



SCHOOL OF MECHANICAL ENGINEERING - NTUA  
CONTROLSYSTEMSLAB

## ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΣΥΣΤΗΜΑΤΑ ΑΥΤΟΜΑΤΙΣΜΟΥ»

Κατεύθυνση Β: «Συστήματα Αυτομάτου Ελέγχου και Ρομποτικής»

# Μεταπτυχιακή Εργασία

## ΤΙΤΛΟΣ

Παρακολούθηση πεζοδρομίου από ιπτάμενο  
όχημα μέσω οπτικής ανατροφοδότησης

(Pavement tracking by an aerial vehicle via visual  
servoing)

Χρήστος Αντωνίου

Επιβλέπων Καθηγητής: Κωνσταντίνος Κυριακόπουλος

Σχολή Μηχανολόγων Μηχανικών, Control Systems Lab

Αθήνα, Οκτώβριος 2020

## Περίληψη

Η συγκεκριμένη εργασία, αφορά την παρακολούθηση πεζοδρομίου από ένα ιπτάμενο μη-επανδρωμένο όχημα, μέσω οπτικής ανατροφοδότησης. Αρχικά, συγκεντώθηκαν τα απαραίτητα ηλεκτρονικά στοιχεία για την κατασκευή του ρομπότ, συνδέθηκαν σε ένα πλαίσιο έξι κινητήρων και πραγματοποιήθηκαν δοκιμές πτήσης τόσο με χειροκίνητο έλεγχο, όσο και σε αυτόματες αποστολές. Στη συνέχεια, εκπαιδεύτηκε ένα νευρωνικό δίκτυο, το οποίο αναγνωρίζει τη διαχωριστική γραμμή δρόμου-πεζοδρομίου μέσω οπτικού ερεθίσματος, δηλαδή μέσω μιας κάμερας που έχει εφαρμοστεί στο ρομπότ και κοιτά κάθετα προς τα κάτω, παράγοντας μια μάσκα/εικόνα που περιλαμβάνει μόνο τη γραμμή σε ουδέτερο φόντο. Αφού το ρομπότ αναγνωρίζει την διαχωριστική αυτή γραμμή σε πραγματικό χρόνο, εξάγεται μετά από επεξεργασία της μάσκας μια μαθηματικοποιημένη μορφή της γραμμής και σχεδιάστηκε κατάλληλο σύστημα ελέγχου της κλίσης και της απόστασής της από το κέντρο της εικόνας ώστε να μπορεί το ρομπότ να την παρακολουθεί . Δοκιμές και προσομοιώσεις πραγματοποιήθηκαν με τη βοήθεια του ROS.

## Abstract

This work concerns the tracking of a pavement by an unmanned air vehicle (UAV), via visual servoing. Initially, all the necessary electronic components for the construction of the robot were collected and connected to a frame of six motors (Hexacopter). Flight tests were performed both manually and in autonomous missions. Next, a neural network was trained to solve an image segmentation problem and recognize the road-pavement dividing line through a visual stimulus, i.e. through a downwards camera applied to the robot, producing a mask / image that shows the line on neutral background. Since the robot recognizes this dividing line in real time, a mathematical form of the line is extracted after processing this mask and an appropriate control system to regulate its slope and distance from the center of the image is designed so that the robot is able to follow it. Tests and simulations were performed using ROS.

## Ευχαριστίες

Με τη παρούσα διπλωματική εργασία ολοκληρώνονται έπειτα από δύο έτη οι μεταπτυχιακές σπουδές μου στο διατμηματικό πρόγραμμα Συστημάτων Αυτοματισμού του Εθνικού Μετσόβιου Πολυτεχνείου.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Κωνσταντίνο Κυριακόπουλο, για την ευκαιρία που μου έδωσε να αποτελέσω μέλος της ερευνητικής του ομάδας και για την εμπιστοσύνη που μου έδειξε αναθέτωντάς μου την υλοποίηση ενός καινοτόμου έργου, που συνάδει πλήρως με τα ενδιαφέροντά μου.

Επιπλέον, θα ήθελα να εκφράσω θερμές ευχαριστίες σε όλα τα μέλη της εργαστηριακής του ομάδας και ιδιαίτερα στον διδάκτορα, πλέον, κύριο Γιώργο Ζωγόπουλο, για τη καθοριστική βοήθεια και καθοδήγηση που μου παρείχε κατά τα πρώτα στάδια της εργασίας στην ανακατασκευή και ρύθμιση του ιπτάμενου ρομπότ που χρησιμοποίησα.

Τέλος, ευχαριστώ πάρα πολύ τους γονείς μου, που με στήριξαν ψυχολογικά και οικονομικά καθόλη τη διάρκεια των σπουδών μου, καθώς και όλους τους φίλους που ήταν δίπλα μου τα δύο αυτά χρόνια.

# Περιεχόμενα

Περίληψη .....	2
Abstract .....	3
Περιεχόμενα .....	5
Κατάλογος Εικόνων .....	6
Κατάλογος Συντημήσεων .....	7
1 Εισαγωγή .....	8
1.1 Σκοπός Εργασίας .....	9
1.2 Βιβλιογραφική Ανασκόπηση .....	10
1.3 Δομή Εργασίας .....	11
2 Στοιχεία Θεωρίας Σχεδιασμού Συστήματος .....	12
2.1 Αναγνώριση Διαχωριστικής Γραμμής .....	12
2.2 Παρακολούθηση Γραμμής .....	15
3 Σύστημα Παρακολούθησης με Οπτική Ανατροφοδότηση .....	18
3.1 Κατασκευή και Ρύθμιση UAV .....	18
3.2 Εκπαίδευση Νευρωνικού Δικτύου .....	20
3.3 Παρακολούθηση Γραμμής .....	24
3.4 Robot Operating System (ROS) .....	25
4 Πειράματα και Αποτελέσματα .....	27
4.1 Προσομοιώσεις .....	27
4.2 Πειράματα στον Πραγματικό Κόσμο .....	31
5 Συμπεράσματα και Μελλοντική Έρευνα .....	37
Βιβλιογραφία .....	38
Παράρτημα Α .....	40

## Κατάλογος Εικόνων

Εικόνα 1.1 Το UAV παρακολουθεί το πεζοδρόμιο .....	9
Εικόνα 2.1 Τυπική δομή CNN .....	12
Εικόνα 2.2 Τυπική δομή NN vs CNN.....	13
Εικόνα 2.3 Τυπική δομή CNN για τμηματοποίηση εικόνας .....	14
Εικόνα 2.4 Δομή και διασύνδεση νευρώνων σε ένα NN .....	14
Εικόνα 2.5 Διαδικασία εκπαίδευσης NN .....	15
Εικόνα 2.6 UAV local coordinate frame & roll, pitch, yaw angles .....	15
Εικόνα 2.7 Angle & Distance error of detected line.....	16
Εικόνα 2.8 Αναγνώριση θέσης της γραμμής μέσα στην εικόνα & ιδανική θέση που προσπαθούμε να πετύχουμε.....	17
Εικόνα 2.9 Κλασική δομή PID ελέγχου .....	17
Εικόνα 2.10 Διάγραμμα συνολικού συστήματος .....	17
Εικόνα 3.1 AscTec Hexacopter .....	18
Εικόνα 3.2 Pixhawk2.1 Cube Black.....	18
Εικόνα 3.3 Nvidia Jetson Nano .....	18
Εικόνα 3.4 Εικόνες του UAV.....	19
Εικόνα 3.5 Διαδικασία labeling .....	20
Εικόνα 3.6 Μάσκα που αντιστοιχεί στην labeled εικόνα.....	21
Εικόνα 3.7 Αρχική εικόνα & augmentations .....	21
Εικόνα 3.8 Προβλέψεις νευρωνικού δικτύου/ Αναγνώριση γραμμής.....	22
Εικόνα 3.9 Αναγνώριση θέσης & προσανατολισμού γραμμής μέσα στη μάσκα του NN .....	24
Εικόνα 3.10 Διάγραμμα συνολικού συστήματος στο ROS .....	26
Εικόνα 4.1 Gazebo world .....	27
Εικόνα 4.2 iris model.....	27
Εικόνα 4.3 ROS topics ως προς το χρόνο.....	27
Εικόνα 4.4 ROS topics (2) .....	28
Εικόνα 4.5 Frame-Prediction-Error&Angle .....	28
Εικόνα 4.6 Διάγραμμα angle & yaw rate στη διάρκεια μιας προσομοίωσης.....	29
Εικόνα 4.7 Διάγραμμα error & velocity.y στη διάρκεια μιας προσομοίωσης.....	29
Εικόνα 4.8 Διαγράμματα angle&error σε προσομοίωση με P ελέγχους .....	30
Εικόνα 4.9 Στιγμιότυπα κατά τη διάρκεια προσομοίωσης .....	31
Εικόνα 4.10 Frames & Predictions κατά τη διάρκεια αυτόνομης αποστολής.....	32
Εικόνα 4.11 ROS topics ως προς το χρόνο σε αυτόνομη πτήση .....	33
Εικόνα 4.12 ROS topics ως προς το χρόνο σε αυτόνομη πτήση (2).....	33
Εικόνα 4.13 Διάγραμμα angle & yaw rate στη διάρκεια μιας αυτόνομης πτήσης .....	34
Εικόνα 4.14 Διάγραμμα error & velocity.y στη διάρκεια μιας αυτόνομης πτήσης .....	34
Εικόνα 4.15 Στιγμιότυπα κατά τη διάρκεια αυτόνομης πτήσης .....	36

## Κατάλογος Συντμήσεων

UAV	Unmanned Air Vehicle
GPS	Global Positioning System
ROS	Robot Operating System
MAVs	Micro Air Vehicles
SITL	Software In The Loop
NN	Neural Network
CNN	Convolutional Neural Network
DNN	Deep Neural Network
2D	2-Dimensions

# 1 Εισαγωγή

Τα drones είναι μη επανδρωμένα αεροσκάφη, ενώ είναι πιο επίσημα γνωστά ως μη επανδρωμένα εναέρια οχήματα (Unmanned Air Vehicles/UAVs). Ουσιαστικά, ένα drone είναι ένα ρομπότ που μπορεί να ελεγχθεί από απόσταση ή να πετάξει αυτόνομα, μέσω προγραμμάτων πτήσης, ελεγχόμενο από λογισμικό στα ενσωματωμένα συστήματά του, σε συνδυασμό με διάφορους αισθητήρες και GPS.

Στο πρόσφατο παρελθόν, τα UAVs συνδέονταν συχνότερα με τον στρατό, όπου χρησιμοποιήθηκαν αρχικά σαν στόχοι σε αντιαεροπορικές ασκήσεις, καθώς και για συγκέντρωση πληροφοριών. Πλέον, χρησιμοποιούνται σε ένα ευρύ φάσμα πολιτικών εφαρμογών όπως αναζήτηση και διάσωση, παρακολούθηση της κυκλοφορίας, παρακολούθηση καιρού και πυρόσβεση, προσωπικά drones με κάμερα, λήψη βίντεο, σκηνοθεσία, γεωργία και υπηρεσίες παράδοσης. Επιπλέον, χρησιμοποιούνται όλο και περισσότερο σε εφαρμογές οι οποίες απαιτούν αυτόνομη παρακολούθηση ενός αντικειμένου ή διαδικασίας, εναέρια χαρτογράφηση ή επιτήρηση μιας περιοχής, χαρτογράφηση δύσβατων ή απομακρυσμένων περιοχών (μονοπάτια, σπηλιές, βραχώδεις περιοχές) κ.α. [12,13,14]

Η διεκπεραίωση τέτοιων αυτόνομων αποστολών, επιτυγχάνεται επί τω πλείστον με τη βοήθεια GPS, το οποίο είναι ενσωματωμένο στο drone και δίνει πληροφορία για τη θέση του. Οι συντεταγμένες του GPS χρησιμοποιούνται σε μια αυτόνομη αποστολή σαν WayPoints τα οποία το drone πρέπει να ακολουθήσει. Συχνά, όμως, είναι δύσκολο να γνωρίζουμε εκ των προτέρων ακριβώς τις συντεταγμένες που θέλουμε να ακολουθήσει το ρομπότ, ενώ σε εφαρμογές παρακολούθησης, ειδικά κινούμενου στόχου, δεν μπορούμε να τις γνωρίζουμε. Ακόμα, η ακρίβεια του GPS εξαρτάται από πολλές συνθήκες, όπως οι ατμοσφαιρικές και μπορεί να δώσει λανθασμένες ή και καθόλου μετρήσεις. Τέλος, είναι σημαντικό να βεβαιωθούμε πως η διαδρομή μέσω WayPoints δεν περιλαμβάνει εμπόδια στα οποία το ρομπότ υπάρχει περίπτωση να προσκρούσει, αφού το GPS δεν μας δίνει σχετική πληροφορία.

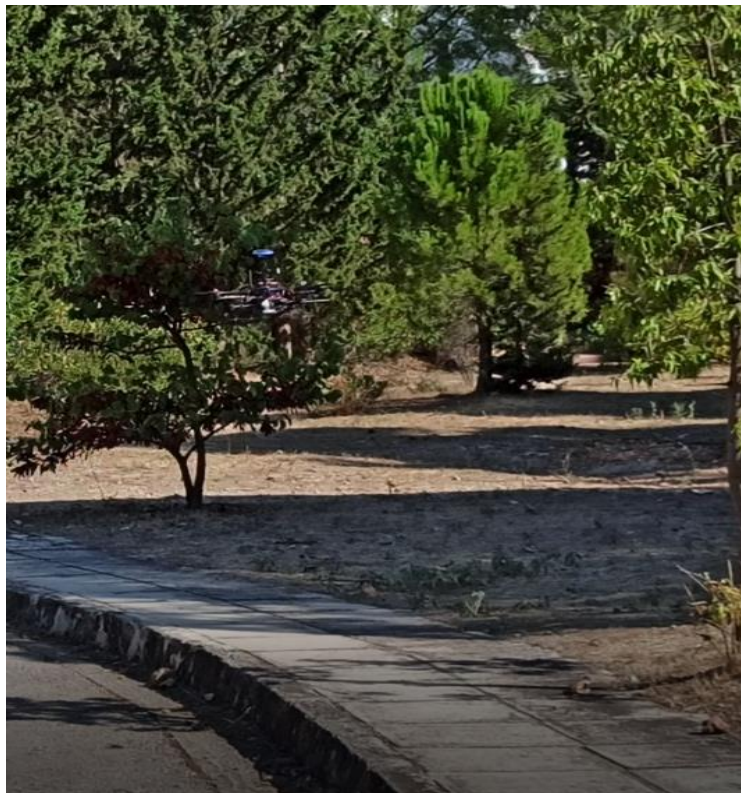
Τα παραπάνω προβλήματα οδήγησαν στο σχεδιασμό συστημάτων που βασίζονται στην οπτική ανατροφοδότηση για την επίτευξη αυτόνομων αποστολών, δηλαδή σε συστήματα που χρησιμοποιούν μια κάμερα ως βασικό αισθητήρα και οι εντολές για κίνηση του ρομπότ (επενέργεια) εξαρτώνται αποκλειστικά από το frame που “βλέπει” η κάμερα, αφού αυτό επεξεργαστεί κατάλληλα [10]. Η επεξεργασία μιας εικόνας για την αναγνώριση και την εξαγωγή των κατάλληλων χαρακτηριστικών (features) από αυτή, είναι ένα πρόβλημα που έχει απασχολήσει τους ερευνητές ανά τον κόσμο και αντιμετωπίζεται συνήθως είτε με κλασικές τεχνικές ψηφιακής επεξεργασίας εικόνας είτε με τεχνικές τεχνητής νοημοσύνης, μηχανικής μάθησης και χρήση νευρωνικών δικτύων, τα οποία κερδίζουν διαρκώς έδαφος τα τελευταία χρόνια, χάρη στην υψηλή ακρίβεια των αποτελεσμάτων τους και ευκολία εφαρμογής σε τεράστια ποικιλία προβλημάτων σε συνδυασμό με τη διαρκώς αυξανόμενη επεξεργαστική ικανότητα των υπολογιστών, είτε με συνδυασμούς των παραπάνω τεχνικών [15].

Αφού εξαχθούν τα χαρακτηριστικά ενδιαφέροντος μιας εικόνας, με το σχεδιασμό κατάλληλου συστήματος ελέγχου, κατευθύνεται το drone ανάλογα την αποστολή που του έχει ανατεθεί. Σε εφαρμογές παρακολούθησης μέσω οπτικής ανατροφοδότησης (visual servoing) [11] ο ελεγκτής συνήθως προσπαθεί να μηδενίσει σφάλματα που έχουν να κάνουν με την απόσταση σε pixels του αντικειμένου, του οποίου η θέση στην εικόνα έχει αναγνωριστεί, με το κέντρο της εικόνας.



## 1.1 Σκοπός Εργασίας

Στην παρούσα εργασία, κατασκευάστηκε ένα drone σε πλαίσιο έξι κινητήρων (Hexarotor), με autopilot Pixhawk2.1 Cube Black και εφοδιασμένο με ένα Nvidia Jetson Nano ως companion computer το οποίο προγραμματίστηκε με στόχο να εκτελεί την αυτόνομη αποστολή παρακολούθησης της διαχωριστικής γραμμής δρόμου-πεζοδρομίου μέσω οπτικής ανάδρασης.



Εικόνα 1.1 Το UAV παρακολουθεί το πεζοδρόμιο

Πιο συγκεκριμένα, ο βασικός αισθητήρας του συστήματος είναι μια USB camera τοποθετημένη να κοιτά κάθετα προς τα κάτω. Για την επεξεργασία των frames της κάμερας και την αναγνώριση του αντικειμένου ενδιαφέροντος, δηλαδή της διαχωριστικής γραμμής, εκπαιδεύτηκε ένα Convolutional Neural Network (CNN), η έξοδος του οποίου είναι μια εικόνα (μάσκα) ίδιων διαστάσεων με την αρχική, όπου κάθε pixel χρωματίζεται ανάλογα με την πιθανότητά του, την οποία υπολόγισε το νευρωνικό δίκτυο, να ανήκει ή όχι στο αντικείμενο που θέλουμε να αναγνωρίσουμε. Η εκπαίδευση πραγματοποιήθηκε μέσω δεδομένων που συλλέχθηκαν από τον πραγματικό κόσμο και χρησιμοποιήθηκαν για μάθηση υπό επίβλεψη (supervised learning) [38].

