ο χρήστης μπορεί να ξεκινήσει το λειτουργικό Activity με το πλήκτρο START.

```

protected BroadcastReceiver mReceiver = (context, intent) -> {
    refreshSDKRelativeUI();
};

private void refreshSDKRelativeUI() {
    BaseProduct mProduct = RegnmoreApplication.getProductInstance();

    if (null != mProduct && mProduct.isConnected()) {

        Log.v(TAG, msg: "refreshSDK: True");
        mBtnOpen.setEnabled(true);

        String str = mProduct instanceof Aircraft ? "DJI Aircraft" : "DJI HandHeld";
        if (null != mProduct.getModel()) {
            str=(mProduct.getModel().getDisplayName());
        }
        mTextConnectionStatus.setText("Status: " + str + " connected");

    } else {

        Log.v(TAG, msg: "refreshSDK: False");
        mBtnOpen.setEnabled(false);

        //mTextProduct.setText(R.string.product_information);
        mTextConnectionStatus.setText("Status: No Product Connected");

    }
}

```

Σχήμα 5.1.1.4: Κώδικας ελέγχου για συνδεδεμένο προϊόν

Για τον παραπάνω έλεγχο χρησιμοποιείται η κλάση BaseProduct. Όλα τα προϊόντα της DJI είναι υποκλάσεις της παραπάνω. Μπορούν να προσπελαστούν μέσω της μεθόδου getProduct της κλάσης DJISDKManager. Με αυτόν τον τρόπο λειτουργεί και η μέθοδος getProductInstance που φαίνεται στον κώδικα. Πιο συγκεκριμένα:

```

/**
 * This function is used to get the instance of DJIBaseProduct.
 * If no product is connected, it returns null.
 */
public static synchronized BaseProduct getProductInstance() {
    if (null == mProduct) {
        mProduct = DJISDKManager.getInstance().getProduct();
    }
    return mProduct;
}

```

Σχήμα 5.1.1.5: Κώδικας getProduct

Έτσι αν δεν επιστρέψει null και ελεγχθεί και τυπικά μέσω της boolean μεθόδου isConnected που πρέπει να επιστρέψει true, σημαίνει ότι υπάρχει συνδεδεμένο αεροσκάφος και η εφαρμογή μπορεί να προχωρήσει στο επόμενο Activity. Ενεργοποιείται το πλήκτρο START ενώ ταυτόχρονα εμφανίζεται το μοντέλο που έχει συνδεθεί.

Στο Main Activity παρέχονται όλες οι διαθέσιμες λειτουργίες και οι πληροφορίες σχετικά με την κατάσταση του αεροσκάφους. Μέσω του UX SDK παρέχονται κάποια έτοιμα widgets που παρέχουν κυρίως ενημέρωση στον χρήστη. Δεν χρειάζεται η συγγραφή κώδικα παρά μόνο η ενσωμάτωση τους στο layout file. Αναλυτικότερα για το status bar έχουν προστεθεί: Pre Flight Status Widget, Flight Mode Widget, GPS Signal Widget, Vision Widget, Remote Control Signal Widget, WiFi Signal Widget, Battery Widget και Connection Widget. Επιπροσθέτως περιλαμβάνονται το Return Home Widget, Camera Controls Widget, Camera Setting Exposure Panel, Camera Setting Advanced Panel και PreFlight CheckList Panel. Η χρήση τους αναλύεται στην επόμενη ενότητα.

Πολύ βασική είναι η συνάρτηση που χρησιμοποιείται για τη λήψη φωτογραφιών. Η κάμερα ρυθμίζεται σε λειτουργία SINGLE, εισάγεται μια μικρή καθυστέρηση για να εφαρμοστούν οι ρυθμίσεις και στη συνέχεια γίνεται η λήψη. Έπειτα εμφανίζεται στην οθόνη το γεωγραφικό μήκος και πλάτος που υπήρχε κατά την χρονική στιγμή που έγινε η κλήση της συνάρτησης.

```
private void captureAction() {
    final dji.sdk.camera.Camera camera = RegnmoreApplication.getCameraInstance();
    if (camera != null) {
        SettingsDefinitions.ShootPhotoMode photoMode = SettingsDefinitions.ShootPhotoMode.SINGLE;
        camera.setShootPhotoMode(photoMode, (djiError) -> {
            if (null == djiError) {
                handler.postDelayed(() -> {
                    camera.startShootPhoto((djiError) -> {
                        if (djiError == null) {
                            double latitude = mFlightController.getState().getAircraftLocation().getLatitude();
                            double longitude = mFlightController.getState().getAircraftLocation().getLongitude();
                            setResultToToast("Latitude:" + latitude + ",Longitude:" + longitude);
                            coordinates.add("Latitude:" + latitude + ",Longitude:" + longitude);
                        } else {
                            setResultToToast(djiError.getDescription());
                        }
                    });
                }, delayMillis: 2000);
            }
        });
    } else {
        setResultToToast("getCameraInstance returned null");
    }
}
```

Σχήμα 5.1.1.6: Κώδικας μεθόδου για λήψη φωτογραφιών

Με ανάλογο τρόπο όπως πριν με την μέθοδο getCamera επιστρέφεται δείκτης που δείχνει στην κάμερα του αεροσκάφους. Εάν δεν είναι null ξεκινάει η λήψη. Ελέγχεται η επιτυχία μέσω Completion callback Σε περίπτωση επιτυχίας εκτυπώνεται το γεωγραφικό μήκος και πλάτος, αλλιώς το σφάλμα που προέκυψε.

```

public static synchronized Camera getCameraInstance() {

    if (getProductInstance() == null) return null;

    Camera camera = null;

    if (getProductInstance() instanceof Aircraft){
        camera = ((Aircraft) getProductInstance()).getCamera();
    } else if (getProductInstance() instanceof HandHeld) {
        camera = ((HandHeld) getProductInstance()).getCamera();
    }

    return camera;
}

```

Σχήμα 5.1.1.7: Κώδικας getCamera

Για να μπορέσει να γίνει upload της αποστολής υπάρχουν κάποιες προϋποθέσεις. Το waypointMissionBuilder πρέπει να έχει αρχικοποιηθεί και να μην είναι NULL. Αυτό γίνεται ή κατά τη διάρκεια που ορίζονται τα waypoints στο χάρτη ή κατά την εκτέλεση της συνάρτησης configWayPointMission που καλείται μόλις γίνει το configuration της αποστολής.

```

@Override
public void onMapClick(LatLng point) {
    if (isAdd == true) {
        markWaypoint(point);
        Waypoint mWaypoint = new Waypoint(point.latitude, point.longitude, altitude);
        if (waypointMissionBuilder != null) {
            waypointList.add(mWaypoint);
            waypointMissionBuilder.waypointList(waypointList).waypointCount(waypointList.size());
        } else {
            waypointMissionBuilder = new WaypointMission.Builder();
            waypointList.add(mWaypoint);
            waypointMissionBuilder.waypointList(waypointList).waypointCount(waypointList.size());
        }
    } else {
        setResultToToast("Enable ADD to Add Waypoint");
    }
}

private void configWayPointMission() {
    if (waypointMissionBuilder == null) {
        waypointMissionBuilder = new WaypointMission.Builder().finishedAction(mFinishedAction)
            .headingMode(mHeadingMode)
            .autoFlightSpeed(mSpeed)
            .maxFlightSpeed(mSpeed)
            .flightPathMode(WaypointMissionFlightPathMode.NORMAL);
    } else {
        waypointMissionBuilder.finishedAction(mFinishedAction)
            .headingMode(mHeadingMode)
            .autoFlightSpeed(mSpeed)
            .maxFlightSpeed(mSpeed)
            .flightPathMode(WaypointMissionFlightPathMode.NORMAL);
    }
}

```

Σχήμα 5.1.1.8: Κώδικας αρχικοποίησης mission builder

Αφού δοθούν τα waypoints και οι σχετικές ρυθμίσεις ορίζονται στο waypointMissionBuilder η ενέργεια που ακολουθεί μετά την περάτωση της αποστολής, ο προσανατολισμός του αεροσκάφους, το ύψος και η ταχύτητα. Για όλα τα παραπάνω χρησιμοποιείται η κλάση WaypointMissionOperator. είναι το μόνο αντικείμενο που ελέγχει, εκτελεί και παρακολουθεί τις Waypoint Missions. Μέσω αυτής πραγματοποιείται η μεταφόρτωση, η εκκίνηση και κάποια πιθανή διακοπή της αποστολής. Αναλυτικότερα σχετικά με το Waypoint Mission:

Το αεροσκάφος θα ταξιδέψει μεταξύ σημείων αναφοράς, θα εκτελέσει ενέργειες στα σημεία αυτά και θα προσαρμόσει την κατεύθυνση του και το υψόμετρο μεταξύ των σημείων. Τα waypoints είναι φυσικές τοποθεσίες στις οποίες θα πετάξει το αεροσκάφος. Δημιουργώντας μια σειρά σημείων αναφοράς, στην πραγματικότητα, θα προγραμματιστεί ένα σχέδιο πτήσης για να ακολουθήσει το αεροσκάφος. Μπορούν επίσης να προστεθούν δράσεις στα σημεία αυτά, οι οποίες θα πραγματοποιηθούν όταν το αεροσκάφος φτάσει εκεί. Το UAV μετακινείται αυτόματα μεταξύ σε βασική ταχύτητα. Ωστόσο, ο χρήστης μπορεί να αλλάξει την ταχύτητα χρησιμοποιώντας το joystick pitch. Το τηλεχειριστήριο μπορεί να πατηθεί προς τα κάτω για να σταματήσει το αεροσκάφος και να πατηθεί περαιτέρω για να αρχίσει να κάνει το αεροσκάφος να ταξιδέψει πίσω κατά μήκος του μονοπατιού που προήλθε. Όταν ταξιδεύει σε αντίστροφη σειρά, δεν θα εκτελέσει τις προκαθορισμένες ενέργειες που αναφέρθηκαν παραπάνω. Αν το joystick απελευθερωθεί, το αεροσκάφος θα ταξιδέψει ξανά μέσω των σημείων αναφοράς στην αρχική σειρά και θα συνεχίσει να εκτελεί τις ενέργειες σημείου (ακόμη και αν έχει εκτελεστεί προηγουμένως). Αν μέσω της διαδικασίας που αναφέρθηκε φτάσει στο πρώτο σημείο, τότε θα παραμείνει εκεί μέχρι να απελευθερωθεί το joystick. Η συγκεκριμένη διαδικασία δεν υποστηρίζεται από το mav pro όταν χρησιμοποιείται σύνδεση wifi, ενώ δεν υποστηρίζεται καθόλου από το Spark[21].

