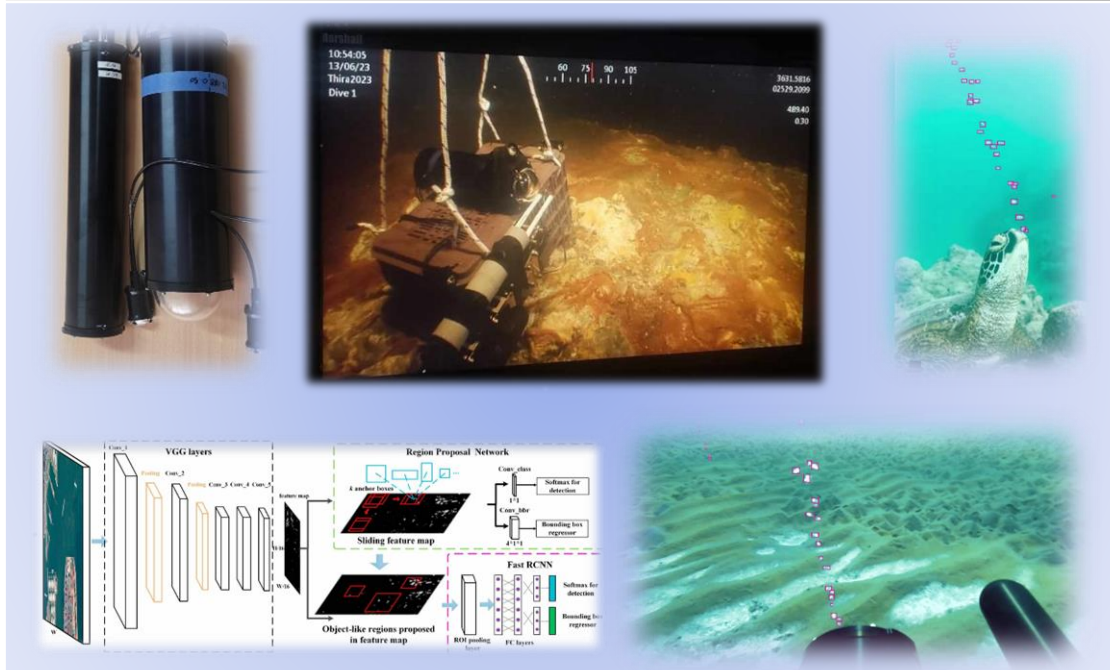




Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων και Τοπογράφων Μηχανικών –
Μηχανικών Γεωπληροφορικής
Τομέας Τοπογραφίας



Διπλωματική Εργασία

*Αυτόνομο υποθαλάσσιο σύστημα βιντεοσκόπησης και
λογισμικό για την ανίχνευση φυσαλίδων*

Επιμέλεια: Σπανός Σωτήριος (Α.Μ.: rs16013)

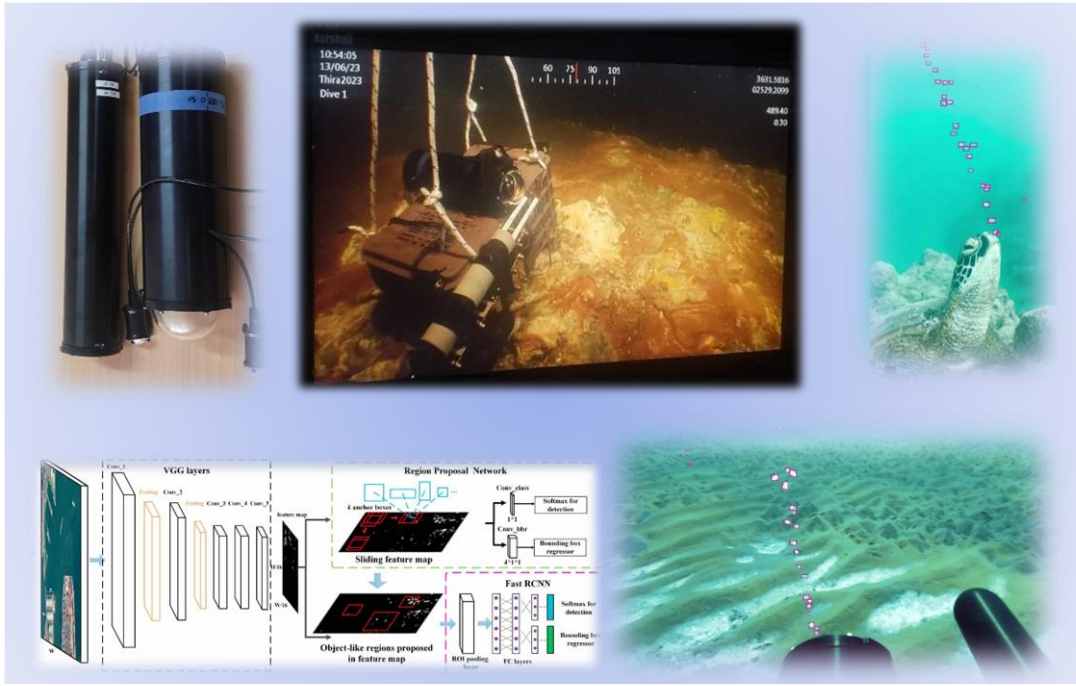
Επιβλέπων: Καράντζαλος Κωνσταντίνος

Τριμελής Επιτροπή: Καράντζαλος Κωνσταντίνος,
Καραθανάση Βασιλεία,
Ιωαννίδης Χαράλαμπος

Αθήνα, Σεπτέμβριος 2023



National Technical University of Athens
*School of Rural Surveying and Geoinformatics
Engineering
Department of Topography*



Diploma Thesis

*Autonomous Submarine Video System for Bubble
Detection.*

Author: Spanos Sotirios (U.C.: rs16013)

Supervisor: Karantzalos Konstantinos

Three Member Committee: Karantzalos Konstantinos,
Karathanassi Vasilias,
Ioannidis Charalabos

Athens, September 2023

Περίληψη

Η μελέτη υποθαλάσσιων φαινομένων αποτελεί έναν σημαντικό επιστημονικό κλάδο, με πληθώρα πρωτότυπων ερευνών. Για την παρακολούθηση των περισσότερων φαινομένων απαιτείται η υλοποίηση ενός συστήματος παρακολούθησης. Στο πλαίσιο της παρούσας ερευνητικής εργασίας υλοποιήθηκε αυτόνομο υποθαλάσσιο σύστημα με μετρητικό όργανο κάμερα για την παρακολούθηση φυσαλίδων στο πυθμένα. Η υλοποίηση ενός υποθαλάσσιου συστήματος δημιούργησε ένα μεγάλο αριθμό σύνθετων προκλήσεων στο υλισμικό και το λογισμικό σκέλος, οι οποίες αντιμετωπίστηκαν μέσα από πειραματικές διατάξεις και λαμβάνοντας υπόψιν την υπάρχουσα βιβλιογραφία.

Χάρης στο σύστημα που υλοποιήθηκε έγινε δυνατή η συλλογή ενός σετ δεδομένων που απουσιάζει από την βιβλιογραφία. Με αυτόν τον τρόπο στοχεύουμε να συμβάλουμε στην υλοποίηση μεθόδων μηχανικής μάθησης από διάφορους επιστημονικούς κλάδους, οι οποίοι δεν θα μπορούσαν να έχουν πρόσβαση σε δεδομένα in-situ. Στο πλαίσιο της παρούσας ερευνητικής εργασίας υλοποιήθηκε μοντέλο με την αρχιτεκτονική του Faster R-CNN για την ανίχνευση φυσαλίδων το οποίο έδειξε πολύ υποσχόμενα ποιοτικά και ποσοτικά στοιχεία όσον αφορά την αξιολόγηση. Συνολικά, η παρούσα εργασία αντιμετώπισε ένα ζήτημα σε όλο του το φάσμα από την κατασκευή του μετρητικού οργάνου, στην συλλογή και επεξεργασία των δεδομένων έως την υλοποίηση ενός λειτουργικού μοντέλου βαθιάς μάθησης. Μέσα από την λεπτομερή ανάλυση όλων επιμέρους σταδίων της εφαρμογής που υλοποιήσαμε, στοχεύουμε να προτρέψουμε και άλλες ερευνητικές ομάδες να κατασκευάσουν τα δικά τους ερευνητικά όργανα.

Abstract

Researching of underwater phenomena is an expanding field in science that provides the opportunities for innovative research. In order to monitor the most interesting phenomena, it is necessary to construct an observatory station. In this paper we describe the development of a Submarine Imaging System for bubble detection. The implementation of such a system presented numerous challenges, both in terms of hardware and software. We successfully addressed these challenges through a series of lab experiments and by leveraging knowledge from scientific publications.

Using the system we created, we were able to collect a dataset that is currently missing from the literature. This dataset has the potential to facilitate future machine learning application more accessible for various scientific branches that aren't able to get an in-situ dataset themselves. Using our dataset, we constructed a model using the Faster R-CNN architecture to detect bubbles. Our model exhibited promising results both in terms of evaluation metrics and qualitative assessments. In summary this paper presents a comprehensive approach to addressing an issue, starting with the construction of the imaging system followed by data collection and preprocessing, and concluding with the development of a functional deep learning model. Through a detailed analysis of each project stage, we aim to inspire more research to embark on the construction of their own scientific instruments.

Ευχαριστίες

Η παρούσα διπλωματική εργασία αποτέλεσε προϊόν συνεργασίας αρκετών ανθρώπων, χωρίς τους οποίους δεν θα μπορούσε να είχε υλοποιηθεί. Αρχικά, θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητή της διπλωματικής μου κύριο Κωσταντίνο Καραντζάλο, ο οποίος πέρα από την βοήθεια του στο πλαίσιο των μαθημάτων αλλά και στην εκπόνηση της παρούσας διπλωματικής εργασίας με στήριξε ιδιαίτερος στις προσπάθειες μου για να ενταχθώ στον κλάδο της έρευνας. Επίσης, θα ήθελα να ευχαριστήσω τον συνεπιβλέποντα της διπλωματικής μου Δρ. Βαλσάμη Ντούσκο για την συνολική του συνεισφορά στην διπλωματική μου εργασία, τόσο στο κομμάτι της υλοποίησης όσο και στο σκέλος των ιδεών και της επίλυσης προβλημάτων. Ένα μεγάλο ευχαριστώ οφείλω να δώσω και στα άτομα που με βοήθησαν με την υλοποίηση του υλιστικού σκέλους της διπλωματικής, τον Δρ. Άγγελο Μάλλιο και τον υποψήφιο Δρ. Χρήστο Αντωνίου. Δεν θα μπορούσα να παραλείψω από τις ευχαριστίες τον φίλο και συνάδελφο Δημήτρη Φλωρόπουλο ο οποίος με βοήθησε ιδιαίτερα κατά την συγκεκριμένη εργασία, καθότι μου παρείχε τον ηλεκτρονικό του υπολογιστή για την υλοποίηση του μοντέλου μου. Τέλος, θέλω να ευχαριστήσω θερμά την οικογένεια μου και την κοπέλα μου για την στήριξη τους σε όλη αυτή την προσπάθεια των τελευταίων ετών.

Πίνακας περιεχομένων

| | |
|---|-----|
| Περίληψη | 1 |
| Abstract..... | 2 |
| Ευχαριστίες | 3 |
| Κατάλογος Συνοτομογραφιών | 6 |
| Κεφάλαιο 1 | 7 |
| Εισαγωγή | 7 |
| 1.1 Εισαγωγή στο αντικείμενο μελέτης. | 7 |
| 1.2 Κύριες προκλήσεις | 8 |
| 1.3 Συνεισφορά | 10 |
| 1.4 Δομή εργασίας | 11 |
| Κεφάλαιο 2 | 11 |
| Επιστημονικό υπόβαθρο..... | 11 |
| 2.1 Βιβλιογραφία εφαρμογών υλιστικού..... | 12 |
| 2.2 Αρχές υλοποίηση μοντέλου ανίχνευσης..... | 19 |
| 2.3 Σχετικές εργασίες ανίχνευσης φυσαλίδων..... | 27 |
| Κεφάλαιο 3 | 34 |
| Ανάλυση υλισμικού και ηλεκτρονικών στοιχείων..... | 34 |
| 3.1 Εισαγωγή..... | 34 |
| 3.2 Μεμονωμένη ανάλυση τεχνικών στοιχείων..... | 36 |
| 3.3 Συνδυασμός τεχνικών στοιχείων και λογισμικού..... | 51 |
| 3.4 Πειραματικές διατάξεις, σχεδιασμός και έλεγχοι..... | 60 |
| 3.5 Διαδικασίες επί του πεδίου | 71 |
| Κεφάλαιο 4 | 75 |
| Μεθοδολογίες που εφαρμόστηκαν στα δεδομένα | 75 |
| 4.1 Εισαγωγή..... | 75 |
| 4.2 Επεξεργασία των δεδομένων | 77 |
| 4.3 Υλοποίηση μοντέλου | 84 |
| 4.4 Αποτελέσματα και αξιολόγηση | 94 |
| Κεφάλαιο 5 | 108 |
| Συμπεράσματα και μελλοντικοί στόχοι | 108 |
| Αναφορές..... | 110 |

Κατάλογος Συντομογραφιών

| | |
|------|-------------------------------|
| IoU | Intersection over Union |
| GPIO | General-Purpose Input/Output |
| SOD | Single Object Detection |
| CNN | Convolutional Neural Networks |
| RPN | Region Proposal Network |
| BB | Bounding Boxes |
| AP | Average Precision |
| mAP | Mean Average Precision |
| CF | Confusion Matrix |
| SOD | Small Object Detection |
| ΥΠ | Υποβρύχιο Περίβλημα |
| ΠΘ | Παράθυρο Θέασης |
| ROV | Remote Operated Vehicle |
| YOLO | You Only Look Once |
| ΦΠ | Φωτεινές Πηγές |
| FOV | Field of View |
| RTC | Real Time Clock |
| ΠΕ | Πηγές Ενέργειας |
| SSD | Solid State Drive |
| OS | Operating System |
| FPS | Frames Per Second |
| COCO | Common Object In Context |
| RoI | Regions of Interest |
| IS | Instance Segmentation |
| LF | Loss function |
| TS | Test Set |

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή στο αντικείμενο μελέτης

Τα τελευταία χρόνια η εξερεύνηση και αξιολόγηση υποθαλάσσιων φαινομένων αποτελεί ένα αναπτυσσόμενο κλάδο της έρευνας (όπως υποδεικνύουν και οι μελέτες που παρατίθενται στην παρούσα εργασία ([1,2,3,7,14,19])). Ειδικά η Ελλάδα όντας μια χώρα με μεγάλο τμήμα της επικράτειας της να είναι παραθαλάσσιο, οφείλει να πρωτοστατήσει σε αυτή την τάση. Εξαιτίας αυτού του φαινομένου και λόγω της ιδιαίτερης προσωπικής αγάπης για την θάλασσα επιλέχθηκε το θέμα της συγκεκριμένης εργασίας να αφορά της υλοποίηση συστήματος για την παρακολούθηση υποθαλάσσιων φαινομένων και της ανάλυσης τους.

Τα φαινόμενα που βρίσκονται υπό το πρίσμα της συγκεκριμένης εργασίας κρίθηκε ότι θα ήταν βέλτιστο να αξιολογηθούν και να μελετηθούν με οπτικά μέσα, πιο συγκεκριμένα με κάμερες (λόγω και του αντικειμένου της σχολής μας). Τα κυριότερα αντικείμενα που παρουσίασαν ενδιαφέρον ήταν η χαρτογράφηση, η παρακολούθηση υποθαλάσσιων αερίων με την μορφή φυσαλίδων (μαγματικά αέρια, αέρια που σχετίζονται με την ραδιενέργεια, άνθρακας, CO₂ κ.α), ραδιομετρικές διορθώσεις των εικόνων κ.α. Στην συγκεκριμένη εργασία το φαινόμενο που επιλέχθηκε, λόγω του μεγάλου ενδιαφέροντος που παρουσιάζει αλλά και του εκτενούς εύρους εφαρμογών του, είναι η ανίχνευση υποθαλάσσιων αερίων (φυσαλίδων) μιας και μπορεί συνεισφέρει σε πληθώρα επιστημονικών κλάδων (γεωλογία, βιολογία, φυσική). Γενικά, ιδιαίτερο ενδιαφέρον μου προκάλεσε η ιδέα να μελετηθεί ένα αντικείμενο που να έχει χρηστικό χαρακτήρα και να είναι όσο το δυνατόν πιο ευρύ. Για αυτό και προέκυψε σαν ιδέα η υλοποίηση ενός μετρητικού οργάνου το οποίο πέρα από τα δεδομένα και τις ανάγκες της παρούσας εργασίας θα συνεχίσει να συνεισφέρει και να είναι λειτουργικό για άλλους μελετητές μελλοντικά. Επιπλέον, ακολουθώντας την ίδια νοοτροπία επιλέχθηκε να κατασκευαστεί ένα πρωτοποριακό σετ δεδομένων φυσαλίδων το οποίο δεν υπάρχει μέχρι στιγμής στην υπάρχουσα βιβλιογραφία, καθώς καμία ερευνητική ομάδα που μελέτησε εφάμιλλο ζήτημα δεν παρέχει τα δεδομένα που συνέλεξε. Πέρα από αυτό, θελήσαμε να δημιουργήσουμε και ένα «εργαλείο» για να αξιολογούνται οι φυσαλίδες του σετ δεδομένων μας. Για αυτό το σκοπό επιλέξαμε να χρησιμοποιήσουμε μια αρχιτεκτονική νευρωνικού δικτύου ώστε να επιτύχουμε την ανίχνευση τους, εκπαιδύοντας ένα μοντέλο. Συνολικά, η συγκεκριμένη διπλωματική

εργασία βασίζεται στην μελέτη θαλάσσιων φαινομένων και προσπαθεί να συνεισφέρει με τον τρόπο της ως προς την μελέτη κάποιων εξ αυτών.

1.2 Κύριες προκλήσεις

Για να φέρουμε σε πέρας τους στόχους που θέσαμε, αντιμετωπίσαμε τρία βασικά ζητήματα, το υλισμικό, την επεξεργασία και ανάλυση των δεδομένων και τις ερευνητικές αποστολές (πεδίο), ενώ εξαιτίας αυτών προέκυψε μεγάλος αριθμός προκλήσεων. Αρχικά, όσον αφορά το πεδίο, ήρθαμε αντιμέτωποι με την δυσκολία προσβασιμότητας στην περιοχή μελέτης, διότι οι αποστολές ήταν σε τοποθεσίες υποθαλάσσιες και δυσπρόσιτες. Επίσης, για να εξαχθούν τα ζητούμενα δεδομένα, ώστε να μελετηθούν ενδιαφέροντα φαινόμενα (όπως τα μαγματικά αέρια), έπρεπε να γίνει επιτόπια έρευνα σε συγκεκριμένες περιοχές (όπως η Μήλος, η Σαντορίνη (Κολούμπο)), μόνο στις οποίες εμφανίζεται το φαινόμενο προς παρακολούθηση. Αυτό οδήγησε στην ανάγκη μεγάλων μετακινήσεων και σε κάποιες περιπτώσεις χρήσης και ερευνητικών σκαφών. Εκτός αυτού η εγκατάσταση του συστήματος απαιτούσε την χρήση ROV το οποίο δυσχέραινε ακόμα περισσότερο τις εργασίες πεδίου καθώς υπήρχε ανάγκη εύρεσης χειριστών. Εξαιτίας αυτού ο αριθμός των μελών της αποστολής που θα έπρεπε να συνεργαστούν για την αποπεράτωση των in-situ μετρήσεων αυξήθηκε, με αποτέλεσμα οποιοδήποτε λάθος στην επικοινωνία μεταξύ τους να οδηγούσε σε ζημίες στο υλισμικό και πιθανώς στην αδυναμία λήψης δεδομένων. Συνολικά οι εργασίες πεδίου αποτέλεσαν ένα σχετικά μικρό σκέλος της παρούσας εργασίας (και χρονικά αλλά και σαν τμήμα του κειμένου στο παρόν τεύχος) αλλά αρκετά σύνθετο, αφού όπως σχολιάστηκε υπήρχαν πολλές προκλήσεις που έπρεπε να αντιμετωπιστούν.