Στη συνέχεια, η μάσκα αυτή επεξεργάζεται ώστε να υπολογιστούν η κλίση σε μοίρες από τον κάθετο άξονα και η απόσταση σε pixels, του κέντρου της γραμμής που αναγνώρισε το CNN από το κέντρο της εικόνας. Για κάθε μία από αυτές τις δύο μεταβλητές σχεδιάστηκε και ρυθμίστηκε ένας PD ελεγκτής, με σημείο αναφοράς το μηδέν, η έξοδος του οποίου καθορίζει το διάνυσμα της ταχύτητας που στέλνεται σαν εντολή στο drone.

Η επικοινωνία ανάμεσα στις επιμέρους διαδικασίες και υποσυστήματα επιτυγχάνεται με τη βοήθεια του Robot Operating System (ROS) [1]. Το ROS είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα για ρομπότ. Κάθε διαδικασία από τη λήψη του frame μέχρι την επενέργεια στους κινητήρες

εκτελείται από ένα κόμβο (node) στο ROS που είτε δημοσιεύει (publish) είτε εγγράφεται ως παραλήπτης (subscriber) σε κάποιο τίτλο (topic), ο οποίος μεταφέρει δεδομένα με μορφή μηνυμάτων (messages).

Δοκιμές πραγματοποιήθηκαν αρχικά στο περιβάλλον προσομοίωσης Gazebo και στη συνέχεια και στον πραγματικό κόσμο.

## 1.2 Βιβλιογραφική Ανασκόπηση

Οι αυτόνομες αποστολές σε UAVs μέσω οπτικής ανατροφοδότησης, έχουν απασχολήσει πολλούς ερευνητές. Αρκετά προηγούμενα έργα που ασχολήθηκαν με την αναγνώριση αντικειμένων έλυσαν ένα πρόβλημα τμηματοποίησης (segmentation), δηλαδή, με στόχο τον προσδιορισμό των περιοχών της εικόνας εισόδου που αντιστοιχούν σε διαφορετικά αντικείμενα. Για να επιλυθεί αυτό το πρόβλημα, πρέπει να οριστεί ρητά ποια οπτικά μοτίβα χαρακτηρίζουν ένα αντικείμενο.

Ο Rasmussen [2] βασίστηκε στην αντίθεση της εικόνας, ενώ ο Santana στην ευκρίνεια [3].

Ο Venugopalan [4] εξετάζεται η παρακολούθηση του ωκεανού. Οι συγγραφείς παρουσιάζουν ένα σύστημα ελέγχου για αυτόματη προσγείωση ενός drone σε ένα σκάφος χρησιμοποιώντας πληροφορίες που αποκτήθηκαν από μία προς τα κάτω τοποθετημένη κάμερα στο UAV.

Με τη σειρά τους, στο [5] οι συγγραφείς ασχολούνται με το πρόβλημα της μακροχρόνιας στερεοσκοπικής οπτικής οδομετρίας για UAV. Υποστηρίζουν ότι η πλοήγηση μέσω της όρασης μειώνει την εξάρτηση από το GPS και άλλα παγκόσμια δορυφορικά συστήματα πλοήγησης, ενισχύοντας την ανθεκτικότητα της πλοήγησης σε εφαρμογές χαμηλού.

Στο Sotomayor [6] παρουσιάστηκε ένα σύστημα ελέγχου που να οδηγεί ένα αυτόνομο εναέριο όχημα με εφαρμογή σε επιθεώρηση καλλιέργειας. Ο προσανατολισμός των γραμμών που πρέπει να ακολουθήσει το drone εξάγεται από μια μέθοδο computer vision βασισμένη στις προσανατολισμένες υφές. Η γωνία που λαμβάνεται από το σύστημα όρασης είναι η αναφορά για τους ελεγκτές PID που καθοδηγούν το UAV.

Στο [7] προτείνεται ένας έλεγχος με οπτική ανατροφοδότηση βασισμένος στη θεωρία Lyapunov για την παρακολούθηση καλλιέργειών. Τα frames της κάμερας περνούν από επεξεργασία ώστε να ανιχνευτούν ακμές, ο προσανατολισμός των οποίων καθορίζει την τροχιά που θα ακολουθήσει το drone.

Στο [8] προτείνεται μια τεχνική παρακολούθησης μονοπατιού σε δάσος η οποία διαφέρει από τις υπόλοιπες αφού χρησιμοποιεί ένα νευρωνικό δίκτυο που δεν επεξεργάζεται την εικόνα για να αναγνωρίσει αντικείμενα ενδιαφέροντος αλλά έχει εκπαιδευτεί ώστε να αποφασίζει αν πρέπει να στρίψει προς κάποια κατεύθυνση ή να πάει ευθεία, απευθείας από την εικόνα που βλέπει.

Το [9] ασχολείται με μια τεχνική εντοπισμού και παρακολούθησης κινούμενων αντικειμένων από UAV.

Όμοια με προηγούμενες εφαρμογές, στην εργασία αυτή παρουσιάζεται μια τεχνική segmentation της εικόνας που βλέπει το ρομπότ όμως η διαδικασία επιτυγχάνεται με τη βοήθεια νευρωνικού δικτύου σε αντίθεση με το συντριπτικό ποσοστό εφαρμογών τέτοιου είδους που χρησιμοποιούν τεχνικές κλασικής, ψηφιακής επεξεργασίας εικόνας. Η χρήση νευρωνικού δικτύου επιτρέπει την αναγνώριση οποιουδήποτε αντικειμένου/δομής εκπαιδευτεί να αναγνωρίζει. Οι επιφάνειες και τα χρώματα των δρόμων και πεζοδρομίων ποικίλουν ακόμα και μέσα στην ίδια

πόλη, οπότε τεχνικές που βασίζονται σε διαφορά χρώματος, στην αντίθεση ή αναγνώριση ακμών είναι πολύ πιθανό να έχουν πρόβλημα στην αναγνώριση όλων των διαχωριστικών επιφανειών δρόμου/πεζοδρομίου ενώ το ίδιο νευρωνικό δίκτυο μπορεί εύκολα να επανεκπαιδευτεί για αναγνώριση και άλλων διαχωριστικών γραμμών (πχ ακτή/ θάλασσα) ή ακόμα και διαφορετικής εντελώς φύσεως αντικειμένων (όπως δέντρα ή ανθρώπους).

Οι έλεγχοι που επιλέχθηκαν είναι PD λόγω της ευκολίας στην εφαρμογή και ρύθμισή τους, προσφέροντας αποτελέσματα το ίδιο ικανοποιητικά με άλλες τεχνικές ελέγχου. Εφαρμόζονται ώστε να μηδενίσουν το σφάλμα κλίσης και απόστασης από το κέντρο της εικόνας ενώ το UAV κινείται με σταθερή ταχύτητα ως προς την (local) κατεύθυνση που κοιτά το μπροστινό μέρος του. Επιπλέον, ο αλγόριθμος οδήγησης, είναι σχεδιασμένος ώστε το drone να μπορεί να ανακτήσει τη θέση του πάνω από τη διαχωριστική γραμμή δρόμου πεζοδρομίου σε περίπτωση που τη χάσει (πχ από διαταραχή ανέμου) με βάση την τελευταία φορά που ανίχνευσε γραμμή, σε αντίθεση με τους περισσότερους συγγενείς αλγόριθμους, οι οποίοι στέλνουν εντολές επενέργειας μόνο όταν ανιχνεύουν το αντικείμενο ενδιαφέροντος. Η τροποποίηση αυτή είναι σημαντική για χαμηλές πτήσεις όπου μικρές μεταβολές της θέσης του ρομπότ, έχουν σημαντική επίπτωση στο οπτικό πεδίο της κάμερας, χωρίς να επηρεάζεται η λειτουργία σε μεγαλύτερα ύψη πτήσης.

Τα αποτελέσματα με το πέρας της εργασίας ήταν πολύ ικανοποιητικά. Το νευρωνικό δίκτυο δεν έχει κανένα πρόβλημα στην αναγνώριση του χαρακτηριστικού μοτίβου το οποίο εκπαιδεύτηκε να αναγνωρίζει, ενώ ο αλγόριθμος υπολογισμού των προς έλεγχο μεταβλητών και στη συνέχεια των εξόδων των PD ελεγκτών, οδηγεί διαρκώς το drone σε θέση ακριβώς πάνω από την επιθυμητή γραμμή με κλίση 0° από την κατεύθυνση που “κοιτά” το drone, πάνω στην οποία κινείται διαρκώς με σταθερή ταχύτητα.

### 1.3 Δομή Εργασίας

Συνοπτικά, η δομή της εργασίας είναι η ακόλουθη:

- Στο κεφάλαιο 2, γίνεται αναφορά στα μαθηματικά και υπολογιστικά εργαλεία που θα χρησιμοποιηθούν για την αναγνώριση της γραμμής ενδιαφέροντος και τίθενται οι βάσεις της τεχνικής παρακολούθησής της και του σχεδιασμού του συστήματος ελέγχου.
- Στο κεφάλαιο 3, παρουσιάζεται η προσέγγιση με την οποία η συγκεκριμένη εργασία αντιμετωπίζει το πρόβλημα παρακολούθησης της διαχωριστικής γραμμής δρόμου/πεζοδρομίου από ένα UAV. Αναλύονται τα επιμέρους υποσυστήματα που σχεδιάστηκαν και εφαρμόστηκαν καθώς και ο ρόλος τους στο συνολικό σύστημα.
- Στο 4° κεφάλαιο, παρουσιάζονται και σχολιάζονται τα πειραματικά αποτελέσματα του συστήματος. Πειράματα πραγματοποιήθηκαν αρχικά σε περιβάλλον προσομοίωσης και στη συνέχεια και στον πραγματικό κόσμο.
- Το 5° κεφάλαιο, αποτελεί μια σύνοψη της εργασίας και περιλαμβάνει προτάσεις για βελτιώσεις και μελλοντική έρευνα σε προβλήματα παρόμοιας φύσεως.
- Τέλος, στο Παράρτημα Α, υπάρχουν πληροφορίες σχετικά με τους αλγόριθμους που υλοποιήθηκαν στα διάφορα στάδια της εργασίας και τη λειτουργία τους.

## 2 Στοιχεία Θεωρίας Σχεδιασμού Συστήματος

Για την παρακολούθηση της διαχωριστικής γραμμής δρόμου-πεζοδρομίου από ένα UAV, δύο είναι τα κύρια προβλήματα που πρέπει να αντιμετωπιστούν. Αυτά είναι:

- Η αναγνώριση της γραμμής που μας ενδιαφέρει μέσα σε μια ψηφιακή εικόνα.
- Η καθοδήγηση του drone ανάλογα την εικόνα που βλέπει, ώστε να ακολουθεί τη γραμμή.

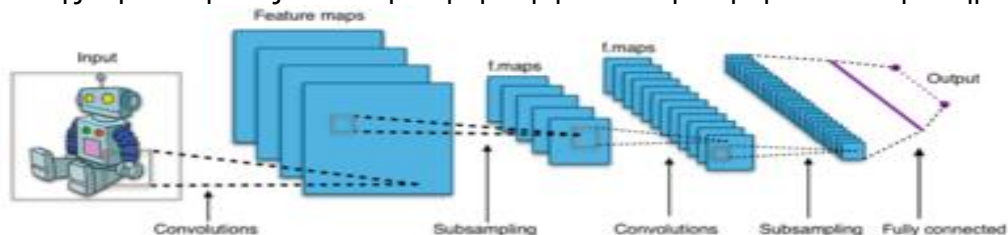
Στο κεφάλαιο αυτό, πραγματοποιείται μια θεωρητική εδραίωση των τεχνικών που υλοποιήθηκαν για να επιτευχθεί ο τελικός στόχος, που είναι η παρακολούθηση της γραμμής. Η αναγνώριση της γραμμής ανήκει στη γενικότερη κατηγορία προβλημάτων μηχανικής όρασης (computer vision) που αφορούν την εύρεση ενός αντικειμένου ή δομής και τη θέση της σε μια εικόνα, ενώ η καθοδήγηση του drone μέσω οπτικής ανάδρασης και ο σχεδιασμός του συστήματος ελέγχου που την επιτυγχάνει, ανήκουν στην κατηγορία προβλημάτων visual servoing.

### 2.1 Αναγνώριση Διαχωριστικής Γραμμής

Η σύγχρονη τεχνολογία Computer Vision, βασισμένη σε μεθόδους AI και deep learning, έχει εξελιχθεί δραματικά την τελευταία δεκαετία. Χρησιμοποιείται για εφαρμογές όπως η ταξινόμηση εικόνας (classification), η αναγνώριση αντικειμένων σε εικόνες (recognition), η ανάλυση βίντεο και η ταξινόμηση και επεξεργασία εικόνας σε ρομπότ και αυτόνομα οχήματα. Οι σημερινές τεχνικές τμηματοποίησης χρησιμοποιούν μοντέλα deep learning για να προσεγγίσουν την “όραση” σε έναν υπολογιστή δίνοντάς τους τη δυνατότητα να αναγνωρίζουν ακριβώς πιο αντικείμενο πραγματικού κόσμου αντιπροσωπεύεται από κάθε pixel μιας εικόνας [16].

Ένα Deep Neural Network μπορεί να μάθει μοτίβα σε οπτικές εισόδους προκειμένου να προβλέψει την κλάση (class) στην οποία ανήκουν τα αντικείμενα που αποτελούν μια εικόνα. Η κύρια αρχιτεκτονική βαθιάς μάθησης που χρησιμοποιείται για την επεξεργασία εικόνων είναι ένα Convolutional Neural Network (CNN). Το Convolutional Neural Network (CNN) έχει αποδειχθεί πολύ καλό στην επεξεργασία διασυνδεδεμένων δεδομένων. Χρησιμοποιεί μια τρισδιάστατη δομή, με τρία εξειδικευμένα νευρωνικά δίκτυα που αναλύουν το κόκκινο, πράσινο και μπλε στρώμα (RGB) μιας έγχρωμης εικόνας. Σαρώνει μία εικόνα κάθε φορά και αναγνωρίζει και εξάγει σημαντικά χαρακτηριστικά της [17].

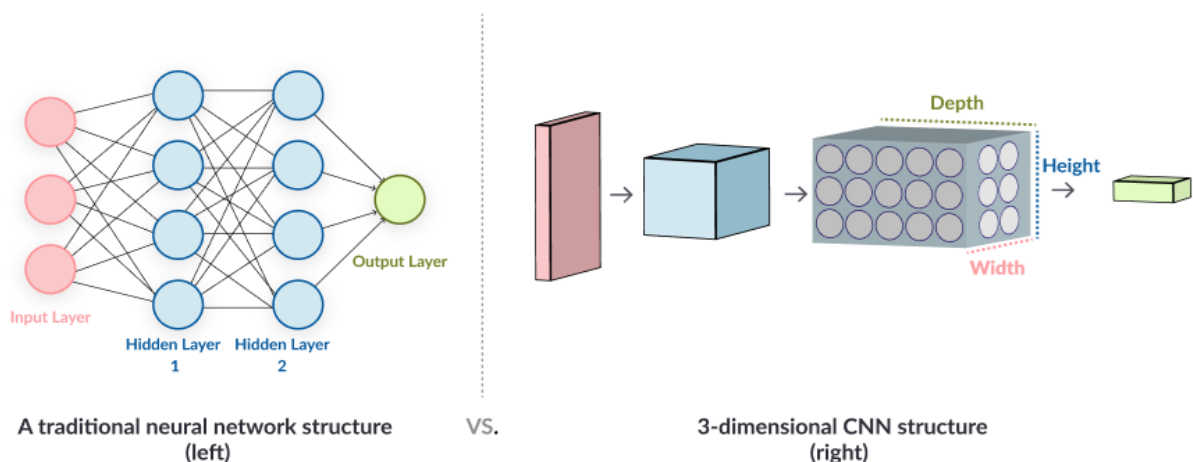
Στην εργασία αυτή, χρησιμοποιήθηκε η τεχνική της σημασιολογικής τμηματοποίησης (semantic segmentation) [18], η οποία με τη βοήθεια ενός CNN ταξινομεί όλα τα pixel μιας εικόνας σε σημαντικές κλάσεις αντικειμένων. Αυτές οι κλάσεις είναι σημασιολογικές ερμηνείες και αντιστοιχούν σε πραγματικές κατηγορίες. Μπορούμε δηλαδή, να απομονώσουμε όλα τα pixel που σχετίζονται με μια γάτα για παράδειγμα και να τα χρωματίσουμε πράσινα. Αυτή η τεχνική είναι επίσης γνωστή ως πυκνή πρόβλεψη επειδή προβλέπει τη σημασία κάθε pixel.



Εικόνα 2.1 Τυπική δομή CNN

Τα μοντέλα βαθιάς μάθησης για την όραση του υπολογιστή συνήθως εκπαιδεύονται και εκτελούνται σε εξειδικευμένες μονάδες επεξεργασίας γραφικών (GPU) για τη μείωση του χρόνου υπολογισμού.

**Αρχιτεκτονική CNN** Ένα απλό full connected νευρωνικό δίκτυο, στο οποίο όλοι οι νευρώνες σε ένα στρώμα επικοινωνούν με όλους τους νευρώνες στο επόμενο στρώμα είναι αναποτελεσματικό στην ανάλυση μεγάλων εικόνων και βίντεο. Για μια μεσαίου μεγέθους εικόνα με εκατοντάδες pixels και τρία κανάλια χρώματος (κόκκινο, πράσινο, μπλε), ο αριθμός των παραμέτρων που χρησιμοποιεί ένα παραδοσιακό νευρωνικό δίκτυο είναι στα εκατομμύρια. Για να περιορίσει τον αριθμό των παραμέτρων και να εστιάσει το νευρωνικό δίκτυο σε σημαντικά μέρη της εικόνας, ένα CNN χρησιμοποιεί μια τρισδιάστατη δομή στην οποία κάθε σύνολο νευρώνων αναλύει μια μικρή περιοχή ή "χαρακτηριστικό" της εικόνας. Αντί να έχουν όλοι οι νευρώνες να περάσουν τις αποφάσεις τους στο επόμενο νευρικό στρώμα, κάθε ομάδα νευρώνων ειδικεύεται στην αναγνώριση ενός μέρους της εικόνας. Το τελικό αποτέλεσμα είναι ένα διάνυσμα βαθμολογίας πιθανοτήτων, το οποίο αντιπροσωπεύει το πόσο πιθανό είναι κάθε χαρακτηριστικό να είναι μέρος μιας κλάσης (αντικειμένου).