Μια άλλη εξίσου σημαντική συνάρτηση που χρησιμοποιείται είναι η onNewFile. Καλείται κάθε φορά που εντοπίζεται νέο αρχείο στη μνήμη, δηλαδή κάθε φορά που γίνεται λήψη φωτογραφίας. Κατά την εκτέλεση της γίνεται αποστολή του τελευταίου αρχείου που δημιουργήθηκε στη συσκευή. Μόλις η αποστολή της εικόνας ολοκληρωθεί, αν έχει δοθεί IP υπολογιστή εντός του δικτύου ακολουθεί η αντιγραφή της εικόνας και στον υπολογιστή με χρήση της κλάσης FileSender. Η FileSender χρησιμοποιεί το Server Socket της Java και αναζητά αν είναι διαθέσιμη η διεύθυνση και η πόρτα που έχει δοθεί. Για να πραγματοποιηθεί με επιτυχία η αποστολή ο αντίστοιχος κώδικας θα πρέπει να εκτελείται συνεχώς στον υπολογιστή που είναι διαθέσιμος. Σε αντίθετη περίπτωση θα προκύψει κάποιο exception ενημερώνοντας το χρήστη πως ο κώδικας δεν εκτελείται ή η IP που δόθηκε δεν είναι σωστή.

```

//Download every new file created by camera
@Override
public void onNewFile(@NonNull final MediaFile mediaFile) {
    if ((mediaFile.getMediaType() == MediaFile.MediaType.PANORAMA) ||
        (mediaFile.getMediaType() == MediaFile.MediaType.SHALLOW_FOCUS)) {
        return;
    }
    filenames.add(mediaFile.getFileName());
    names_index = names_index + 1;
    mediaFile.fetchFileData(destDir, s: null, new DownloadListener<String>() {
        @Override
        public void onFailure(DJIErrors error) {
            setResultToToast("Download File Failed" + error.getDescription());
        }

        @Override
        public void onSuccess(String filePath) {
            if(!ip.isEmpty()) {
                File ToSend = new File( pathname: filePath + "/" + mediaFile.getFileName());
                new FileSender(ToSend, ip).execute("");
            }
        }
    })
}

```

Σχήμα 5.1.1.9: Κώδικας μεθόδου OnNewfile

Η FileSender καλεί τη μέθοδο send που πραγματοποιεί την αποστολή του αρχείου στον υπολογιστή.

```

public void Send() {
    try {
        Socket socket = new Socket(IP, port: 2445);
        byte[] bytes = new byte[(int) ToSend.length()];
        InputStream in = new FileInputStream(ToSend);
        OutputStream out = socket.getOutputStream();

        BufferedWriter bufferedWriter = new BufferedWriter(new OutputStreamWriter(out));
        bufferedWriter.write(ToSend.getName());
        bufferedWriter.newLine();
        bufferedWriter.flush();

        out.write(bytes, off: 0, in.read(bytes));

        socket.close();
        in.close();
        out.close();
    }
}

```

Σχήμα 5.1.1.10: Κώδικας για αποστολή φωτογραφιών σε υπολογιστή

Χρησιμοποιείται η πόρτα 2445 και η IP που έχει δοθεί. Πραγματοποιείται η αρχικοποίηση του socket και το αρχείο προς αποστολή αποθηκεύεται στον buffer. Αφού ολοκληρωθεί η διαδικασία όλοι οι πόροι που χρησιμοποιήθηκαν αποδεσμεύονται.

5.1.2 Receiver class

Με μια `while true` συνθήκη ο κώδικας εκτελείται συνεχώς. Απαιτείται η εκκίνηση του από το τερματικό. Διακοπή πραγματοποιείται χρησιμοποιώντας τον συνδυασμό πλήκτρων `Ctrl + C`. Αποδέχεται το socket και διαβάζει νέα αρχεία. Μόλις η εικόνα αποθηκευτεί στον ίδιο φάκελο που βρίσκεται και το αρχείο Java, δίνεται εντολή να εκτελεστεί ο αλγόριθμος εντοπισμού. Η εικόνα εισόδου είναι κάθε φορά η τελευταία που έχει ληφθεί. Για να αποφευχθούν προβλήματα συντονισμού, αφού πραγματοποιηθεί η λήψη της εικόνας, υπάρχει μια μικρή καθυστέρηση (200 ms) προτού ξεκινήσει ο αλγόριθμος.

```
ServerSocket serverSocket = new ServerSocket(2445);
Scanner scanner = null;
while (true) {
    try {
        Socket socket = serverSocket.accept();
        scanner = new Scanner(socket.getInputStream());
        String Name = scanner.nextLine();

        byte[] bytes = readStream(socket.getInputStream());
        FileOutputStream fileOutputStream = new FileOutputStream
        |(new File(Name.replace("FromMyPsshone", "")));
        fileOutputStream.write(bytes, 0, bytes.length);
        fileOutputStream.flush();
        fileOutputStream.close();
    }
}
```

Σχήμα 5.1.2.1: Κώδικας για λήψη φωτογραφιών από υπολογιστή

5.1.3 Object detection algorithm

Για λόγους επεκτασιμότητας, παρόλο που το μοντέλο που έχει επιλεγεί είναι το YOLO με darknet framework, παρέχεται η δυνατότητα να χρησιμοποιηθούν και άλλα. Αναλυτικότερα το python script μπορεί να δεχθεί: `caffe`, `tensorflow`, `torch` και `darknet`. Σε κάθε περίπτωση όμως εκτός από το darknet απαιτούνται και αρχεία που δεν παρέχονται όπως το εκπαιδευμένο μοντέλο και οι ρυθμίσεις του δικτύου. Η διαδικασία είναι ίδια για όλες τις περιπτώσεις, αυτό που αλλάζει είναι η ταχύτητα και η μέθοδος εντοπισμού. Αρχικά φορτώνεται το αρχείου κειμένου που περιέχει τα ονόματα των κλάσεων του μοντέλου αν αυτό δίνεται. Στην συνέχεια ακολουθεί η φόρτωση του εκάστοτε δικτύου χρησιμοποιώντας πληροφορίες που έχουν δοθεί ως ορίσματα. Ο αλγόριθμος εκτελείται, και αν εντοπίσει κάποια κλάση μέσα στο εύρος που έχει δοθεί ακολουθεί ο σχεδιασμός παραλληλογράμμου γύρω από το αντικείμενο. Αν πρόκειται για όχημα ή άνθρωπο αυξάνεται ο μετρητής που

χρησιμοποιείται για την τελική ενημέρωση σχετικά με τον αριθμό ατόμων και οχημάτων που εντοπίστηκαν.

```
def drawPred(classId, conf, left, top, right, bottom):
    # Draw a bounding box.
    return_value=0
    global persons
    global cars
    assert(classId < len(classes))
    if(classId<=23):
        cv.rectangle(frame, (left, top), (right, bottom), (0, 255, 0))
        label = '%.2f' % conf
        # Print a label of class.
        label = '%s: %s' % (classes[classId], label)
        labelSize, baseLine = cv.getTextSize(label, cv.FONT_HERSHEY_SIMPLEX, 0.5, 1)
        top = max(top, labelSize[1])
        cv.rectangle(frame, (left, top - labelSize[1]), (left + labelSize[0], top + baseLine),
            (255, 255, 255), cv.FILLED)
        cv.putText(frame, label, (left, top), cv.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))
        return_value=1
        if(classId == 0):
            persons=persons+1
        if(classId == 2 or classId == 5 or classId == 7):
            cars=cars+1
    return return_value
```

Σχήμα 5.1.3.1: Κώδικας για μέτρηση εντοπισμένων αντικειμένων

Στα οχήματα περιλαμβάνονται Ι.Χ, λεωφορεία και φορτηγά. Στην παραπάνω εικόνα παρατίθεται το κομμάτι κώδικα που αφορά την διαδικασία σχεδιασμού μόλις εντοπιστεί μια από τις επιθυμητές κλάσεις αντικειμένων. Αφού ολοκληρωθεί η χάραξη στα συγκεκριμένα σημεία αναγράφεται στο τερματικό ο αριθμός των αντικειμένων που εντοπίστηκαν καθώς και η κατηγορία τους.

```
Persons found at DJI_0018.JPG : 0
Cars found at DJI_0018.JPG : 2
Persons found at DJI_0019.JPG : 1
Cars found at DJI_0019.JPG : 1
Persons found at DJI_0020.JPG : 2
Cars found at DJI_0020.JPG : 0
Persons found at DJI_0021.JPG : 1
Cars found at DJI_0021.JPG : 0
Persons found at DJI_0022.JPG : 0
Cars found at DJI_0022.JPG : 16
Persons found at DJI_0023.JPG : 0
Cars found at DJI_0023.JPG : 9
Persons found at DJI_0024.JPG : 0
Cars found at DJI_0024.JPG : 11
Persons found at DJI_0025.JPG : 0
Cars found at DJI_0025.JPG : 5
Persons found at DJI_0026.JPG : 0
Cars found at DJI_0026.JPG : 18
Persons found at DJI_0027.JPG : 1
Cars found at DJI_0027.JPG : 10
Persons found at DJI_0028.JPG : 0
Cars found at DJI_0028.JPG : 10
```