Όσον αφορά το υλισμικό κομμάτι οι προκλήσεις προέκυψαν πριν ακόμα ξεκινήσει η υλοποίηση καθώς δεν ήταν εφικτό να πραγματοποιηθούν πειράματα σε περιβάλλον που θα προσομοίαζε τις συνθήκες των αποστολών. Αυτό θα οδηγούσε σε έλλειψη πληροφορίας όσον αφορά τις δυνατότητες αλλά και τα ζητήματα του υλοποιημένου συστήματος. Προφανώς θα μπορούσαμε για λίγα πειράματα να μεταφερθούμε σε κάποια τοποθεσία παραθαλάσσια, παρόλα αυτά, όπως θα φανεί στην συνέχεια, θέλαμε να κάνουμε εκτεταμένα πειράματα. Έτσι, το κομμάτι της προσομοίωσης των αποστολών αποτέλεσε ζήτημα καθώς η εύρεση χώρου που θα μπορούσαν να τοποθετηθούν οι εξοπλισμοί σε περιβάλλον παρόμοιο με τις πραγματικές συνθήκες, ειδικά για μεγάλα χρονικά διαστήματα, θα ήταν αδύνατη αν δεν παρεχόταν από το εργαστήριο πησίνα μεγάλων διαστάσεων. Ακόμα και με την χρήση των

συγκεκριμένων υποδομών δεν κατέστη δυνατόν να προσομοιαστούν κάποια ζητήματα που παρουσιάστηκαν στην αποστολή, όπως ήταν το όξινο περιβάλλον εξαιτίας του οποίου προκλήθηκε μερική διάβρωση σε κάποια τμήματα του συστήματος, αλλά και η αλατότητα. Αφού βρέθηκαν οι κατά το δυνατόν βέλτιστες λύσεις σε αυτά τα ζητήματα ξεκίνησε η κατασκευή του συστήματος. Οι βασικότερες προκλήσεις σε αυτό το στάδιο χωρίστηκαν σε δύο βασικά σκέλη της αυτονομίας και της εξισορρόπησης δυνατοτήτων του συστήματος σε σχέση με το κόστος του (τόσο οικονομικό όσο και ενεργειακό). Όσον αφορούσε την αυτονομία, έπρεπε το σύστημα να έχει επαρκή ενέργεια αλλά και αποθηκευτικό χώρο (μνήμη) ώστε να συλλέξει τον μεγαλύτερο δυνατό όγκο δεδομένων για ευρεία χρονικά διάστημα μιας και με αυτή την μεθοδολογία θα μπορούσαν να παρατηρηθούν στοιχεία όπως η περιοδικότητά των φαινομένων, οι αλλαγές που προκύπτουν από άλλους παράγοντες (όπως οι σεισμοί) κ.α. Η δυσκολία στην βελτιστοποίηση της αυτονομίας οφείλεται κυρίως στον χωρικό περιορισμό του συστήματος μιας και για να αγοραστούν μεγαλύτερα προστατευτικά περιβλήματα το κόστος αυξανόταν εκθετικά. Αφού αξιολογήθηκαν αυτά τα ζητήματα, έπρεπε να ληφθούν σημαντικές αποφάσεις που θα αφορούσαν την εξισορρόπηση της υπολογιστικής ισχύος και της φωτεινότητας που παρείχαν τα φώτα που τοποθετήθηκαν, σε σχέση με το ενεργειακό αλλά και με το οικονομικό κόστος. Τέλος, οφείλαμε να επιλύσουμε το σύνθετο σκέλος της τηλεμετρίας, ώστε να μπορούμε να συλλέξουμε τα δεδομένα μας αλλά και να “επικοινωνήσουμε” με το σύστημα χωρίς την χρήση καλωδίου και λαμβάνοντας υπόψιν ότι το σήμα απορροφάται με γρήγορους ρυθμούς όταν το μέσο διάδοσης είναι το νερό.

Έχοντας ολοκληρώσει τα ζητήματα των αποστολών και του υλιστικού, το οποίο συνεπάγεται ότι έχουμε την δυνατότητα να εξάγουμε τα ζητούμενα δεδομένα, τέθηκαν ζητήματα που αφορούσαν την διαχείριση τους. Αρχικά, έπρεπε να αποφασιστεί η μεθοδολογία για την αποθήκευση των δεδομένων κατά την λήψη τους από την κάμερα (επιλογή format), ώστε να εξάγουμε την βέλτιστη δυνατή πληροφορία που θα καταλάμβανε τον λιγότερο δυνατό αποθηκευτικό χώρο. Στην συνέχεια παρουσιάστηκαν ζητήματα που αφορούσαν την επιλογή της διαμόρφωσης των δεδομένων ώστε να είναι αξιοποιήσιμα από το νευρωνικό δίκτυο το οποίο θα επιλεγόταν. Η επιλογή της αρχιτεκτονικής του νευρωνικού δικτύου αποτέλεσε ένα από τα σημαντικότερα ζητήματα που έπρεπε να αντιμετωπιστούν καθώς τα δεδομένα είχαν αρκετές ιδιαιτερότητες, όπως το μικρό τους μέγεθος, το μη σταθερό σχήμα, τις διαφορές που παρουσίαζαν ραδιομετρικά μεταξύ τους (ανάλογα την γωνία λήψης) κ.α. Η τελευταία πρόκληση που αντιμετωπίστηκε ήταν η επιλογή των δεδομένων που θα χρησιμοποιούνταν κατά την εκπαίδευση του τελικού μοντέλου, καθώς από τις αποστολές που έγιναν συλλέχθηκε μεγάλος αριθμός δειγμάτων τα οποία θα έπρεπε να

εξεταστούν υπό το πρίσμα κάποιων κριτηρίων μιας και δεν ήταν δυνατή η ετικετοποίηση του συνόλου τους. Συνολικά, παρουσιάστηκαν αρκετές προκλήσεις οι οποίες όμως αντιμετωπίστηκαν στο πλαίσιο της συγκεκριμένης εργασίας, οδηγώντας μας στην «εξερεύνηση» εναλλακτικών λύσεων.

1.3 Συνεισφορά

Στόχος της συγκεκριμένης εργασίας ήταν να συνεισφέρει σε μεγάλο εύρος επιστημών. Αρχικά, ο πρώτος τομέας συνεισφοράς ήταν αυτός της υλοποίησης ενός αυτόνομου συστήματος παραγωγής δεδομένων (εικόνων) ικανοποιητικής ανάλυσης για μεγάλο εύρος βάθους (μηδέν έως περισσότερα των πεντακοσίων μέτρων κάτω από τη στάθμη της θάλασσας). Λόγω της αναλυτικής περιγραφής που γίνεται στο τρίτο κεφάλαιο αρκετές ερευνητικές ομάδες θα μπορούσαν να αναπαράγουν το σύστημα μας, το οποίο παρουσιάζει αρκετά μικρό κόστος σε σχέση με τις προδιαγραφές του. Εκτός αυτού παρέχουμε την δυνατότητα με μια απλή αφαίρεση του μετρητικού μας οργάνου το σύστημα μας να αποτελέσει βάση για τοποθέτηση άλλων αισθητήρων αφού η εταιρία παραγωγής του μικροϋπολογιστή που επιλέχθηκε, παρέχει μεγάλο εύρος επιπρόσθετων τμημάτων. Συνολικά, λόγω του τρόπου υλοποίησης του συστήματος μπορούν να γίνουν πολλές αλλαγές ώστε να ικανοποιήσουν τις ανάγκες διάφορων επιστημονικών κλάδων. Πέραν από το κατασκευαστικό κομμάτι στο πλαίσιο της συγκεκριμένης εργασίας υλοποιήθηκε σετ δεδομένων με φυσαλίδες μέσα από το οποίο παρέχονται οι απαραίτητες ετικέτες ώστε να χρησιμοποιηθεί για την εκπαίδευση μοντέλων διαφόρων αρχιτεκτονικών (νευρωνικών δικτύων). Αυτή η προσφορά δεν υφίσταται στην υπάρχουσα βιβλιογραφία. Εκτός αυτού, η συγκεκριμένη εργασία συνεισφέρει με ένα μοντέλο για ανίχνευση φυσαλίδων το οποίο παρουσιάζει πολλά υποσχόμενα στοιχεία όσον αφορά την ποσοτική (μετρικές) και ποιοτική αξιολόγηση του. Η αξία του συγκεκριμένου μοντέλου είναι μεγάλη καθώς λαμβάνοντας υπόψιν τα αντικείμενα που ανιχνεύει μπορεί να γίνει αξιολόγηση μεγάλου εύρους φαινομένων (πέραν των μαγματικών αερίων που αποτέλεσε στόχο της εργασίας).

Η συγκεκριμένη εργασία αποτέλεσε ένα αρκετά μεγάλο έργο με συνεργασία αρκετών ατόμων. Σε αυτή την παράγραφο θα γίνει επακριβής ανάλυση της προσωπικής μου συνεισφοράς. Όσον αφορά το υλιστικό είχα μερική συνεισφορά σε όλες τις επιλογές των επιμέρους τμημάτων αλλά και στην συνδεσμολογία. Πλήρης προσωπική υλοποίηση αποτέλεσαν οι πειραματικές διατάξεις που παρουσιάζονται στο Κεφάλαιο 3, καθώς και το λογισμικό που υλοποιήθηκε για τις βασικές λειτουργίες του

συστήματος. Τέλος, όλες οι μεθοδολογίες που εφαρμόστηκαν στα δεδομένα (Κεφάλαιο 4) αποτέλεσαν επίσης πλήρως δικιά μου συνεισφορά.

1.4 Δομή εργασίας

Στο κεφάλαιο 2 που ακολουθεί παρουσιάζεται το επιστημονικό υπόβαθρο στο οποίο βασίστηκε τόσο η υλοποίηση του υλιστικού όσο και η εκπαίδευση του τελικού μοντέλου. Στόχος ήταν το μέγεθος του να μην είναι πολύ εκτενές αλλά παρόλα αυτά να παρέχει στον αναγνώστη όλη την απαραίτητη πληροφορία για να κατανοήσει επί του συνόλου την αξία της παρούσας εργασίας. Στα κεφάλαια 3 και 4 που ακολουθούν αντιμετωπίζονται οι προκλήσεις που αναφέρθηκαν προηγουμένως. Πιο συγκεκριμένα, στο κεφάλαιο 3 παρουσιάζονται αναλυτικά όλα τα στοιχεία του υλιστικού και παρέχεται όλη η απαραίτητη πληροφορία για την υλοποίηση του συστήματος μας, ενώ στο τέλος του γίνεται αναφορά στις αποστολές που επιτελέστηκαν για την συλλογή δεδομένων. Όσον αφορά το κεφάλαιο 4, αναλύεται η μεθοδολογία σύνθεσης του σετ δεδομένων που αποτέλεσε μια από της σημαντικές συνεισφορές της συγκεκριμένης εργασίας. Εκτός αυτού, παρουσιάζεται η μεθοδολογία εκπαίδευσης και αξιολόγησης του μοντέλου που υλοποιήθηκε για την ανίχνευση των φυσαλίδων. Τέλος, στο κεφάλαιο 5 εξάγονται συμπεράσματα και αναλύονται οι μελλοντικοί μας στόχοι.

Κεφάλαιο 2

Επιστημονικό υπόβαθρο

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση του επιστημονικού υποβάθρου στο οποίο βασίστηκε η παρούσα εργασία. Το κεφάλαιο θα χωριστεί σε τρία βασικά τμήματα. Στο πρώτο θα αναλυθούν ερευνητικές εργασίες που παρείχαν τα εφόδια για την υλοποίηση του συστήματος παρακολούθησης, ενώ στο δεύτερο θα γίνει παρουσίαση για την μηχανική μάθηση, την βαθιά μάθηση και θα παρατεθούν δύο αρχιτεκτονικές οι οποίες αποτέλεσαν τις επικρατέστερες για να λύσουν το πρόβλημα της ανίχνευσης που αντιμετωπίστηκε. Όσον αφορά το τρίτο τμήμα, θα αναλυθούν ερευνητικές εργασίες που προσπάθησαν να επιλύσουν το ίδιο ζήτημα και θα σχολιαστούν. Στόχος είναι με αυτό το κεφάλαιο ο αναγνώστης να έρθει σε επαφή με

το αντικείμενο της εργασίας και να γίνει επίλυση πιθανών αποριών οι οποίες αφορούν τα βασικά στοιχεία της διαδικασίας.

2.1 Βιβλιογραφία εφαρμογών υλιστικού

Ο βασικότερος στόχος της παρούσας εργασίας είναι η υλοποίηση ενός αυτόνομου συστήματος για την εξαγωγή δεδομένων. Μια κατασκευή σαν αυτή μπορεί να αποτελέσει αρκετά σύνθετη εργασία καθώς περιλαμβάνει πολύ μεγάλο αριθμό αποφάσεων αλλά και γνώσεων τεχνικής φύσεως. Χωρίς, όμως την συγκεκριμένη υλοποίηση δεν θα μπορούσαν να επιτευχθούν οι υπόλοιποι στόχοι της συγκεκριμένης εργασίας, για αυτό χρειάστηκε να γίνει μελέτη της υπάρχουσας βιβλιογραφίας ώστε να βρεθεί ένα σύστημα με τις απαιτούμενες προδιαγραφές. Έτσι, λαμβάνοντας υπόψιν την μεθοδολογία που ακολουθήθηκε σε εφάμιλλές περιπτώσεις και εφαρμόζοντας τις απαραίτητες προσαρμογές και βελτιώσεις κατέστη δυνατή η υλοποίηση του δικού μας αυτόνομου συστήματος. Για την επίτευξη του συστήματος μας μεγάλη συνεισφορά παρείχαν οι κατασκευάστριες εταιρίες των επιμέρους τμημάτων (είτε μέσω επικοινωνίας είτε με χρήση των εγχειριδίων που παρείχαν) αλλά και άλλοι συνεργάτες.

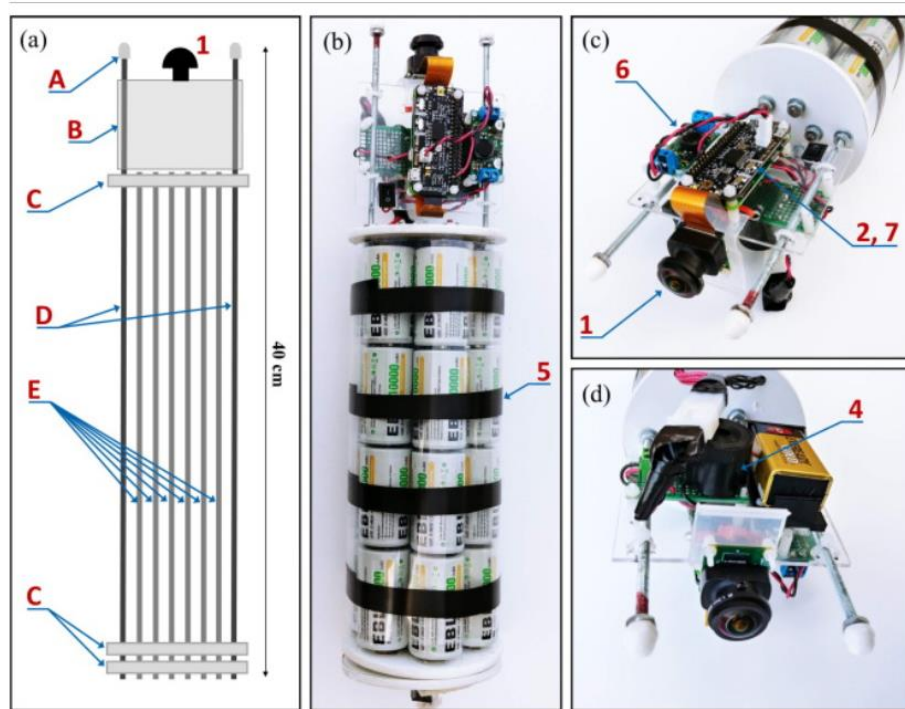
Η πρώτη ερευνητική εργασία η οποία αποτέλεσε θεωρητική βάση για το σύστημα που υλοποιήσαμε ήταν η [1]. Σε αυτή την εργασία η προβλεπόμενη εφαρμογή του συστήματος ήταν εφάμιλλη με αυτή που θέλαμε να επιτύχουμε και η κατασκευή του έγινε με αρκετά μικρό κόστος. Αυτό μας ενθάρρυνε καθώς έγινε εμφανές ότι η υλοποίηση ενός συστήματος με τις προδιαγραφές που αποζητούσαμε ήταν δυνατή και μπορούσε να επιτευχθεί χωρίς υπέρογκα κόστη. Μεγάλο βάρος της συγκεκριμένης εργασίας δόθηκε στο τεχνικό κομμάτι, δηλαδή την υλοποίηση του υλιστικού αλλά και στον τρόπο με τον οποίο το κόστος θα ελαχιστοποιούνταν. Στο Σχήμα 2.1 που ακολουθεί φαίνονται οι βασικές αρχές της συγκεκριμένης εργασίας. Αρχικά, η υλοποίηση έγινε με εύκολα ως προς την εύρεση τους υλικά ενώ η κατασκευή δεν ήταν σύνθετη και περιεγράφηκε αναλυτικά. Το κόστος ήταν μικρότερο των 500 δολαρίων και παρόλα αυτά η αυτονομία του συστήματος ήταν έως και 212 ώρες βίντεο. Το σύστημα που κατασκεύασαν (αριστερό τμήμα του Σχήματος 2.1) αποτελούνταν από το μετρητικό όργανο (1) την κάμερα Raspberry Pi Camera v2 με φακό 110 μοιρών η οποία ελεγχόταν με Python μέσα από την βιβλιοθήκη picamera. Ο μικροϋπολογιστής που επέλεξαν ήταν ο Raspberry Pi Zero W (2) κυρίως λόγω της μικρής κατανάλωσης ενέργειας, του λειτουργικού και της παροχής ενσωματωμένων wifi και Bluetooth. Η microSD κάρτα του συστήματος ήταν 200 GB (3) στην οποία γινόταν η αποθήκευση των βίντεο που

συνέλεξαν. Επίσης, τοποθέτησαν ένα κύκλωμα (4) ώστε να επιτευχθεί συγχρονισμός με ακουστικά δεδομένα που στόχευαν να συλλέξουν (μέσα από GPIO pin του Raspberry). Το επόμενο στοιχείο του συστήματος τους ήταν η πηγή ενέργειας όπου τοποθέτησαν 4 πυρήνες μπαταρίας (5) στην σειρά και στην συνέχεια 7 τέτοιες συστοιχίες παράλληλα. Οι μπαταρίες ήταν D-Shell με συνολική χωρητικότητα 10.000 mAh. Οι μπαταρίες συνδέονταν με ένα ρυθμιστή τάσης (6) ο οποίος παρείχε την απαιτούμενη τάση στο Witty Pi mini (7), το οποίο με την σειρά του ενεργοποιούσε και απενεργοποιούσε το Raspberry βάση του προγράμματος που έθετε ο προγραμματιστής, ενώ ταυτόχρονα έκανε και την διαχείριση της ενέργειας.

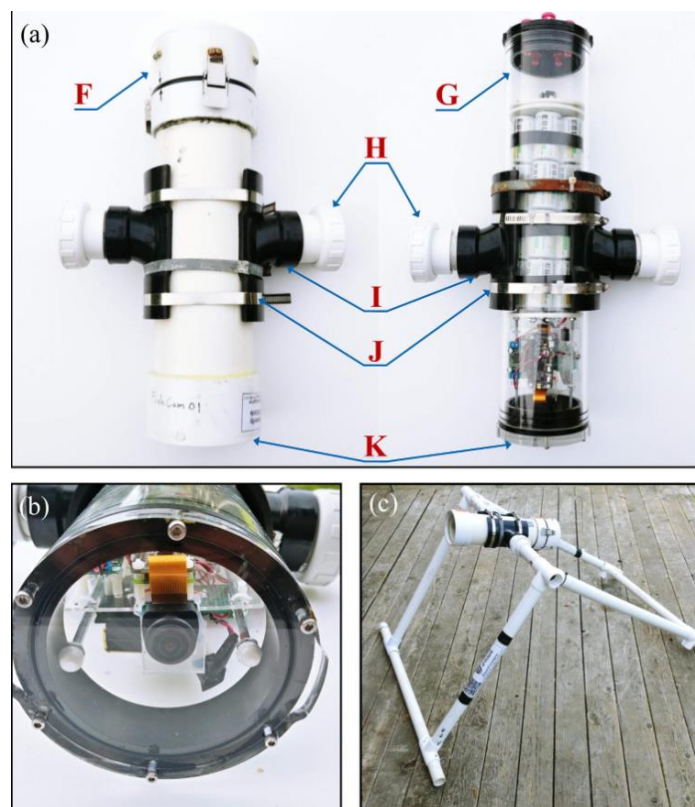


Σχήμα 2.1: Βασικά στοιχεία ερευνητικής εργασίας

Μεγάλο ενδιαφέρον παρουσίασε και το μηχανολογικό σχέδιο, βάση του οποίου «στήθηκαν» όλα τα επιμέρους τμήματα εντός του περιβλήματος. Εκτός αυτού παρουσιάστηκε η μεθοδολογία, βάσει της οποίας υλοποιήθηκε το προστατευτικό περίβλημα (Σχήματα 2.1, 2.2, 2.3). Το υλικό κατασκευής του ήταν κυρίως από ράβδους αλουμίνιου (Σχήμα 2.2 D,E) οι οποίες διαχωρίζονταν από τρεις δίσκους (C) καθώς και από ένα ακρυλικό επίπεδο (B). Οι μικρές ράβδοι (E) αποτέλεσαν την στήριξη των μπαταριών ενώ οι δύο μεγάλες έφταναν μέχρι και τον φακό (1) για να παρέχουν στήριξη στο ακρυλικό φύλλο (B) αλλά και στο βάρος της κάμερας. Στην κορυφή τους τοποθετήθηκε λάστιχο για να μην χαραχθεί το παράθυρο θέασης της κάμερας. Πάνω στο ακρυλικό φύλλο (B) τοποθετήθηκαν τα ηλεκτρονικά συστήματα. Οι δίσκοι (C) είχαν κονέκτορες και καλώδια σε κάθε πλευρά τους για να συνδέουν παράλληλα τις μπαταρίες. Όλο το εσωτερικό σύστημα που φαίνεται στο Σχήμα 2.2 τοποθετήθηκε εντός του προστατευτικού περιβλήματος για να διασφαλιστεί η στεγανότητα του. Σε αυτό το σημείο οι συγγραφείς υλοποίησαν δύο διαφορετικά προστατευτικά περιβλήματα. Ένα από αυτά το κατασκεύασαν με απλά υλικά ενώ το άλλο το αγόρασαν από κατασκευαστή (Σχήμα 2.3). Η αιτία που έγινε αυτό είναι για να δώσουν την δυνατότητα μείωσης τους κόστους.