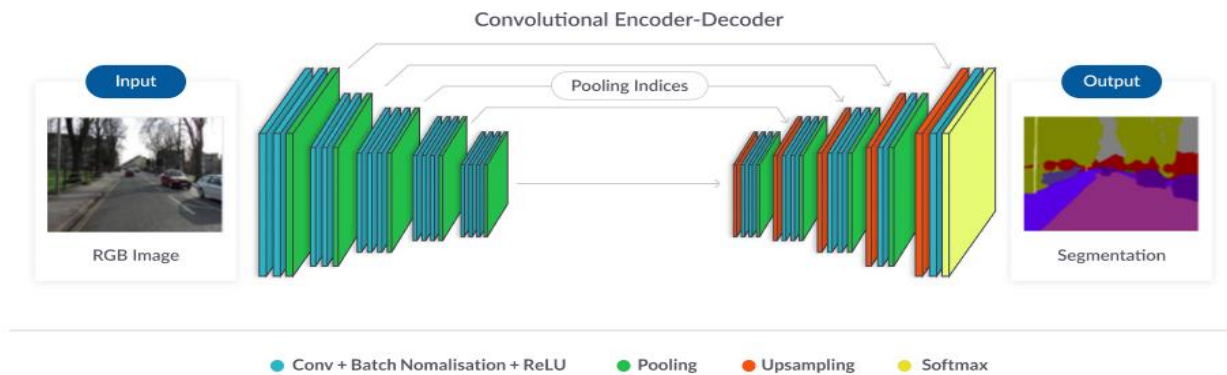


Εικόνα 2.2 Τυπική δομή NN vs CNN

**Λειτουργία CNN** Ένα CNN λειτουργεί σε τρία στάδια [19].

- Το πρώτο είναι μια συνέλιξη (convolution), στην οποία η εικόνα σαρώνεται ανά μερικά pixels τη φορά, και δημιουργείται ένας χάρτης χαρακτηριστικών με την πιθανότητα ότι κάθε χαρακτηριστικό ανήκει στην απαιτούμενη κλάση.
- Το δεύτερο στάδιο είναι η συγκέντρωση (concentration) ή δειγματοληψία, η οποία μειώνει τη διάσταση κάθε χαρακτηριστικού διατηρώντας παράλληλα τις πιο σημαντικές πληροφορίες.
- Το στάδιο συναρμολόγησης (assembly) δημιουργεί μια «περίληψη» των πιο σημαντικών χαρακτηριστικών της εικόνας.

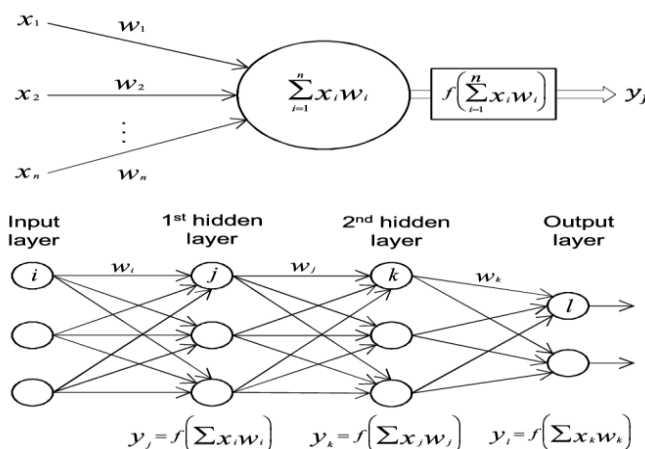




Εικόνα 2.3 Τυπική δομή CNN για τμηματοποίηση εικόνας

Τα περισσότερα CNN χρησιμοποιούν την τακτική μέγιστης συγκέντρωσης (max pooling), κατά την οποία λαμβάνεται από κάθε περιοχή pixel η υψηλότερη τιμή που σαρώνεται από το CNN. Τέλος, το CNN μπαίνει στο τρίτο στάδιο, το οποίο αποτελείται από ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο που αναλύει τις τελικές πιθανότητες και αποφασίζει σε ποια κατηγορία ανήκει κάθε χαρακτηριστικό που εντοπίστηκε στην εικόνα.

**Διαδικασία εκπαίδευσης** Για την αναγνώριση μοντέλων σε εικόνες με τεχνικές supervised learning, το αρχικό στάδιο είναι ο προσδιορισμός από το σχεδιαστή του μοτίβου που θέλει να αναγνωρίσει (labeling). Συλλέγοντας ένα ικανοποιητικά μεγάλο dataset εικόνων που περιέχουν το μοτίβο και προσδιορίζοντας χειροκίνητα την ύπαρξη και τη θέση του μοτίβου σε κάθε εικόνα, μπορούμε να εκπαιδεύσουμε ένα νευρωνικό δίκτυο, το οποίο θα επεξεργαστεί κατάλληλα τις εικόνες που του χορηγήσαμε και θα μπορεί να προβλέπει την ύπαρξη και θέση του μοτίβου σε νέες εικόνες που δεν έχει ξαναδεί. Τα νευρωνικά δίκτυα είναι μια διασυνδεδεμένη συλλογή κόμβων που ονομάζονται νευρώνες ή perceptrons. Κάθε νευρώνας παίρνει ένα κομμάτι των δεδομένων εισόδου, συνήθως ένα pixel ή ομάδες γειτονικών pixel της εικόνας και εφαρμόζει έναν

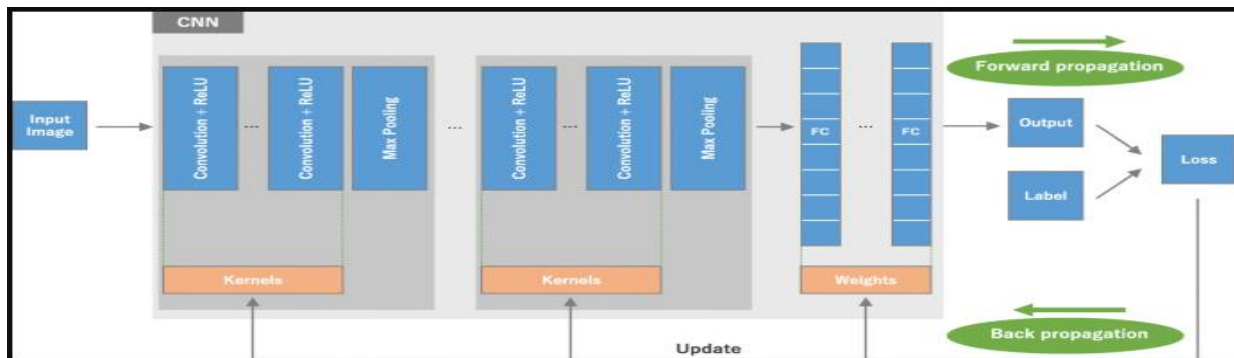


Εικόνα 2.4 Δομή και διασύνδεση νευρώνων σε ένα NN

Κατά την εκπαίδευση, τα αρχικά βάρη έχουν τυχαίες τιμές και η πρόβλεψη του νευρωνικού για κάθε εικόνα του training set, συγκρίνεται με την πραγματική τιμή που έχει δώσει ο χρήστης κατά το labeling. Το νευρωνικό δίκτυο μαθαίνει, δηλαδή ρυθμίζει τα βάρη που πρέπει να έχει κάθε νευρώνας με μια διαδικασία που ονομάζεται backpropagation, ανάλογα με το μέτρο του σφάλματος (συνάρτηση κόστους) της πραγματικής (labeled) τιμής από αυτή που προέβλεψε το νευρωνικό. Τα βάρη ανανεώνονται σύμφωνα με την

απλό υπολογισμό, που ονομάζεται συνάρτηση ενεργοποίησης για να δημιουργήσει ένα αποτέλεσμα (έξοδος νευρώνα). Κάθε νευρώνας έχει ένα αριθμητικό βάρος που επηρεάζει το αποτέλεσμα. Αυτό το αποτέλεσμα τροφοδοτείται σε επιπλέον νευρικά στρώματα έως ότου στο τέλος της διαδικασίας το νευρωνικό δίκτυο δημιουργεί μια πρόβλεψη, δηλαδή υπολογίζει την πιθανότητα για κάθε είσοδο ή pixel, να ανήκει στο αντικείμενο που ψάχνουμε.

τεχνική βελτιστοποίησης που έχει εφαρμοστεί στη συνάρτηση κόστους. Μόλις εκπαιδευτεί ένα μοντέλο, εφαρμόζεται σε ένα νέο σύνολο εικόνων που δεν συμμετείχαν στην εκπαίδευση (ένα σύνολο δοκιμών ή επικύρωσης), για να ελέγξει την ακρίβεια του. Το μοντέλο, πλέον, μπορεί να χρησιμοποιηθεί για την τμηματοποίηση των πραγματικών εικόνων, αφού τα βάρη του, που καθορίζουν άμεσα την απόκρισή του, έχουν λάβει την κατάλληλη τιμή [20].

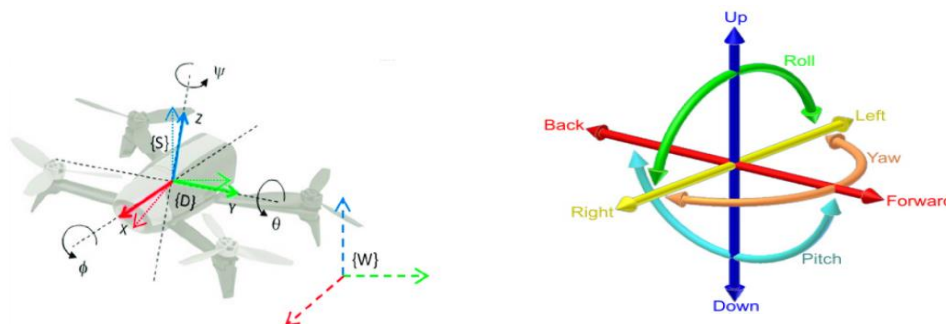


Εικόνα 2.5 Διαδικασία εκπαίδευσης NN

## 2.2 Παρακολούθηση Γραμμής

Η έξοδος του νευρωνικού δικτύου είναι μια εικόνα δύο χρωμάτων, με κάθε pixel να χρωματίζεται ανάλογα με την πρόβλεψη του νευρωνικού δικτύου αν ανήκει ή όχι στην κλάση της διαχωριστικής γραμμής. Έχοντας την πληροφορία αυτή, καλούμαστε να αντιμετωπίσουμε το πρόβλημα παρακολούθησης της γραμμής αυτής. Στην παρούσα εργασία, το προηγούμενο πρόβλημα αντιμετωπίστηκε με μια κλασική τεχνική visual servoing για UAVs. Η βασική της ιδέα είναι το drone να προσαρμόζει τη θέση του ώστε η διαχωριστική γραμμή να διέρχεται πάντα από το κέντρο της εικόνας που βλέπει και σε γωνία μηδέν μοιρών από τον κάθετο άξονά της, καθώς το drone κινείται με σταθερή ταχύτητα στον άξονα αυτό. Αφού το ύψος πτήσης είναι σταθερό, πρόκειται για ένα πρόβλημα οδήγησης σε δύο άξονες (x,y) οι οποίοι έχουν την βάση τους πάνω στο drone.

Μπορούμε να ελέγχουμε τόσο τη γραμμική, όσο και τη γωνιακή ταχύτητα του drone γύρω και από τους τρεις άξονες x,y,z, ρυθμίζοντας τις γωνίες Euler pitch,roll και yaw αντίστοιχα. Ταχύτητα στον άξονα των x, σημαίνει πως το drone κινείται μπροστά ή πίσω σχετικά πάντα με το πού κοιτά, στον άξονα y, κινείται δεξιά ή αριστερά και στον z πάνω ή κάτω. Οι ταχύτητες αυτές δεν είναι συζευγμένες μεταξύ τους και μπορεί να υλοποιηθεί έλεγχος κάθε μίας ξεχωριστά χωρίς να επηρεάζονται οι υπόλοιπες. Πληροφορίες σχετικά με τη δυναμική του UAV, βρίσκονται στο [21].

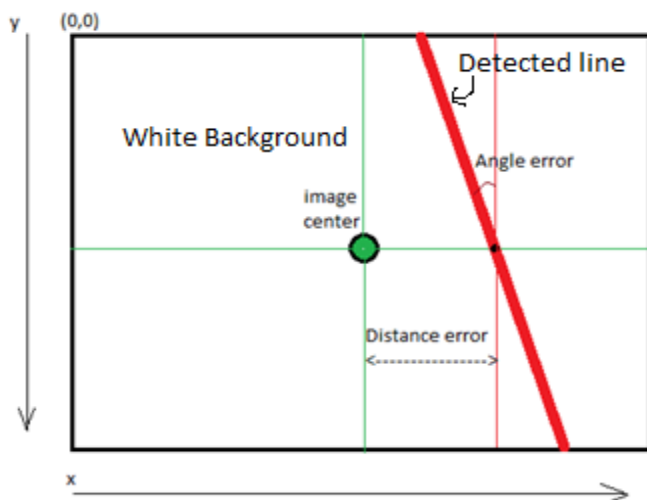


Εικόνα 2.6 UAV local coordinate frame & roll, pitch, yaw angles

Οι δύο μεταβλητές που πρέπει να υπολογιστούν και να ελεγχθούν για την κατάλληλη προσαρμογή της θέσης του drone είναι:

- Η οριζόντια απόσταση σε pixels της κάθετης στον οριζόντιο άξονα και τέμνει τη γραμμή από το κεντρικό σημείο της εικόνας.

και η γωνιακή απόκλιση σε μοίρες της γραμμής από τον κάθετο άξονα



Στόχος, λοιπόν, είναι ο σχεδιασμός συστήματος ελέγχου που να επιτυγχάνει μηδενισμό του σφάλματος απόστασης και γωνίας της γραμμής από το κέντρο της εικόνας, όπως αυτά ορίστηκαν παραπάνω. Οι δύο αυτές μεταβλητές υπολογίζονται εύκολα, μετά από επεξεργασία της μάσκας/εξόδου του νευρωνικού δικτύου. Ο αλγόριθμος που υλοποιεί τον υπολογισμό αυτό, ακολουθεί το παρακάτω βασικά βήματα:

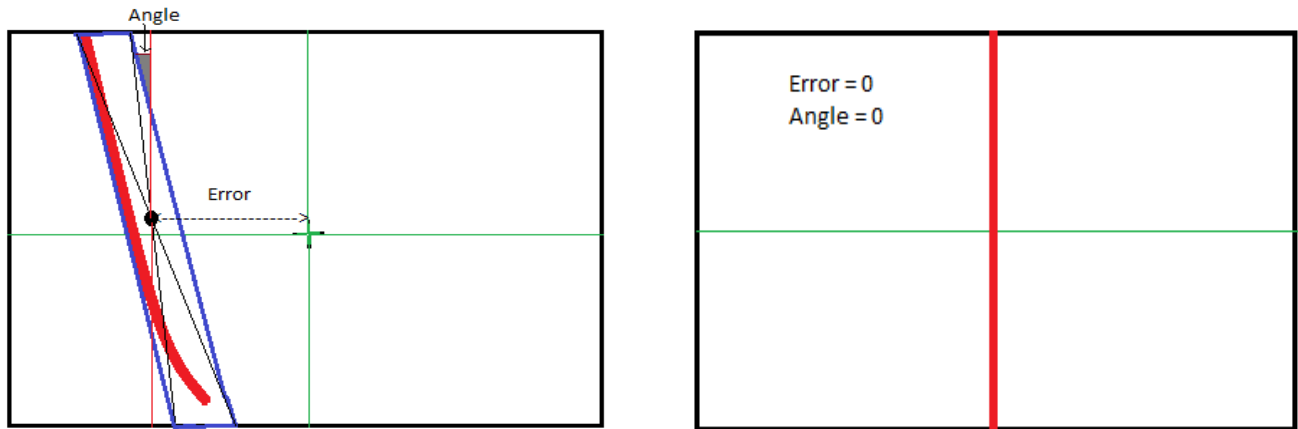
Εικόνα 2.7 Angle & Distance error of detected line

- Εύρεση των τεσσάρων ακμών του παραλληλογράμμου, ως σημεία στον διδιάστατο χώρο  $(x,y)$  της εικόνας, που εμπεριέχει όλη την καμπύλη που ανίχνευσε το νευρωνικό δίκτυο στο εμβαδόν του, το οποίο είναι το ελάχιστο δυνατό. Τα  $(x,y)$  αναφέρονται στην απόσταση σε pixels του σημείου θεωρώντας το πάνω αριστερά pixel ως το σημείο  $(0,0)$  και τον άξονα  $y$  ανεστραμμένο. Η εύρεση αυτή, πραγματοποιείται μέσω της βιβλιοθήκης OpenCV και βασίζεται στη χρωματική διαφορά της γραμμής από το background της.
- Με βάση τα τέσσερα σημεία που εντοπίστηκαν προηγουμένως, υπολογίζεται γεωμετρικά το κέντρο του παραλληλογράμμου ως σημείο τομής των διαγωνίων του και έπειτα, υπολογίζεται η απόστασή του από την ευθεία που είναι παράλληλη στον άξονα  $y$  και διέρχεται από το κέντρο της εικόνας. Η μετρική αυτή, είναι στην ουσία το σφάλμα απόκλισης της γραμμής από το κέντρο της εικόνας που βλέπει το drone και αποτελεί την πρώτη προς έλεγχο μεταβλητή (error).

$$error = x_{line} - x_{image\_center}$$

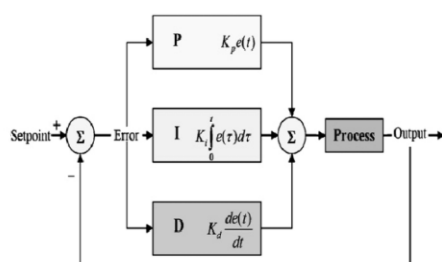
- Υπολογίζοντας την κλίση της ευθείας που διέρχεται από μια μεγάλη πλευρά του παραλληλογράμμου, μέσω της μαθηματικής της εξίσωσης που βρίσκεται εύκολα έχοντας τις συντεταγμένες των τεσσάρων ακμών του, μπορούμε να υπολογίσουμε και τη γωνία της σε σχέση με τον οριζόντιο και τον κάθετο άξονα. Η γωνία σε σχέση με τον κάθετο άξονα είναι η δεύτερη μεταβλητή που πρέπει να οδηγούμε στο μηδέν (angle).