Σχήμα 5.1.3.2: Έξοδος τερματικού μετά την εκτέλεση του αλγορίθμου

Η εικόνα αποθηκεύεται με όνομα ίδιο με αυτό της αρχικής. Ο χρήστης μπορεί να αλλάξει την τοποθεσία αποθήκευσης μεταβάλλοντας το παρακάτω path στον κώδικα:

```
label = 'Inference time: %.2f s' % (t / cv.getTickFrequency())
label2= "Persons: "+str(persons)
label3= "Cars: " +str(cars)
cv.putText(frame, label, (15, 80), cv.FONT_HERSHEY_SIMPLEX, 3.0, (0, 0, 0), 7)
cv.putText(frame, label2, (15, 180), cv.FONT_HERSHEY_SIMPLEX, 3.0, (0, 0, 0), 7)
cv.putText(frame, label3, (15, 280), cv.FONT_HERSHEY_SIMPLEX, 3.0, (0, 0, 0), 7)
print("Persons found at "+args.input[-12:]+ " : "+str(persons))
print("Cars found at "+args.input[-12:]+ " : "+str(cars))

#cv.imshow(winName, frame)
out_img='/home/tzagkarakis/diploma/code/opencv_samples/out_768' # Path of save location
out_img=out_img+'/'+args.input[-12:]
```

Σχήμα 5.1.3.3: Κώδικας για αλλαγή τοποθεσίας αποθήκευσης εικόνας

Έχουν δοκιμαστεί 3 διαφορετικά μεγέθη εικόνας. Τα αποτελέσματα των μετρήσεων παρατίθενται σε ενότητα που ακολουθεί. Το μέγεθος μπορεί να μεταβληθεί από την εντολή που δίνεται για εκτέλεση αλλάζοντας τα πεδία width και height. Η εντολή που δίνεται στο τερματικό είναι:

```
python3 object_detection_2.py --input "$file" --config ./yolov3.cfg --model  
./yolov3.weights --framework darknet --classes ./class.txt --width=768  
--height=768 --scale=0.00392
```

Σε περίπτωση που οι φωτογραφίες αποστέλλονται ταυτόχρονα με την εκτέλεση της αποστολής, η εντολή εκτελείται αυτόματα και η αλλαγή θα πρέπει να πραγματοποιηθεί στο ReceiveServer.java στη γραμμή 40 του κώδικα. Τα μεγέθη θα πρέπει να είναι τετραγωνικά, δηλαδή ίδιο ύψος και πλάτος, αλλιώς το yolo μοντέλο δεν θα λειτουργήσει.

```
ExecuteShellCommand com = new ExecuteShellCommand();
System.out.println(com.executeCommand("python3 object_detection.py --input "+ path+"/"
+Name+" --config ./yolov3.cfg --model ./yolo/yolov3.weights --framework darknet --classes ./class.txt
--scale=0.00392 --width=512 --height=512"));
}
```

Σχήμα 5.1.3.4: Κώδικας για αλλαγή μεγέθους εικόνας

5.2 Γραφικό περιβάλλον

5.2.1 Περιήγηση

Κατά την πρώτη εκκίνηση παραχωρούνται τα δικαιώματα που χρειάζεται η εφαρμογή (χρήση τοποθεσίας συσκευής και πρόσβαση στον αποθηκευτικό χώρο). Επιπροσθέτως απαιτείται σύνδεση στο διαδίκτυο κατά την πρώτη εκκίνηση για να είναι δυνατή η εγγραφή της εφαρμογής. Σε αντίθετη περίπτωση είναι αδύνατη η εκτέλεση και εμφανίζεται μήνυμα λάθος που ζητάει να ελεγχθεί η σύνδεση. Όταν η εγγραφή πραγματοποιηθεί εμφανίζεται αντίστοιχο μήνυμα και αναμένεται η σύνδεση του Drone.



Σχήμα 5.2.1.1: Λήψη δικαιωμάτων και εγγραφή

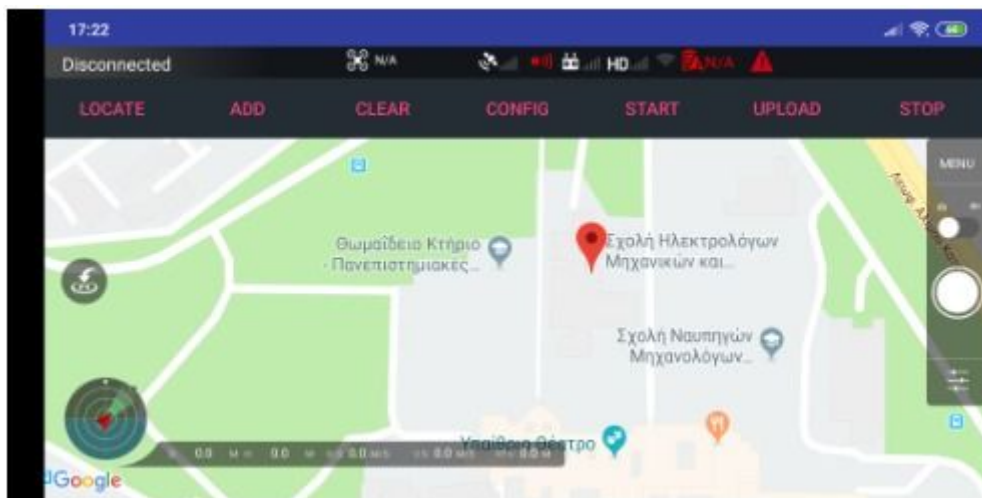
Σε αυτό το στάδιο το πλήκτρο START δεν μπορεί να πατηθεί προτού να γίνει η σύνδεση, και σε αντίθετη περίπτωση ο χρήστης μπορεί να κλείσει την εφαρμογή πατώντας 2 φορές το back button. Μόλις συνδεθεί το drone το START γίνεται διαθέσιμο, ενώ στην οθόνη εμφανίζονται πληροφορίες σχετικές με το μοντέλο αεροσκάφους.



Σχήμα 5.2.1.2: Σύνδεση προϊόντος

Πατώντας το START βρισκόμαστε στο λειτουργικό περιβάλλον της εφαρμογής και πλέον μπορεί να ρυθμιστεί η αποστολή. Προβάλλεται η τοποθεσία του αεροσκάφους μέσω των google maps, ενώ διατίθεται και το πλήκτρο Locate για την εστίαση στο χάρτη.

Για τη δημιουργία μια αποστολής ο χρήστης αρχικά πρέπει να επιλέξει τα σημεία στο χάρτη που επιθυμεί να πετάξει το drone. Αυτό γίνεται με τη χρήση του ADD, ακολουθούμενο από taps στα αντίστοιχα σημεία του χάρτη. Αφού επιλεγθούν τα waypoints, πρέπει να γίνουν ρυθμίσεις που αφορούν την πτήση και την ενέργεια που θα ακολουθήσει μετά την ολοκλήρωση της αποστολής.



Σχήμα 5.2.1.3: Κεντρικό περιβάλλον εφαρμογής

Στο κάτω μέρος της οθόνης υπάρχει μια πυξίδα που υποδεικνύει τον προσανατολισμό. Δίπλα ακριβώς υπάρχουν πληροφορίες σχετικές με την απόσταση από το Home point, το ύψος καθώς και την ταχύτητα του μη επανδρωμένου. Εκτός από όλα τα παραπάνω, στην κορυφή υπάρχει μια μπάρα κατάστασης με διάφορα στοιχεία του αεροσκάφους για την μπαταρία, την σύνδεση κλπ, ενώ πατώντας πάνω εμφανίζονται επιπλέον πληροφορίες. Τέλος παρέχεται η δυνατότητα να γίνει λήψη φωτογραφιών εκτός των προγραμματισμένων, με το κουμπί της κάμερας που υπάρχει στο δεξί μέρος της οθόνης. Στο ίδιο σημείο ο χρήστης μπορεί να κάνει διαφορες ρυθμίσεις της κάμερας. Κατά τη διάρκεια της πτήσης η ακριβής τοποθεσία του αεροσκάφους ενημερώνεται διαρκώς και φαίνεται στο χάρτη. Για όλα τα παραπάνω ακολουθεί λεπτομερή περιγραφή στις επόμενες ενότητες.

Αφού ολοκληρωθούν οι ρυθμίσεις μέσω του config, η αποστολή είναι έτοιμη για μεταφόρτωση. Κάνοντας upload, αν ακολουθήσει το σχετικό μήνυμα επιτυχίας, όλα είναι έτοιμα και η αποστολή ξεκινάει με το πλήκτρο START.