Σχήμα 2.2: Εσωτερική κατασκευή. Πηγή: [1]



Σχήμα 2.3: Προστατευτικό περίβλημα. Πηγή: [1]

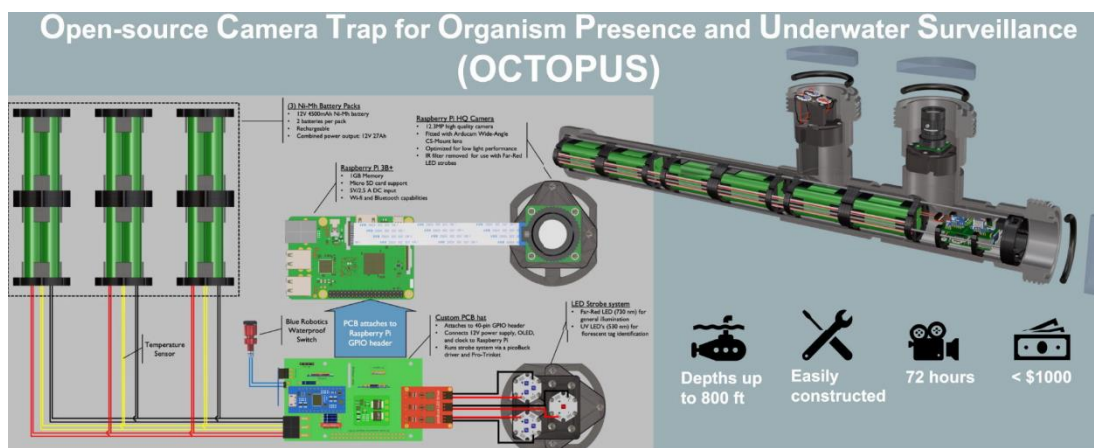
Όσον αφορά το λογισμικό, το σύστημα τους υλοποιήθηκε με τα ελάχιστα δυνατά προγράμματα ώστε να απελευθερωθεί όσο το δυνατόν περισσότερη μνήμη στη micro SD. Εκτός αυτού λειτουργήσαν το σύστημα τους headless, δηλαδή χωρίς

γραφικό περιβάλλον για να μειώσουν την απαιτούμενη υπολογιστική ισχύ. Ο τρόπος που συνδεόντουσαν σε αυτό ήταν χρησιμοποιώντας το πρωτόκολλο SSH. Ο χρόνος λειτουργίας και οι κύκλοι του συστήματος, δηλαδή πόσο παρέμενε σε λειτουργία και κάθε πότε απενεργοποιούνταν, ρυθμιζόντουσαν μέσα από το Witty. Σημαντική λεπτομέρεια είναι ότι το σύστημα τους λειτουργούσε μόνο την μέρα με φυσικό φωτισμό (σε σχετικά ρηγά νερά, 8-12 μέτρα). Στην συνέχεια, έγινε λεπτομερής παρουσίαση για την υλοποίηση του υλισμικού και δίνονται οι απαραίτητες οδηγίες για την αναπαραγωγή του συστήματος. Ακολούθως, σχολιάστηκαν οι περιορισμοί του συστήματος που υλοποίησαν. Αυτοί οφείλονταν κυρίως στην έλλειψη μνήμης χωρίς να προέκυψε ενεργειακή έλλειψη, αφού σε όλες τις αποστολές που επιχείρησαν η κάρτα μνήμης δεν είχε άλλο αποθηκευτικό χώρο. Ταυτόχρονα τόνισαν ότι το σύστημα τους είναι εύκολα προσαρμόσιμο στις ανάγκες άλλων ομάδων μιας και το Raspberry που τοποθέτησαν έχει πολλές δυνατότητες για αλληλεπίδραση. Κλείνοντας, συμπλήρωσαν ότι τέτοιου τύπου κάμερες μπορούν να έχουν μεγάλο εύρος εφαρμογών όπως είναι η παρακολούθηση της συμπεριφοράς των ψαριών, ο έλεγχος της βιοποικιλότητας τους αλλά και γενικότερα η παρακολούθηση διαφόρων φαινομένων.

Από την συγκεκριμένη ερευνητική εργασία λάβαμε πολύτιμες πληροφορίες τόσο για τις βασικές ανάγκες υλοποίησης ενός αυτόνομου συστήματος όσο και για τις εταιρίες που παράγουν προϊόντα που θα μπορούσαμε και εμείς να απευθυνθούμε. Ενδεικτικά παραδείγματα είναι ο μικροϋπολογιστής και το προστατευτικό υποβρύχιο περίβλημα. Εκτός αυτού αντλήσαμε ιδέες για την μεθοδολογία επικοινωνίας με το σύστημα μας όσο αυτό βρισκόταν εντός της θάλασσα αλλά και χωρικά για τον τρόπο που θα μπορούσαμε να τοποθετήσουμε τα ηλεκτρονικά μέρη του συστήματος μας εντός του προστατευτικού περιβλήματος. Συνολικά λάβαμε αρκετά ενδιαφέροντα στοιχεία που συνείσφεραν ιδιαίτερος στην προσπάθεια μας.

Μια ακόμα ερευνητική εργασία που μελετήσαμε πριν ξεκινήσουμε την διαδικασία της υλοποίησης ήταν η [2], η οποία επίσης αποσκοπούσε στην κατασκευή ενός συστήματος υποθαλάσσιας κάμερας για παρακολούθηση και εύρεση οργανισμών (χρησιμοποιώντας ένα σύστημα που ενεργοποιεί την κάμερα όταν ανιχνεύεται κίνηση στον περιβάλλοντα χώρο). Ένα πρόβλημα που προσπάθησε να αντιμετωπίσει και η δικιά τους ομάδα ήταν η έλλειψη (στην υπάρχουσα βιβλιογραφία) συστημάτων παρακολούθησης με μεγάλη αυτονομία, γεγονός που περιορίζει τις δυνατότητες μελέτης πολλών υποθαλάσσιων φαινομένων. Παρατηρώντας το Σχήμα 2.4 φαίνονται τα βασικότερα χαρακτηριστικά του συστήματος και αρχικά το μεγάλο βάθος που μπορούσε να αντέξει το προστατευτικό περίβλημα (περίπου 250 μέτρα), το οποίο όμως δεν ήταν αρκετό για τις δικές μας εφαρμογές. Εκτός αυτού το κόστος για την

υλοποίηση του ήταν μικρό (<1000\$) και η αυτονομία του σχετικά μεγάλη (72 ώρες). Επίσης, χρησιμοποίησαν LED φωτισμό, (μιας και το σύστημα τους τοποθετείται σε μεγάλα βάθη) το φως του οποίου έχει ονομαστικές τιμές 730 και 530 nm στο φάσμα. Το συγκεκριμένο σύστημα εκμεταλλεύτηκε τις δυνατότητες του μικροϋπολογιστή Raspberry Pi 3B+ (μεγαλύτερη υπολογιστική ισχύ αλλά και ενεργειακά πιο κοστοβόρο σε σχέση με την προηγούμενη ερευνητική εργασία) ο οποίος συνδέθηκε σε προσαρμοσμένη πλακέτα PCB. Ενδιαφέρον στην συγκεκριμένη εργασία παρουσίασαν τα ηλεκτρονικά συστήματα που τοποθέτησαν όπως το RTC, ο μετρητής τάσης και ρεύματος (INA 219), ο διακόπτης (power switch), ο μικροελεγκτής (Adafruit Pro Trinket) αλλά και ο μετρητής θερμότητας. Οι συγγραφείς παρείχαν εξαιρετική ανάλυση της τοποθέτησης όλων των επιμέρους τμημάτων στην πλακέτα PCB (κάτι που εμείς δεν υλοποιήσαμε κατά αυτόν τον τρόπο). Το σύστημα που υλοποίησαν είχε διπλή συνεισφορά ως προς εμάς, αρχικά ανέδειξε ότι η ανάγκη που στοχεύουμε να καλύψουμε είναι υπαρκτή και ότι ακόμα πολλές ομάδες βρίσκονται σε διαδικασία αντιμετώπισης της. Επίσης, μας παρείχε ιδέες όσον αφορά το ηλεκτρολογικό κομμάτι και τους αισθητήρες που θα έπρεπε να τοποθετηθούν για να εξασφαλισθεί η λειτουργική ζωή του συστήματος, αλλά και κάποια στοιχεία στην συνδεσμολογία.



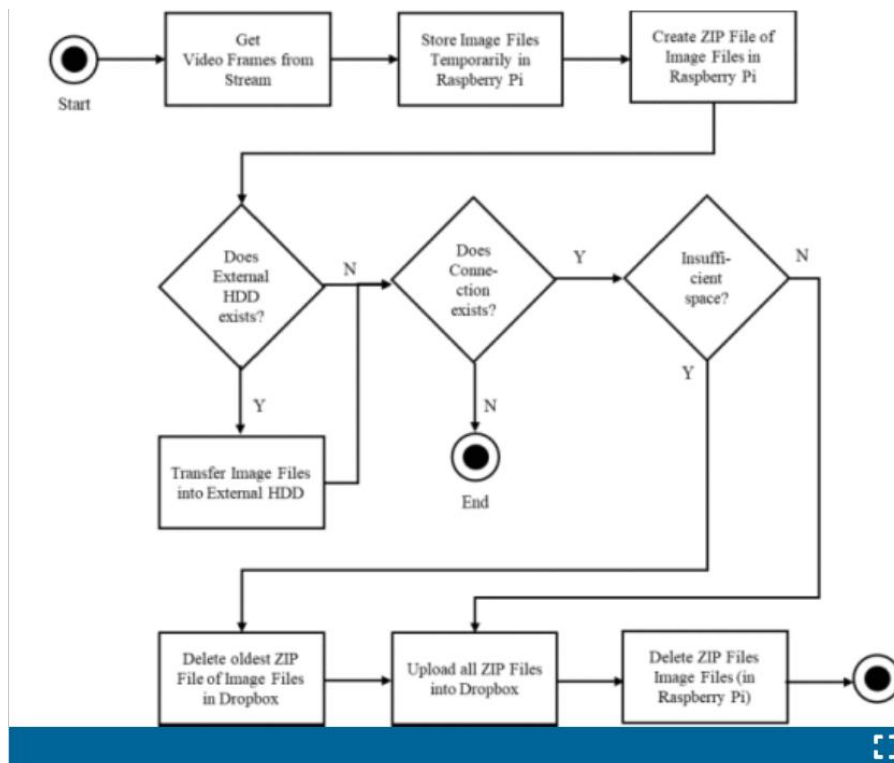
Σχήμα 2.4: Γενικά χαρακτηριστικά του συστήματος. Πηγή: [2]

Ένα ακόμα αντίστοιχο σύστημα που αξιολογήθηκε ήταν στην εργασία [3], η οποία παρουσίασε διαφορές σε σχέση με τις υπόλοιπες. Αρχικά, η μελέτη έγινε σε γλυκό νερό και όχι υποθαλάσσια και εκτός αυτού η συγκεκριμένη ομάδα προσπάθησε να μειώσει το μεγαλύτερο περιορισμό που υπάρχει σε τέτοια συστήματα (κυρίως στα ρηγά νερά αφού συνήθως δεν υπάρχει η ανάγκη φωτισμού) που είναι η έλλειψη χώρου αποθήκευσης δεδομένων ενώ ταυτόχρονα αυτοματοποίησε και την εξαγωγή των δεδομένων (ζητούμενο που μας αφορά και αναλύεται και στην [1]). Ο κύριος τομέας ενδιαφέροντος της συγκεκριμένης εργασίας πέρα από το υλισμικό είναι η παρακολούθηση ιχθυοκαλλιέργειών γλυκού νερού. Για να επιτευχθεί αυτό όμως ήταν αναγκαίο να προστατευθούν τα ψάρια και οι φυσικοί πόροι μέσα από κάποια

πρωτόκολλα. Για την δημιουργία των πρωτοκόλλων απαιτούνταν πληθώρα δεδομένων της συμπεριφοράς της τοπικής πανίδας και σε αυτό το σημείο εισάγεται η ανάγκη για την κατασκευή του συστήματος παρακολούθησης. Το σύστημα τους βασίστηκε και αυτό στον μικροϋπολογιστή Raspberry Pi Model 4B2 Gb RAM καθώς και την κάμερα Arducam IMX219 Wide Angle Camera Drop-in (με φακό 145° οριζόντια με 77° κάθετα) η οποία λειτουργεί πλήρως σε συνεργασία με τον μικροϋπολογιστή. Τα συγκεκριμένα στοιχεία τοποθετήθηκαν πάνω σε βάση που υλοποιήθηκε μέσω της τεχνικής τρισδιάστατης εκτύπωσης, ενώ το προστατευτικό περίβλημα επιλέχθηκε να προμηθευθεί από την εταιρία Blue Robotics (όπως και στην [2]) διότι παρουσίαζε εξαιρετική αντοχή ακόμα και σε μεγάλα βάθη (το συγκεκριμένο έως τα 400m). Τα δεδομένα από την κάμερα μπορούσαν να εμποτευθούν σε πραγματικό χρόνο μέσω streaming ενώ ταυτόχρονα αποθηκευόντουσαν είτε σε αποθετήριο στον ιστό, είτε στην κάρτα micro SD. Το πιο ενδιαφέρον στοιχείο της συγκεκριμένης εργασίας αποτέλεσε η μεταφορά των δεδομένων στο cloud ώστε να μην ασκεί πίεση ο φόρτος δεδομένων στην micro SD. Αυτό επιτεύχθηκε με πολύ απλό τρόπο όπως φαίνεται και στο διάγραμμα που ακολουθεί (Σχήμα 2.5).

Αρχικά, το Raspberry λάμβανε τις εικόνες από την κάμερα, τις αποθήκευε (προσωρινά) και τις συμπιέζε (zip) ενώ στην συνέχεια αν υπήρχε εξωτερικός σκληρός γινόταν αποθήκευση εκεί (τελική) αλλιώς ο μικροϋπολογιστής έκανε έλεγχο αν υπήρχε σύνδεση στο διαδίκτυο. Αν υπήρχε σύνδεση αλλά και ελεύθερος χώρος στο ηλεκτρονικό αποθετήριο, αποθήκευε το zip αρχείο εκεί και το διέγραφε από το Raspberry. Διαφορετικά, αν το αποθετήριο δεν είχε άλλο αποθηκευτικό χώρο, διέγραφε το παλαιότερο αρχείο από το αποθετήριο και στην συνέχεια αποθήκευε το καινούργιο. Παρότι ήταν μια πολύ απλή μεθοδολογία η ιδέα για καταγραφή σε ηλεκτρονικό αποθετήριο πράγματι θα μπορούσε να λύσει αρκετά προβλήματα που παρουσιάζονται σε τέτοιου είδους εργασίες.

Συνολικά και αυτή η εργασία μας έδωσε αρκετές ιδέες κάποιες από τις οποίες μπορεί να μην υλοποιήσαμε αλλά σίγουρα θα θελήσουμε να τις δοκιμάσουμε σε επόμενες βελτιωμένες εκδόσεις του συστήματος μας. Εκτός αυτού μας ανέδειξε ένα ακόμα κλάδο που δεν είχαμε συνειδητοποιήσει ότι το σύστημα μας θα μπορούσε να χρησιμοποιηθεί ο οποίος ήταν οι ιχθυοκαλλιέργειες.



Σχήμα 2.5: Σειρά εργασιών αποθήκευσης εικόνας. Πηγή: [3]

Γενικότερα, στο πλαίσιο της συγκεκριμένης εργασία λήφθηκαν υπόψιν και τμήματά από διαφορετικές εργασίες αλλά και από τα εγχειρίδια κατασκευαστών των διάφορων τμημάτων του συστήματος, για αυτό και δεν κατέστη δυνατόν να αναλυθούν όλα σε βάθος. Παρόλα αυτά στο κεφάλαιο 3 που ακολουθεί θα παρατεθούν και άλλες ενδιαφέρουσες πηγές. Η βιβλιογραφία που μελετήθηκε μας ενθάρρυνε καθώς τόνισε την ανάγκη υλοποίησης του αυτόνομου συστήματος μας, διότι ήταν πολύ μικρός ο αριθμός αντίστοιχων συστημάτων, ενώ η αξία τους είναι εμφανής καθώς είναι χρηστικά σε πολλές ερευνητικές και εμπορικές εφαρμογές. Εκτός αυτού κανένα από τα υλοποιημένα συστήματα δεν στόχευε σε παρατηρήσεις που θα γινόντουσαν σε μεγάλα βάθη, ούτε είχε αρκετή αυτονομία για να επιτύχει τις αποστολές που καλούμαστε να φέρουμε εις πέρας εμείς. Παρόλα αυτά αντλήσαμε στοιχεία από διάφορες ερευνητικές εργασίες, τα οποία θα γίνουν αντιληπτά στο κεφάλαιο που ακολουθεί.

2.2 Αρχές υλοποίηση μοντέλου ανίχνευσης

2.2.1 Εισαγωγή στην Βαθιά Μάθηση

Η τεχνητή νοημοσύνη αποτελεί τον βασικό όρο «ομπρέλα» που περιλαμβάνει ένα μεγάλο εύρος ζητημάτων τα οποία αφορούν τον τρόπο που ο υπολογιστής μπορεί να πλησιάσει και υπό μια έννοια να «μιμηθεί» την ανθρώπινη συμπεριφορά και τρόπο σκέψης, ακόμα και με σύνθετους τρόπους όπως είναι η αναγνώριση προτύπων, ανίχνευση αντικειμένων κ.α. Σε αυτή την διπλωματική εργασία δεν θα γίνει ανάλυση επί του συνόλου των ζητημάτων που απασχολούν την τεχνητή νοημοσύνη αλλά θα εμβαθύνουμε στην υποκατηγορία της μηχανικής μάθησης και στην συνέχεια της βαθιάς μάθησης (που αποτελεί «υποκατηγορία» της μηχανικής).