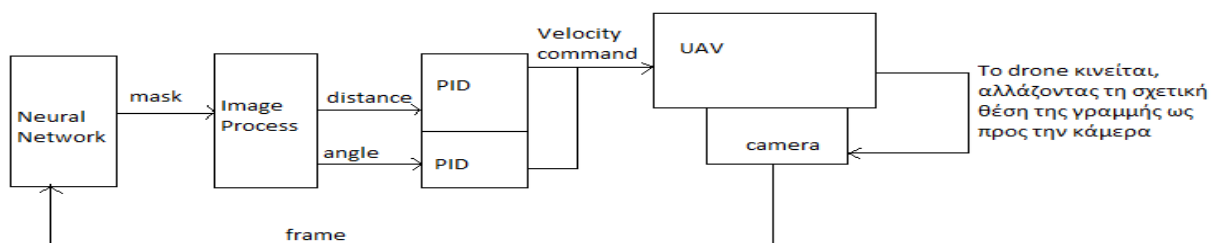
Εικόνα 2.8 Αναγνώριση θέσης της γραμμής μέσα στην εικόνα & ιδανική θέση που προσπαθούμε να πετύχουμε

Πετώντας σε σταθερό ύψος, μπορούμε ρυθμίζοντας τη γραμμική ταχύτητα του drone στον y άξονά του (αντιστοιχεί στον άξονα x της εικόνας), να μηδενίσουμε το σφάλμα απόστασης και ρυθμίζοντας τη γωνιακή του ταχύτητα ως προς τον άξονα z, να μηδενίσουμε το σφάλμα γωνίας, διατηρώντας τις ταχύτητες ως προς τους υπόλοιπους άξονες στο μηδέν. Θεωρώντας θετικό error δεξιά του κέντρου της εικόνας και θετική γωνία δεξιόστροφη, αν υπολογιστεί θετικό error, αρκεί το drone να κινηθεί δεξιά σε σχέση με τον άξονα που κοιτά, ενώ αν υπολογιστεί θετική γωνία αρκεί να περιστραφεί δεξιόστροφα. Ιδανικά σκόπος είναι να πετύχουμε θέση του UAV σχετικά με τη γραμμή όπως φαίνεται στην παραπάνω εικόνα (δεξιά) ώστε το UAV να ακολουθεί τη γραμμή όταν κινείται με σταθερή ταχύτητα στον x άξονα του local συστήματος συντεταγμένων του (y άξονα της εικόνας) [22].



Εικόνα 2.9 Κλασική δομή PID ελέγχου

Έτσι, σχεδιάστηκαν δύο ανεξάρτητοι ελεγκτές PID, καθένας με είσοδο την τιμή σφάλματος μίας από τις μεταβλητές (απόσταση, γωνία) και έξοδο την τιμή που πρέπει να πάρει η ταχύτητα, γύρω από τον αντίστοιχο άξονα, προσπαθούμε επομένως να πετύχουμε  $|error| \rightarrow 0$  και  $|angle| \rightarrow 0$ . Καθώς το drone πετά με σταθερή ταχύτητα στον x άξονα, η διόρθωση του σφάλματος απόστασης κρατά πάντα τη γραμμή εντός οπτικού πεδίου του drone, ενώ η διόρθωση της γωνίας προσαρμόζει την κατεύθυνσή του, ώστε να συμπίπτει με την κατεύθυνση της γραμμής που πρέπει να ακολουθηθεί και μάλιστα κινούμενο όσο πιο ομαλά. Σε θεωρητικό περιβάλλον χωρίς διαταραχές, θα αρκούσε μόνο ο έλεγχος του σφάλματος γωνίας για να παρακολουθήσουμε τη γραμμή.



Εικόνα 2.10 Διάγραμμα συνολικού συστήματος

### 3 Σύστημα Παρακολούθησης με Οπτική Ανατροφοδότηση

Για να επιτευχθεί η παρακολούθηση της διαχωριστικής γραμμής δρόμου/πεζοδρομίου από ένα UAV, έπρεπε να ολοκληρωθούν τέσσερις βασικές υποκατηγορίες του συνολικού προβλήματος:

- Η κατασκευή και ρύθμιση ενός drone με πλαίσιο έξι κινητήρων (hexarotor)
- Η εκπαίδευση ενός νευρωνικού δικτύου για την αναγνώριση της γραμμής ενδιαφέροντος μέσα σε μια εικόνα
- Ο σχεδιασμός συστήματος ελέγχου, ώστε το drone να παρακολουθεί τη γραμμή που ανίχνευσε το νευρωνικό δίκτυο
- Η διασύνδεση όλων των υποσυστημάτων και η λειτουργία μέσω του Robot Operating System (ROS)

#### 3.1 Κατασκευή και Ρύθμιση UAV

Το drone που χρησιμοποιήθηκε στη συγκεκριμένη εργασία είναι μια ανακατασκευή του AscTec Hexacopter [23]. Πρόκειται για ένα μικρό UAV με πλαίσιο έξι κινητήρων. Οι έλικές του είναι διαστάσεων 8x4.5 (διάμετρος x pitch) inches και οι κινητήρες του είναι brushless 2300KV (KV = rpm/pickVolts), καθένας συνδεδεμένος με ένα ESC (Electronic Speed Controller).



Είναι εφοδιασμένο με:

- ένα Pixhawk 2.1(Cube Black) [24] ως Autopilot (μικροελεγκτή πτήσης) και
- ένα Nvidia Jetson Nano [25] ως companion computer.

Εικόνα 3.1 AscTec Hexacopter



Εικόνα 3.2 Pixhawk2.1 Cube Black



Εικόνα 3.3 Nvidia Jetson Nano

Οι δύο αυτοί υπολογιστές επικοινωνούν μέσω του πρωτοκόλλου MavLink [26].

Οι επιπρόσθετοι αισθητήρες που είναι συνδεδεμένοι με τον Autopilot είναι:

- ένα GPS,
- ένα telemetry module για RC χειρισμό ή χειρισμό μέσω GCS.
- ένας αισθητήρας optical flow και
- ένας LIDAR Rangefinder, τοποθετημένοι και οι δύο να κοιτούν κάθετα προς τα κάτω, για να έχουμε πληροφορία σχετικά με τη θέση του drone αν δεν λειτουργεί το GPS, ενώ
- μια USB camera (480x640), είναι συνδεδεμένη στο Jetson Nano.

Για την τοποθέτηση του Pixhawk, αλλά και του GPS, εκτυπώθηκαν σε 3D printer δύο κατάλληλες βάσεις με υλικό ABS.



Εικόνα 3.4 Εικόνες του UAV

Η κατασκευή του drone και η διασύνδεση των επιμέρους ηλεκτρονικών στοιχείων, πραγματοποιήθηκε σύμφωνα με το ArduPilot (open source autopilot system) [27].

Η ρύθμισή του (calibrations, tuning, flight modes κτλ) πραγματοποιήθηκε με τη βοήθεια του Mission Planner, ενός Ground Control Station, για αυτόνομα οχήματα ArduPilot [28].

Πριν τη χρήση του σε πειράματα ακολούθησης γραμμής, πραγματοποιήθηκε πλήθος πτήσεων και δοκιμών της λειτουργίας του drone τόσο με χειροκίνητες πτήσεις μέσω Radio Control όσο και με αυτόματες αποστολές μέσω Mission Planner για κάλυψη διαδρομής συγκεκριμένων WayPoints όπου απαιτείται χρήση του GPS και μέσω dronekit [29], μιας βιβλιοθήκης της python που επικοινωνεί με τον autopilot με μηνύματα MavLink. Στη δεύτερη περίπτωση μπορούμε να ορίσουμε αποστολές στο drone που δεν απαιτούν χρήση GPS, αλλά βασίζονται στους υπόλοιπους αισθητήρες. (πχ το drone να κινηθεί ανατολικά, ή δεξιά σε σχέση με το που κοιτά με σταθερή ταχύτητα).

## 3.2 Εκπαίδευση Νευρωνικού Δικτύου

Στο προηγούμενο κεφάλαιο έγινε μια σύντομη περιγραφή του τρόπου που ένα Convolutional Neural Network μπορεί να αναγνωρίσει την ύπαρξη και τη θέση ενός αντικειμένου σε μια εικόνα. Στην παράγραφο αυτή, αναλύεται η μέθοδος αναγνώρισης της διαχωριστικής γραμμής δρόμου/πεζοδρομίου που εφαρμόστηκε στην παρούσα εργασία, με τη βοήθεια ενός νευρωνικού δικτύου το οποίο εκπαιδεύτηκε για την επίτευξη αυτής της συγκεκριμένης αποστολής.

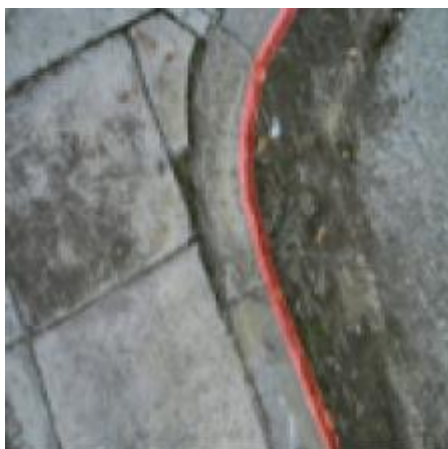
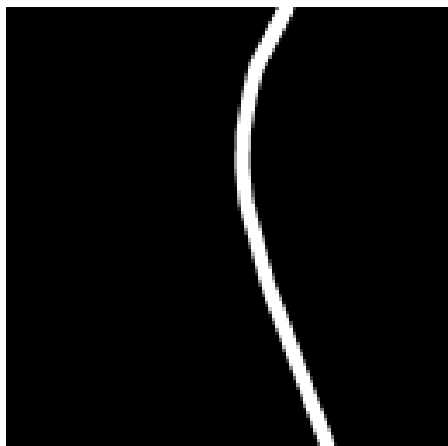
Πρώτο βήμα σε κάθε εφαρμογή που περιλαμβάνει εκπαίδευση νευρωνικών δικτύων με supervised learning τεχνικές, είναι η συλλογή των απαραίτητων δεδομένων σε ικανοποιητική ποσότητα. Στην συγκεκριμένη περίπτωση, τα δεδομένα αυτά είναι εικόνες που περιέχουν γραμμή δρόμου πεζοδρομίου. Οι εικόνες αυτές συλλέχθηκαν μέσω της κάμερας που είναι εφοδιασμένη στο drone και κοιτά κάθετα προς τα κάτω, από χειροκίνητες πτήσεις πάνω από πεζοδρόμια με όσο το δυνατόν περισσότερη ποικιλομορφία.

Επόμενο βήμα είναι, αφού πραγματοποιηθεί επιλογή των πιο αντιπροσωπευτικών εικόνων από όσες τραβήχτηκαν, να ορισθούν χειροκίνητα τα κομμάτια κάθε εικόνας που περιέχουν τη διαχωριστική γραμμή ώστε να χρησιμοποιηθούν αργότερα ως αναφορά για την εκπαίδευση του νευρωνικού δικτύου. Η διαδικασία αυτή ονομάζεται labeling.



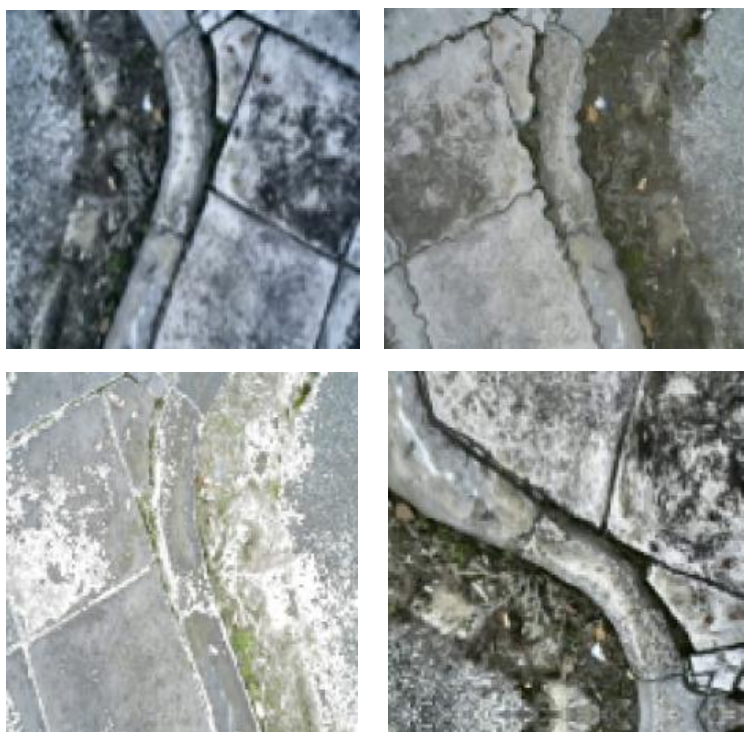
Εικόνα 3.5 Διαδικασία labeling

Συνολικά, πραγματοποιήθηκε labeling σε 2000 εικόνες, μερικές από τις οποίες (10%) δεν χρησιμοποιήθηκαν στο training αλλά για την αξιολόγηση του νευρωνικού. Συγκεκριμένα, σε κάθε εικόνα επιλέχθηκαν χειροκίνητα κάποια σημεία που ανήκουν στην καμπύλη ενδιαφέροντος. Έπειτα, με τη βοήθεια της βιβλιοθήκης OpenCV [30], ενώνουμε τα σημεία που επιλέχθηκαν νωρίτερα με γραμμές μεταξύ τους, δημιουργώντας μια καμπύλη που αντιπροσωπεύει τη διαχωριστική γραμμή που θέλουμε. Δημιουργώντας μια εικόνα ίδιων διαστάσεων με την αρχική, όπου το background είναι μαύρο, ενώ η καμπύλη της διαχωριστικής γραμμής λευκή, έχουμε κατασκευάσει μια μάσκα αναφοράς για κάθε εικόνα με λευκά τα pixels που περιέχουν τη γραμμή, δηλαδή ανήκουν στην κλάση ενδιαφέροντος.



Εικόνα 3.6 Μάσκα που αντιστοιχεί στην labeled εικόνα

Έπειτα, πραγματοποιήθηκε augmentation των data, δηλαδή από κάθε labeled frame παρήχθησαν 20 ακόμα labeled frames, αλλάζοντας ψηφιακά τον φωτισμό, την περιστροφή της εικόνας, zoom κ. α. παραμέτρους καθώς και συνδυασμούς τους.

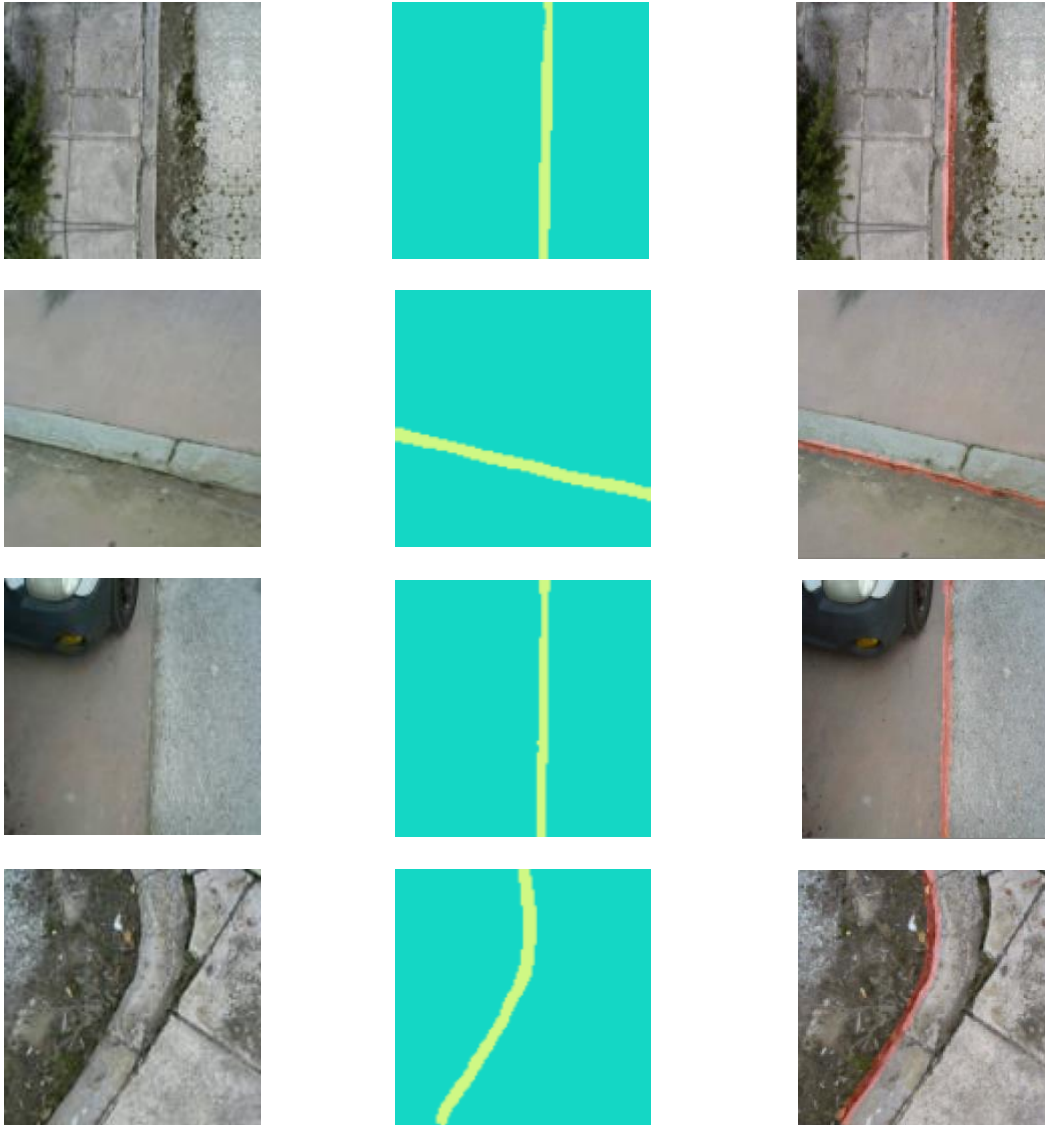


Εικόνα 3.7 Αρχική εικόνα & augmentations

Στο σημείο αυτό, έχει πραγματοποιηθεί η απαραίτητη προετοιμασία πριν την εκπαίδευση του νευρωνικού δικτύου. Το νευρωνικό έχει είσοδο ένα διάνυσμα 224x224, όπως οι διαστάσεις των εικόνων που δέχεται σαν είσοδο. Κάθε στοιχείο του διανύσματος εισόδου αντιστοιχεί σε ένα pixel της εικόνας.