5.2.2 Ανάλυση widgets



Σχήμα 5.2.2.1: Pre Flight Status Widget

Pre Flight Status Widget: Το κείμενο παρουσιάζει μια σύνοψη της κατάστασης ελέγχου κατά την εκκίνηση του αεροσκάφους. Το χρώμα του φόντου αλλάζει ανάλογα με το πόσο σοβαρή είναι η κατάσταση. Το πράσινο δείχνει ότι όλα είναι εντάξει. Το κίτρινο δείχνει ότι κάτι βρίσκεται σε προειδοποιητικό επίπεδο. Το κόκκινο

δείχνει ότι κάτι είναι λάθος και το αεροσκάφος δεν θα απογειωθεί. Πατώντας στο κείμενο θα εμφανιστεί ή εξαφανιστεί το PreFlight CheckList Panel όπως φαίνεται στην εικόνα.

PreFlight CheckList Panel: Πίνακας που εμφανίζει τη λίστα ελέγχου πριν από την πτήση, η οποία περιλαμβάνει μια λίστα στοιχείων (όπως IMU, GPS κ.λπ.) και την κατάστασή τους, τα οποία εμφανίζονται σε μια γενική κατάσταση.

Flight Mode Widget: Η τρέχουσα κατάσταση πτήσης του αεροσκάφους.

GPS Signal Widget: Μικρός αριθμός πάνω από το δορυφορικό εικονίδιο δείχνει τον αριθμό δορυφόρων που είναι διαθέσιμοι. Οι 5 κάθετες γραμμές αντιπροσωπεύουν την τρέχουσα ισχύ σήματος GPS. Το μικρό γράμμα 'R' που εμφανίζεται ή είναι κρυφό δείχνει ότι ο weather RTK είναι ενεργοποιημένος ή όχι.

Vision Widget: Το Widget έχει δύο οπτικές καταστάσεις για να δείξει εάν χρησιμοποιείται οπτική τοποθέτηση. Το λευκό εικονίδιο υποδεικνύει ότι χρησιμοποιείται οπτική τοποθέτηση. Το κόκκινο εικονίδιο υποδεικνύει ότι η οπτική τοποθέτηση δεν χρησιμοποιείται.



No Positioning (Atti)		ATTI	
Overall Status	No Positioning (Atti)		
Flight Mode	ATTI		
Compass	Unknown		
IMU Status	Abnormal		
ESC Status	Normal		
Vision Sensors	Normal		
Radio Channel Quality	Poor		
Remote Controller Mode	Mode 2		
Aircraft Battery	56%		
Remote Controller Battery	46%		
Aircraft Battery Temperature	33° C		
Remaining Storage Capacity	6813MB		
Gimbal Status	Normal		

Σχήμα 5.2.2.2: PreFlight CheckList Panel

Remote Control Signal Widget: Εικονίδιο με συνολικά 5 κάθετες γραμμές αντιπροσωπεύει την ισχύ σήματος μεταξύ RC και αεροσκάφους.

WiFi Signal Widget: Widget που εμφανίζει την ισχύ και το κανάλι του WiFi του αεροσκάφους.

Battery Widget: Το εικονίδιο μπαταρίας έχει δύο χρωματικές καταστάσεις. Το λευκό χρώμα δείχνει ότι όλα είναι εντάξει. Το εικονίδιο θα αλλάξει σε κόκκινο χρώμα εάν το αεροσκάφος χρειάζεται να επιστρέψει στο σπίτι, να προσγειωθεί αμέσως ή η σύνδεση της μπαταρίας είναι κακή. Το κείμενο δείχνει το ποσοστό της φόρτισης που παραμένει στην μπαταρία. Το κείμενο έχει επίσης δύο χρωματικές καταστάσεις. Το πράσινο δείχνει ότι το ποσοστό είναι εντάξει. Το κόκκινο δείχνει ότι το ποσοστό είναι είτε κάτω είτε μόνο για να επιστρέψετε στο σπίτι ή να προσγειωθείτε αμέσως.

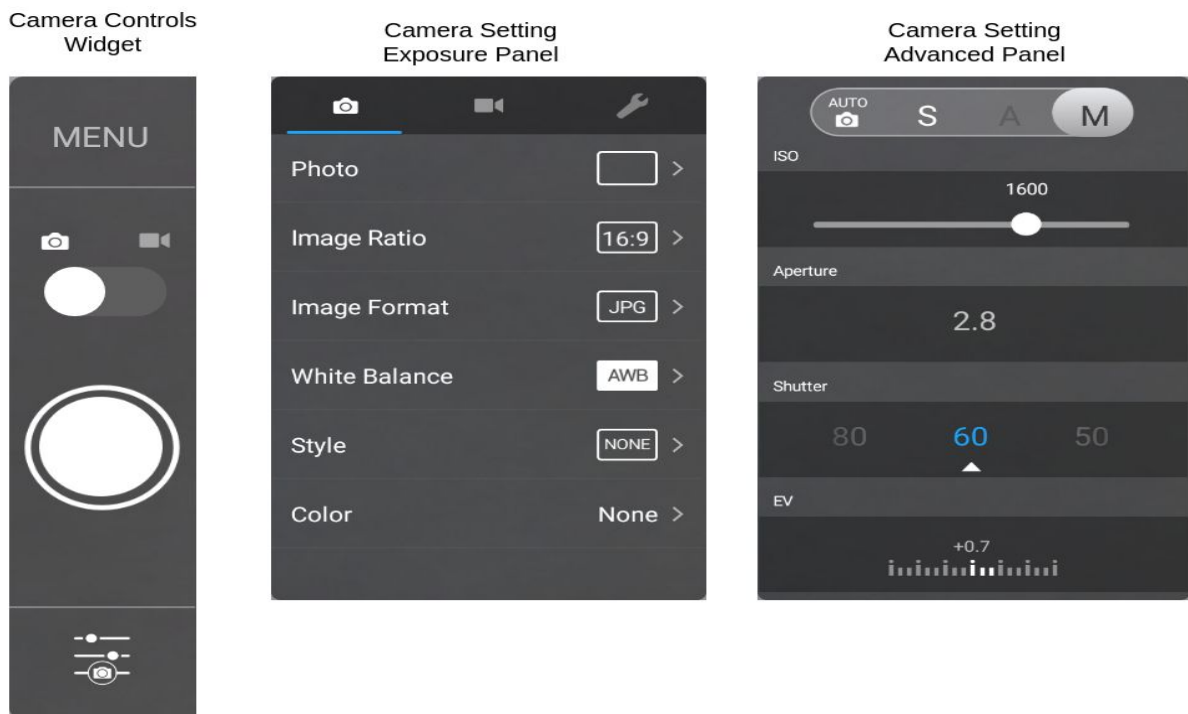
Connection Widget: Το Widget έχει δύο καταστάσεις που αντιπροσωπεύονται με δύο εικονίδια. Το εικονίδιο κόκκινου τριγώνου υποδεικνύει ότι δεν υπάρχει σύνδεση με το προϊόν. Το εικονίδιο του πράσινου κύκλου υποδεικνύει ότι υπάρχει σύνδεση με το προϊόν.

Return Home Widget: Το Widget έχει δύο καταστάσεις με δύο διαφορετικές αλληλεπιδράσεις. Το λευκό εικονίδιο με το βέλος κάτω δείχνει ότι το αεροσκάφος πετάει και η αλληλεπιδράση του widget είναι να αρχίσει να επιστρέφει στο σπίτι. Το κόκκινο εικονίδιο υποδεικνύει ότι το αεροσκάφος επιστρέφει στο σπίτι και η αλληλεπιδράση του widget είναι να ακυρώσει την επιστροφή στο σπίτι.

Camera Controls Widget: Πατώντας το κουμπί Μενού θα γίνει εναλλαγή μεταξύ εμφάνισης και απόκρυψης του Camera Setting Advanced Panel. Πατώντας το κουμπί διακόπτη, θα μεταβείτε στη λειτουργία φωτογραφικής μηχανής ανάμεσα στη λήψη φωτογραφιών και την εγγραφή βίντεο. Πατώντας το κάτω κουμπί θα γίνει εναλλαγή μεταξύ εμφάνισης και απόκρυψης του Camera Setting Exposure Panel.

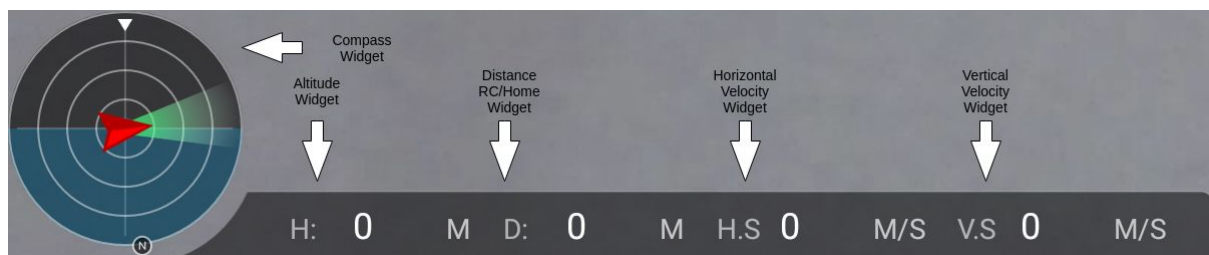
Camera Setting Exposure Panel: Αυτός ο πίνακας εμφανίζει όλες τις ρυθμίσεις της μηχανής που σχετίζονται με την έκθεση. Επιτρέπει την επιλογή λειτουργίας έκθεσης (αυτόματη, προτεραιότητα διαφράγματος, προτεραιότητα κλείστρου και χειροκίνητη) ανάλογα με τη συνδεδεμένη κάμερα. Ανάλογα με τη λειτουργία, μπορεί επίσης να ρυθμιστεί η τιμή ISO, διάφραγμα, ταχύτητα κλείστρου και τιμή αντιστάθμισης έκθεσης.