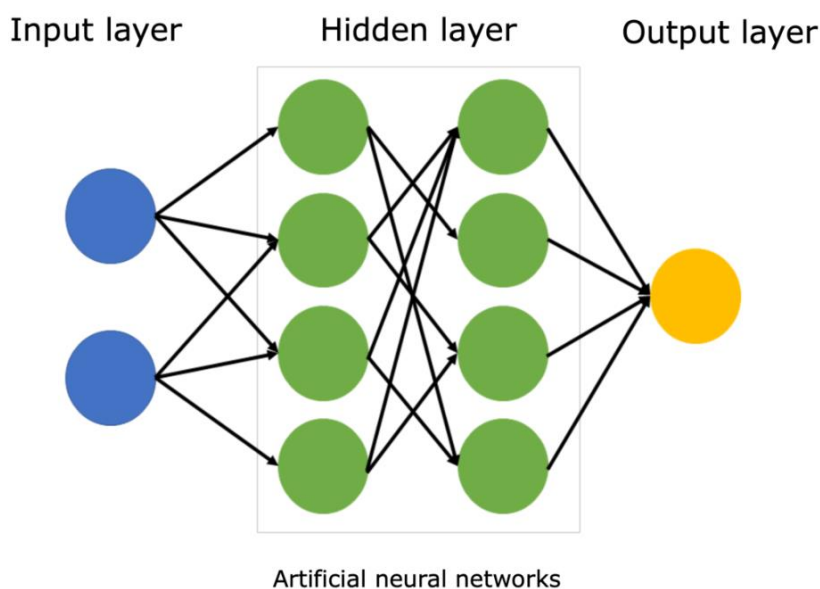
Η μηχανική μάθηση αποτελεί το «πεδίο μελέτης που δίνει στους υπολογιστές της δυνατότητα να μαθαίνουν χωρίς να έχουν ρητά προγραμματιστεί» (- Arthur Samuel, 1959). Τι σημαίνει όμως αυτός ο ορισμός; Ένας τυπικός κώδικας έχει προγραμματιστεί να υλοποιεί μια πολύ συγκεκριμένη διαδικασία με κάποιες εισροές και εκροές. Όσον αφορά την μηχανική μάθηση ο τρόπος που από τις εισροές φτάνει στις εκροές δεν είναι τόσο άμεσος και περνά από ένα σύνολο διαδικασιών που αρκετές φορές και ειδικά σε κάποιον που δεν έχει εντρυφήσει με εφάμιλλά ζητήματα μπορεί να φαντάζει ότι δεν έχει ξεκάθαρη σχέση με τον αρχικό σκοπό. Πρακτικά η μηχανική μάθηση αποτελείται από ένα σύνολο αλγορίθμων που ασχολούνται αποκλειστικά με τα δεδομένα χωρίς να ενδιαφέρονται ούτε από που έχουν προκύψει ούτε τι μορφή παρουσιάζουν (είτε δώσουμε ως εισροή εικόνες από ακτινογραφίες είτε εικόνες από το κινητό μας η αντιμετώπιση θα είναι η ίδια). Στην συνέχεια «εκπαιδεύονται»/ «μαθαίνουν» από τα δεδομένα εισαγωγής και βάσει αυτών μπορούν να επιλύσουν ζητήματα, να πάρουν αποφάσεις κ.α. Η μηχανική μάθηση χωρίζεται σε τρεις βασικές κατηγορίες:

- **Επιβλεπόμενη:** Στην οποία ο χρήστης δίνει ένα σύνολο δεδομένων εισόδου (εισροών) για τα οποία γνωρίζει εξ αρχής το αποτέλεσμα εξόδου (κλάσεις). Αυτά χαρακτηρίζονται και ως δεδομένα αληθείας (ground truth). Ο χρήστης διαχωρίζει το σετ δεδομένων του σε αυτά της μάθησης και αυτά του ελέγχου (ένα σύνθητες νούμερο είναι 70-80% για μάθηση και 20-30% για έλεγχο αλλά αυτό αλλάζει ανάλογα την εφαρμογή) και στην συνέχεια δημιουργεί και αξιολογεί το μοντέλο του. Μερικά παραδείγματα επιβλεπόμενης μάθησης είναι η παλινδρόμηση και η ταξινόμηση.

- Μη Επιβλεπόμενη: Σε αυτή ο χρήστης παρέχει δεδομένα εισόδου των οποίων δεν γνωρίζει το αποτέλεσμα εξόδου και μέσα από κάποια «χαρακτηριστικά» ο αλγόριθμός κάνει την «κατηγοριοποίηση» τους. Ένα χαρακτηριστικό παράδειγμα είναι αν τα δεδομένα εισόδου δημιουργούν συστάδες (clusters) λόγω των τιμών τους (οι οποίες πολλές φορές αντιμετωπίζονται χωρικά, σαν συντεταγμένες) σε κάποιο φάσμα όπως συμβαίνει και με τον πολύ γνωστό αλγόριθμο k-mean. Η μη επιβλεπόμενη ταξινόμηση μπορεί να χρησιμοποιηθεί και σαν εργαλείο εύρεσης του βέλτιστου αριθμού κλάσεων πριν γίνει μια επιβλεπόμενη ταξινόμηση (π.χ σε δασικές εκτάσεις μπορούμε να βρούμε τον βέλτιστο, κατά ένα βαθμό, αριθμό κλάσεων τον οποίο θα μπορούσαμε να αναγνωρίσουμε με κάποιον αλγόριθμο επιβλεπόμενης ταξινόμησης).
- Ενισχυτική: Είναι λίγο πιο σύνθετη από τις άλλες μορφές μιας και ο υπολογιστής μαθαίνει μέσα από ένα περιβάλλον που έχει «δυναμικό» χαρακτήρα όπως είναι ένα παιχνίδι, ή η αυτόματη οδήγηση, χωρίς όμως να έχει ρητές εντολές για τον σκοπό που πρέπει να επιτευχθεί. Η αλληλεπίδραση είναι αυτή που πρακτικά εκπαιδεύει το μοντέλο και σε αρκετές περιπτώσεις προκύπτει μια μορφή δοκιμής και λάθους (trial and error).

Αφού παρουσιάστηκε η ευρύτερη κατηγορία της μηχανικής μάθησης, θα γίνει εμβάθυνση σε αλγορίθμους βαθιάς μάθησης που αποτέλεσαν και το κομμάτι που αξιολογεί η παρούσα εργασία. Η βαθιά μάθηση βασίζεται κυρίως σε εκμάθηση από δεδομένα (αναπαραστάσεις) υψηλού επιπέδου (deep features). Το βασικό «δομικό» της στοιχείο είναι ο νευρώνας (perceptron, ο οποίος πρακτικά υλοποιεί μια δυαδική ταξινόμηση). Ένα σύνολο νευρώνων συνδεδεμένων σε πολλά διαφορετικά επίπεδα υλοποιούν ένα τεχνητό νευρωνικό δίκτυο (Σχήμα 2.6). Κάποια από τα μεγάλα πλεονεκτήματα των συγκεκριμένων δικτύων αφορούν την διαλειτουργικότητα τους μέσα από την μεταφορά μάθησης εντός του ίδιου χώρου προτύπων ή και τον επανασυντονισμό (fine tuning) ώστε να προσαρμοστούν τα προεκπαιδευμένα δίκτυα σε διαφορετικούς αλλά εφάμιλλους χώρους προτύπων σε σχέση με το αρχικό δίκτυο. Στο πλαίσιο της συγκεκριμένης εργασίας εμβαθήναμε σε δίκτυα που έχουν άμεση σχέση με την όραση υπολογιστών μιας και τα δεδομένα που θέλαμε να αξιολογήσουμε αφορούσαν την εικόνα. Δομές όπως αυτές που εμφανίζονται στα βαθιά νευρωνικά δίκτυα έχουν μεγάλες απαιτήσεις σε δεδομένα για να «παράξουν» ικανοποιητικά μοντέλα. Ο τρόπος που η βαθιά μάθηση αντιμετωπίζει τα ζητήματα στην όραση υπολογιστών είναι μέσα από τα συνελκτικά νευρωνικά δίκτυα (CNN). Μερικές

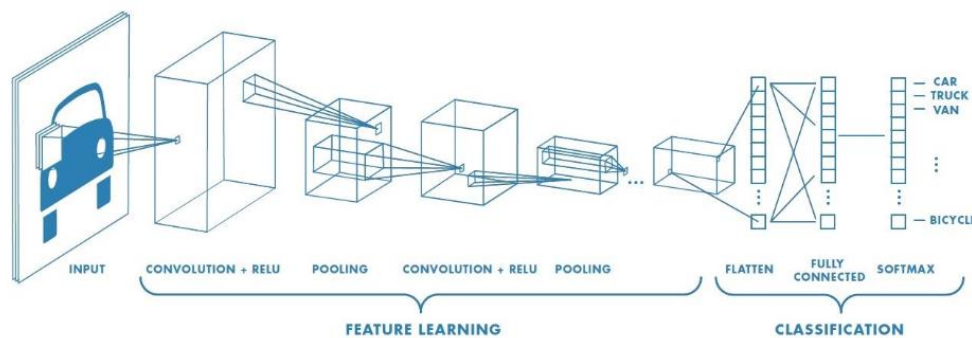
εφαρμογές της βαθιάς μάθησης στην όραση υπολογιστών είναι η σημασιολογική κατάτμηση, η ανίχνευση αντικειμένων, η μεταφορά στυλ κ.α. Ενδιαφέρον έχει να σχολιαστεί το Σχήμα 2.6 που παρουσιάζεται στην συνέχεια. Κάθε κόμβος στο σχήμα αποτελεί ένα νευρώνα με τους μπλε να συντελούν νευρώνες εισόδου, τον κίτρινο εξόδου και τους πράσινους να είναι τα κρυφά επίπεδα (hidden layers). Παρατηρώντας το Σχήμα γίνεται αντιληπτό ότι τα επίπεδα είναι πλήρως συνδεδεμένα (fully connected). Εκτός αυτού κάθε βέλος μεταξύ δύο επιπέδων αναπαριστά ένα βάρος, όλα τα βέλη που παρουσιάζονται θα μπορούσαν να γραφούν σε ένα κοινό πίνακα βαρών καθώς αποτελούν ένα γραμμικό μετασχηματισμό μεταξύ δύο διανυσμάτων με τις αντίστοιχες διαστάσεις. Το μη γραμμικό κομμάτι του συστήματος είναι στην έξοδο των κόμβων.



Σχήμα 2.6: Πολυεπίπεδο νευρωνικό δίκτυο (Πηγή : <https://www.sciencelearn.org.nz/images/5156-neural-network-diagram>)

Πριν γίνει η παρουσίαση των τεχνικών ανιχνεύσεων θα παρουσιαστούν λίγο πιο αναλυτικά τα συνελκτικά νευρωνικά δίκτυα μιας και αποτελούν την βέλτιστη προσφερόμενη λύση στη διαχείριση εικόνων. Οι συνελκτικές στρώσεις δέχονται ως είσοδο εικόνες σε μορφή τανυστών 2^{ης} τάξης αν η εικόνα είναι grayscale ή 3^{ης} αν η εικόνα αποτελεί φυσικό έγχρωμο σύνθετο RGB. Οι συγκεκριμένες στρώσεις τοποθετούνται ανάμεσα στα κρυφά επίπεδα και εξάγουν πληροφορία υψηλού επιπέδου (μέσα από τους πίνακες χαρακτηριστικών). Συνήθως μετά από τις συνελκτικές στρώσεις παρουσιάζονται και στρώσεις επιπέδων συγκέντρωσης (pooling) με στόχο να μειώσουν τις διαστάσεις της εικόνας ούτως ώστε να επιτευχθεί μείωση των διαστάσεων του πίνακα χαρακτηριστικών. Το τμήμα του δικτύου στο οποίο εμφανίζονται οι συνελκτικές και οι pooling στρώσεις αποτελεί το τμήμα εξαγωγής χαρακτηριστικών

και συνήθως εφαρμόζεται πριν γίνει η ταξινόμηση. Παρατηρώντας και το Σχήμα 2.7 φαίνεται ότι μετά από το τμήμα εξαγωγής χαρακτηριστικών γίνεται επιπεδοποίηση σε τανυστή 1ου επιπέδου ώστε μέσα από τις πλήρως συνδεδεμένες στρώσεις του δικτύου να γίνει η διαδικασία την ταξινόμησης. Η Softmax (που φαίνεται στο Σχήμα) αποτελεί συνάρτηση ενεργοποίησης.



Σχήμα 2.7: Τυπικό CNN (Πηγή : <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>)

2.2.2 Ανίχνευση αντικειμένων

Η ανίχνευση αντικειμένων ορίζεται ως ένα πρόβλημα με διττή φύση. Αρχικά, τίθεται το ζήτημα της ταξινόμησης, δηλαδή ποια κλάση είναι το αντικείμενο που ανιχνεύεται και στην συνέχεια η «τοπικοποίηση» (παλινδρόμηση), δηλαδή ποια είναι η ακριβής θέση του αντικείμενου στην εικόνα. Γενικά, η ανίχνευση αντικειμένων χωρίζεται σε single object και σε multiple object. Στην περίπτωση της εργασίας μας η ανίχνευση των φουσαλίδων αποτελεί ένα πρόβλημα ανίχνευσης αντικειμένων μιας κλάσης (single object detection). Η διαφορά μεταξύ των δύο κατηγοριών οφείλεται στον αριθμό των κλάσεων εξόδου. Σημαντικός όρος για την ανίχνευση αντικειμένων αποτελούν τα κουτιά περικλεισης (bounding boxes), τα οποία ορίζονται ως ορθογώνια παραλληλόγραμμα τα οποία περικλείουν το αντικείμενο ενδιαφέροντος. Οι συντεταγμένες των bb μπορεί να παρουσιάζονται με διάφορους τρόπους ανάλογα την μορφοποίηση του απαιτεί η κάθε εφαρμογή ώστε να αποτελέσουν ορθά δεδομένα εισόδου, μερικά χαρακτηριστικά παραδείγματα είναι:

- $(X_{min}, Y_{min}, X_{max}, Y_{max})$ -Pascal Format
- $(X_{min}, Y_{min}, Width, Height)$ -COCO Format
- $(X_{centre}, Y_{centre}, Width, Height)$ -YOLO Format

Η ανίχνευση των αντικειμένων χωρίζεται σε δύο βασικές υποκατηγορίες:

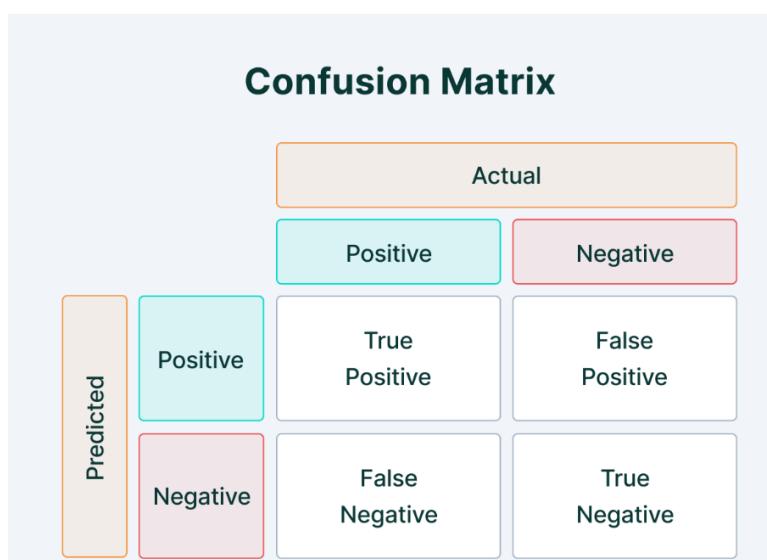
- Αρχικά υπάρχουν οι ανιχνευτές ενός σταδίου (1 stage detector), με πιο χαρακτηριστικά παραδείγματα την οικογένεια YOLO και το Retina Net. Στις περιπτώσεις αυτές οι ανιχνευτές ενός σταδίου αποτελούν πλήρως συνελκτικές αρχιτεκτονικές. Επίσης, όπως γίνεται αντιληπτό και από την ονομασία τους κάνουν απευθείας ανίχνευση των αντικείμενων με ένα «πέρασμα» από το συνεχόμενο δίκτυο.
- Επίσης, υπάρχουν οι ανιχνευτές δυο σταδίων (2 stage detector), με πιο χαρακτηριστικά παραδείγματα την οικογένεια R-CNN και τα Pyramid Networks. Οι συγκεκριμένοι ανιχνευτές ορίζουν περιοχές ενδιαφέροντος (ROI) και ανιχνεύουν βάση αυτών. Το πρώτο τους στάδιο ορίζει ένα δίκτυο από πιθανές περιοχές ενδιαφέροντος (RPN) βάσει του πίνακα χαρακτηριστικών που προκύπτει από ένα CNN. Έτσι ορίστηκε ο όρος «objectness» ο οποίος δηλώνει την ύπαρξη αντικειμένων ανεξαρτήτων της κατηγοριών (κλάσης) που ταξινομούνται αυτά, εντός της περιοχής αξιολόγησης. Βάση αυτής της μετρικής ορίζονται οι περιοχές ενδιαφέροντος και επαναυπολογίζονται οι πίνακες χαρακτηριστικών. Όσον αφορά το δεύτερο στάδιο εντός των περιοχών ενδιαφέροντος γίνεται η ταξινόμηση και η παλινδρόμηση των αντικειμένων.

Σε αυτό το στάδιο οφείλει να αναλυθεί μια από τις βασικότερες μετρητικές στην ανίχνευση αντικειμένων, η average precision (AP), μιας και θα χρησιμοποιηθεί ιδιαίτερα στην αξιολόγηση του μοντέλου που θα παρουσιαστεί. Αρχικά, υλοποιείται ο πίνακας σύγχυσης (confusion matrix) (Σχήμα 2.8) στον οποίο αντιπαρατίθενται τα πραγματικά δεδομένα εξόδου με αυτά που έχει προβλέψει το μοντέλο. Έτσι προκύπτουν 4 βασικές κατηγορίες:

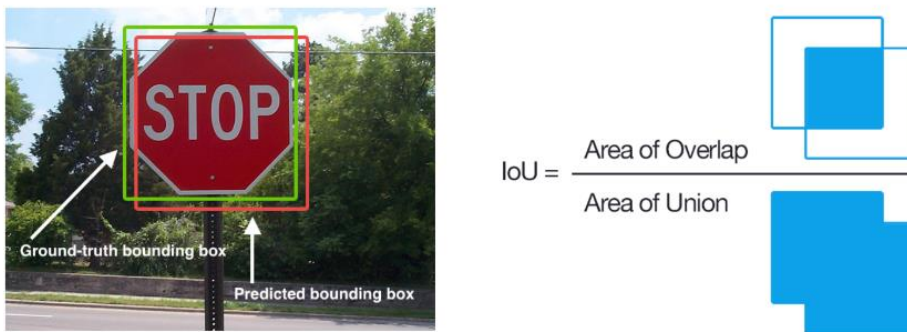
1. True Positive ή Ορθός Θετικό (TP): Όταν ένα δείγμα προβλέπεται ως θετικό από τον αλγόριθμο και είναι όντως θετικό στα πραγματικά δεδομένα εξόδου.
2. True Negative ή Ορθός Αρνητικό (TN): Όταν ένα δείγμα προβλέπεται ως αρνητικό από τον αλγόριθμο και είναι όντως αρνητικό στα πραγματικά δεδομένα εξόδου.
3. False Positive ή Λανθασμένα Θετικό (FP): Όταν ένα δείγμα προβλέπεται ως θετικό από τον αλγόριθμο και είναι αρνητικό στα πραγματικά δεδομένα εξόδου.

- False Negative ή Λανθασμένα Αρνητικό (FN): Όταν ένα δείγμα προβλέπεται ως αρνητικό από τον αλγόριθμο και είναι θετικό στα πραγματικά δεδομένα εξόδου.

Από το συγκεκριμένο πίνακα υπολογίζεται η $IoU = \frac{\text{Περιοχή επικάλυψης}}{\text{Περιοχή ένωσης}}$ (Σχήμα 2.9) η οποία αποτελεί και κατώφλι για τον διαχωρισμό στον πίνακα σύγχυσης και παίρνει τιμές από 0 έως 1. Στην συνέχεια ορίζονται ακόμα δύο μετρικές το $precision = \frac{True\ Positives}{False\ Positives + True\ Positives}$ και το $recall = \frac{True\ Positives}{False\ Negatives + True\ Positives}$, οι οποίες επίσης παρουσιάζουν τιμές στο διάστημα 0 έως 1 (και δηλώνουν καλύτερα αποτελέσματα όσο προσεγγίζουν την μονάδα). Τα δύο συγκεκριμένα μεγέθη οπτικοποιούνται (συνήθως ανά κατηγορία) δημιουργώντας μια καμπύλη, για διαφορετικές τιμές του IoU. Από την συγκεκριμένη καμπύλη υπολογίζοντας το εμβαδόν ανάμεσα σε αυτή και τον οριζόντιο άξονα προκύπτει η μετρητική AP (πρακτικά αποτελεί το σταθμισμένο μέσος όρος των precision που επιτυγχάνονται για διαφορετικά recall). Τέλος μπορεί να υπολογιστεί η μετρική mAP (Mean Average Precision) η οποία ορίζεται ως ο μέσος όρος των AP για κάθε κλάση και αποτελεί μετρική που χρησιμοποιείται αρκετά σε περιπτώσεις ανίχνευσης.