Μετά την εκπαίδευση μετρήθηκε η ακρίβεια (accuracy), η οποία αποτελεί το λόγο των ορθών προβλέψεων προς τις συνολικές και οι απώλειες (loss), δηλαδή μια μετρική του σφάλματος της πρόβλεψης από την πραγματική τιμή, με βάση κάποιες εικόνες που είχαν γίνει κανονικά labeled, όμως δεν χρησιμοποιήθηκαν κατά το training για αυτόν ακριβώς το λόγο. Το νευρωνικό, λοιπόν, δοκιμάστηκε σε εικόνες που δεν είχε επεξεργαστεί ξανά και πέτυχε 0.9852 accuracy και 0.0378 loss, έπειτα από 10 εποχές εκπαίδευσης, ενώ από την 3<sup>η</sup> κιάλας εποχή πέτυχε accuracy 0.9721 και loss 0.0639 αποτελέσματα ιδιαίτερα ικανοποιητικά.



*Εικόνα 3.8 Προβλέψεις νευρωνικού δικτύου/ Αναγνώριση γραμμής*

Η την εκπαίδευση και χρήση του νευρωνικού δικτύου για αναγνώριση της γραμμής γίνονται με τη βοήθεια της βιβλιοθήκης Keras [32] της Python και της TensorFlow [33], μιας δηλαδή, ολοκληρωμένης πλατφόρμας ανοιχτού κώδικα για μηχανική μάθηση.

Μερικές χρήσιμες πληροφορίες σχετικά με τη δομή και την εκπαίδευση του νευρωνικού δικτύου, παρουσιάζονται στη συνέχεια:

- Το νευρωνικό δίκτυο που χρησιμοποιήθηκε ονομάζεται mobilenet\_segnet [34] και πρόκειται για ένα convolutional neural network ικανό στην τμηματοποίηση εικόνων. Δέχεται σαν είσοδο εικόνες διαστάσεων 224x224 pixels.
- Εκπαιδεύτηκε να αναγνωρίζει 2 κλάσεις, δηλαδή, διαχωριστική γραμμή ή όχι.
- Αποτελείται από 100 layers (τύπων ZeroPadding2D, Convolution2D, BatchNormalization, DepthwiseConvolution2D, Activation, UpSampling2D και ένα Reshape στην έξοδο) και την απόκρισή του καθορίζουν 5.541.378 παράμετροι από τις οποίες οι 5.524.738 είναι trainable.
- Συνολικά, πραγματοποιήθηκε labeling σε 1400 εικόνες, όπου κάθε μια επαυξήθηκε x25 δίνοντας ένα σύνολο 35.000 εικόνων και άλλων τόσων масκών. Περίπου το 10% από αυτές, μεταφέρθηκε στο validation set.
- Εκπαιδεύτηκε για 10 εποχές (epoch: υποδεικνύει τον αριθμό που έχει επεξεργαστεί όλο το dataset από το νευρωνικό δίκτυο),
- με batch size 10 (αναφέρεται στον αριθμό των παραδειγμάτων εκπαίδευσης που χρησιμοποιούνται σε μία επανάληψη),
- 512 βήματα ανά εποχή (δηλώνει τον αριθμό των παρτίδων--batches που θα επεξεργάζονται ταυτόχρονα ανά κάθε εποχή).
- Σαν Cost Function (μετρική απόκλισης πρόβλεψης από την πραγματική τιμή), επιλέχτηκε η 'Categorical Crossentropy' ή 'Softmax Loss'.
- ενώ, σαν συνάρτηση βελτιστοποίησης του Cost Function (loss) επιλέχτηκε η 'Adam'.
- Η μετρική που χρησιμοποιήθηκε για την αξιολόγηση της απόκρισης του συστήματος, επιλέχτηκε να είναι η ακρίβεια, δηλαδή ο αριθμός των ορθά υπολογισμένων δειγμάτων προς τον συνολικό αριθμό τους.

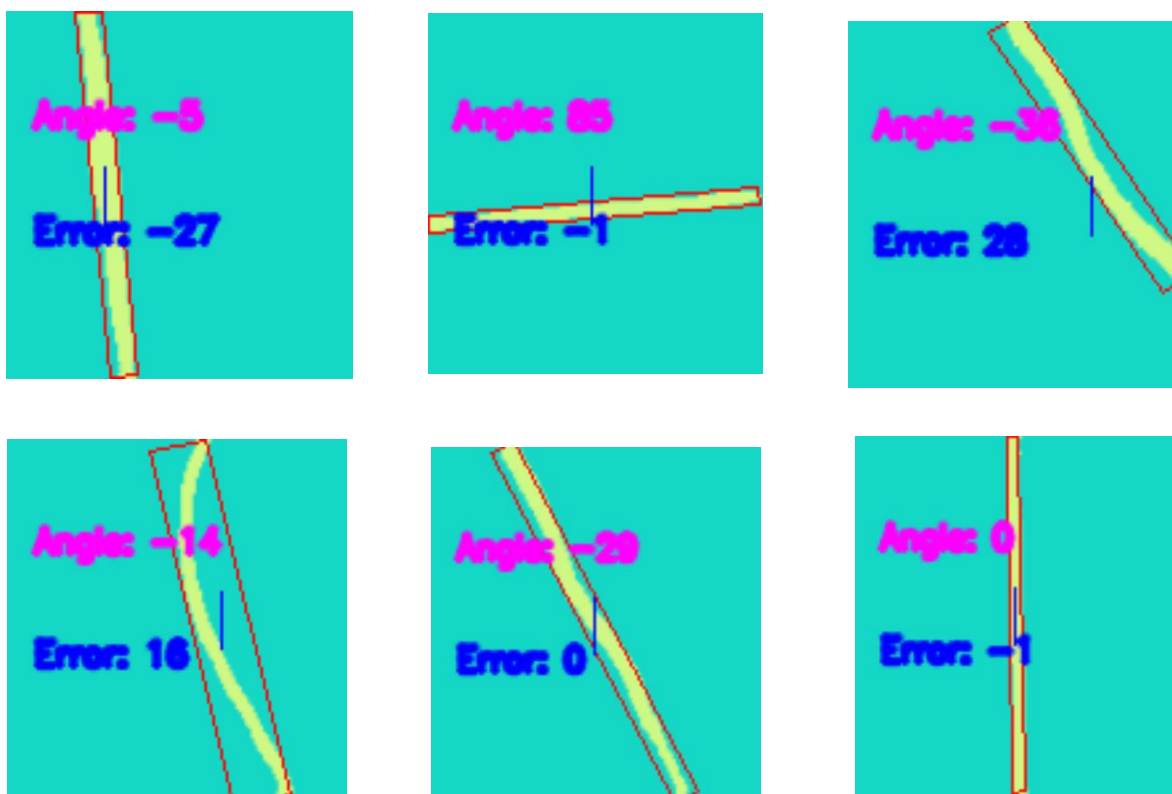
### 3.3 Παρακολούθηση Γραμμής

Στο προηγούμενο κεφάλαιο, επεξηγήθηκε σε θεωρητικό επίπεδο η τεχνική παρακολούθησης της διαχωριστικής γραμμής δρόμου/πεζοδρομίου που χρησιμοποιήθηκε στην εργασία. Στην παράγραφο αυτή, θα αναλυθεί η εφαρμογή της τεχνικής αυτής στην πράξη, θα παρουσιαστούν, δηλαδή, τα βήματα προς την υλοποίησή της και οι απαραίτητες ρυθμίσεις στις μεθόδους και το σύστημα ελέγχου που σχεδιάστηκε.

Αρχικά, πρέπει να υπολογιστούν οι δύο μεταβλητές που θα υποβληθούν σε έλεγχο, δηλαδή η γωνία της γραμμής που αναγνώρισε το νευρωνικό δίκτυο ως προς τον κάθετο άξονα (angle) και η οριζόντια απόσταση του μέσου της από το κέντρο της εικόνας σε pixels (error).

Θεωρούμε θετικό error αν η γραμμή βρίσκεται δεξιά του κέντρου της εικόνας και θετικό angle, το οποίο γενικά κινείται στο διάστημα  $[-90,90]$ , όταν η γωνία γυρνά δεξιόστροφα σε σχέση με τον κάθετο άξονα.

Παρακάτω φαίνονται μερικά παραδείγματα εύρεσης του παραλληλογράμμου και υπολογισμού των angle και error σε μερικές καμπύλες που αναγνώρισε το νευρωνικό δίκτυο. Η μπλε γραμμή αντιστοιχεί στην ευθεία που είναι παράλληλη με τον κάθετο άξονα και διέρχεται από το κέντρο του παραλληλογράμμου. Η εύρεση των σημείων πραγματοποιείται με τη βοήθεια της βιβλιοθήκης OpenCV και η αναγνώριση βασίζεται στη χρωματική διαφορά της γραμμής από το background.



Εικόνα 3.9 Αναγνώριση θέσης & προσανατολισμού γραμμής μέσα στη μάσκα του NN



Γνωρίζοντας, πλέον, τα δύο σφάλματα που ανατροφοδοτούνται και πρέπει να μηδενιστούν, σχεδιάζεται ο νόμος ελέγχου. Οι τιμές των *error* και *angle* ανατροφοδοτούνται και υποβάλλονται σε ένα PD έλεγχο η καθεμία της μορφής:

$$linearvelocity_y = -[K_P * error + K_D * (error - error_{previous})]$$

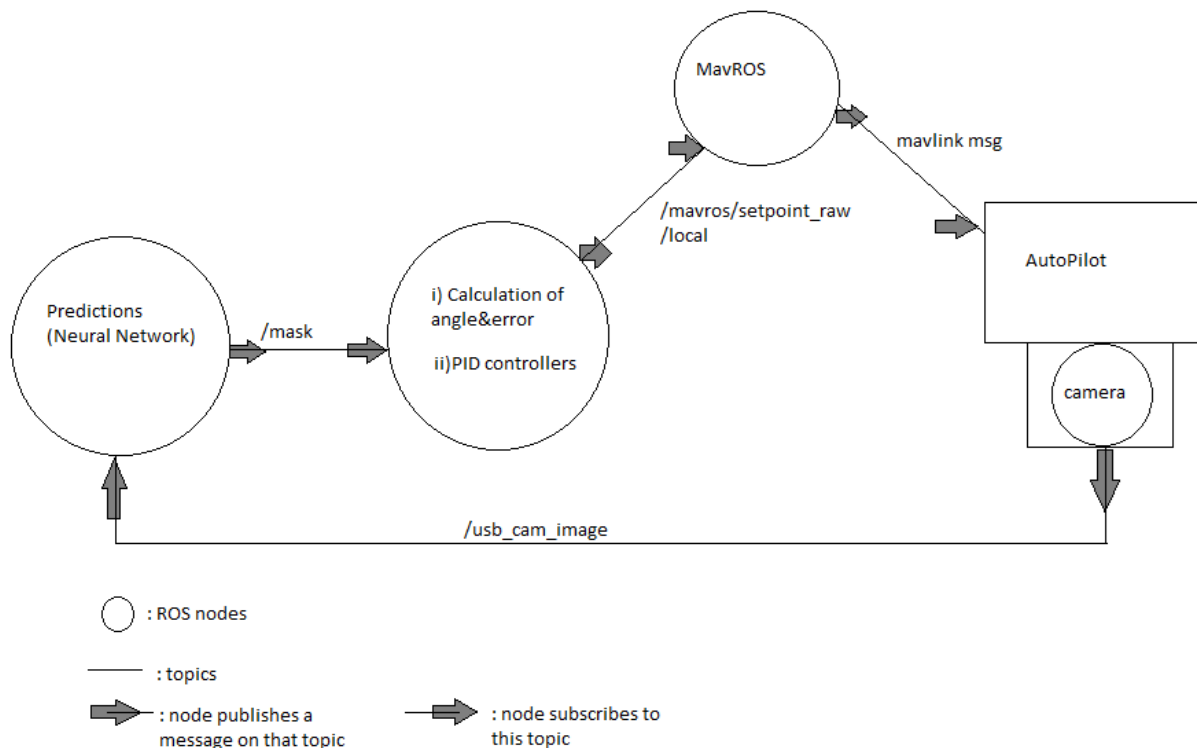
$$angularvelocity_z = -[K_{P_{angle}} * angle + K_{D_{angle}} * (angle - angle_{previous})]$$

Η έξοδος κάθε PD αποστέλλεται στον μικροελεγκτή του drone, σαν εντολή ταχύτητας, ως προς τον αντίστοιχο άξονα, δηλαδή γωνιακή ταχύτητα γύρω από τον z από το *angle* και γραμμική ταχύτητα στον y από το *error*, θεωρώντας τοπικό σύστημα συντεταγμένων όπου η αρχή των αξόνων συμπίπτει με το κέντρο βάρους του drone και ο άξονας x με τον άξονα που το drone κοιτά κάθε στιγμή. Αν το drone χάσει τη γραμμή από το οπτικό του πεδίο λόγω κάποιας εξωτερικής διαταραχής, (πχ ανέμου) δηλαδή αν η έξοδος του νευρωνικού δικτύου είναι μια κενή μάσκα, δίνεται στο drone εντολή να κινηθεί στον y άξονα με σταθερή ταχύτητα, ανάλογα την τιμή (θετική ή αρνητική) του *error* την τελευταία φορά που ανιχνεύτηκε γραμμή.

### 3.4 Robot Operating System (ROS)

Η διασύνδεση όλων των παραπάνω υποσυστημάτων, η μεταφορά πληροφοριών μεταξύ τους και η επικοινωνία με τον Autopilot του ρομπότ, ώστε να αποστέλλονται εντολές κίνησης, επιτεύχθηκε με τη βοήθεια του Robot Operating System (ROS) Melodic [1,35]. Συγκεκριμένα, το σύστημα αποτελείται από 4 κόμβους (executable scripts):

- **Camera:** Δημοσιεύει μηνύματα με μορφή εικόνων στο topic */usb\_cam\_image*. Τα μηνύματα αυτά είναι τα frames που βλέπει η κάμερα, με συχνότητα 20Hz.
- **Prediction:** Κάνει Subscribe στο */usb\_cam\_image*, περνά την εικόνα από το νευρωνικό δίκτυο και δημοσιεύει την μάσκα που αυτό προέβλεψε, σαν μήνυμα εικόνας, στο topic */mask* στα 7Hz.
- **MAVROS [40]:** Πρόκειται για ένα κόμβο του ROS που διασυνδέει μηνύματα MavLink με συγκεκριμένα ROS topics επιτρέποντας σε οχήματα ArduPilot να επικοινωνούν με το ROS. Από τα πολλά topics που προσφέρει, αυτό που χρησιμοποιήθηκε στην εργασία αυτή, είναι το */mavros/setpoint\_raw/local* που επιτρέπει να στέλνονται εντολές ταχύτητας στο drone, γραμμικές και γωνιακές ως προς τους άξονες του τοπικού συστήματος συντεταγμένων του.
- **Servoing:** Κάνει Subscribe στο */mask*, υπολογίζει τα *angle* και *error* όπως αυτά ορίστηκαν στην προηγούμενη παράγραφο και δημοσιεύει την ταχύτητα που υπολογίστηκε πως πρέπει να έχει το drone μέσω των δύο PD ελέγχων στο topic */mavros/setpoint\_raw/local*, ώστε το drone να κινηθεί αντίστοιχα.



Εικόνα 3.10 Διάγραμμα συνολικού συστήματος στο ROS

### Συνολικό Σύστημα και Προβλήματα

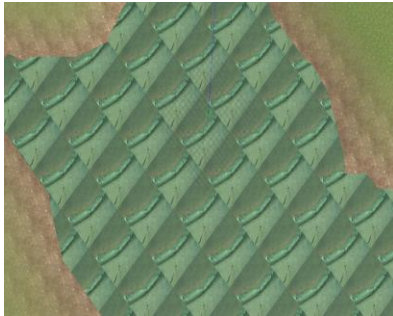
Εξετάζοντας ένα ένα τα επιμέρους υποσυστήματα που σχεδιάστηκαν και συντελούν στη λύση του προβλήματος παρακολούθησης μιας γραμμής μέσω οπτικής ανάδρασης, μπορούμε να ισχυριστούμε πως λειτουργούν πλήρως ικανοποιητικά, κάτι που επαληθεύεται στην πράξη, όπως θα συζητηθεί στα επόμενα κεφάλαια. Το νευρωνικό δίκτυο προβλέπει με  $accuracy > 98\%$  και  $loss < 0.05$ . Ο υπολογισμός των  $angle$  και  $error$  γίνεται με μεγάλη ακρίβεια και η συνολική χρονική καθυστέρηση από τη στιγμή που θα δημοσιευθεί ένα frame μέχρι να αποσταλεί στον autopilot εντολή ταχύτητας είναι  $< 0.05$  δευτερόλεπτα σε πραγματικές συνθήκες.

Βασικό πρόβλημα αποτελεί, βέβαια, πως για να παράγει το νευρωνικό δίκτυο προβλέψεις, χρησιμοποιεί οριακά ολόκληρη την GPU του Jetson Nano και την 4GB RAM του, εισάγοντας καθυστερήσεις στο σύστημα (η κάμερα δημοσιεύει εικόνες στα 20Hz, ενώ ο κόμβος του νευρωνικού δικτύου στα 7Hz) γεγονός που έχει επιπτώσεις και στην ομαλή παρακολούθηση της γραμμής, ειδικά όταν αυξάνεται η επιθυμητή ταχύτητα παρακολούθησης. Σίγουρα με έναν πιο ισχυρό υπολογιστή, η λειτουργία του συστήματος θα ήταν ταχύτερη και η παρακολούθηση πιο ομαλή και σθεναρή σε εξωτερικές διαταραχές και αυξημένη ταχύτητα. Επιπλέον, το πρόβλημα αυτό αποτέλεσε περιοριστικό παράγοντα στο σχεδιασμό των επιμέρους αλγορίθμων.