Camera Setting Advanced Panel: Αυτός ο πίνακας εμφανίζει όλες τις ρυθμίσεις της κάμερας που δεν σχετίζονται με την έκθεση. Διαχωρίζεται σε τρία μέρη: - Ρυθμίσεις εικόνας - Ρυθμίσεις βίντεο - Κοινές ρυθμίσεις.



Σχήμα 5.2.2.3: Camera controls widget, exposure panel και settings advanced panel

Dashboard Widget: Σύνθετο widget που συγκεντρώνει σημαντικές πληροφορίες φυσικής κατάστασης του αεροσκάφους σε πίνακα οργάνων. Περιλαμβάνει το κυκλικό Compass Widget, το Distance Home Widget, το Horizontal Velocity Widget, το Distance RC Widget, το Vertical Velocity Widget και το Altitude Widget.



Σχήμα 5.2.2.4: Dashboard Widget

Compass Widget: Σύνθετο γραφικό στοιχείο που συγκεντρώνει τα δεδομένα θέσης και ύψους σε ένα γραφικό στοιχείο. Το γραφικό στοιχείο δείχνει: Κατεύθυνση του αεροσκάφους σε σχέση με τον πιλότο, βορά σε σχέση με πιλότο και αεροσκάφος Roll και pitch, yaw και pitch του gimbal, θέση αεροσκάφους σε σχέση με τον πιλότο, απόσταση αεροσκάφους από πιλότο και αρχική τοποθεσία

Distance Home Widget: Απόσταση μεταξύ της τρέχουσας θέσης του αεροσκάφους και του home σε μέτρα.

Horizontal Velocity Widget: Οριζόντια ταχύτητα αεροσκάφους σε m / s.

Distance RC Widget: Απόσταση μεταξύ τρέχουσας θέσης του αεροσκάφους και RC (πιλότος) σε μέτρα.

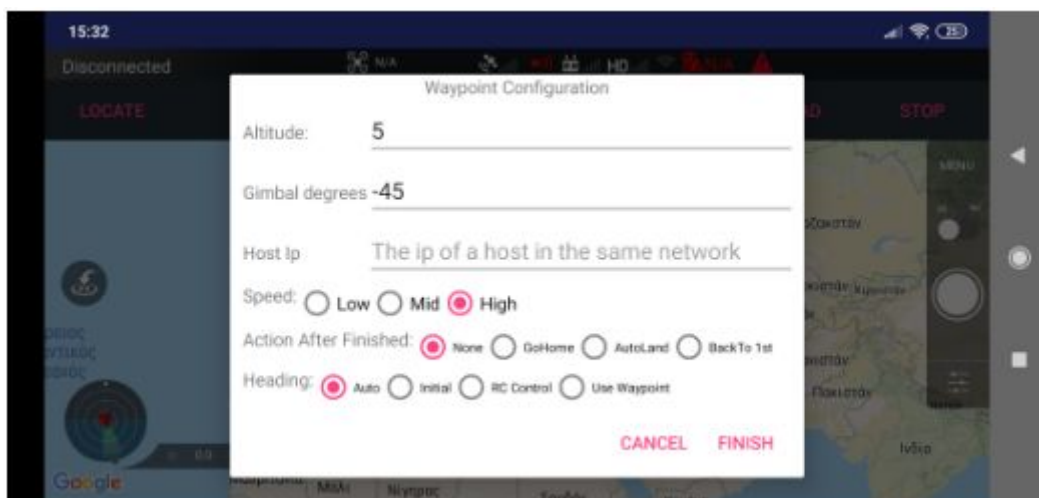
Vertical Velocity Widget: Το κείμενο δείχνει την τρέχουσα κάθετη ταχύτητα του αεροσκάφους σε m / s. Το βέλος δείχνει αν το αεροσκάφος ανεβαίνει ή κατεβαίνει.

Altitude Widget: Τρέχον υψόμετρο του αεροσκάφους σε μέτρα.

5.2.3 Ανάλυση ρυθμίσεων αποστολής

Altitude: Είναι το ύψος που θα πραγματοποιηθεί η πτήση. Ως μηδέν θεωρείται το σημείο απογείωσης. Ποικίλει ανάλογα την εκάστοτε αποστολή, αλλά στην περίπτωση που δεν υπάρχουν εμπόδια, και δεν απαιτείται μεγάλο ύψος είναι καλό να διατηρείται σε χαμηλά επίπεδα ώστε η αναγνώριση να έχει μεγαλύτερη επιτυχία.

Gimbal degrees: Είναι οι μοίρες που περιστρέφεται το gimbal της κάμερας. Προτείνονται τιμές μεταξύ -40 και -60 ώστε να φωτογραφίζονται σωστά οι ανθρώπινες οντότητες και να είναι δυνατός ο εντοπισμός τους.



Σχήμα 5.2.3.1: WayPoint configuration

Host IP: Η Ip του υπολογιστή που βρίσκεται στο ίδιο δίκτυο με την Android συσκευή και θα χρησιμοποιηθεί για την αποστολή και επεξεργασία των εικόνων.

Speed: Είναι η ταχύτητα του αεροσκάφους. Διατίθενται 3 επιλογές: **i)Low** με αντιστοιχία 3 m/s, **ii)Mid** 5 m/s, και **iii)High** 10 m/s.

Action after finished: Η ενέργεια που ακολουθεί μετά το πέρας της αποστολής.
i)None: Το drone παραμένει στο τελευταίο waypoint χωρίς να προσγειωθεί και μπορεί πλέον να ελεγχθεί και απο το controller. **ii)Go Home:** Εάν βρίσκεται περισσότερα από 20 μέτρα μακριά από το home point, θα γυρίσει πίσω και θα προσγειωθεί. Διαφορετικά θα προσγειωθεί στο σημείο που βρίσκεται. **iii)AutoLand:** Προσγειώνεται αυτόματα στο τελευταίο waypoint. **iv)BackTo1st:** Επιστρέφει στο πρώτο waypoint και παραμένει στον αέρα.

Heading: Ο προσανατολισμός καθώς πετάει μεταξύ των waypoints. **i)Auto:** το μπροστά μέρος του αεροσκάφους έχει πάντοτε την ίδια κατεύθυνση με την πτήση. **ii)Initial:** Ο προσανατολισμός παραμένει σταθερός και ίδιος με αυτόν που αποκτήθηκε στο πρώτο waypoint. **iii)RC Control:** Ελέγχεται από το τηλεχειριστήριο. **iv)UseWaypoint:** Ρυθμίζεται σταδιακά καθώς προχωράει σε γειτονικά σημεία.

Αφού οριστούν όλα τα παραπάνω η αποστολή είναι έτοιμη για εκτέλεση. Χρησιμοποιείται το upload για τη μεταφόρτωση στο drone και στη συνέχεια το Start για εκκίνηση. Μόλις το αεροσκάφος απογειωθεί και φτάσει το ζητούμενο ύψος πραγματοποιείται λήψη φωτογραφιών κάθε 5 δευτερόλεπτα, εμφανίζεται το γεωγραφικό μήκος και πλάτος, και αν έχει δοθεί σωστή Ip όλες οι φωτογραφίες αποστέλλονται στον υπολογιστή που εκτελείται ο αλγόριθμος αναγνώρισης. Κατά την διάρκεια της αποστολής ο χρήστης μπορεί να την ακυρώσει χρησιμοποιώντας την επιλογή Stop. Επίσης σε περίπτωση που επιθυμεί να επιστρέψει πίσω το αεροσκάφος και να προσγειωθεί, μπορεί να το κάνει πατώντας την επιλογή Return Home.

Κεφάλαιο 6

Μετρήσεις και αποτελέσματα

Για την διεξαγωγή των μετρήσεων και αποτελεσμάτων χρησιμοποιήθηκε ο επεξεργαστής AMD phenom II X4 965 που διαθέτει ρολόι στα 3.4 GHz. Ακόμα, το αεροσκάφος που συνδέθηκε ήταν το DJI Mavic Pro. Διαθέτει gimbal με δυνατότητα περιστροφής 3 αξόνων. Επιπροσθέτως μπορεί να βγάλει βίντεο 4K/30fps, ενώ η κάμερα του είναι 12 megapixels. Σε ανάλυση αυτό μεταφράζεται ως 4000 pixels πλάτος επί 3000 pixels ύψος. Για λόγους απλότητας και καλύτερης εστίασης ανιχνεύθηκαν μόνο άνθρωποι και οχήματα, ώστε η παρουσίαση των αποτελεσμάτων να είναι περισσότερο εξειδικευμένη. Οι μετρήσεις πραγματοποιήθηκαν στο χώρο του πολυτεχνείου αφού εκεί υπάρχει μεγάλη προσέλευση ατόμων καθώς και οχημάτων. Μέσω της εφαρμογής επιλέχθηκαν αρκετά σημεία στο χάρτη ώστε να πραγματοποιηθεί σχετικά μεγάλης διάρκειας πτήση και να γίνει επαρκής λήψη φωτογραφιών. Επειδή υπήρχαν αρκετά εμπόδια (κτίρια, δέντρα κλπ) που θα μπορούσαν να οδηγήσουν σε συντριβή του μη επανδρωμένου χρησιμοποιήθηκε ύψος 14 μέτρων. Το gimbal της κάμερας ρυθμίστηκε στις 45 μοίρες. Συνεπώς η απόσταση της κάμερας του μη επανδρωμένου από το έδαφος υπολογίζεται στα 19.7 μέτρα σε ευθεία γραμμή. Έγινε διαχωρισμός σε μετρήσεις που αφορούν εντοπισμό ανθρώπινων οντοτήτων και οχημάτων.