Σχήμα 2.8 Πίνακας σύγχυσης (Πηγή: <https://www.v7labs.com/blog/mean-average-precision#:~:text=Average%20Precision%20is%20calculated%20as,mAP%20varies%20in%20different%20contexts>)



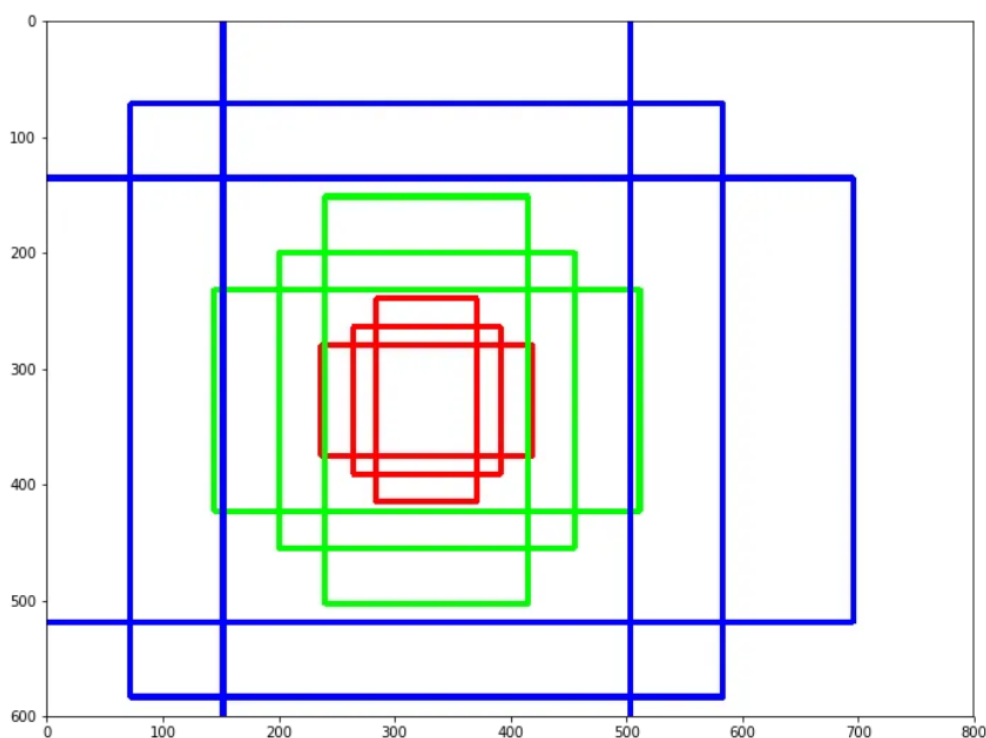
Σχήμα 2.9 IoU (Πηγή: <https://www.v7labs.com/blog/mean-average-precision#:~:text=Average%20Precision%20is%20calculated%20as,mAP%20varies%20in%20different%20contexts>)

Αφού έχει γίνει η παρουσίαση του γενικού θεωρητικού πλαισίου, θα παρατεθούν οι δύο αρχιτεκτονικές που αξιολογήθηκαν για την παρούσα εργασία. Αρχικά, θα παρουσιαστεί ο αλγόριθμος Faster R-CNN [4] ο οποίος αποτελεί μέλος της αρκετά ευρείας οικογένειας R-CNN και είναι ανιχνευτής δύο σταδίων. Ο συγκεκριμένος αλγόριθμος προέκυψε ως βελτιωμένη έκδοση του Fast R-CNN. Όσον αφορά τα στάδια του, λειτουργούν όπως σε έναν τυπικό 2 stage detector, με το πρώτο να βρίσκει την ύπαρξη των αντικειμένων μέσα σε περιοχές ενδιαφέροντος δημιουργώντας τα proposal regions, ενώ το δεύτερο αναλύει τα συγκεκριμένα proposals και τα ταξινομεί στις κλάσεις (κατηγορίες) που αυτά αντιστοιχούν.

Ο αλγόριθμος Faster R-CNN ήταν αυτός που όρισε σαν έννοια τα δίκτυα της μορφής RPN, το οποίο αφορά μονάχα προτάσεις αντικειμένων, ενώ σε αντίθεση με την προηγούμενη έκδοση του (Fast R-CNN) είναι πολύ πιο γρήγορο. Η πιο μεγάλη συνεισφορά του συγκεκριμένου αλγορίθμου ήταν ότι εισήγαγε την έννοια των αγκυρώσεων (anchors) οι οποίες αποτελούν παραλληλόγραμμα πολλών διαφορετικών διαστάσεων βάσει των οποίων προτείνονται και οι περιοχές ενδιαφέροντος (έτσι ο χρήστης μπορεί να «επιλέξει» το μέγεθος των αντικειμένων που θέλει να ανιχνεύσει). Οι παλαιότερες μέθοδοι ήταν η «πυραμίδα» εικόνων (εικόνες διαφορετικών κλιμάκων) ή τα φίλτρα διαφορετικών μεγεθών. Ο αλγόριθμος χρησιμοποιώντας κοινές πράξεις συνέλιξης μεταξύ του ανιχνευτή και του RPN κατάφερε να μειώσει αισθητά το υπολογιστικό κόστος. Στο Σχήμα 2.10 που ακολουθεί φαίνεται ένα παράδειγμα με 9 διαφορετικά anchors πολλαπλών μεγεθών.

Το RPN εναποθέτει τα δεδομένα εισόδου (εικόνες) σε ένα (backbone) CNN δίκτυο αφού πρώτα ελεγχθούν και μεταβληθούν οι διαστάσεις τους. Στην συνέχεια εισάγονται οι «αγκυρώσεις» στα δεδομένα εισόδου και ορίζονται οι περιοχές ενδιαφέροντος. Μετά

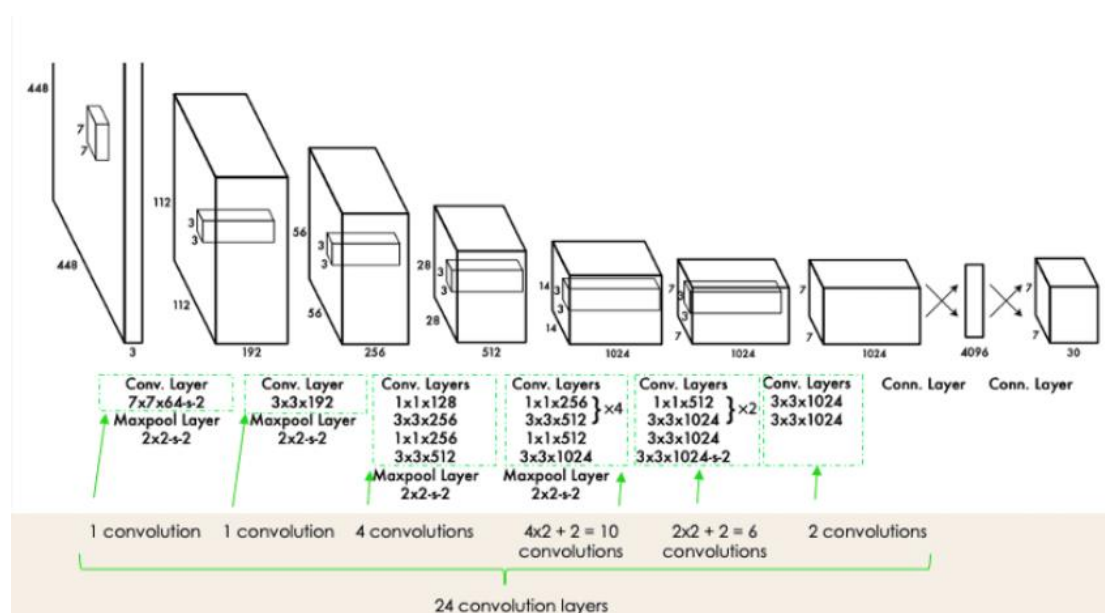
γίνεται έλεγχος των αγκυρώσεων και υπολογίζονται τα bb. Πρακτικά το RPN χωρίζεται στα τμήματα της ταξινόμησης (έλεγχος αν υπάρχει αντικείμενο) και στην παλινδρόμηση των κουτιών περικλείσις (εύρεση συντεταγμένων των «αγκυρώσεων»). Το μεγαλύτερο ζήτημα της συγκεκριμένης μεθόδου είναι ότι «δημιουργεί» πυκνές και επικαλυπτόμενες περιοχές ενδιαφέροντος. Για αυτό και χρησιμοποιείται ο αλγόριθμος Non-Maximum Suppression (NMS) βάσει του οποίου επιλέγονται οι περιοχές ενδιαφέροντος με την υψηλότερη πιθανότητα ορθής πρόβλεψης.



Σχήμα 2.10: Παράδειγμα 9 διαφορετικών anchors (Πηγή : <https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8>)

Ένας ακόμα ενδεδειγμένος αλγόριθμος για ανίχνευση αντικειμένων είναι ο You Only Look Once (YOLO) [5] ο οποίος αποτελείται από ένα στάδιο. Ο τρόπος που λειτουργεί ο συγκεκριμένος αλγόριθμος είναι τμηματοποιώντας την εικόνα σε κελιά εντός των οποίων γίνεται η ανίχνευση και η κατηγοριοποίηση των αντικειμένων. Αντίστοιχα με τον αλγόριθμο Faster-RCNN γίνονται προβλέψεις για τις συντεταγμένες των κουτιών περικλείσις (εντός των κελιών), καθώς και για την κλάση του κάθε αντικειμένου, σε συνδυασμό με την πιθανότητα ύπαρξης του εντός του συγκεκριμένου κελιού. Ένα βασικό ζήτημα που προκύπτει είναι ότι πολλά κελιά που γειτνιάζουν προβλέπουν το ίδιο αντικείμενο και έτσι θα πρέπει να χρησιμοποιηθεί η NMS όπως και στο προηγούμενο αλγόριθμο. Τεχνικά, ο συγκεκριμένος αλγόριθμος (Σχήμα 2.11) προκύπτει από 24 στρώσεις (συνεκτικές) καθώς και δύο fully connected. Τα αρχικά

βάρη του αλγορίθμου έχουν προκύψει από την εκπαίδευση που έχει γίνει στο ImageNet. Εκτός αυτού, ενδιαφέρον παρουσιάζει και η συνάρτηση κόστους (loss function) του, καθώς έχει 5 όρους, 2 για την παλινδρόμηση, 2 για την ταξινόμηση και έναν κοινό που αφορά και τις δύο. Στα νευρωνικά δίκτυα η συνάρτηση κόστους αποτελεί υπό μια έννοια κριτήριο τερματισμού καθώς ο στόχος είναι με την σύγκλιση των παραμέτρων να επιτευχθεί ελαχιστοποίηση (και σύγκλιση) της συγκεκριμένης συνάρτησης. Το YOLO παρουσιάζει ως κύρια πλεονεκτήματα την μεγάλη ταχύτητα στην υλοποίηση και τις υψηλές δυνατότητες γενίκευσης του. Έτσι, καθίσταται δυνατό σε αρκετές περιπτώσεις να εφαρμοστεί η τεχνική του domain adaptation. Ως κύρια μειονεκτήματα του λογίζεται η αδυναμία ανίχνευσης αντικειμένων πολλών διαφορετικών κλιμάκων, καθώς και το γεγονός ότι εξαιτίας του τρόπου που ορίστηκε η συνάρτηση κόστους αντιμετωπίζει με τον ίδιο τρόπο τις αστοχίες που γίνονται σε μικρά και μεγάλα κουτιά περικλείσις με αποτέλεσμα να υπερτονίζονται τα λάθη στην τοποθεσία των αντικειμένων.



Σχήμα 2.11: Αρχιτεκτονική του YOLO (Πηγή : <https://www.datacamp.com/blog/yolo-object-detection-explained>)

2.3 Σχετικές εργασίες ανίχνευσης φυσαλίδων

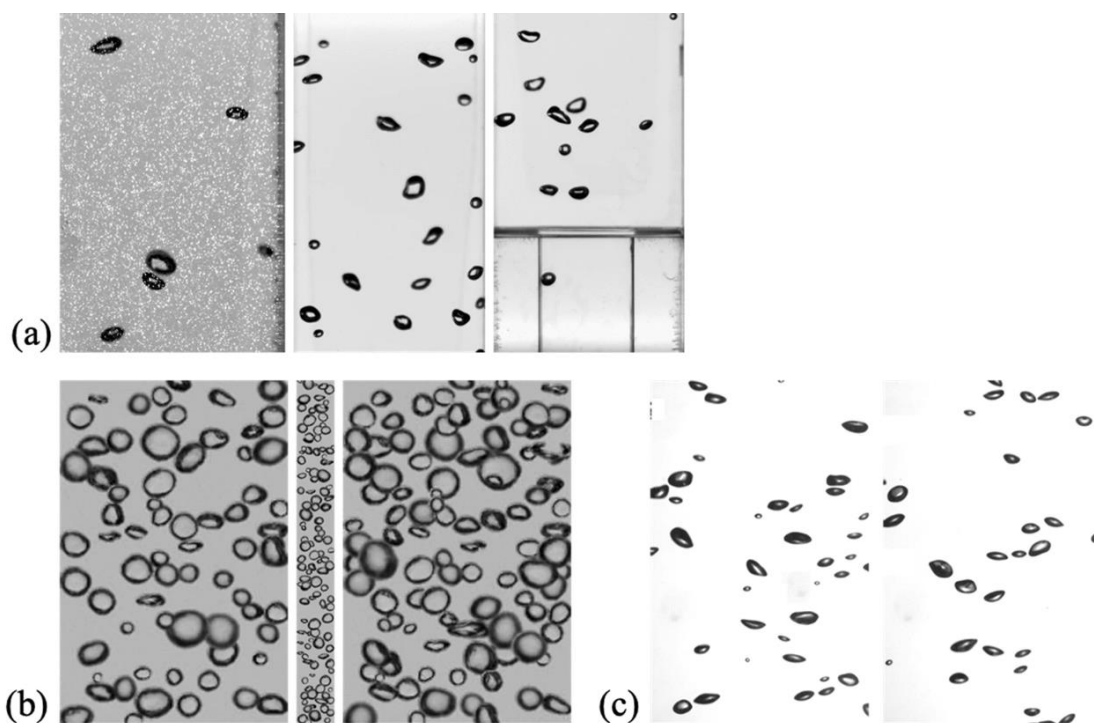
Η ανίχνευση φυσαλίδων αποτελεί ένα ζήτημα που έχει αξιολογηθεί στο παρελθόν, καθώς η επίλυση του μπορεί να συνεισφέρει σε αρκετούς επιστημονικούς κλάδους. Σε αυτό το κεφάλαιο θα παρουσιαστεί η βιβλιογραφία που αποτέλεσε πηγή έμπνευσης για την δική μας μεθοδολογία, αλλά και άλλες ενδιαφέρουσες προσεγγίσεις, κυρίως υπό

το πρίσμα της βαθιάς μάθησης. Πριν ξεκινήσει η παρουσίαση των προσεγγίσεων, δεν θα μπορούσε να παραληφθεί το γεγονός ότι καμία από αυτές που θα παρουσιαστούν, αλλά και γενικότερα από όσες υφίστανται στην υπάρχουσα βιβλιογραφία, δεν παρέχουν πραγματικά δεδομένα για ανάλυση, παρότι σε κάποιες περιπτώσεις τα χρησιμοποιούν (τα οποία όμως έχουν εργαστηριακή προέλευση). Εκτός αυτού εκμεταλλεύονται συνθετικά δεδομένα που έχουν δημιουργήσει για να ενισχύσουν την στιβαρότητα των μοντέλων τους (κυρίως βελτιώνοντας την γενίκευση) αλλά και για να αυξήσουν το πλήθος των δεδομένων που χρησιμοποιούν για την εκπαίδευση.

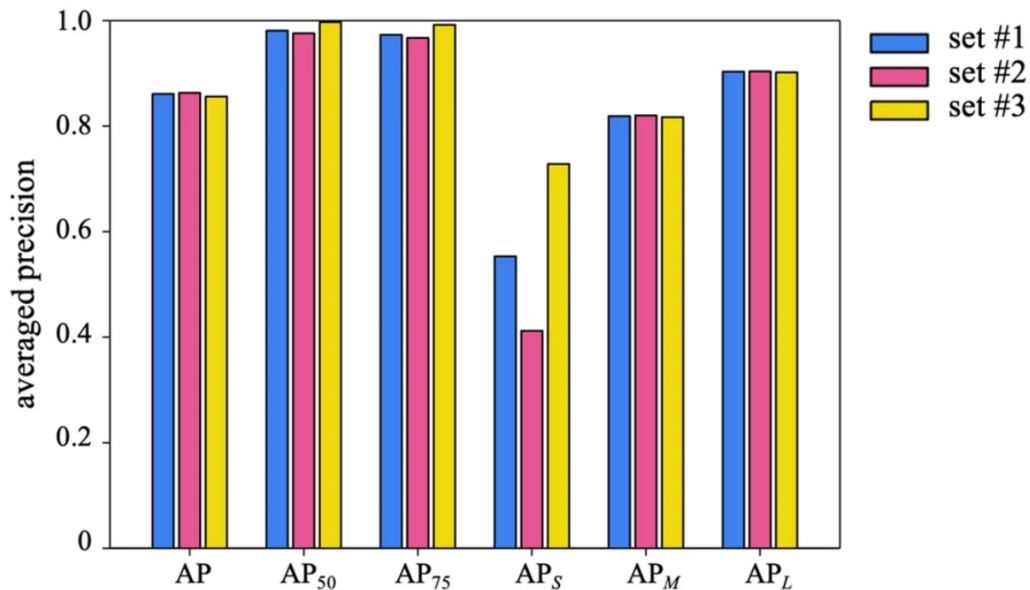
Η πρώτη ερευνητική εργασία που θα παρουσιαστεί είναι η [6], η ομάδα της οποίας εκπαίδευσε τον αλγόριθμο mask R-CNN (χρησιμοποιώντας ResNet-101 ως backbone και εφαρμόζοντας μεταφορά μάθησης από προ εκπαιδευμένα βάρη COCO) ο οποίος εφαρμόζει την τεχνική instance segmentation (τμηματοποίηση κατά περίπτωση) η οποία ανιχνεύει τα αντικείμενα και παράγει μάσκες σαν αποτελέσματα, σε αντίθεση με τις μεθόδους που μελετήθηκαν στο προηγούμενο κεφάλαιο όπου το αποτέλεσμα ήταν bb. Ο συγκεκριμένος αλγόριθμος ανήκει στην ευρεία οικογένεια R-CNN η οποία εμφανίζει πολύ καλά αποτελέσματα στην ανίχνευση εικόνων (είτε με detection είτε με IS). Πράγματι τα μετρικά αποτελέσματα αποτελέσματά τους ήταν αρκετά εντυπωσιακά (με average precision₅₀ πάνω 95% ακόμα και σε ροές που δεν υπήρχαν κατά την εκπαίδευση). Ένα από τα σημαντικότερα στοιχεία της συγκεκριμένης εργασίας ήταν η ιδιαίτερη προσοχή που έδωσαν στο σετ δεδομένων τους ώστε να επιτύχουν τα βέλτιστα δυνατά αποτελέσματα με τα λιγότερα δυνατά δεδομένα. Αυτό επιτεύχθηκε με έναν συνδυασμό από συνθετικά (BubGAN) και πραγματικά (εργαστηριακά) δεδομένα και μια σειρά μεθόδων ενίσχυσης της πληροφορίας. Εκτός αυτού χρησιμοποίησαν δικιά τους συνάρτηση κόστους (loss function), χάρις στην οποία κατέστη δυνατό να «επιλέξει» ο χρήστης πόσο μικρές φυσαλίδες θέλει να ανιχνεύσει. Στο Σχήμα 2.12 που ακολουθεί φαίνονται κάποια δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση του αλγορίθμου. Με (a),(c) παρουσιάζονται δεδομένα που συλλέχθηκαν εργαστηριακά. Οι μέθοδοι που χρησιμοποιήθηκαν για τα δεδομένα ήταν η σκιαγράφηση και η μέτρηση ταχύτητας ροής με απεικόνιση σωματιδίων. Στο (b) παρουσιάζονται φυσαλίδες που έχουν παραχθεί συνθετικά. Όσον αφορά τα συνθετικά δεδομένα επιλέχθηκε κατά την παραγωγή τους το IoU μεταξύ δύο κουτιών περικλεισης να είναι 0,11-0,16 ώστε να μην έχουν πολύ μεγάλες επικαλύψεις. Όλα οι εικόνες μετασχηματίστηκαν σε διαστάσεις 640×640 εικονοστοιχείων, ασχέτως του τρόπου εξαγωγής των αρχικών δεδομένων.

Όσον αφορά την εκπαίδευση το μοντέλο τους εκπαιδεύτηκε μόλις για 24 εποχές χρησιμοποιώντας μέγεθος batch 1 και με αρχική ταχύτητα μάθησης 10^{-4} (τυπική τιμή).