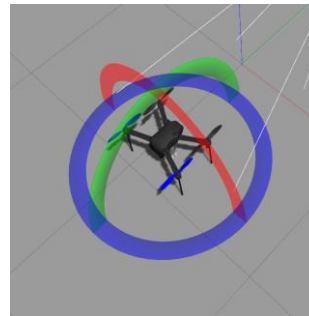
## 4 Πειράματα και Αποτελέσματα

### 4.1 Προσομοιώσεις

Αφού σχεδιάστηκαν και συνδέθηκαν τα επιμέρους υποσυστήματα, πραγματοποιήθηκαν πειραματικές δοκιμές του συνολικού συστήματος, αρχικά, σε περιβάλλον προσομοίωσης Gazebo [36], με τη βοήθεια του ROS και SITL [41]. Το μοντέλο UAV στο οποίο έγιναν δοκιμές είναι το iris quadcopter [37]. Το έδαφος του κόσμου προσομοίωσης δημιουργήθηκε από εικόνες που συλλέχθηκαν από πτήσεις του πραγματικού drone. Οι κόμβοι και τα ROS topics που χρησιμοποιήθηκαν είναι αντίστοιχα με αυτά που περιγράφηκαν στο προηγούμενο κεφάλαιο.

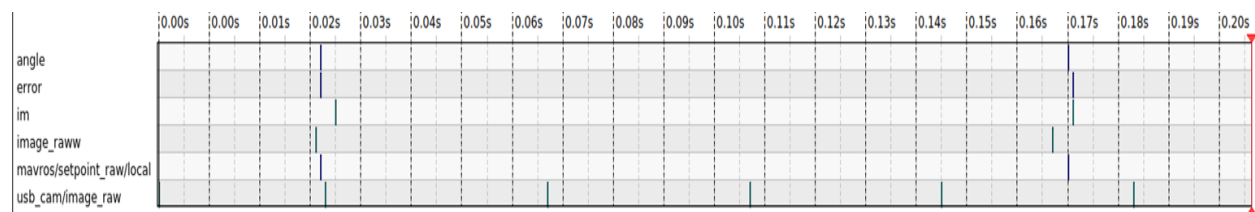


Εικόνα 4.1 Gazebo world



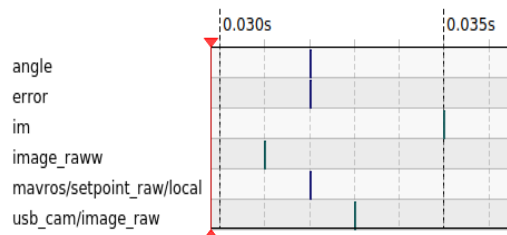
Εικόνα 4.2 iris model

Στη συνέχεια φαίνονται τα αποτελέσματα του πειράματος παρακολούθησης της διαχωριστικής γραμμής δρόμου και πεζοδρομίου, όπως αυτά αποθηκεύτηκαν κατά την εκτέλεση της αυτόνομης αποστολής, στον κόσμο που φαίνεται παραπάνω.



Εικόνα 4.3 ROS topics ως προς το χρόνο

Παραπάνω, φαίνεται η χρονική στιγμή που δημοσιεύεται κάποιο μήνυμα σε κάθε ROS topic. Η κάμερα δημοσιεύει frames στο usb\_cam/image\_raw, το νευρωνικό δίκτυο στο image\_raw, ενώ τα υπόλοιπα topics δημοσιεύονται από τον ίδιο κόμβο και αφορούν τα σφάλματα γωνίας (angle) και απόστασης (error), τον εντοπισμό του παραλληλογράμμου που περικλείει τη γραμμή (im) και την εντολή ταχύτητας προς το drone (mavros/setpoint\_raw/local).

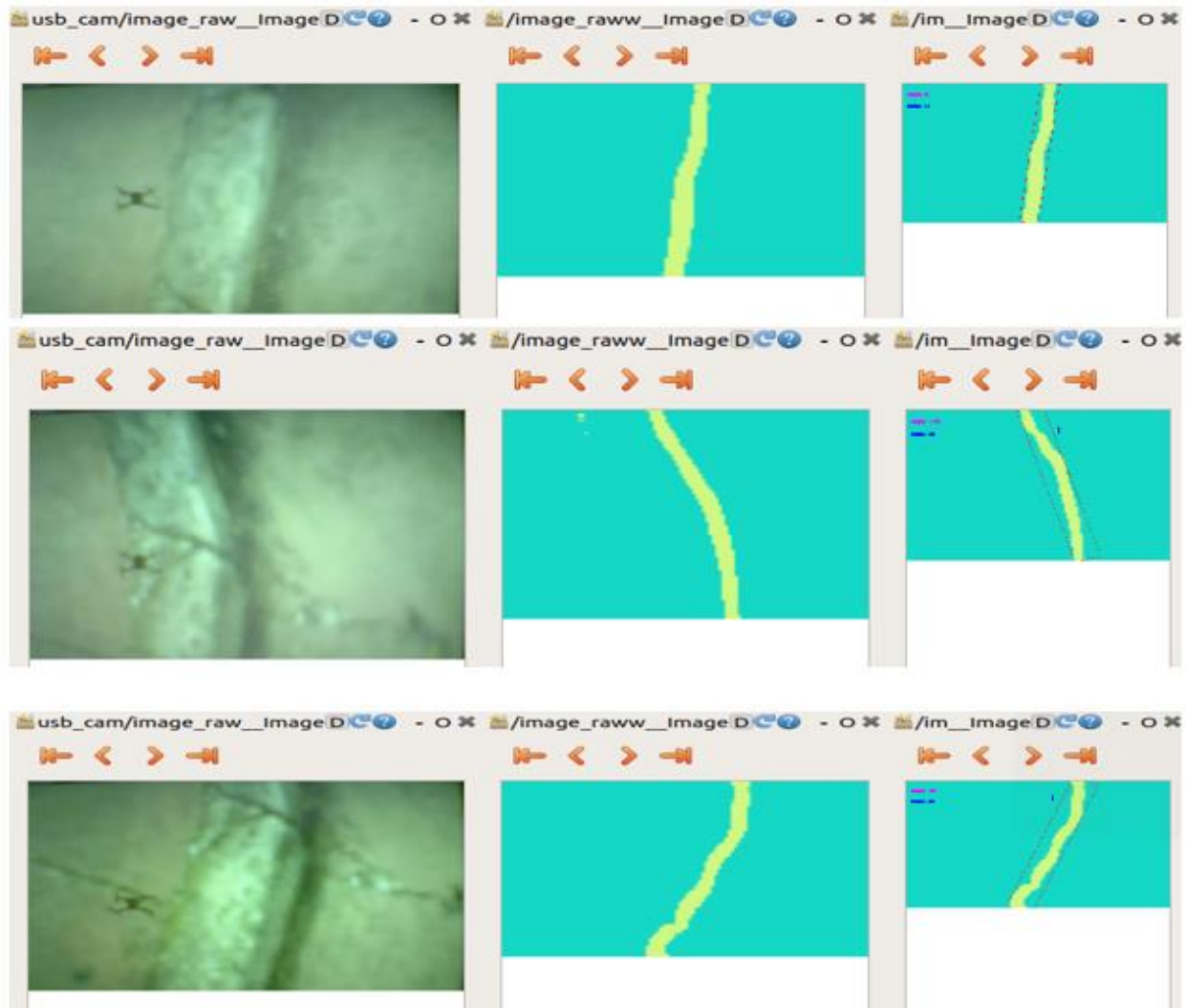


Εικόνα 4.4 ROS topics (2)

του οποίου εξαρτάται από τη συχνότητα δημοσίευσης του νευρωνικού δικτύου, δημοσιεύει στα 4-6hz κατά μέσο όρο.

Παρατηρούμε πως το χρονικό διάστημα από την στιγμή δημοσίευσης μιας εικόνας, μέχρι να δοθεί εντολή ταχύτητας στο drone είναι περίπου 0.025 sec, ενώ από τη δημοσίευση της μάσκας, μέχρι την εντολή είναι μόλις 0.001 sec. Ο κόμβος που διαχειρίζεται το νευρωνικό δίκτυο εισάγει τη σημαντικότερη καθυστέρηση στο σύστημα και δημοσιεύει στα 7hz κατά μέσο όρο σε σχέση με τα 20hz που δημοσιεύει ο κόμβος της κάμερας. Ενδεικτικά, ο κόμβος που δημοσιεύει τα error, angle, im και mavros/setpoint\_raw/local, η συχνότητα

στη συνέχεια φαίνονται στη σειρά τα μηνύματα των topics που αφορούν το frame, την αντίστοιχη πρόβλεψη από το νευρωνικό δίκτυο και τον εντοπισμό της γραμμής, σε τυχαίες χρονικές στιγμές.



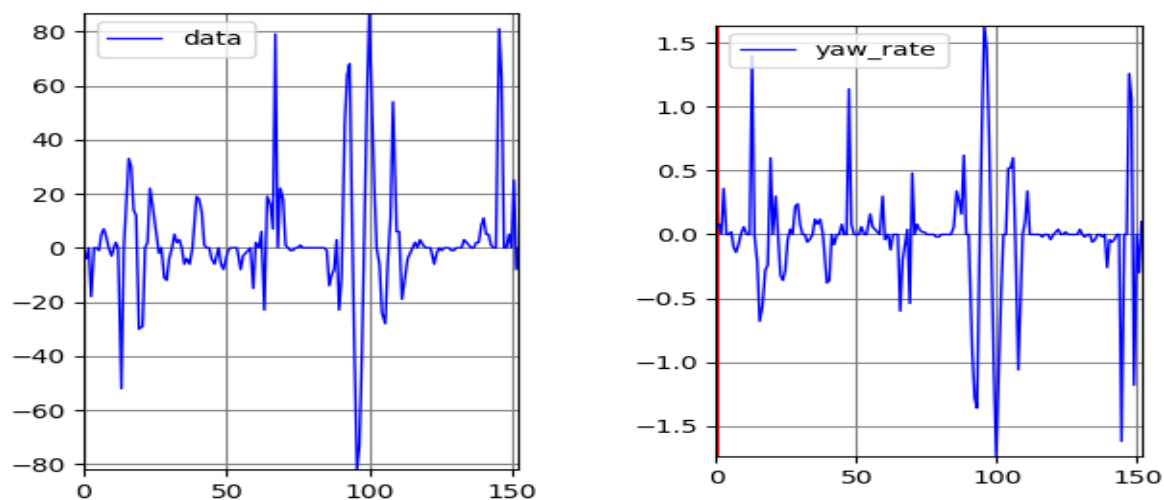
Εικόνα 4.5 Frame-Prediction-Error&Angle

Η αναγνώριση της γραμμής από το νευρωνικό δίκτυο λειτουργεί με μεγάλη ακρίβεια και μάλιστα σε διαφορετικά εδάφη (εικόνες πεζοδρομίων) στον κόσμο προσομοίωσης. Το ίδιο συμβαίνει και με την αναγνώριση της θέσης της στην εικόνα από τον επόμενο κόμβο.

Παρουσιάζονται στη συνέχεια τα γραφήματα που μας δείχνουν τις τιμές των *angle* και *error* ως προς το χρόνο, κατά τη διάρκεια της αυτόνομης πτήσης, για δύο διαφορετικά πειράματα. Φαίνεται ξεκάθαρα η προσπάθεια των ελεγκτών να μηδενίσουν τα *angle* και *error*. Επιπλέον, παρουσιάζονται τα γραφήματα για τις αντίστοιχες τιμές των ταχυτήτων που στέλνονται σαν εντολές στο drone. Εύκολα επαληθεύεται η αναλογική σχέση ανάμεσα στις εισόδους και εξόδους των ελεγκτών.

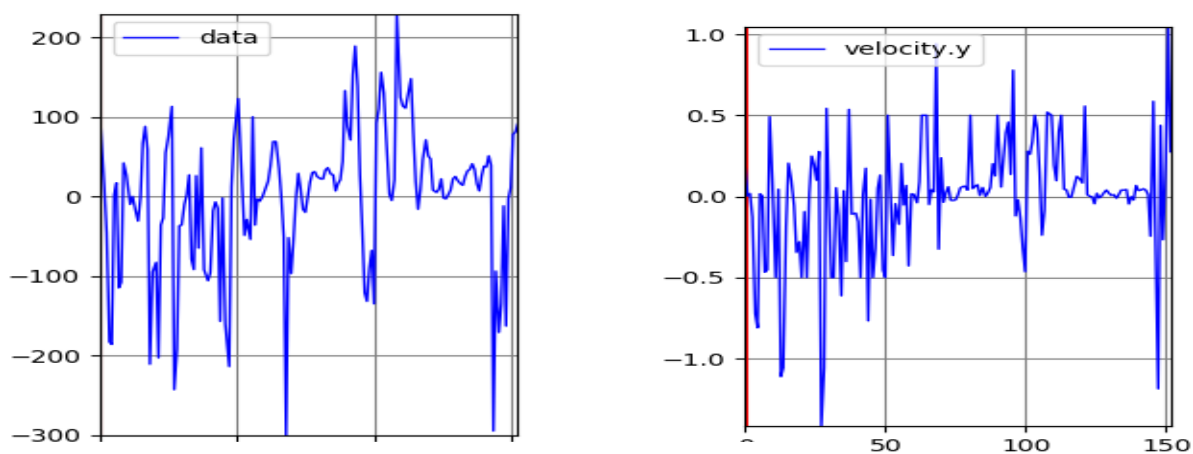
Ι) Πείραμα με δύο PD ελέγχους και υψηλή ταχύτητα παρακολούθησης (2m/s)

Angle/yaw\_rate:



Εικόνα 4.6 Διάγραμμα *angle* & *yaw rate* στη διάρκεια μιας προσομοίωσης

Error/velocity.y:

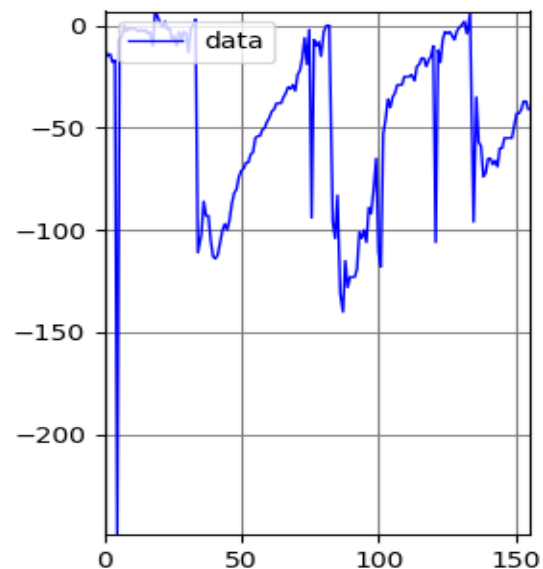
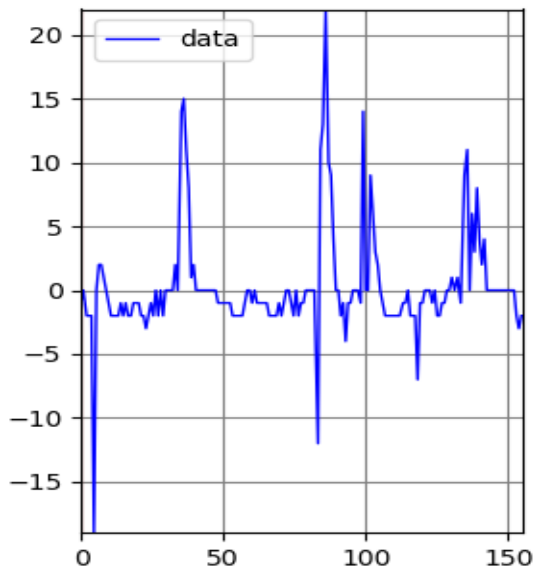


Εικόνα 4.7 Διάγραμμα *error* & *velocity.y* στη διάρκεια μιας προσομοίωσης

## II) Πείραμα με δύο P ελέγχους και χαμηλή ταχύτητα παρακολούθησης (0.4m/s)

Angle:

Error:

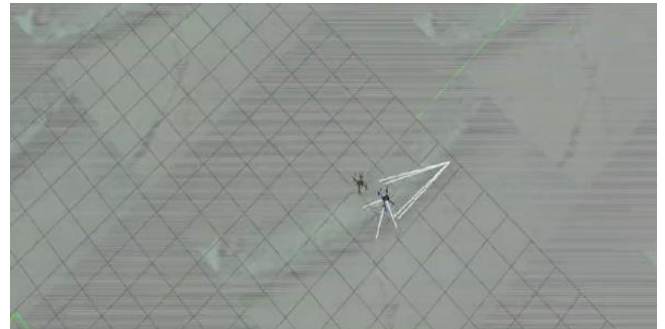


Εικόνα 4.8 Διαγράμματα  $angle$  &  $error$  σε προσομοίωση με P ελέγχους

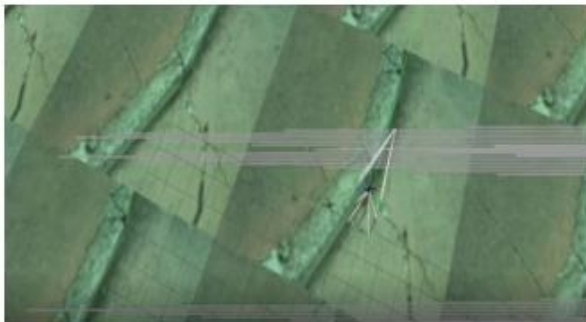
Παρατηρούμε από τα δύο παραπάνω πειράματα, πως υψηλότερες ταχύτητες παρακολούθησης, προκαλούν περισσότερες ταλαντώσεις στις αποκρίσεις των  $error$  και  $angle$ .

Ακόμα, είναι εμφανής η ευεργετική επηρροή του PD ελέγχου της πρώτης περίπτωσης σε σχέση τον απλό αναλογικό της δεύτερης, ακόμα και αν η ταχύτητα παρακολούθησης είναι υψηλότερη στην πρώτη περίπτωση. Ειδικά στο  $error$ , παρατηρούμε πως η χρήση αναλογικού ελέγχου οδηγεί στην ύπαρξη ενός μόνιμου σφάλματος από την επιθυμητή τιμή (μηδέν).