6.1 Μετρήσεις

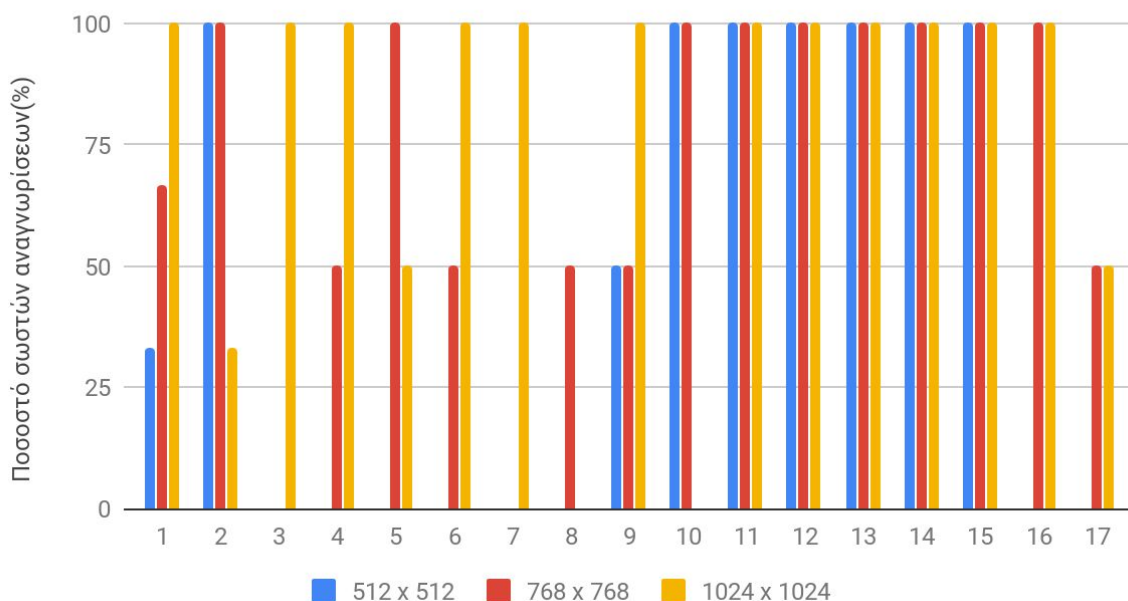
6.1.1 Εντοπισμός ατόμων

Ο αλγόριθμος εντοπισμού αντικειμένων εκτελέστηκε για 3 διαφορετικά μεγέθη εικόνας: 512 x 512, 768 x 768, 1024 x 1024. Τα κριτήρια σύγκρισης είναι η ακρίβεια του μοντέλου, δηλαδή πόσα αντικείμενα εντοπίστηκαν σωστά από το σύνολο αντικειμένων που εμφανίζονται στις εικόνες, και ο χρόνος εκτέλεσης. Αναλυτικότερα σε 14 εικόνες εμφανίζονται κάποια άτομα, ενώ σε 27 υπήρχαν οχήματα. Όλα τα παραπάνω παρουσιάζονται στη συνέχεια με μορφή πινάκων καθώς και διαγραμμάτων. Το μοντέλο YOLO είναι γνωστό για την ταχύτητα του. Η σύγκριση που ακολουθεί εστιάζει περισσότερο στις διαφορές που προκύπτουν όταν μεταβάλλεται το μέγεθος της εικόνας. Σχετικά με την ακρίβεια του, παρόλο που διατηρείται σε υψηλά επίπεδα, πολλές φορές αίτια για μη επιτυχής ανίχνευση ήταν μικρά εμπόδια που κάλυπταν κάποιο μικρό μέρος του αντικειμένου ή η μεγάλη απόσταση. Αρχικά παρουσιάζονται τα ποσοστά ακρίβειας σχετικά με τον επιτυχή εντοπισμό ανθρώπων και οχημάτων.

ID Εικόνας	512 x 512	768 x 768	1024 x 1024
1	33,33333333	66,66666667	100
2	100	100	33,33333333
3	0	0	100
4	0	50	100
5	0	100	50
6	0	50	100
7	0	0	100
8	0	50	0
9	50	50	100
10	100	100	0
11	100	100	100
12	100	100	100
13	100	100	100
14	100	100	100
15	100	100	100
16	0	100	100
17	0	50	50
AVG	46,07843137	71,56862745	78,43137255

Σχήμα 6.1.1.1: Πίνακας μετρήσεων για εντοπισμό ατόμων

Εντοπισμός Ατόμων



Σχήμα 6.1.1.2: Διάγραμμα επιτυχούς εντοπισμού ατόμων

Παρατηρώντας τον πίνακα και το διάγραμμα που παρουσιάζεται προκύπτουν αρκετά συμπεράσματα. Γενικά το μοντέλο φαίνεται να ανταποκρίνεται πολύ καλά αφού στις περισσότερες περιπτώσεις με ανάλυση 1024×1024 εντοπίζει σχεδόν όλα τα άτομα. Είναι πολύ σημαντικό και πρέπει να σημειωθεί ότι σε καμία εικόνα δεν υπήρξε λανθασμένος εντοπισμός. Όσες φορές εντοπίστηκε κάτι ήταν σωστό. Ο εντοπισμός ατόμων είναι αρκετά δυσκολότερος σε σχέση με τα οχήματα. Αυτό γίνεται κατανοητό συγκρίνοντας τα ποσοστά που ακολουθούν και αφορούν τα οχήματα. Η απόκλιση αυτή προκύπτει λόγω της μεγάλης διαφοράς μεγέθους μεταξύ των 2 αντικειμένων.

Όπως ήταν αναμενόμενο υπάρχουν μικρές διακυμάνσεις ανάμεσα στις 3 διαφορετικές μετρήσεις που πραγματοποιήθηκαν. Η πρώτη με ανάλυση εικόνας εισόδου 512×512 φαίνεται να υστερεί αρκετά σε ακρίβεια αφού κατά μέσο όρο εντοπίζει μόνο περίπου το 46 % των ατόμων. Αυτό το αποτέλεσμα είναι αρκετά λογικό αφού η εικόνα είναι πολύ μικρότερη από ότι η αρχική. Παρόλα αυτά το ποσοστό διατηρείται σε πολύ χαμηλά επίπεδα, και είναι σχεδόν απαγορευτική η χρήση της συγκεκριμένης ανάλυσης. Ιδιαίτερα σε περιπτώσεις που σκοπός είναι η καταμέτρηση κατά προσέγγιση, τα αποτελέσματα θα είχαν μεγάλη απόκλιση από την πραγματικότητα. Το συγκεκριμένο μέγεθος προτιμάται μόνο σε περιπτώσεις που τα περιθώρια χρόνου είναι υπερβολικά στενά και δεν υπάρχει εναλλακτική επιλογή.

Αυξάνοντας το μέγεθος τα αποτελέσματα είναι αισθητά βελτιωμένα. Για μέγεθος 768×768 το ποσοστό είναι 71.5 %. Εδώ αξίζει να εξεταστεί αν ο παραπάνω χρόνος που απαιτείται για μεγαλύτερο μέγεθος είναι ανάλογος του βελτιωμένου ποσοστού. Σε αρκετές περιπτώσεις που η προηγούμενη εκτέλεση δεν κατάφερε να εντοπίσει ορισμένα άτομα, η αύξηση της ανάλυσης κατά το ήμισυ κατάφερε να δώσει λύση. Αρκετά εμπόδια καθώς και η απόσταση από το στόχο δυσχεραίνουν την διαδικασία. Για να είναι σε θέση ο αλγόριθμος να εντοπίζει επιτυχώς άτομα σε τέτοια σενάρια απαιτείται μεγαλύτερη ανάλυση, που οδηγεί εν τέλει σε αύξηση των περιοχών ενδιαφέροντος κατά την επεξεργασία της εικόνας.

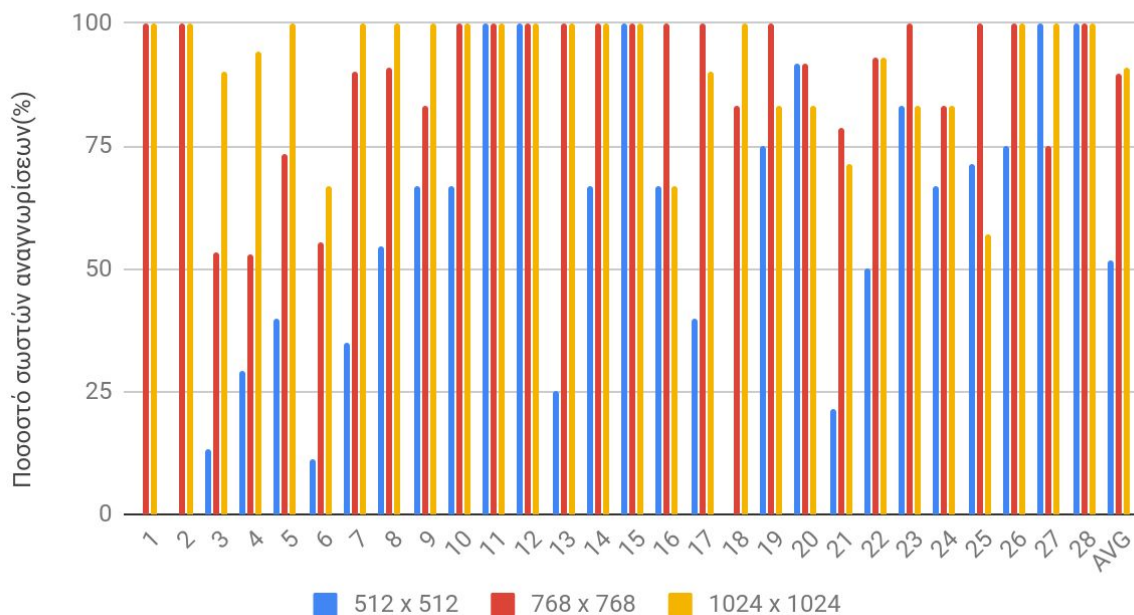
Τέλος για 1024×1024 η επιτυχία κυμαίνεται στο 78.43 %. Λαμβάνοντας υπόψη το διάγραμμα οι 2 εκτελέσεις είναι πολύ κοντά, με την τελευταία να υπερτερεί όταν οι οντότητες είναι είτε πολύ μακριά, είτε βρίσκονται κοντά τους εμπόδια. Τελικά προκύπτει το συμπέρασμα ότι για μεγαλύτερη λεπτομέρεια και ακρίβεια είναι προτιμότερο να χρησιμοποιείται το μέγεθος 1024×1024 , έχοντας όμως ως μέτρο σύγκρισης μόνο τα ποσοστά ακρίβειας. Όταν ο χρόνος εκτέλεσης αποτελεί σημαντικό κριτήριο τότε ίσως η τελευταία θα είναι καλό να αποφεύγεται.