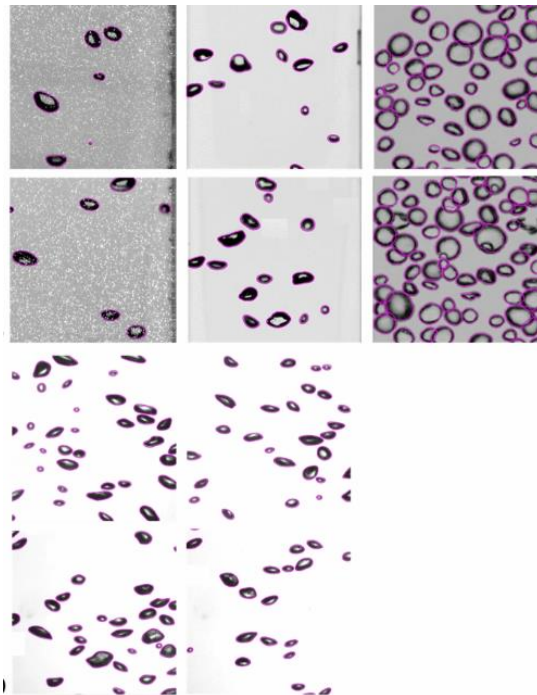
Σημαντική λεπτομέρεια είναι ότι από το πλήρες μοντέλο, μονάχα το στάδιο 5 του Resnet καθώς και η κεφαλή (head layer) επανεκπαιδεύτηκαν. Όσον αφορά τις μεθόδους ενίσχυσης που χρησιμοποιήθηκαν ήταν η περιστροφή, η αντιστροφή της εικόνας και η προσθήκη gaussian θορύβου τυχαία στο δείγμα τους. Οι μετρικές του μοντέλου τους έδειξαν πολλά υποσχόμενα αποτελέσματα (Σχήμα 2.13). Ιδιαίτερα εντυπωσιακά ήταν τα αποτελέσματα στο δεύτερο σετ δεδομένων, όπου τα δεδομένα ελέγχου έμοιαζαν πολύ με αυτά της εκπαίδευσης, αλλά και το τρίτο σετ ελέγχου όπου τα δεδομένα είχαν αρκετές διαφορές. Σε όλα τα σετ ελέγχου φαίνεται ότι το AP κινείται σε πολύ υψηλά επίπεδα (για κατώφλι IoU=0.5 οι τιμές κοντεύουν την μονάδα). Μερικά ποιοτικά αποτελέσματα (για IoU>0.5) παρουσιάζονται στο Σχήμα 2.14 που ακολουθεί. Στο οποίο οι μάσκες «εφαρμόζουν» πολύ ικανοποιητικά στα σχήματα των φυσαλίδων, παρότι τα συνθετικά με τα εργαστηριακά δεδομένα παρουσιάζουν πολύ έντονες γεωμετρικές διαφορές.



Σχήμα 2.12: Φυσαλίδες που χρησιμοποιήθηκαν κατά την εκπαίδευση [6]



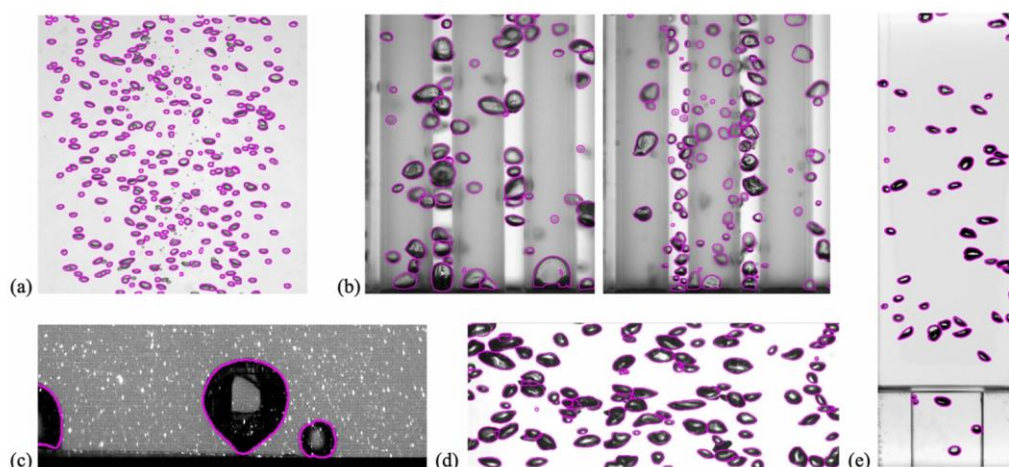
Σχήμα 2.13: Μετρικές αξιολόγησης του μοντέλου [6]



Σχήμα 2.14: Μάσκες που προέκυψαν από το μοντέλο [6]

Αφού εξήγαγαν τα αποτελέσματα θέλησαν να ελέγξουν πόσο επηρεάζει η πυκνότητα των φυσαλίδων τα μοντέλα (δηλαδή το πόσο μεγάλο μέρος της εικόνας καταλαμβάνουν και πόσο επικαλυπτόμενες είναι). Από τα αποτελέσματα έγινε αντιληπτό ότι όταν αυξάνεται η επικάλυψη, η απόδοση του μοντέλου μειωνόταν και οι μετρικές έφθιναν. Δοκιμάζοντας το μοντέλο τους σε εντελώς διαφορετικά σενάρια ελέγχου από αυτά που εκπαιδεύτηκε έδειξε ότι οι ικανότητες γενίκευσης του ήταν πολύ ικανοποιητικές (Σχήμα 2.15). Συνολικά οι συγγραφείς πέτυχαν την υλοποίηση ενός πολύ στιβαρού μοντέλου ανίχνευσης φυσαλίδων με αποτελέσματα εξόδου να είναι

μάσκες, οι οποίες παρέχουν την δυνατότητα χρησιμοποίησης σε μεγαλύτερο εύρος εφαρμογών από ότι τα bb.



Σχήμα 2.15: Μάσκες που προέκυψαν από το μοντέλο σε διαφορετικά δεδομένα από αυτά που εκπαιδεύτηκε [6]

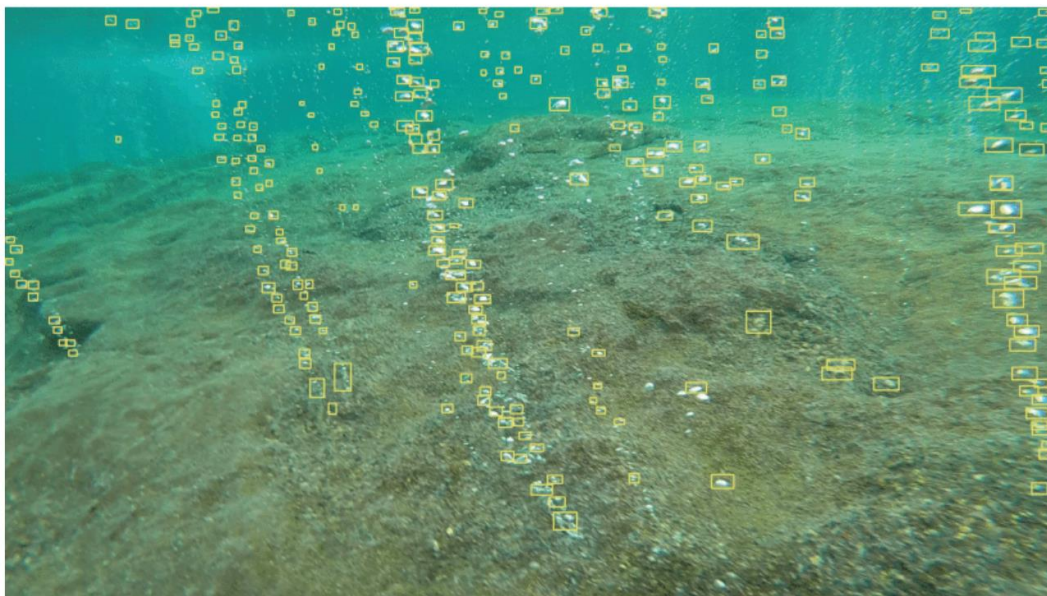
Μια διαφορετική προσέγγιση με βαθιά μάθηση παρουσιάζεται στην ερευνητική εργασία [7]. Η επιστημονική ομάδα εφάρμοσε τον αλγόριθμο YOLO σε δεδομένα που προέκυψαν από το ηφαιστειό Vulcano (στην Ιταλία), καθώς και σε δεδομένα που βρήκαν στο διαδίκτυο. Όσον αφορά το ηφαιστειό, παρουσιάζει μεγάλο αριθμό ενεργών καμινάδων, οι οποίες λειτούργησαν ως πηγές για την εξαγωγή των δεδομένων τους (Σχήμα 2.16). Η επιλογή του YOLO ως αλγόριθμο ανίχνευσης από την συγκεκριμένη ομάδα έγινε κυρίως λόγω της ταχύτητας του συγκεκριμένου ανιχνευτή αλλά και των καλών αποτελεσμάτων που παράγει σε αντικείμενα παρόμοιας κλίμακας (συνολικά λίγο χειρότερη ακρίβεια από ότι η οικογένεια R-CNN).

Το σετ δεδομένων που χρησιμοποιήθηκε στην εκπαίδευση του αλγορίθμου προέκυψε από 15 εικόνες υποθαλάσσιου βίντεο της ευρύτερης περιοχής του ηφαιστείου και 20 εικόνες από φυσαλίδες σε διαφορετικά περιβάλλοντα (για να αυξηθεί η γενίκευση και η στιβαρότητα του μοντέλου). Επίσης, 15 εικόνες προέκυψαν από τμηματοποίηση των αρχικών εικόνων και χρησιμοποιήθηκαν σε δεύτερη φάση της εκπαίδευσης. Η ετικετοποίηση των δεδομένων έγινε χειροκίνητα χρησιμοποιώντας κουτιά περίκλεισης (σε αντίθεση με την προηγούμενη ερευνητική εργασία που χρησιμοποιήθηκαν μάσκες) με PASCAL VOC format. Το μοντέλο που υλοποιήθηκε από την συγκεκριμένη ομάδα απέδωσε αρκετά ικανοποιητικά αποτελέσματα όπως φαίνεται και στο Σχήμα 2.17 όπου ο μεγαλύτερος αριθμός των φυσαλίδων της εικόνας έχει ανιχνευθεί ορθά. Το μεγαλύτερο πρόβλημα εντοπίστηκε σε φυσαλίδες που απείχαν μεγάλες αποστάσεις από τον αισθητήρα με αποτέλεσμα να λογίζονται ως μικρά αντικείμενα (small object), η ανίχνευση των οποίων μπορεί να αποτελέσει πρόκληση.

Οι συγγραφείς υπογραμμίζουν ότι τα αποτελέσματα του μοντέλου που εκπαιδεύτηκε στο σετ με τις 50 εικόνες (χρησιμοποιώντας και τις 15 που προέκυψαν με τμηματοποίηση) ήταν καλύτερα σε σχέση με αυτά που προέκυψαν από το σετ των 35 εικόνων. Συνολικά, η συγκεκριμένη ερευνητική εργασία είχε δύο βασικά ζητήματα, τον μικρό αριθμό δεδομένων εκπαίδευσης που χρησιμοποίησαν, αλλά και την έλλειψη μετρικών ούτως ώστε να επιτευχθεί ορθή αξιολόγηση των αποτελεσμάτων τους. Παρόλα αυτά έγινε αντιληπτό ότι ο αλγόριθμος YOLO θα μπορούσε να αποτελέσει ικανοποιητική επιλογή για την επίτευξη της ανίχνευσης φυσαλίδων.



Σχήμα 2.16: Φυσαλίδες σε βάθος 5 μέτρων



Σχήμα 2.17: Ανίχνευση φυσαλίδων που προέκυψε με τον αλγόριθμο YOLO

Η τελευταία ερευνητική εργασία που θα παρουσιαστεί είναι η [8]. Παρότι σε αυτή δεν έγινε χρήση κάποιας τεχνικής βαθιάς μάθησης, παρουσίασε ενδιαφέρον ως προς την ανίχνευση των φυσαλίδων. Ο αλγόριθμος ανίχνευσης που χρησιμοποιήθηκε βασίστηκε στο μετασχηματισμό Hough, ο οποίος αποτελεί μια τεχνική εξαγωγής χαρακτηριστικών υψηλού επιπέδου βασιζόμενος στην αντιστοίχιση σχημάτων. Κυρίως

χρησιμοποιείται για να γίνει εξαγωγή σχημάτων όπως κύκλοι ευθείες και άλλα. Μεγάλο ενδιαφέρον στην συγκεκριμένη εργασία παρουσίασε η προσέγγιση της ανίχνευσης των φυσαλίδων, βάση της γεωμετρίας και του σχήματος τους. Εκτός αυτού το σύστημα που χρησιμοποίησαν για την εξαγωγή δεδομένων θα μπορούσε να αποτελέσει την πηγή αρκετών υποθαλάσσιων δεδομένων καθώς παρείχε την δυνατότητα λήψης 15 εικόνων ανά δευτερόλεπτο σε βάθος 8 ωρών. Παρότι πιο κλασική μέθοδος στην όραση υπολογιστών, η τεχνική που εφαρμόστηκε από την συγκεκριμένη ομάδα, απέδωσε αρκετά καλά αποτελέσματα για ιδιαίτερα μεγάλο αριθμό δειγμάτων (850.000), γεγονός που αποδεικνύει ότι δεν είναι απαραίτητη η χρήση τεχνικών βαθιάς μάθησης για την επίλυση εφάμιλλων ζητημάτων. Παρόλα αυτά μεθοδολογίες όπως αυτή παρουσιάζουν ζητήματα κυρίως στον χρόνο που απαιτείται για να περατωθεί ο αλγόριθμος.

Συνολικά, η μελέτη της υπάρχουσας βιβλιογραφίας ανέδειξε ότι το ζήτημα της ανίχνευσης των φυσαλίδων έχει αντιμετωπιστεί με διάφορους τρόπους στην όραση υπολογιστών. Οι προσεγγίσεις που σε αυτό το στάδιο έχουν παράξει τα βέλτιστα αποτελέσματα, είναι ακολουθώντας τεχνικές βαθιάς μάθησης και εκπαιδύοντας αλγόριθμους προς ανίχνευση αντικειμένων. Μεγάλη έλλειψη της βιβλιογραφίας ήταν η αποχή σετ δεδομένων in-situ για εκπαίδευση περεταίρω αλγορίθμων. Οι προσεγγίσεις που εφαρμόστηκαν από κάθε ερευνητική ομάδα ήταν αρκετά διαφορετικές τόσο ως προς την μεθοδολογία όσο και ως προς το εύρος των δεδομένων (αλλά στην μέθοδο συλλογής τους). Εξαιτίας αυτού κατέστη δύσκολο να υπάρξει κάποια αντικειμενική σύγκριση ανάμεσα στα αποτελέσματα που προέκυψαν από τις μεθόδους της βιβλιογραφίας ώστε να κριθεί ότι κάποια μεθοδολογία αποτελεί την βέλτιστη. Κλείνοντας, ακολουθώντας το υπάρχον επιστημονικό υπόβαθρο ευελπιστούμε να προσεγγίσουμε εξίσου καλά αποτελέσματα με αυτά που παρατηρήθηκαν και να μπορέσουμε να παρέχουμε σετ δεδομένων ώστε επόμενοι ερευνητές να καταφέρουν να εκπαιδεύσουν δικούς τους αλγορίθμους ευκολότερα και με περισσότερη από την μέχρι στιγμής υπάρχουσα πληροφορία.

Κεφάλαιο 3

Ανάλυση υλισμικού και ηλεκτρονικών στοιχείων

Σε αυτό το κεφάλαιο θα γίνει ανάλυση όλων των υλικών που χρησιμοποιήθηκαν για την υλοποίηση της αυτόνομης υποθαλάσσιας κάμερας. Θα παρατεθούν όλα τα τεχνικά μέρη, οι μεταξύ τους αλληλεπιδράσεις ο τρόπος και η αιτία που επιλέχθηκε το καθένα από αυτά καθώς και διάφορες εναλλακτικές επιλογές που αξιολογήθηκαν αλλά απορρίφθηκαν. Εκτός αυτού θα αναλυθούν ζητήματα που προέκυψαν από τις εν λόγω αλληλεπιδράσεις καθώς και ο τρόπος που αυτά αντιμετωπίστηκαν. Επίσης, θα παρουσιαστούν τα πειράματα που διενεργήθηκαν για την αξιολόγηση των δυνατοτήτων των υλικών που χρησιμοποιήθηκαν αλλά και για την βελτιστοποίηση της λειτουργικής «ζωής» του συστήματος στο υποθαλάσσιο περιβάλλον. Πέραν τούτου θα παρατεθούν τα στοιχεία που αξιολογήθηκαν πριν γίνουν πειραματικές αποστολές σε ελεγχόμενο περιβάλλον (πισίνα). Τέλος, θα γίνει παρουσίαση του τελικού συστήματος και θα αναλυθούν οι διαδικασίες που έγιναν κατά τις αποστολές στο πεδίο.

3.1 Εισαγωγή

Η δημιουργία ενός στιβαρού συστήματος ηλεκτρονικών στοιχείων ήταν ένας από τους βασικότερους στόχους της συγκεκριμένης ερευνητικής εργασίας. Η υλοποίηση του παρείχε πληθώρα δυνατοτήτων, καθιστώντας την πραγματοποίηση αρκετών, σύνθετων πειραμάτων εφικτή, στο απαιτητικό υποθαλάσσιο περιβάλλον. Επιλέχθηκε για το συγκεκριμένο εγχείρημα ως κεντρική υπολογιστική μονάδα, ο μικροϋπολογιστής Raspberry Pi zero 2 W, κυρίως λόγω του μεγέθους αλλά και της μικρής ποσότητας ενέργειας που καταναλώνει η λειτουργία του, σε σύγκριση με άλλους μικροϋπολογιστές (Jetson nano, Raspberry Pi 4 κ.α). Ως κύριο μετρητικό όργανο χρησιμοποιήθηκε η κάμερα Arducam imx477, διότι παρείχε την δυνατότητα, αρκετές από τις βασικές λειτουργίες της κάμερας όπως είναι η εστίαση, το (οπτικό) ζουμ αλλά ακόμα και η κίνηση του αισθητήρα, να γίνονται μέσω τηλεχειρισμού. Για να προστατευθεί το σύστημα από εισροές νερού αλλά και τις μεγάλες πιέσεις που ασκούνται σε μεγάλα βάθη υλοποιήθηκε κατάλληλο υποβρύχιο περίβλημα (housing). Σημαντική απόφαση σε αυτό το στάδιο αποτέλεσε η επιλογή της γεωμετρίας του παράθυρου θέασης στο πρόσθιο τμήμα του υποβρύχιου περιβλήματος, το οποίο επιλέχθηκε να είναι θόλος ενώ θα μπορούσε να είναι και επίπεδο. Πέρα από αυτά τα

βασικά στοιχεία που αναφέρθηκαν, το υλισμικό σκέλος της εργασίας αποτελούνταν ακόμα από ένα σύνολο στοιχείων που χρησιμοποιήθηκαν για τροφοδοσία, ελέγχους, αυτοματισμούς, διασύνδεση και περισυλλογή δεδομένων του συστήματος. Σε αυτό το κεφάλαιο θα γίνει ανάλυση και αιτιολόγηση των στοιχείων του υλιστικού αλλά και των αποφάσεων που λήφθηκαν για την υλοποίηση του τελικού συστήματος.

Το λογισμικό που υλοποιήθηκε για τις απαραίτητες λειτουργίες του συστήματος αναφοράς αποτέλεσε σημαντικό σκέλος των εργασιών που έγιναν στο παρόν κεφάλαιο καθώς στα υποθαλάσσια συστήματα δεν υπάρχει η δυνατότητα διορθωτικών δράσεων αφού ολοκληρωθεί η τοποθέτηση τους. Για αυτό και όλα τα τμήματα έπρεπε να λειτουργούν με αυτοματισμούς και τα δεδομένα που εξάγονται να αποθηκεύονται χωρίς να χαθεί καθόλου πληροφορία, καθώς η εκ των υστέρων επίλυση ζητημάτων είναι αρκετά δύσκολη.

Για να γίνει έλεγχος ότι το σύστημα λειτουργούσε όπως ήταν σχεδιασμένο, έγινε ένα σύνολο πειραμάτων τόσο εκτός θαλάσσιου περιβάλλοντος όσο και σε ελεγχόμενο υδάτινο περιβάλλον που προσομοίαζε το θαλάσσιο (εκτός από τις πιέσεις, την αλατότητα και την οξύτητα). Τα δεδομένα που εξάχθηκαν από αυτά τα πειράματα καθώς και η μεθοδολογία υλοποίησής τους παρουσιάζονται στην συνέχεια αυτού του κεφαλαίου. Όλη η εμπειρία που αποκτήθηκε από τα εφαρμοσμένα πειράματα χρησιμοποιήθηκε κατά τις αποστολές που έγιναν, για την εξαγωγή των δεδομένων που παρουσιάζονται σε επόμενο κεφάλαιο.