Τέλος, παρουσιάζονται στιγμιότυπα της προσομοίωσης στο Gazebo, όπου το UAV φαίνεται να ακολουθεί τη διαχωριστική γραμμή δρόμου/πεζοδρομίου, προσπαθώντας να τη διατηρεί στο κέντρο του οπτικού του πεδίου, με προσανατολισμό παράλληλο σε αυτόν του UAV. Σημειώνεται, πως η μέγιστη ταχύτητα παρακολούθησης με την οποία έχουμε ικανοποιητικά αποτελέσματα είναι m/s, ενώ το ύψος πτήσης δεν επηρεάζει σημαντικά την ορθή λειτουργία του συστήματος.



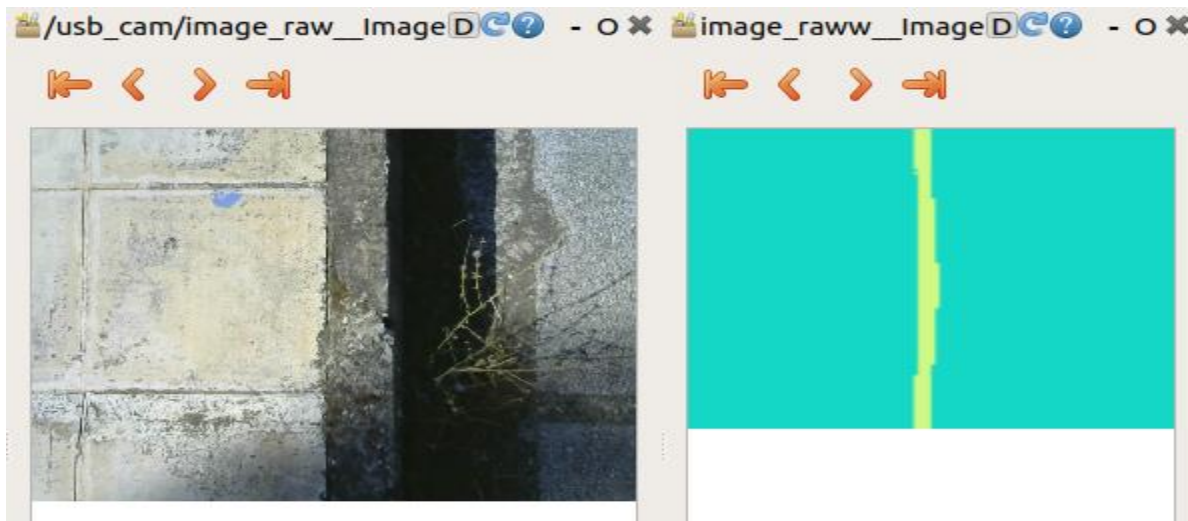


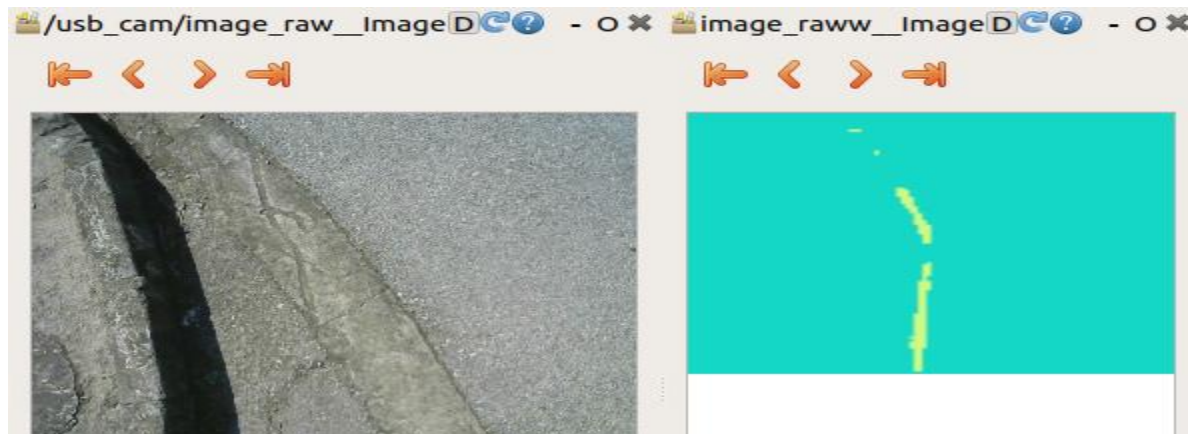
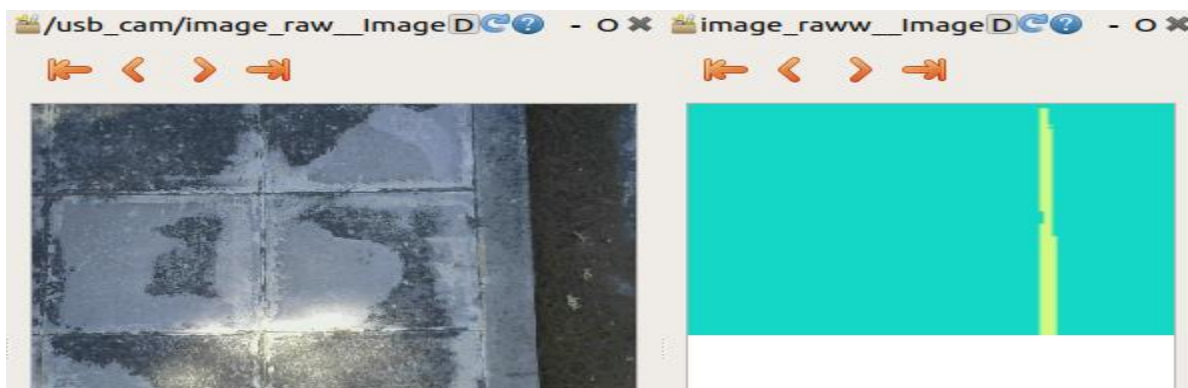
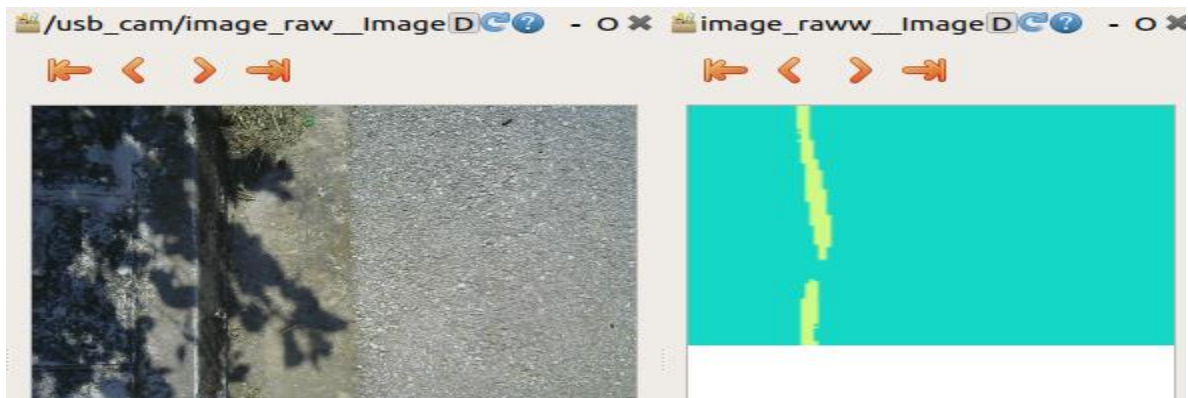


Εικόνα 4.9 Στιγμιότυπα κατά τη διάρκεια προσομοίωσης

## 4.2 Πειράματα στον Πραγματικό Κόσμο

Παρακάτω, φαίνονται κάποια frames από την κάμερα του UAV και οι αντίστοιχες μάσκες που προέβλεψε το νευρωνικό δίκτυο.

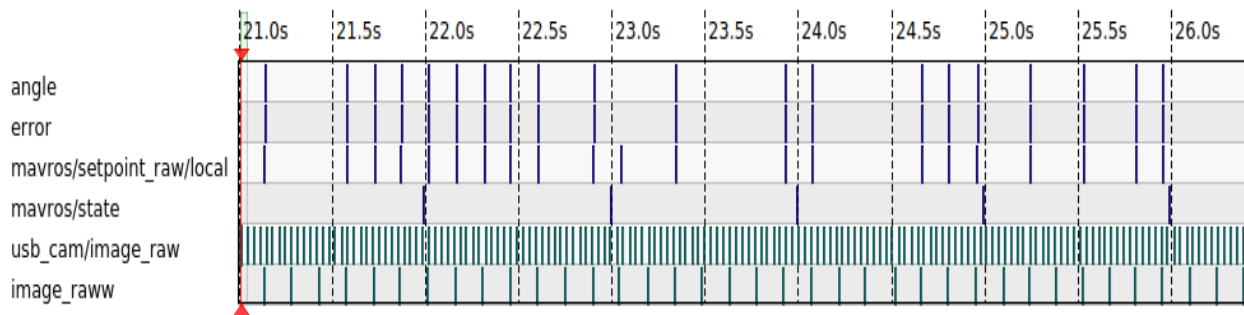




Εικόνα 4.10 Frames & Predictions κατά τη διάρκεια αυτόνομης αποστολής

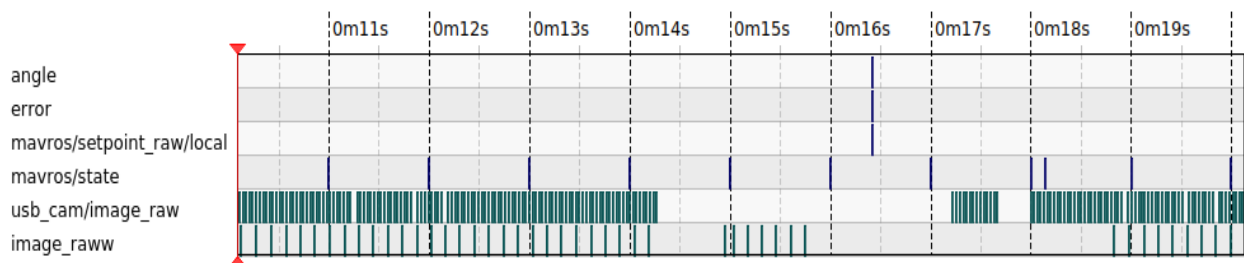
Όπως φαίνεται από τα παραπάνω στιγμιότυπα frame/μάσκας, το νευρωνικό δίκτυο αναγνωρίζει τη γραμμή του πεζοδρομίου με μεγάλη ακρίβεια. Παρόλα αυτά, παρατηρείται μια μετατόπιση της γραμμής στη μάσκα σε σχέση με την πραγματική της θέση. Όπως θα σχολιαστεί και στη συνέχεια, η μετατόπιση αυτή οφείλεται στη χρονική καθυστέρηση που εισάγει το νευρωνικό δίκτυο στο σύστημα. Ουσιαστικά η μάσκα που φαίνεται δεν αντιστοιχεί στο frame που βλέπουμε δίπλα της αλλά σε κάποιο προηγούμενο.





Εικόνα 4.11 ROS topics ως προς το χρόνο σε αυτόνομη πτήση

Στην παραπάνω εικόνα φαίνονται οι στιγμές που δημοσιεύεται κάποιο μήνυμα σε ένα ROS topic. Το average rate που δημοσιεύει frames η κάμερα, είναι 20hz, το νευρωνικό δίκτυο 7hz, ενώ στα υπόλοιπα topics 5hz. Βέβαια, ενώ υπό φυσιολογικές περιπτώσεις η καθυστέρηση ανάμεσα στη δημοσίευση του frame και στη δημοσίευση της επιθυμητής ταχύτητας είναι περίπου 0.03s, υπάρχουν περιπτώσεις που λόγω υπερφόρτωσης της GPU του companion computer Jetson Nano, εισάγονται σημαντικές καθυστερήσεις στο σύστημα και επηρεάζεται αρνητικά η απόδοσή του (κολλάει). Όπως φαίνεται παρακάτω, η κάμερα σταματά να δημοσιεύει frames ανά διαστήματα, προκαλώντας προβλήματα στην ορθή λειτουργία του συστήματος και μειώνοντας τη μέγιστη ταχύτητα παρακολούθησης.



Εικόνα 4.12 ROS topics ως προς το χρόνο σε αυτόνομη πτήση (2)

Ένας τρόπος να αποφευχθούν περιπτώσεις υπολειτουργίας λόγω υπερφόρτωσης της GPU, είναι να μην δημοσιεύονται μηνύματα εικόνων και να μην χρησιμοποιούνται (όσο το δυνατόν) εντολές επεξεργασίας εικόνας στους διάφορους αλγορίθμους. Ενδεικτικά, αν πραγματοποιούνταν δημοσίευση και της εικόνας όπου έχει αναγνωριστεί η γραμμή από την μάσκα της, εισάγεται επιπλέον καθυστέρηση 1s περίπου από την δημοσίευση του frame μέχρι την εντολή ταχύτητας προς το UAV.

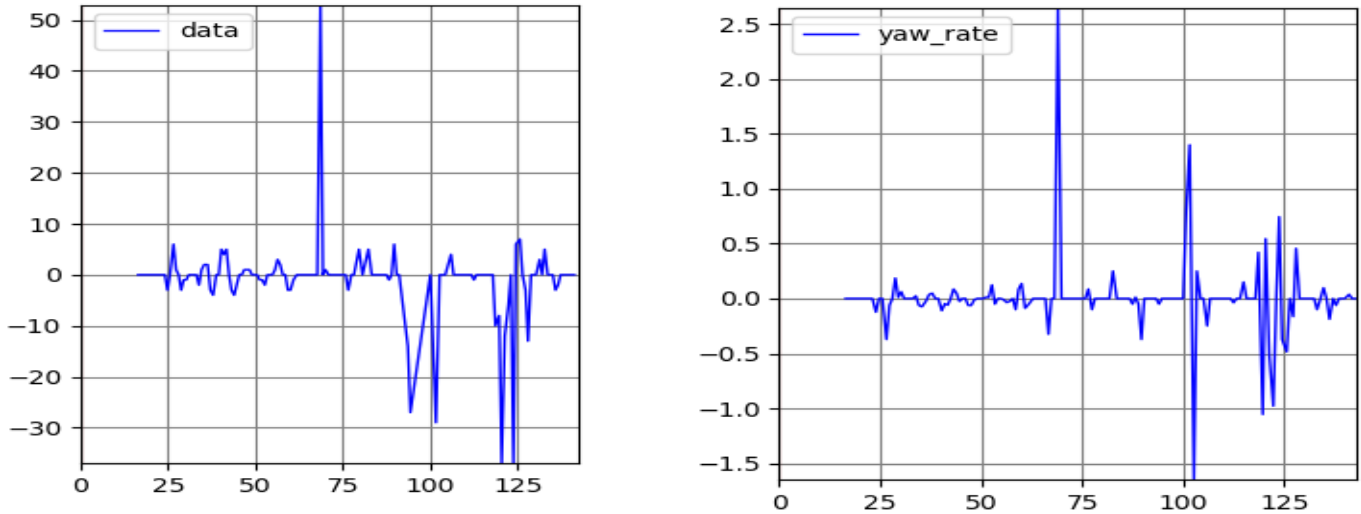
Έπειτα από δοκιμές και διαφορετικές ρυθμίσεις, επιλέχθηκαν ελεγκτές τύπου PD με κέρδη:

Angle: P-->0.02, D-->1

Error: P-->0.2, D-->0.8

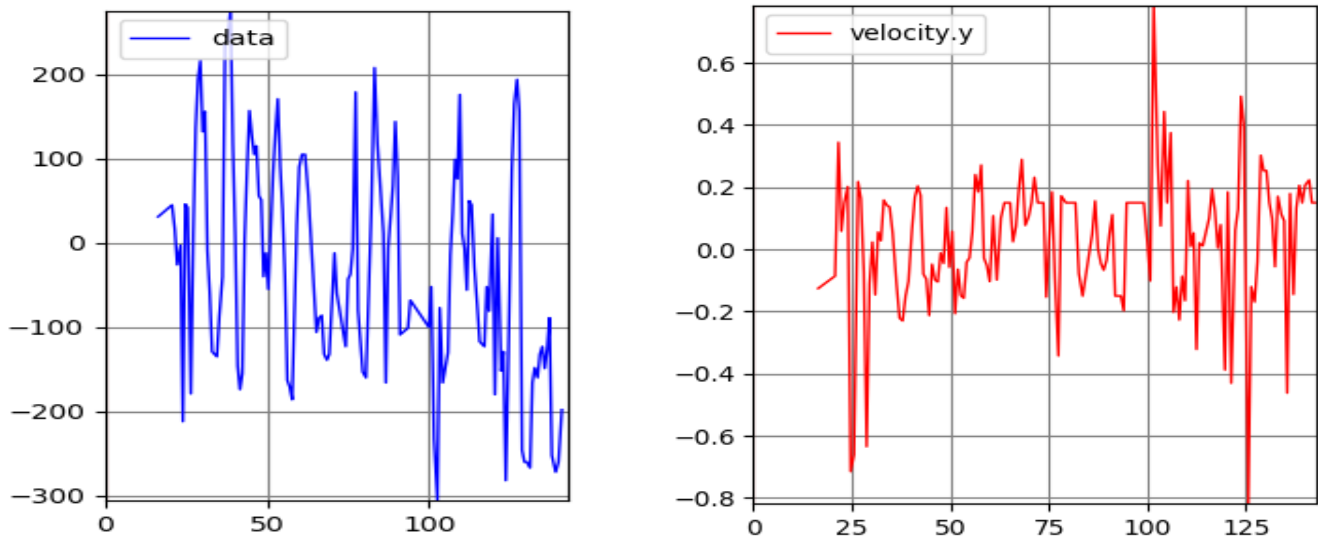
Ο όρος Ki διατηρήθηκε στο μηδέν, ώστε να μην οδηγηθούμε σε αστάθεια από τη συσσώρευση του σφάλματος που αυτός ο όρος 'αποθηκεύει'. Στη συνέχεια, παρουσιάζονται τα διαγράμματα των angle και error (είσοδοι των PD ελεγκτών) και οι αντίστοιχη ταχύτητα (έξοδοι PD) κατά τη διάρκεια της αυτόνομης αποστολής.