6.1.2 Εντοπισμός οχημάτων

ID Εικόνας	512 x 512	768 x 768	1024 x 1024
1	0	100	100
2	0	100	100
3	13,33333333	53,33333333	90
4	29,41176471	52,94117647	94,11764706
5	40	73,33333333	100
6	11,11111111	55,55555556	66,66666667
7	35	90	100
8	54,54545455	90,90909091	100
9	66,66666667	83,33333333	100
10	66,66666667	100	100
11	100	100	100
12	100	100	100
13	25	100	100
14	66,66666667	100	100
15	100	100	100
16	66,66666667	100	66,66666667
17	40	100	90
18	0	83,33333333	100
19	75	100	83,33333333
20	91,66666667	91,66666667	83,33333333
21	21,42857143	78,57142857	71,42857143
22	50	92,85714286	92,85714286
23	83,33333333	100	83,33333333
24	66,66666667	83,33333333	83,33333333
25	71,42857143	100	57,14285714
26	75	100	100
27	100	75	100
28	100	100	100
AVG	51,90739	89,58337414	90,85434174

Σχήμα 6.1.2.1: Πίνακας μετρήσεων για εντοπισμό οχημάτων

Εντοπισμός Οχημάτων



Σχήμα 6.1.2.2: Διάγραμμα επιτυχούς εντοπισμού οχημάτων

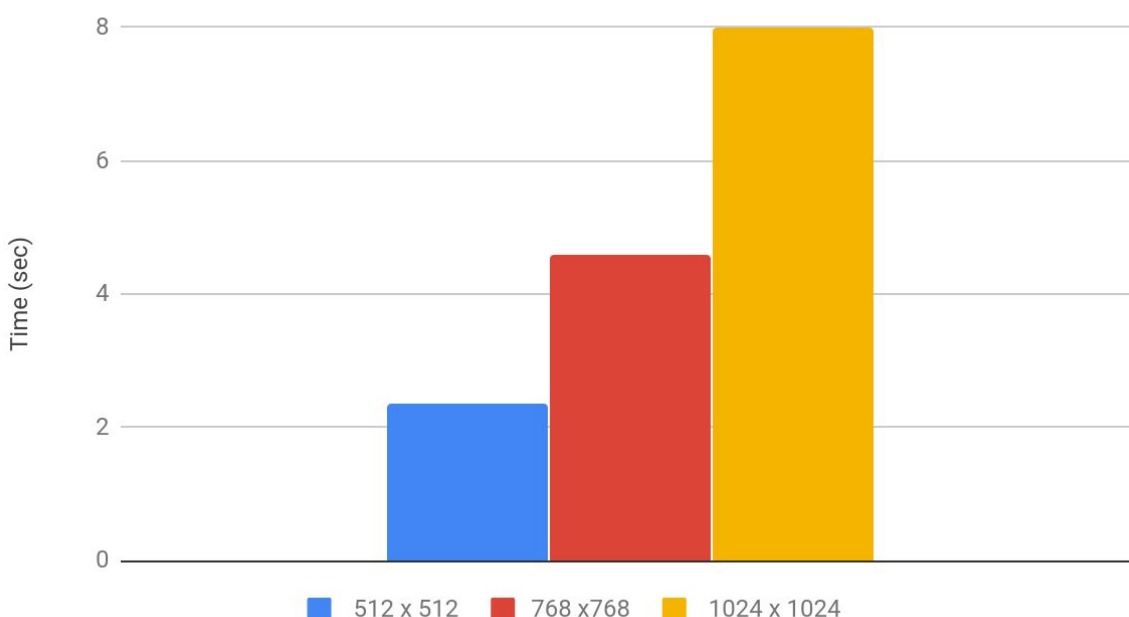
Στην περίπτωση των οχημάτων τα ποσοστά είναι αυξημένα. Παρόλα αυτά στην μικρότερη ανάλυση συνεχίζει να παρατηρείται πολύ μικρή επιτυχία που αγγίζει μόλις το 51.9 %. Παρόμοια με την περίπτωση του εντοπισμού ατόμων, ανάλογα με τη χρήση που προβλέπεται αυτό το ποσοστό μπορεί να είναι αποδεκτό ή όχι. Πιο συγκεκριμένα όταν χρειάζεται εντοπισμός οχημάτων χωρίς να παρευρίσκονται πολλά εμπόδια αλλά και να μην είναι τοποθετημένα πολύ κοντά το ένα στο άλλο, ακόμα και η πιο χαμηλή ανάλυση ανταποκρίνεται επαρκώς.

Στις 2 μεγαλύτερες αναλύσεις ο σωστός εντοπισμός είναι σχεδόν ίδιος, με διαφορά μόνο 1%. Για μέγεθος 768 * 768 έχει επιτευχθεί 89,58 % ενώ για 1024 * 1024 90.85 %. Συνεπώς το προηγούμενο δίλημμα μεταξύ επιλογής μεγέθους δεν υπάρχει εδώ. Ο επιπλέον χρόνος που θα απαιτήσει η 3η εκτέλεση δεν προσφέρει κάποια αισθητή διαφορά. Όπως αναφέρθηκε παραπάνω, η βελτίωση που παρατηρείται οφείλεται στη διαφορά μεγέθους μεταξύ των 2 αντικειμένων, με τα οχήματα να είναι λιγότερο απαιτητικά στον εντοπισμό. Στον εντοπισμό οχημάτων, όταν ο χρόνος εκτέλεσης θέλουμε να διατηρηθεί μικρός μπορεί να χρησιμοποιηθεί εικόνα 512 * 512, ιδιαίτερα σε ανοιχτές περιοχές όπως αυτοκινητόδρομους που δεν παρεμβάλλονται εμπόδια.

6.1.3 Χρόνος εκτέλεσης αλγορίθμου

Στις προηγούμενες υποενότητες παρουσιάστηκε η μεταβολή της απόδοσης συγκριτικά με το μέγεθος της εικόνας εισόδου. Ένα σημαντικό ερώτημα είναι αν αξίζει να θυσιαστεί χρόνος εκτέλεσης για την επίτευξη καλύτερων αποτελεσμάτων. Όσο αυξάνεται το μέγεθος, συνεπάγεται και η αύξηση του χρόνου που απαιτείται για να εκτελεστεί το μοντέλο. Η διαφορά αυτή φαίνεται στο διάγραμμα που ακολουθεί.

Εκτέλεση Αλγορίθμου



Σχήμα 6.1.3.1: Διάγραμμα χρόνου εκτέλεσης

Όπως φαίνεται στο σχήμα η αύξηση του χρόνου είναι γραμμική. Για αύξηση της εικόνας κατά 0.5 φορές ο χρόνος εκτέλεσης σχεδόν διπλασιάζεται, ενώ για διπλασιασμό, ο χρόνος τετραπλασιάζεται. Για ακόμα καλύτερη κατανόηση των αποτελεσμάτων, ακολουθεί σύγκριση μεταξύ της μεταβολής του ποσοστού επιτυχίας και του χρόνου.

	Μεταβολή ποσοστού επιτυχίας(άτομα)	Μεταβολή ποσοστού επιτυχίας(οχήματα)	Μεταβολή χρόνου εκτέλεσης
768 x 768	55,31914894 %	72,58308335 %	95,74468085 %
1024 x 1024	9,589041096 %	1,418753878 %	73,91304348 %

Σχήμα 6.1.3.2: Πίνακας σύγκρισης αποτελεσμάτων

Τα αποτελέσματα είναι ξεκάθαρα. Η αλλαγή από ανάλυση 512 x 512 σε 768 x 768, οδηγεί σε μια ποσοστιαία μεταβολή περίπου 55 % και 72.58 %, με τον χρόνο σχεδόν

να διπλασιάζεται. Συνεπώς ο συμβιβασμός μεταξύ χρόνου και απόδοσης μπορεί να χαρακτηριστεί αναγκαίος σε αυτήν την περίπτωση, αφού αλλιώς τα ποσοστά επιτυχούς ανίχνευσης διατηρούνται πολύ μικρά. Αντίθετα, περαιτέρω αύξηση σε 1024×1024 θεωρείται περιττή. Σε ανθρώπους και οχήματα το ποσοστό δεν αλλάζει αισθητά, ενώ το τίμημα του χρόνου είναι μεγάλο. Άρα αθροίζοντας όλα τα παραπάνω, προκύπτει ότι η περίπτωση μεγέθους που συνδυάζει ακριβή αποτελέσματα χωρίς να εξαντλούνται τα χρονικά περιθώρια είναι το 768×768 .