Η δομή του κεφαλαίου είναι η ακόλουθη. Στο Κεφάλαιο 3.2 γίνεται παρουσίαση όλων των τεχνικών στοιχείων για την υλοποίηση του συστήματος αυτόνομης θαλάσσιας κάμερας. Στην συνέχεια στο Κεφάλαιο 3.3 γίνεται ανάλυση των αλληλεπιδράσεων των στοιχείων, δηλαδή πως από μεμονωμένα στοιχεία έγιναν ενιαίο σύστημα, καθώς και παρουσίαση του λογισμικού που χρησιμοποιήθηκε για την λειτουργία του συστήματος. Ακολούθως στο Κεφάλαιο 3.4 παρουσιάζονται τα πειράματα που έγιναν και ο τρόπος που αυτά προγραμματίστηκαν. Τελικά, στο κεφάλαιο 3.5 παρουσιάζονται οι διαδικασίες που έγιναν στις αποστολές και αναλύεται η πιο ολοκληρωμένη από αυτές.

3.2 Μεμονωμένη ανάλυση τεχνικών στοιχείων

3.2.1 Επιλογή μετρητικού οργάνου

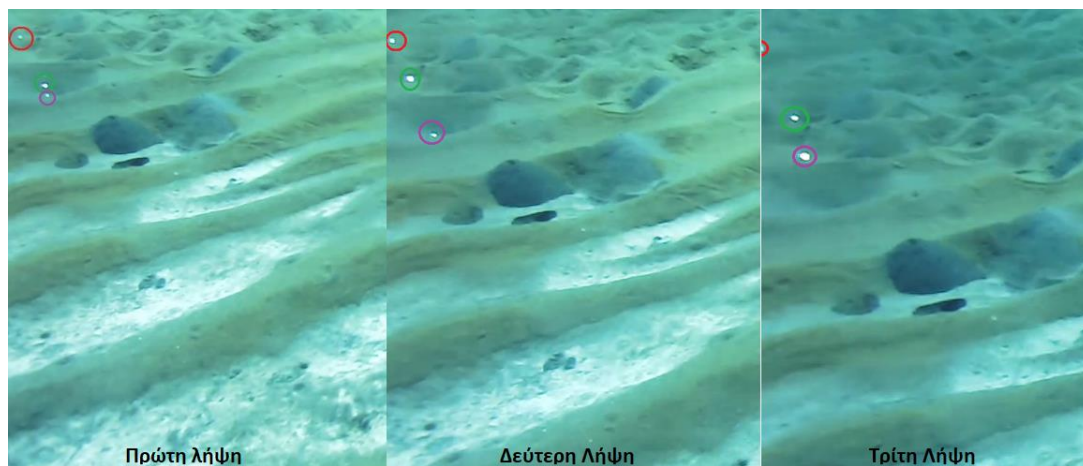
Για κάθε σύστημα που κατασκευάζεται με σκοπό την εξαγωγή δεδομένων, η κρισιμότερη απόφαση αφορά την επιλογή του μετρητικού οργάνου. Στην περίπτωση της παρούσας εργασίας, καθότι ο στόχος ήταν η εξαγωγή εικόνων, το όργανο που επιλέχθηκε ήταν κάμερα. Για να γίνει η συγκεκριμένη επιλογή έπρεπε να τεθούν κάποιες παράμετροι ώστε να βρεθεί η βέλτιστη δυνατή. Αρχικά, απαιτήθηκε η εξέταση του υπό μελέτη φαινομένου, το οποίο ήταν η ανίχνευση υποθαλάσσιων αερίων, μέσα από τις φυσαλίδες που αυτά προκαλούν. Μετά από παρακολούθηση βίντεο που καταγράφηκαν για την αξιολόγηση του φαινομένου, παρατηρήθηκε ότι η ταχύτητα των φυσαλίδων δεν ήταν μεγάλη (η μέθοδος παρατήρησης του φαινομένου παρουσιάζεται στην συνέχεια), έτσι δεν υπήρξε ανάγκη για εξαγωγή πολλών λήψεων ανά δευτερόλεπτο. Το γεγονός αυτό «έλυσε» πολλά ζητήματα μιας και κατέστη δυνατή η επιλογή μικροϋπολογιστή με σχετικά μικρές υπολογιστικές δυνατότητες.

Το πιο σημαντικό ζήτημα που εξετάστηκε ήταν η αυτονομία του συστήματος στο θαλάσσιο περιβάλλον. Για να επιτευχθεί αυτό έπρεπε αφού γινόταν η εναπόθεση του συστήματος, η κάμερα να παρείχε την δυνατότητα κάποιες από τις λειτουργίες της να γίνονται με τηλεχειρισμό, μιας και το όλο σύστημα θα ήταν δυσπρόσιτο. Οι λειτουργίες που είχαν την μεγαλύτερη σημασία ήταν η εστίαση και το οπτικό ζουμ (μεταβολή της εστιακής απόστασης f). Η παροχή των συγκεκριμένων λειτουργιών ήταν ζωτικής σημασίας ώστε το σύστημα να παρείχε ευκρινή εικόνα μιας και η εστίαση στο υποθαλάσσιο περιβάλλον είναι ιδιαίτερα σύνθετη. Η αιτία που δυσχεραίνει την πραγματοποίηση ορθής εστίασης έγκειται στην τοποθέτηση της κάμερα εντός του προστατευτικού περιβλήματος με αποτέλεσμα η διαδικασία της εστίασης να είναι αδύνατον να υλοποιηθεί χειροκίνητα όπως συμβαίνει με μια τυπική κάμερα. Εκτός αυτού το φως εξασθενεί εντός του νερού (λόγω σκέδασης και απορρόφησης), συνεπώς έπρεπε να τοποθετηθούν ΦΠ που ανάλογα την σκηνή επηρέαζαν αισθητά την εστίαση. Ένας ακόμα παράγοντας που επηρέασε την διαδικασία ήταν η σύνθετη γεωμετρία που προέκυψε εξαιτίας του παραθύρου θέασης. Εκτός από τις προαναφερθείσες λειτουργίες, ζητούμενο ήταν και η δυνατότητα κίνησης της κάμερας σε δύο άξονες, ώστε με μικρομετακινήσεις σε αυτούς να επιτευχθεί το υπό μελέτη φαινόμενο στην βέλτιστη δυνατή θέση εντός του κάδρου θέασης.

Προκειμένου να διασφαλιστεί ότι το φαινόμενο θα καταγραφόταν από τον αισθητήρα, χρησιμοποιήθηκαν παλαιότερα βίντεο για την αξιολόγηση του (που προέκυψαν από τηλεχειριζόμενο υποβρύχιο όχημα). Γνωρίζοντας ότι η κάμερα του συγκεκριμένου οχήματος αντλούσε τριάντα εικόνες ανά δευτερόλεπτο επιλέχθηκε να γίνει μια δειγματοληψία τριών εικόνων με χρονική απόσταση 333 millisecond (άρα 3 λήψεις σε ένα δευτερόλεπτο). Αφού ελέγξαμε ότι το φαινόμενο μπορούσε να παρατηρηθεί ικανοποιητικά με αυτές τις παραμέτρους, εξάχθηκε το συμπέρασμα ότι η κάμερα που θα έπρεπε να επιλεγεί αρκούσε να παρείχε την δυνατότητα λήψης τριών εικόνων ανά δευτερόλεπτο. Σε αυτό το σημείο υπεισέρχεται ένας ακόμα παράγοντας που είναι η απόσταση από τον αισθητήρα. Γεωμετρικά ισχύει ότι το μέγεθος της X,Y κάλυψης (οι οποίες αποτελούν το μέγεθος της σκηνής στο πραγματικό κόσμο) προκύπτουν από την εστιακή απόσταση, το μέγεθος του αισθητήρα αλλά και την απόσταση του αντικειμένου από τον αισθητήρα. Ο τύπος υπολογισμού είναι της μορφής:

$$\text{κάλυψη} = \frac{\text{απόσταση αντικειμένου (m)} \times \text{μέγεθος αισθητήρα (mm)}}{\text{εστιακή απόσταση (mm)}}$$

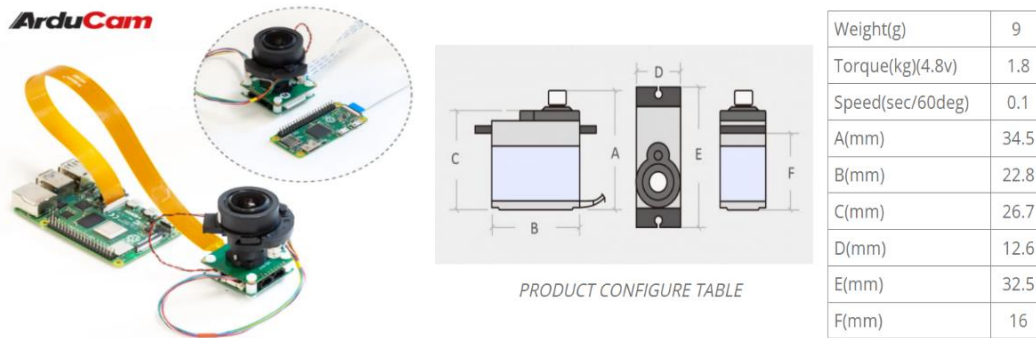
Από τον συγκεκριμένο τύπο (ο οποίος αφορά λήψεις στο αέριο περιβάλλον) έγινε αντιληπτό ότι όσο πιο κοντά βρίσκεται το αντικείμενο τόσο πιο «μικρή» είναι η σκηνή. Αυτό είχε σημασία διότι επιλέχθηκε στις εικόνες που αξιολογήθηκαν (Σχήμα 3.1) οι φυσαλίδες να βρίσκονται κοντά στον αισθητήρα (αποστάσεις μικρότερες από δύο μέτρα), ώστε η απόσταση που έπρεπε να διανύσουν για να «βγουν» από το κάδρο να είναι μικρότερες ή ίσες με την απόσταση που θα διανύουν οι φυσαλίδες κατά την συλλογή δεδομένων στο πεδίο (περί των δύο μέτρων). Όπως φαίνεται στο Σχήμα 3.1 οι φυσαλίδες βρίσκονται εντός του κάδρου και στις τρεις λήψεις (εντός κύκλων ίδιου χρώματος παρουσιάζονται οι φυσαλίδες στις διαφορετικές χρονικές στιγμές). Ο λόγος που επιλέχθηκαν οι συγκεκριμένες λήψεις και όχι άλλες με μεγαλύτερο αριθμό φυσαλίδων, συνέβη καθότι σε περιπτώσεις πυκνών δειγμάτων ήταν αδύνατον να αναγνωριστεί η κίνηση τους λόγω των δυνατών ρευμάτων αλλά και των μεταβολών στην γεωμετρία τους.



Σχήμα 3.1: Εικόνες που χρησιμοποιήθηκαν για έλεγχο της ταχύτητας ανόδου των φυσαλίδων (σε κάθε λήψη προς τα δεξιά γίνεται μεγέθυνση για να είναι πιο εύκολη η παρατήρηση).

Τελικά, αναλύοντας όλα τα στοιχεία που παρατέθηκαν, η κάμερα που επιλέχθηκε ήταν η Arducam 12MP IMX477 Pan Tilt Zoom (Σχήμα 3.2). Η συγκεκριμένη κάμερα βάσει του κατασκευαστή της [9] ήταν συμβατή με τους μικροϋπολογιστές που αξιολογήθηκαν (Raspberry Pi και Jetson Nano), εκτός αυτού παρείχε την δυνατότητα η εστίαση και το ζουμ να γίνονται με τηλεχειρισμό, χάρη σε μικροκινητήρες. Μεγάλο πλεονέκτημα όσον αφορά την εστίαση και το οπτικό ζουμ προέκυψε από την ευαισθησία των κινητήρων (stepper motors), όπου για να φτάσει από την μέγιστη στην ελάχιστη εστιακή απόσταση ο κινητήρας έπρεπε να κάνει 2317 μικροκινήσεις ενώ για την εστίαση 3009, και έτσι παρείχε την δυνατότητα για επίτευξη λεπτομερών αποτελεσμάτων. Οι εστιακές αποστάσεις που μπορούσαν να επιτευχθούν από το συγκεκριμένο μοντέλο ήταν 4.38-10.71 mm \pm 5%. Ένα μικρό ζήτημα που προέκυψε αφορούσε το κλείστρο της μηχανής, το οποίο ήταν κυλιόμενο (rolling shutter), καθώς δεν ενδείκνυται για κινούμενους στόχους μεγάλων ταχυτήτων και για μη στατικές κάμερες, διότι εξαιτίας του δημιουργούνται παραμορφώσεις. Στην δικιά μας εφαρμογή οι ταχύτητες των φυσαλίδων ήταν αρκετά μικρές και η κάμερα τοποθετήθηκε σε σταθερή θέση, ώστε να αποφευχθούν πιθανές παραμορφώσεις.

Όσον αφορά τον αισθητήρα, οι φυσικές του διαστάσεις ήταν 36 mm \times 36 mm ενώ τα ενεργά εικονοστοιχεία 4056 (οριζόντια) \times 3040 (κάθετα) επομένως ήταν 12.3 MP. Ένα ακόμα τεχνικό στοιχείο της κάμερα, αποτέλεσε το οπτικό πεδίο (FOV) το οποίο ήταν 96° – 33° (οριζόντια). Για θαλάσσιες εφαρμογές συνήθως χρησιμοποιούνται κάμερες με οπτικά πεδία όσο το δυνατόν μεγαλύτερα, διότι λόγω της εναλλαγής των μέσων (νερό σε αέρα) αλλά και της γεωμετρίας του παραθύρου θέασης του υποβρυχίου περιβλήματος, το οπτικό πεδίο περιορίζεται (για αυτό για θαλάσσια φωτογραφία επιλέγονται υπερευρυγώνιοι φακοί ή fisheye). Κλείνοντας, μεγάλο ενδιαφέρον παρουσίασε η δυνατότητα εγκατάστασης δύο σερβοκινητήρων (Σχήμα 3.2), με χρήση των οποίων κατέστη εφικτή η κίνηση της κάμερας σε δύο άξονες. Έτσι, η σκηνή μεταβαλλόταν χωρίς να χρειαστεί επανατοποθέτηση του συστήματος. Έτσι, επιτεύχθηκε το φαινόμενο παρατήρησης να παρουσιάζεται στο κέντρο του κάδρου, δίνοντας αυξημένη αυτονομία στο σταθμό που κατασκευάστηκε.



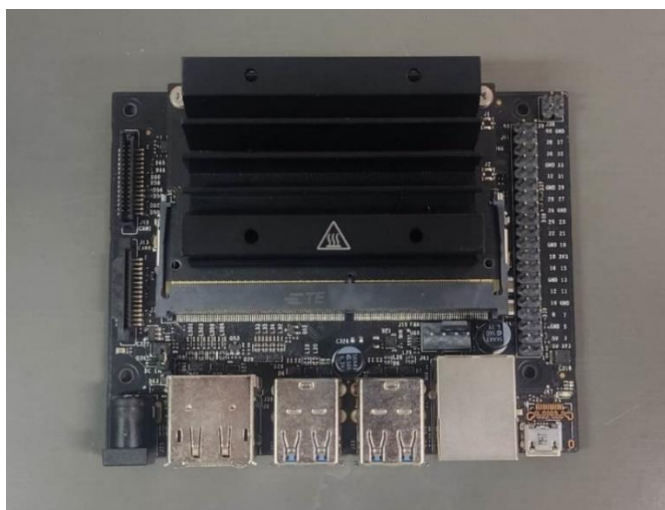
Σχήμα 3.2: Αριστερά κάμερα Arducam 12MP IMX477, πηγή : [9]. Δεξιά: μοντέλο σερβοκινητήρα που επιλέχθηκε με τα βασικά χαρακτηριστικά του, Πηγή: <https://www.towerpro.com.tw/product/sg90-7/>

3.2.2 Μικροϋπολογιστές και αρχές λειτουργίας

Η επιλογή του μικροϋπολογιστή αποτέλεσε ένα από τα σύνθετα ζητήματα της συγκεκριμένης εργασίας, διότι χρησιμοποιήθηκε ως κεντρική μονάδα διαχείρισης των περιφερειακών οργάνων που επιλέχθηκαν. Αφού όλο το σύστημα εδραζόταν στο υποθαλάσσιο περιβάλλον παρουσιάστηκαν κάποια βασικά ζητήματα που έπρεπε να επιλυθούν. Αρχικά, το σύστημα όφειλε να λειτουργεί αυτόνομα για μεγάλα χρονικά διαστήματα με χρήση μπαταρίας για την κάλυψη των ενεργειακών αναγκών του. Αυτό οδήγησε στην ανάγκη εύρεσης μικροϋπολογιστή με ικανή υπολογιστική ισχύ για να «υποστηρίξει» το απαραίτητο λογισμικό των αισθητήρων, ενώ ταυτόχρονα με την μικρότερη δυνατή ενεργειακή κατανάλωση. Εκτός αυτού, το μέγεθος αποτελεί πολύ σημαντική μεταβλητή σε υποθαλάσσιες εργασίες μιας και μεγαλύτερο μέγεθος οδηγεί σε εκθετική αύξηση του κόστους κατασκευής του ΥΠ, ενώ ταυτόχρονα δυσχεραίνει την μεταφορά και τοποθέτηση του συστήματος. Ένα ακόμα βασικό ζήτημα αφορούσε την αποθήκευση των δεδομένων που θα εξάγονταν από τους αισθητήρες. Έτσι η υπολογιστική μονάδα έπρεπε να έχει την δυνατότητα μεταφοράς δεδομένων άμεσα σε εξωτερικό σκληρό ώστε να διασφαλιστεί η αυτονομία του συστήματος. Τέλος, το κόστος της κεντρικής μονάδας αποτέλεσε σημαντικό παράγοντα, μιας και ένας από τους στόχους της συγκεκριμένης εργασίας ήταν το κόστος της υλοποίησης να μην είναι υπέρογκο (χωρίς όμως να μειωθούν οι δυνατότητες του) ώστε η κατασκευή του να είναι προσιτή σε μεγάλο ερευνητικό κοινό.

Για την τελική επιλογή αξιολογήθηκαν τρία μοντέλα μικροϋπολογιστών με αρκετά διαφορετικά χαρακτηριστικά το καθένα. Το Jetson Nano Developer Kit (σχήμα 3.3) το οποίο αποτέλεσε την επιλογή με την μεγαλύτερη «υπολογιστική ισχύ». Έτσι, παρείχε την δυνατότητα υλοποίησης αρκετά απαιτητικών προγραμμάτων, όπως και την χρήση απλών νευρωνικών δικτύων, χάρη στον ισχυρό επεξεργαστή και την κάρτα γραφικών

με τα οποία ήταν εφοδιασμένο. Μεγάλο πλεονέκτημα του αποτέλεσαν οι τέσσερις θήρες USB 3.0, οι οποίες καθιστούσαν δυνατή τη μεταφορά δεδομένων με ταχύτητες της τάξης των 400-480 MB/sec. Παρόλα αυτά ο συνδυασμός των μεγάλων, συγκριτικά με τους άλλους μικροϋπολογιστές διαστάσεων, η απαγορευτική κατανάλωση ενέργειας (Πίνακας 3.1) αλλά και το υψηλό του κόστος κατέστησαν την συγκεκριμένη επιλογή υποδεέστερη από τις υπόλοιπες. Το συγκεκριμένο μοντέλο μικροϋπολογιστή αποτελεί ιδανική επιλογή σε περίπτωση ανάγκης άμεσης (on the spot) επεξεργασίας των δεδομένων, δίνοντας την δυνατότητα στο σύστημα να ανταποκρίνεται (responsive) σε εξωτερικά ερεθίσματα. Μερικά παραδείγματα θα ήταν να κινούνται οι σερβοκινητήρες σε περίπτωση ανίχνευσης κίνησης, ή να ενεργοποιούνται τα φώτα με κάποιο ερέθισμα. Τέτοιου τύπου εφαρμογές θα είχαν πολύ ενδιαφέρον σε μια μελλοντική προσέγγιση του ζητήματος, παρόλα αυτά σε αυτό το στάδιο τα βασικά κριτήρια της επιλογής ήταν διαφορετικά, γεγονός που μας οδήγησε στην απόρριψη του συγκεκριμένου υπολογιστικού συστήματος.