Angle/ yaw rate:



Εικόνα 4.13 Διάγραμμα angle & yaw rate στη διάρκεια μιας αυτόνομης πτήσης

Error/ velocity y:

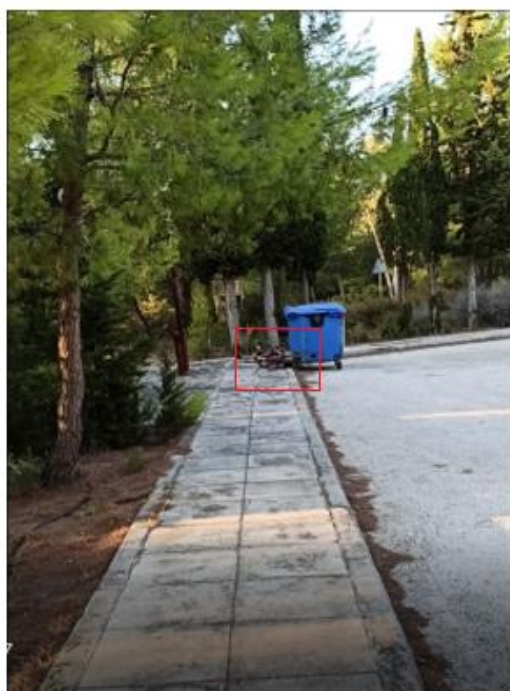
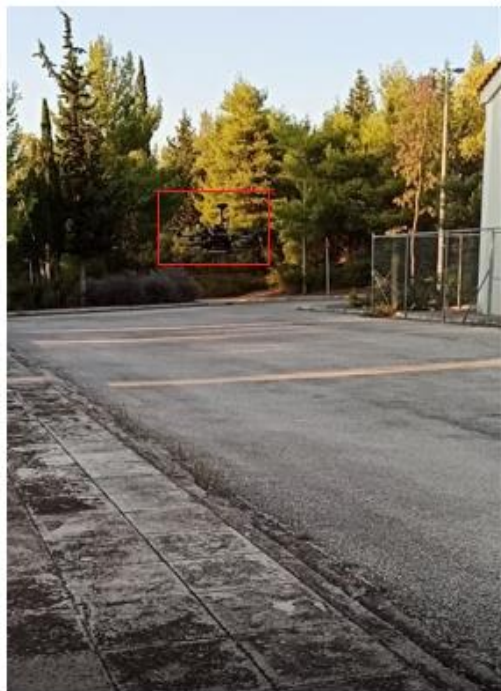


Εικόνα 4.14 Διάγραμμα error & velocity.y στη διάρκεια μιας αυτόνομης πτήσης

Παρατηρούμε πώς παρά τις όποιες διαταραχές από το εξωτερικό περιβάλλον και τις καθυστερήσεις του συστήματος, οι δύο προς έλεγχο μεταβλητές προσπαθούν διαρκώς να ισορροπήσουν γύρω από το μηδέν, επιτυγχάνοντας την παρακολούθηση της επιθυμητής γραμμής από το UAV. Εξ'αιτίας της υπερφόρτωσης της GPU του Jetson Nano, το σύστημα δεν μπορεί να ανταποκριθεί με επιτυχία σε ταχύτητες μεγαλύτερες των 2m/s. Για υψηλότερες ταχύτητες ορθής παρακολούθησης απαιτείται χρήση πιο ισχυρού υπολογιστή. Το ύψος πτήσης (από 0.5 μέχρι 3-4 μέτρα όπου και δοκιμάστηκε) δεν φαίνεται να επηρεάζει την απόκριση του συστήματος.

Οι ταλαντώσεις στο error και η αδυναμία μηδενισμού του, οφείλονται κατά κύριο λόγο σε διαταραχές από το εξωτερικό περιβάλλον (αέρας), ενώ κάποια ricks στο angle, εμφανίζονται όταν υπάρχει στροφή ή λόγω εσφαλμένης πρόβλεψης του νευρωνικού δικτύου, που όμως δεν έχει σοβαρές επιπτώσεις στη συνολική απόκριση του συστήματος και την ικανότητά του να παρατηρεί τη γραμμή.

Τέλος, παρουσιάζονται κάποια στιγμιότυπα, κατά τη διάρκεια αυτόνομων πτήσεων.







Εικόνα 4.15 Στιγμιότυπα κατά τη διάρκεια αυτόνομης πτήσης

## 5 Συμπεράσματα και Μελλοντική Έρευνα

Στην εργασία αυτή, αντιμετωπίστηκε με επιτυχία το πρόβλημα παρακολούθησης της διαχωριστικής γραμμής ανάμεσα σε δρόμο και πεζοδρόμιο μέσω οπτικής ανάδρασης, από ένα UAV, με κάμερα τοποθετημένη να κοιτά κάθετα προς τα κάτω.

Το νευρωνικό δίκτυο που εκπαιδεύτηκε με σκοπό να αναγνωρίζει τη γραμμή αυτή μέσα σε μια εικόνα, εκπληρώνει το ρόλο του με πολύ υψηλή ευστοχία, επιβεβαιώνοντας την ευελιξία, ευκολία στην εφαρμογή και ακρίβεια στα αποτελέσματα που διέπει τα νευρωνικά δίκτυα όταν εφαρμόζονται σε προβλήματα μηχανικής όρασης.

Παράλληλα, η στρατηγική παρακολούθησης της γραμμής, φαίνεται πως λειτούργησε πολύ αποτελεσματικά σε συνδυασμό με την σωστή ρύθμιση των PD ελεγκτών, οι οποίοι αποτελούν μία από τις πιο εύκολα εφαρμόσιμες και αξιόπιστες τεχνικές ελέγχου, προσφέροντας γρήγορο και ταυτόχρονα ομαλό έλεγχο της κίνησης του drone.

Το UAV δεν είχε κανένα πρόβλημα να αναγνωρίζει και να ακολουθεί το πεζοδρόμιο ακόμα και σε περιπτώσεις που το πεζοδρόμιο εμφανίζει μεγάλες καμπύλες ή απότομες γωνίες 90 μοιρών. Αν μετά από κάποια στροφή ή εξωτερική διαταραχή χάσει τη γραμμή από το οπτικό του πεδίο, την ξαναβρίσκει με σχετική ευκολία εκτός από περιπτώσεις ισχυρής διαταραχής (δυνατού ανέμου) και λειτουργεί αποτελεσματικά σε μεγάλο εύρος υψομέτρου πτήσης.

Το παρών έργο, μπορεί να αποτελέσει κίνητρο για τη λύση παρόμοιων προβλημάτων παρακολούθησης μέσω οπτικής ανατροφοδότησης τεθούν μελλοντικά. Το νευρωνικό δίκτυο μπορεί να εκπαιδευτεί για την αναγνώριση οποιουδήποτε αντικειμένου επιθυμεί ο σχεδιαστής να παρακολουθεί, ιδίως όταν το αντικείμενο αυτό είναι μια διαχωριστική γραμμή. Μπορεί, για παράδειγμα, να έχει εφαρμογή στην παρακολούθηση των γραμμών ενός γηπέδου, ακτογραμμών, μονοπατιών, αντιπυρικών ζωνών σε δάση κ.α. με μεγάλη ευκολία.

Επιπλέον, η ρύθμιση των ελεγκτών του είναι αρκετά εύκολη διαδικασία, ενώ με κάποιες αλλαγές στις προς έλεγχο μεταβλητές, το ρομπότ μπορεί να ακολουθεί οποιοδήποτε αντικείμενο, όχι απαραίτητα γραμμή, όπως ανθρώπους, άλλα ρομπότ, αυτοκίνητα κτλ, πρόβλημα που ίσως απασχολήσει έρευνες στο μέλλον.

Μελλοντικές έρευνες με βάση την παρούσα, μπορεί επίσης να αφορούν την παρακολούθηση μιας διαχωριστικής επιφάνειας αλλάζοντας τον νόμο ελέγχου προς βελτιστοποίηση κάποιου συγκεκριμένου κριτηρίου.

Τέλος, κρατώντας αυτούσιο το σύστημα που προτάθηκε στην εργασία αυτή, μπορούν να πραγματοποιηθούν επεκτάσεις στη λειτουργία, όπως αποφυγή αντικειμένων κατά την πτήση με την προσθήκη LIDAR ή δεύτερης κάμερας και εκπαίδευσης του ίδιου νευρωνικού δικτύου να αναγνωρίζει και τα εμπόδια αυτά, εντοπισμός της γραμμής που πρέπει να ακολουθήσει αυτόματα, συνδυασμός πληροφοριών οπτικού αισθητήρα και GPS, λειτουργία ανάκτησης θέσης σε περίπτωση ισχυρής διαταραχής, εφαρμογή σε project μεγαλύτερης κλίμακας, ενώ ιδιαίτερο ενδιαφέρον παρουσιάζει η ιδέα της χρήσης και αλληλεπίδρασης πολλαπλών νευρωνικών δικτύων προς τη δημιουργία ενός “έξυπνου” αυτόνομου UAV.

## Βιβλιογραφία

- [1] <http://wiki.ros.org/>
- [2] C. Rasmussen, Y. Lu, and M. Kocamaz (2009), «Appearance contrast for fast, robust trail-following»
- [3] P. Santana, L. Correia, R. Mendonça, N. Alves, and J. Barata (2013), «Tracking natural trails with swarm-based visual saliency»
- [4] Venugopalan, T., Taher, T., and Barbastathis, G. (2012), «Autonomous landing of an unmanned aerial vehicle on an autonomous marine vehicle in oceans».
- [5] Warren, M., Corke, P., and Upcroft, B. (2015), «Long-range stereo visual odometry for extended altitude flight of unmanned aerial vehicles»
- [6] Sotomayor, J. F., Gomez, A. P., and Castillo, A. (2014), «Visual control of an autonomous aerial vehicle for crop inspection»
- [7] Alexandre S. Brandao, Felipe N. Martins and Higor B. Soneguetti (2015), «A Vision-based Line Following Strategy for an Autonomous UAV»
- [8] Alessandro Giusti , Jérôme Guzzi , Dan C. Cireşan , Fang-Lin He , Juan P. Rodríguez Flavio Fontana , Matthias Faessler , Christian Forster Jürgen Schmidhuber , Gianni Di Caro , Davide Scaramuzza , Luca M. Gambardella (2015), «A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots»
- [9] Roman Bartak Adam Vyskovsky (2015), «Any Object Tracking and Following by a Flying Drone»
- [10] Rafael C. Gonzalez, Richard E. Woods (2011), «Digital Image Processing»
- [11] Koichi Hasimoto (2001), «Real time control of robot manipulators based on visual sensory feedback»
- [12] Valavanis, K.P. (2008), «Advances in unmanned aerial vehicles: state of the art and the road to autonomy»
- [13] Kanellakis, C., Nikolakopoulos, G. (2017), «Survey on computer vision for uavs: Current developments and trends»
- [14] Puri, V., Nayyar, A., Raja, L. (2017), «Agriculture drones: A modern breakthrough in precision agriculture»
- [15] Niall O' Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, Joseph Walsh (2019), «Deep Learning vs. Traditional Computer Vision»
- [16] Zhong-Qiu Zhao, Member, IEEE, Peng Zheng, Shou-tao Xu, and Xindong Wu, Fellow, IEEE (2019), «Object Detection with Deep Learning: A Review»
- [17] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do & Kaori Togashi (2018), «Convolutional neural networks: an overview and application in radiology»
- [18] Jonathan Long Evan Shelhamer Trevor Darrell (2015), «Fully Convolutional Networks for Semantic Segmentation»
- [19] [https://www.researchgate.net/figure/Architecture-of-a-Convolutional-Neural-Network-CNN-The-traditional-CNN-structure-is\\_fig1\\_330106889](https://www.researchgate.net/figure/Architecture-of-a-Convolutional-Neural-Network-CNN-The-traditional-CNN-structure-is_fig1_330106889)
- [20] <https://towardsdatascience.com/training-a-convolutional-neural-network-from-scratch-2235c2a25754>
- [21] Randal W. Beard (2008), «Quadrotor Dynamics and Control»
- [22] Yakov korlansky, Shabi Sabatan, Gabi Davidof (2019), «Autonomous line follower drone using ROS + OpenCV»
- [23] [https://www.researchgate.net/figure/Hexacopter-AscTec-Firefly-25\\_fig8\\_275887696](https://www.researchgate.net/figure/Hexacopter-AscTec-Firefly-25_fig8_275887696)
- [24] [https://docs.px4.io/master/en/flight\\_controller/pixhawk-2.html](https://docs.px4.io/master/en/flight_controller/pixhawk-2.html)
- [25] <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

- [26] <https://mavlink.io/en/>
- [27] <https://ardupilot.org/dev/index.html>
- [28] <https://ardupilot.org/planner/>
- [29] <https://dronekit.io/>
- [30] <https://opencv.org/>
- [31] <https://github.com/wkentaro/labelme>
- [32] <https://keras.io/>
- [33] <https://www.tensorflow.org/>
- [34] <https://github.com/divamgupta/image-segmentation-keras>
- [35] <http://wiki.ros.org/melodic>
- [36] <http://gazebo-sim.org/>
- [37] <http://www.arducopter.co.uk/iris-quadcopter-uav.html>
- [38] <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>
- [39] <https://github.com/aleju/imgaug>
- [40] <http://wiki.ros.org/mavros>
- [41] <https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>
- [42] [http://wiki.ros.org/usb\\_cam](http://wiki.ros.org/usb_cam)



# Παράρτημα Α

## Προετοιμασία και Εκπαίδευση Νευρωνικού Δικτύου

Η προετοιμασία των εικόνων του training set για το νευρωνικό δίκτυο καθώς και η εκπαίδευσή του, πραγματοποιήθηκαν με τη βοήθεια της Python 3.7.3.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι:

- cv2 4.2.0, numpy 1.16.2, json 2.0.9 για την επεξεργασία εικόνων και масκών
- keras 2.3.1, tensorflow 1.15.0 για την εκπαίδευση

Παρακάτω περιγράφεται η διαδικασία που ακολουθήθηκε και η λειτουργία των αλγορίθμων που συνοδεύουν την εργασία.

1. Συγκεντρώθηκαν τα frames της κάμερας που επιλέχθηκαν να απαρτίζουν το training set και αποθηκεύτηκαν σαν .png αρχεία.

2. Rename: Μετονομάζει τα frames σε αριθμούς, για να διευκολυνθεί η αντιστοίχιση με το label τους αργότερα.

3. Μέσω του [31], ορίστηκε σε κάθε frame μια ακολουθία σημείων που αν ενωθούν με ευθύγραμμα τμήματα, δίνουν τη γραμμή ενδιαφέροντος. Οι συντεταγμένες πάνω στην εικόνα κάθε σημείου αποθηκεύεται σε ένα .json αρχείο.

4. TakeMasks: Παίρνει σαν είσοδο τα παραπάνω json αρχεία και σχεδιάζει μια μάσκα για το καθένα, δίνοντας λευκό χρώμα στα pixels που αντιστοιχούν στη διαχωριστική γραμμή και μαύρο στο background.

5. Augment [39]: Για κάθε συνδυασμό frame/μάσκα παράγει νέους συνδυασμούς, οι οποίοι προκύπτουν από επεξεργασία του αρχικού (rotation, mirroring, lights)

6. Resize: Αλλάζει τις διαστάσεις όλων των frame και масκών ώστε να ικανοποιούν τις διαστάσεις εισόδου του νευρωνικού δικτύου.

7. DisplayFrMasks: Εμφανίζει τα frames και τις αντίστοιχες μάσκες σε μια κοινή εικόνα. Ουσιαστικά, οπτικοποιεί τη διαδικασία και διευκολύνει τον εντοπισμό σφαλμάτων.

8. Run ClassMasks: Για κάθε μάσκα, τα pixels της (λευκής) διαχωριστικής γραμμής παίρνουν την τιμή 1, ενώ του background 0, αντιστοιχίζονται δηλαδή τα διαφορετικά χρώματα της μάσκας με τον αριθμό της κλάσης (class) στην οποία ανήκουν.

9. Μερικά από τα frames και τις μάσκες τους, τα διαχωρίζουμε από το training set για να χρησιμοποιηθούν στο validation.

10. ModelTrain (keras\_segmentation): Αφού προετοιμάστηκαν τα ζεύγη frame/μάσκας, ο αλγόριθμος αυτός αναλαμβάνει την εκπαίδευση του νευρωνικού δικτύου που επιλέχθηκε.

11. Predict: Το εκπαιδευμένο νευρωνικό δίκτυο φορτώνεται και μπορεί να χρησιμοποιηθεί για να παράγει προβλέψεις (μάσκες) σε νέα frame που δεν έχει ξαναεπεξεργαστεί. Επιπλέον, πραγματοποιούνται μετρήσεις των accuracy και loss, με χρήση του validation set.



## **Κόμβοι ROS**

Η έκδοση του ROS στην οποία έγιναν τα πειράματα είναι το Melodic. Οι προσομοιώσεις, πραγματοποιήθηκαν σε Gazebo9. Οι κόμβοι που συνθέτουν το συνολικό σύστημα παρακολούθησης με ανατροφοδότηση, είναι εκτελέσιμα Python scripts.

1. MAVROS [40]: Ο κόμβος που εδραιώνει την επικοινωνία ROS και ArduPilot, αντιστοιχώντας ROS topics με μηνύματα MavLink. Έτσι, μπορούμε εύκολα να ελέγξουμε την ταχύτητα του UAV, απλά δημοσιεύοντας ROS μηνύματα στο κατάλληλο MAVROS topic.
2. Usb\_cam [42]: Δημοσιεύει τα frames που βλέπει η κάμερα, στο topic /usb\_cam/image\_raw.
3. subtry.py: Εγγράφεται στο /usb\_cam/image\_raw, το νευρωνικό δίκτυο επεξεργάζεται το frame εισόδου και δίνει σαν έξοδο τη μάσκα/πρόβλεψή του, την οποία δημοσιεύει στο topic /image\_raw.
4. tryimcon.py: Εγγράφεται στο /image\_raw και επεξεργάζεται τη μάσκα που εμφανίζεται κάθε φορά στην είσοδό του για να υπολογίσει τις μεταβλητές προς έλεγχο (angle, error). Στη συνέχεια οδηγεί τις μεταβλητές αυτές σε ένα PID ελεγκτή την καθεμία και η έξοδός τους στέλνεται στον Autopilot του UAV σαν εντολή επιθυμητής ταχύτητας. Το μήνυμα δημοσιεύεται στο topic /mavros/setpoint\_raw/local.