6.1.4 Σχολιασμός αποτελεσμάτων

Συγκρίνοντας όλα τα παραπάνω διαγράμματα, τις μετρήσεις και κοιτώντας προσεκτικά όλες τις φωτογραφίες προκύπτουν ορισμένα σημαντικά συμπεράσματα. Στον εντοπισμό ατόμων για να χρησιμοποιηθεί χαμηλή ανάλυση οι λήψεις θα πρέπει να έχουν πραγματοποιηθεί από σχετικά χαμηλή απόσταση και γενικά η περιοχή ενδιαφέροντος να μην έχει πάρα πολλά εμπόδια. Σε αντίθετη περίπτωση, κατά την εκτέλεση του αλγορίθμου, όταν η εικόνα χωριστεί σε περιοχές τα άτομα ίσως να μην περιλαμβάνονται εξ ολοκλήρου σε μια, με αποτέλεσμα την αποτυχία εντοπισμού. Γενικά η δυσκολία στον εντοπισμό προκύπτει λόγω του μικρού μεγέθους που έχουν τα άτομα όταν η λήψη γίνεται απο ύψος, ιδιαίτερα συγκριτικά με αντικείμενα όπως τα αυτοκίνητα. Για το λόγο αυτό προτιμάται η ανάλυση $768 * 768$, που συνήθως ανταποκρίνεται αποτελεσματικά σε όλες τις περιπτώσεις. Αν όμως απαιτείται μεγάλη λεπτομέρεια, ακρίβεια και η εικόνα περιέχει πάρα πολλά αντικείμενα μπορεί το μέγεθος να αυξηθεί σε 1024, 1280, 1536 με μεγάλο τίμημα πάντα στον χρόνο. Ένα τέτοιο παράδειγμα είναι ο εντοπισμός 2 ατόμων σε οροφή που βρισκόταν σε μεγάλη απόσταση απο το οπτικό πεδίο της κάμερας. Μικρές αναλύσεις δεν κατάφεραν να εντοπίσουν κανένα απο τα 2 άτομα, ενώ λίγο μεγαλύτερες έβρισκαν μόνο το ένα. Αφού ξεπεράστηκε το 1024 όλες οι εκτελέσεις εντόπισαν και τους 2 όπως φαίνεται στην εικόνα που ακολουθεί.



Σχήμα 6.1.4.1: Άτομα σε μεγάλη απόσταση

Αντίθετα με τις ανθρώπινες οντότητες τα οχήματα δεν έχουν τόσο μεγάλες απαιτήσεις σε ανάλυση. Συνήθως μέγεθος 768 * 768 είναι σε θέση να εντοπίσει τα περισσότερα, αν όχι και όλα τα οχήματα, ειδικά σε ευνοϊκές συνθήκες. Όπως και προηγουμένως για μικρό αριθμό αντικειμένων, επιθυμητά αποτελέσματα μπορούν να υπάρξουν και σε χαμηλότερες αναλύσεις. Όσο όμως οι απαιτήσεις και το εύρος περιοχών αυξάνεται, καλό είναι να γίνεται και μια μικρή αύξηση και στο μέγεθος της εικόνας. Πιο συγκεκριμένα ένα τέτοιο παράδειγμα είναι η εικόνα που ακολουθεί. Περιέχει πάρα πολλά οχήματα και κάνει σαφές ότι χαμηλές αναλύσεις δεν μπορούν να διαχειριστούν στο 100% τέτοιες εικόνες. Σε ανάλυση 512 -> 4 οχήματα, 768 -> 16, 1024 -> 27, 1280 -> 33 και 1536 -> 35. Παρατίθεται η τελική εικόνα στη μεγαλύτερη ανάλυση.



Σχήμα 6.1.4.2: Μεγάλος αριθμός οχημάτων σε εικόνα

Κεφάλαιο 7

Συμπεράσματα και μελλοντικές επεκτάσεις

Η παρούσα Διπλωματική Εργασία περιγράφει μια εφαρμογή για τον εντοπισμό διαφόρων αντικειμένων χρησιμοποιώντας την κάμερα μη επανδρωμένων αεροσκαφών. Εστιάζει περισσότερο σε ανθρώπινες οντότητες και οχήματα επειδή συνήθως αυτό εξυπηρετεί τις περισσότερες ανάγκες εντοπισμού. Αυτό όμως δεν επιδρά περιοριστικά αφού είναι σε θέση εντοπίζει επίσης πληθώρα άλλων αντικειμένων. Με αυτόν τον τρόπο μπορεί να καλύψει μεγάλες περιοχές προβλημάτων και να προσφέρει άμεσες και εύκολες λύσεις.

Υπάρχουν αρκετές μελλοντικές κατευθύνσεις για βελτίωση του συστήματος που προτείνεται. Ως προς το χρόνο εκτέλεσης, θα μπορούσε να μειωθεί εστιάζοντας

σε μικρότερο αριθμό κατηγοριών, εξειδικεύοντας έτσι το σκοπό της εφαρμογής σε συγκεκριμένα προβλήματα προς επίλυση. Σε περιπτώσεις που η χρήση προβλέπεται να εξυπηρετεί επαγγελματικούς σκοπούς το σύστημα θα μπορούσε να εξάγει μεγαλύτερο εύρος πληροφοριών. Η διαδικασία θα μπορούσε να γίνει πιο αποτελεσματική χρησιμοποιώντας ταυτόχρονα εκτός την κλασική κάμερα, μια ακόμα θερμική. Η παραπάνω προσέγγιση έχει εφαρμοστεί σε αρκετές αντίστοιχες περιπτώσεις.

Άλλη κατεύθυνση για περαιτέρω αύξηση της ταχύτητας είναι η εκτέλεση του αλγορίθμου εντοπισμού θα μπορούσε να πραγματοποιηθεί onboard χωρίς έτσι να απαιτείται η αποστολή των φωτογραφιών σε κάποιο υπολογιστή. Με αυτό τον τρόπο επιλύονται και προβλήματα απώλειας δεδομένων που μπορούν να προκύψουν κατά την αποστολή. Για να επιτευχθεί αυτό όμως χρειάζεται να τοποθετηθεί πάνω στο αεροσκάφος είτε κάποιος μικροϋπολογιστής όπως ένα raspberry ή για ακόμα ταχύτερα αποτελέσματα κάποια κάρτα γραφικών. Τα παραπάνω όμως αντιμετωπίζουν προβλήματα αυτονομίας αφού για να λειτουργήσουν απαιτούν τροφοδοσία. Επιπλέον θα μπορούσε αντί να χρησιμοποιηθεί κάποιο προεκπεδευμένο μοντέλο, να γινόταν χειροκίνητη εκπαίδευση κάποιου νέου, ώστε ο εντοπισμός να είναι ακόμη πιο στοχευμένος.

Τέλος μια εναλλακτική προσέγγιση είναι η επεξεργασία εικόνων διαδικτυακά μέσω κάποιου server, και χρήση δικτύου για την αποστολή τους. Το τελευταίο αποσκοπεί σε ταχεία ενημέρωση ακόμα και από άτομα που δεν θα έχουν πρόσβαση στον υπολογιστή που πραγματοποιείται η εκτέλεση του αλγορίθμου.

Βιβλιογραφία

- [1] Clarke, Roger. "Understanding the drone epidemic." *Computer Law & Security Review* 30.3 (2014): 230-246.
- [2] Callam, Andrew. "Drone wars: Armed unmanned aerial vehicles." *International Affairs Review* 18.3 (2010): 1.
- [3] Chao, HaiYang, YongCan Cao, and YangQuan Chen. "Autopilots for small unmanned aerial vehicles: a survey." *International Journal of Control, Automation and Systems* 8.1 (2010): 36-44.
- [4] Anderson, Karen, and Kevin J. Gaston. "Lightweight unmanned aerial vehicles will revolutionize spatial ecology." *Frontiers in Ecology and the Environment* 11.3 (2013): 138-146.
- [5] Culjak, Ivan, et al. "A brief introduction to OpenCV." 2012 proceedings of the 35th international convention MIPRO. IEEE, 2012.
- [6] Bradski, Gary, and Adrian Kaehler. "Learning OpenCV: Computer vision with the OpenCV library." (2008).
- [7] Pulli, Kari, et al. "Real-time computer vision with OpenCV." *Communications of the ACM* 55.6 (2012): 61-69.
- [8] Gaszczak, Anna, Toby P. Breckon, and Jiwan Han. "Real-time people and vehicle detection from UAV imagery." *Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques* 7878 (2011): 78780B.
- [9] Tijtgat, Nils, et al. "Embedded real-time object detection for a UAV warning system." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.
- [10] Giitsidis, Themistoklis, et al. "Human and fire detection from high altitude UAV images." 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. IEEE, 2015.
- [11] "DroneSAR | Emergency Response specific software for DJI drones." <https://www.dronesarpilot.com>.

- [12] "THE SMARTEST EYES IN THE SKY - Nightingale Security."
https://www.nightingalesecurity.com/wp-content/uploads/2019/03/Nightingale_Security_Digital_Brochure_031119.pdf
- [13] Shalev-Shwartz, Shai, and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.
- [14] "Object detection - Wikipedia." https://en.wikipedia.org/wiki/Object_detection
- [15] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- [16] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
- [17] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems (2015): 91-99.
- [18] Liu, Wei, et al. "Ssd: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.
- [19] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [20] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
- [21] "DJI Mobile SDK Documentation - DJI Developer."