Σχήμα 3.3: Μικροϋπολογιστής Jetson Nano Developer Kit , Πηγή : [10]

Ένας ακόμα μικροϋπολογιστής που αξιολογήθηκε ήταν ο Raspberry Pi 4 Model B 8GB (Σχήμα 3.4). Το συγκεκριμένο μοντέλο αποτέλεσε οικονομικότερη εναλλακτική σε σχέση με το Jetson Nano Developer Kit, χωρίς όμως να παρέχει εφάμιλλή υπολογιστική ισχύ. Τα βασικά μειονεκτήματα του ήταν η έλλειψη κάρτας γραφικών με αποτέλεσμα να μην μπορούν να υλοποιηθούν βασικά νευρωνικά δίκτυα, καθώς και το μεγάλο σε σχέση με άλλα μοντέλα κόστος του. Όσον αφορά τις φυσικές του διαστάσεις, δεν απείχαν ιδιαίτερα από του Jetson Nano. Η σημαντικότερη διαφορά τους παρατηρήθηκε στην κατανάλωση ενέργειας όπου το Raspberry Pi βάσει του κατασκευαστή καταναλώνει σχεδόν 40% λιγότερη (6.4 έναντι 10 Watt) σε πλήρη λειτουργία. Η διαχείριση ενέργειας αποτελεί ίσως τον πιο σημαντικό παράγοντα σε υποθαλάσσιες εργασίες, για αυτό και ένας από τους βασικότερους στόχους που

θεσπίσαμε, ήταν η επέκταση της διάρκειας ζωής του συστήματος χωρίς να δέχεται υποστήριξη από εξωτερικούς παράγοντες (νέες μπαταρίες, αλλαγές στο λογισμικό κ.α.). Λαμβάνοντας υπόψιν αυτά τα στοιχεία το Raspberry Pi 4 Model B 8GB αποτελεί ικανοποιητική επιλογή σε περίπτωση που υπάρχει ανάγκη άμεσης αλλά όχι ιδιαίτερα σύνθετης επεξεργασίας των δεδομένων από το σύστημα. Ενώ ταυτόχρονα παρέχει «διαχειρίσιμη» ενεργειακή κατανάλωση συγκριτικά με Jetson Nano Developer Kit. Παρόλα αυτά για την συγκεκριμένη εργασία ο συνδυασμός των μεγάλων διαστάσεων και των υψηλών ενεργειακών απαιτήσεων, ιδιαίτως σε σχέση με το Raspberry Pi Zero 2 W που αποτέλεσε τον τελικό μικροϋπολογιστή που αξιολογήθηκε, οδήγησαν στην απόρριψη της συγκεκριμένης επιλογής.



Σχήμα 3.4: Μικροϋπολογιστής Raspberry Pi 4 Model B 8GB, πηγή : [11]

Λαμβάνοντας υπόψιν την ερευνητική εργασία [1], η οποία παρείχε αναλυτικές πληροφορίες για την υλοποίηση υποθαλάσσιου συστήματος με μετρητικό όργανο κάμερα, αξιολογήθηκε ο μικροϋπολογιστής Raspberry Pi Zero 2 W (Σχήμα 3.5). Τα πλεονεκτήματα του συγκεκριμένου συστήματος όπως παρουσιάστηκαν και από την ερευνητική ομάδα της [1] ήταν πολλά. Αρχικά, η ενεργειακή κατανάλωση του μικροϋπολογιστή ήταν πολύ μικρότερη συγκριτικά με τις υπόλοιπες πιθανές επιλογές (κατά τουλάχιστον 70%) και ταυτόχρονα το μέγεθος του ήταν αισθητά μικρότερο, γεγονός που εξοικονόμησε αρκετούς πόρους από την κατασκευή του ΥΠ. Εξαιτίας της συγκεκριμένης επιλογής απελευθερώθηκε χώρος για τα υπόλοιπα ηλεκτρονικά στοιχεία αλλά και για τις πηγές ενέργειας, το μέγεθος των οποίων καθόρισε και την ποσότητα ενέργειας που θα μπορούσαν να παρέχουν. Για πιο χρονοβόρα πειράματα αλλά και τις αποστολές, οι πηγές ενέργειας τοποθετήθηκαν σε διαφορετικό υποβρύχιο περίβλημα. Όσον αφορά το κόστος του συγκεκριμένου μοντέλου ήταν μηδαμινό σε σχέση με τις άλλες επιλογές που αξιολογήθηκαν. Τα βασικά μειονέκτημα του συγκεκριμένου μικροϋπολογιστή αφορούσαν την έλλειψη υπολογιστικής ισχύος, καθώς και την αδυναμία μεταφοράς δεδομένων με μεγάλες ταχύτητες σε εξωτερικό

σκληρό, διότι δεν παρείχε θύρα USB 3.0. Οι δυνατότητες της θύρας micro USB (η οποία αποτέλεσε την μοναδική δίοδο για μεταφοράς δεδομένων) ήταν αρκετά περιορισμένες, ενδεικτικά η ταχύτητα μεταφοράς δεδομένων γινόταν με ταχύτητες σχεδόν δέκα φορές μικρότερες από ότι με την θύρα USB 3.0. Παρόλα αυτά για τις ανάγκες της συγκεκριμένης εργασίας στόχος ήταν η εξαγωγή δεδομένων μικρού μεγέθους από το σύστημα, χωρίς την άμεση επεξεργασία τους. Συνεπώς, λαμβάνοντας υπόψιν τα προαναφερθέντα στοιχεία, το Raspberry Pi Zero 2 W ικανοποιούσε πλήρως τα κριτήρια κόστους, μεγέθους και κατανάλωσης ενέργειας που είχαν τεθεί. Το τελευταίο ζήτημα που αξιολογήθηκε αφορούσε τη δυνατότητα του συγκεκριμένου υπολογιστή να υποστηρίξει την κάμερα που επιλέχθηκε. Ως προς το συγκεκριμένο ζήτημα, το Raspberry Pi Zero 2 αποδείχθηκε συμβατό με το μετρητικό όργανο. Έτσι, κρίθηκε ότι η επιλογή του συγκεκριμένου υπολογιστή ήταν η βέλτιστη.



Σχήμα 3.5: Μικροϋπολογιστής Raspberry Pi Zero 2 W, Πηγή : [11]

| Τεχνικά χαρακτηριστικά | Μοντέλα Μικροϋπολογιστών | | |
|------------------------|------------------------------------|---|--|
| | Jetson Nano Developer Kit | Raspberry Pi 4 Model B 8GB | Raspberry Pi Zero 2 W |
| Βασικά | | | |
| Διαστάσεις -mm- | 100x79 | 85x56 | 65x30 |
| Ισχύς -Watt- | 5 - 10 | 2.7 - 6.4 | 0.6 – 1.4 |
| Κόστος -€- | 149 | 98.90 | 24.90 |
| Επεξεργασίας | | | |
| CPU | 64-bit Quad-core ARM A57 @ 1.43GHz | Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz | Broadcom BCM2710A1, 1GHz quad-core 64-bit Arm Cortex-A53 |

| | | | |
|-------------------------|--|----------------------------------|----------------|
| GPU | 128-core NVIDIA Maxwell @ 921MHz | - | - |
| Κωδικοποιητής Βίντεο | 4Kp30 (4x) 1080p30 (2x) 1080p60 | H264 1080p30 | H264 1080p30 |
| Αποκωδικοποιητής Βίντεο | 4Kp60 (2x) 4Kp30 (8x) 1080p30 (4x) 1080p60 | H.265 4Kp60 decode, H264 1080p60 | H264 1080p30 |
| Διεπαφές | | | |
| USB | 4x USB 3.0 A (Host) USB 2.0 Micro B (Device) | 2x USB 3.0 2x USB 2.0 | Micro USB port |
| Θύρα κάμερας | MIPI CSI-2 x2 | 2-lane MIPI CSI camera port | CSI-2 camera |

Πίνακας 3.1: Τεχνικά χαρακτηριστικά υπό αξιολόγηση μικροϋπολογιστών. Πηγές : [10], [11]

3.2.3 Επιλογή ηλεκτρονικών στοιχείων

Η επιλογή των επιμέρους ηλεκτρονικών στοιχείων αποτέλεσε ένα από τα βασικότερα σκέλη της κατασκευής του συστήματος, καθώς στόχος μας σε αυτό το στάδιο ήταν η ενίσχυση της αυτονομίας του συστήματος με δύο βασικούς τρόπους. Αρχικά, με την παρακολούθηση σημαντικών στοιχείων/πληροφοριών του υποθαλάσσιου σταθμού, όπως η κατανάλωση ενέργειας, η τάση της μπαταρίας, οι χρονικές στιγμές που ανοίγει και κλείνει το σύστημα και άλλα. Εκτός αυτού για να αυξηθεί η αυτονομία του σταθμού κρίθηκε αναγκαία η εγκατάσταση ενός συστήματος που να παρέχει την δυνατότητα να ενεργοποιεί και να απενεργοποιεί τον μικροϋπολογιστή και το μετρητικό όργανο βάσει προγράμματος. Η μεγάλη συνεισφορά αυτής της λειτουργίας ήταν ότι κατέστησε εφικτή την εξαγωγή δεδομένων συγκεκριμένες ώρες της ημέρας, δίνοντας την δυνατότητα παρατήρησης φαινομένων με χρονική περιοδικότητα χωρίς την καταγραφή άχρηστης πληροφορίας. Εκτός αυτού χάρις σε αυτό το σύστημα υπήρξε καλύτερη διαχείριση της διάρκειας ζωής της μπαταρίας.

Για να επιλυθούν τα ζητήματα που αναλύθηκαν επιλέχθηκε το Witty Pi 3 REV2 (Σχήμα 3.6) το οποίο βάσει του κατασκευαστή χρησιμοποιούσε ρολόι πραγματικού χρόνου (RTC), το οποίο είχε αξία διότι στο υποθαλάσσιο περιβάλλον δεν υπήρχε σύνδεση Wifi για έλεγχο της ώρας του μικροϋπολογιστή. Εκτός αυτού παρείχε την δυνατότητα να προγραμματιστούν οι χρονικές στιγμές που το σύστημα θα βρισκόταν σε λειτουργία. Ταυτόχρονα, χάρη στο συγκεκριμένο σύστημα κατέστη δυνατή η δημιουργία αρχείων, με πληθώρα, μεταδεδομένων όπως παρουσιάζεται και στο Σχήμα

3.7. Τα συγκεκριμένα δεδομένα σε συνδυασμό με αυτά που παρήγαγε ο αισθητήρας μέτρησης ρεύματος και τάσης (που θα παρουσιαστεί στην συνέχεια) καταγράφηκαν σε αρχεία μεταδεδομένων (log.files), ώστε μετά το πέρας του πειράματος να μπορεί να αξιολογηθεί η συμπεριφορά του συστήματος, ενώ σε περίπτωση κάποια βλάβης να βρεθεί η χρονική στιγμή αλλά και η αιτία της ώστε σε μετέπειτα πειράματα να διορθωθεί. Το Witty Pi 3 REV2 ήταν διαστάσεων ελάχιστα μεγαλύτερων από τον μικροϋπολογιστή που επιλέχθηκε με αποτέλεσμα να μην καταλάμβανε υπέρογκο χώρο εντός του ΥΠ.

Ο αισθητήρας Power Sense Module (PSM) της Blue Robotics (Σχήμα 3.6) αποτέλεσε το πιο σημαντικό εργαλείο για την διασφάλιση της ζωής των μπαταριών λιθίου που χρησιμοποιήθηκαν μιας και ελέγχοντας την τάση (V) μπορέσαμε να εξασφαλίσουμε ότι η μπαταρία δεν θα «πέσει» χαμηλότερα από κάποιο προκαθορισμένο επίπεδο τάσης. Εφόσον συνέβαινε αυτό οι μπαταρίες δεν θα μπορούσαν να επαναφορτιστούν (θα γίνει σχολιασμός και αξιολόγηση σε επόμενο κεφάλαιο), το κόστος των οποίων ήταν αρκετά υψηλό. Εκτός αυτού η εύρεση της απαιτούμενης ισχύς (Watt) παρείχε την δυνατότητα υπολογισμού του χρόνου του πειράματος εκ των προτέρων γνωρίζοντας και την χωρητικότητα των μπαταριών. Έτσι, γνωρίζοντας τις ώρες λειτουργίας του συστήματος, έως ότου η μπαταρία να φτάσει στο πέρας της, κατέστη δυνατό με χρήση του Witty Pi 3 REV2 να κατασκευαστεί το ιδανικό πρόγραμμα λειτουργίας για την παρακολούθηση του φαινομένου. Επίσης, μαζί με τις μετρήσεις που κάνει το Power Sense Module έγινε καταγραφή της μέτρησης θερμοκρασίας που παρέχει ο μικροϋπολογιστής, ώστε να εξασφαλιστεί η ασφάλεια του. Σε πολύ μεγάλες θερμοκρασίες το σύστημα μπορούσε να υποστεί ανεπανόρθωτες ζημιές. Σε επόμενο κεφάλαιο θα γίνει ανάλυση των υπολογισμών που εφαρμόστηκαν τις τελικές αποστολές.



Σχήμα 3.6: Αριστερά: Witty Pi 3 REV2. Δεξιά: ο αισθητήρας Power Sense Module. Πηγή : <https://bluerobotics.com/>

| | |
|--|--|
| <pre>[2023-05-17 15:20:16] Done :-) [2023-05-17 15:20:23] Pending for incoming shutdown command... [2023-05-17 15:20:23] File "schedule.wpi" not found, skip running schedule script. [2023-05-17 15:49:40] Copying "plan.wpi" to "schedule.wpi"... [2023-05-17 15:49:40] Running the script... [2023-05-17 15:49:40] Schedule next shutdown at: 2023-05-17 16:20:00 [2023-05-17 15:49:41] Schedule next startup at: 2023-05-17 16:12:00 [2023-05-17 15:49:41] Done :-) [2023-05-17 15:54:31] Witty Pi daemon (v3.51) is started. [2023-05-17 15:54:31] Running on Raspberry Pi Zero 2 W Rev 1.0 [2023-05-17 15:54:34] Firmware ID: 0x24 [2023-05-17 15:54:34] Current Vin=13.99, Vout=4.99, Iout=0.75 [2023-05-17 15:54:34] Send out the SYS_UP signal via GPIO-17 pin. [2023-05-17 15:54:34] System startup as scheduled. [2023-05-17 15:54:34] Synchronizing time between system and Witty Pi... [2023-05-17 15:54:34] Writing RTC time to system... [2023-05-17 16:12:20] Done :-)</pre> | <pre>2023-05-17 18:23:29,165 - Voltage: 13.829750 2023-05-17 18:23:29,166 - Current: 0.075758 2023-05-17 18:23:29,167 - CPU_Temp: 33.248000 2023-05-17 18:23:29,167 - 2023-05-17 18:23:59,494 - Voltage: 13.839375 2023-05-17 18:23:59,597 - Current: 0.018939 2023-05-17 18:23:59,598 - CPU_Temp: 32.710000 2023-05-17 18:23:59,599 - 2023-05-17 18:24:29,651 - Voltage: 13.842125 2023-05-17 18:24:29,652 - Current: 0.000000 2023-05-17 18:24:29,653 - CPU_Temp: 32.172000 2023-05-17 18:24:29,653 - 2023-05-17 18:24:59,705 - Voltage: 13.842125 2023-05-17 18:24:59,706 - Current: 0.009470 2023-05-17 18:24:59,707 - CPU_Temp: 32.710000</pre> |
|--|--|

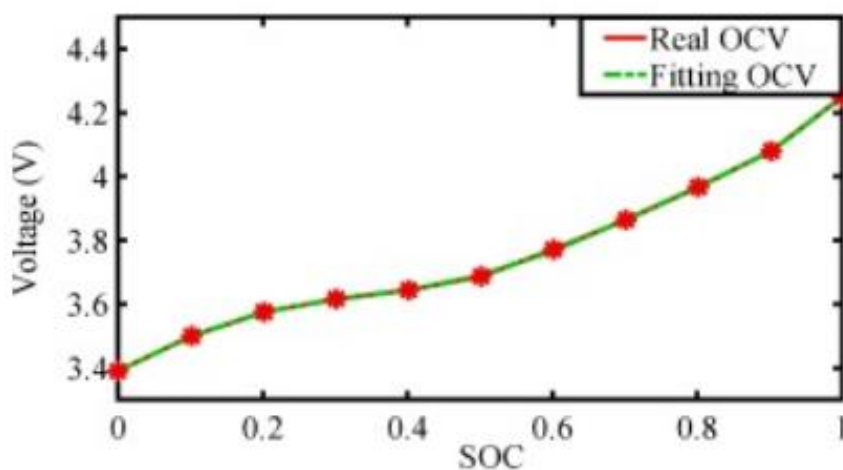
Σχήμα 3.7: Αριστερά: αρχείο μεταδεδομένων που προκύπτει από το Witty Pi 3 REV2. Δεξιά: αρχείο μεταδεδομένων του αισθητήρα Power Sense Module

3.2.4 Περιφερειακά στοιχεία

Το υποθαλάσσιο σύστημα παρουσίασε ανάγκη μεγάλου αριθμού περιφερειακών στοιχείων για την λειτουργία του. Αρχικά, χρειάστηκαν ΠΕ (μπαταρίες) οι οποίες μπορούσαν να παρέχουν αυτονομία για μεγάλα χρονικά διαστήματα. Εκτός αυτού χρειάστηκε ΦΠ λόγω της εξασθένησης του φωτός στο θαλάσσιο περιβάλλον, μιας και στόχος του συστήματος ήταν να έχει την δυνατότητα καταγραφής σε ποίκιλα βάθη. Σημαντική ήταν η επιλογή σκληρού δίσκου για την αποθήκευση δεδομένων, ώστε να αποφευχθεί η επιβάρυνση της SD κάρτας, εντός της οποίας εδραζόταν το λειτουργικό του μικροϋπολογιστή καθώς και οι κώδικες για την λειτουργία όλων των επιμέρους οργάνων.

Όσον αφορά τις μπαταρίες επιλέχθηκε να είναι επαναφορτιζόμενες, ώστε να μπορούν επαναχρησιμοποιηθούν, μειώνοντας το κόστος των επόμενων πειραμάτων. Οι ευρύτερα χρησιμοποιούμενες επαναφορτιζόμενες μπαταρίες [12], [13] είναι οι Ni-Cd, Ni-MH, μόλυβδου-οξέος, ιόντων και πολυμερών λίθιου. Για την συγκεκριμένη εργασία επιλέχθηκαν μπαταρίες πολυμερών/ιόντων (στα αρχικά στάδια χρησιμοποιήθηκαν πολυμερών ενώ στα τελικά πειράματα ιόντων) λιθίου (οι οποίες έχουν πολλά κοινά χαρακτηριστικά μεταξύ τους) γιατί μπορούν να παρέχουν υψηλή απόδοση ισχύος, έχουν μεγάλη διάρκεια ζωής και είναι φιλικές ως προς το περιβάλλον. Το βασικότερο μειονέκτημα τους είναι το γεγονός ότι για εφαρμογές που απαιτούν πολύ μεγάλες ποσότητες ενέργειας χρειάζεται μεγάλος αριθμός μπαταριών σε σειρά και παράλληλα διασυνδεδεμένων. Γενικότερα οι μπαταρίες λιθίου απαιτούν μεγάλη προσοχή στον τρόπο φόρτισης, αλλά και στα όρια της τάσης (Voltage range) καθώς αν ξεπεραστούν θα υπονομευτεί η απόδοση της μπαταρίας. Συνεπώς βάσει της βιβλιογραφίας επιλέχθηκε το κατώτερο όριο ασφαλείας να είναι όταν η μπαταρία φτάσει στο 20% της ενεργειακής χωρητικότητας της, που όπως φαίνεται από το Σχήμα 3.8 αντιστοιχεί σε λίγο περισσότερα από 3.6 Volt ανά Shell μπαταρίας λιθίου (αν και

σε άλλες περιπτώσεις η τιμή αυτή βρέθηκε έως και 3.2 Volt). Αφού τελικά επιλέχθηκαν ως βέλτιστη επιλογή η μπαταρία λιθίου, απομένει μόνο να υπολογιστεί η ενεργειακή ανάγκη του συστήματος για να αξιολογηθεί ο αριθμός και η συνδεσμολογία των μπαταριών. Αυτή η διαδικασία θα παρουσιαστεί σε επόμενο κεφάλαιο όπου θα αναλυθούν οι τελικές αποφάσεις που πάρθηκαν για την υλοποίηση των πειραμάτων.



Σχήμα 3.8: Συνάρτηση τάσης (V) με το ποσοστό φόρτισης της μπαταρίας ιόντων λιθίου, Πηγή: [13]

Η ΦΠ ήταν το επόμενο τμήμα του συστήματος που αξιολογήσαμε. Η ορθή επιλογή ΦΠ μπορεί να αυξήσει κατακόρυφα την αυτονομία ενός υποβρυχίου συστήματος καθώς αποτελεί το πιο ενεργειακά κοστοβόρο τμήμα του. Πέρα από αυτό μεγάλη σημασία είχε τα επιλεγμένα φώτα να είναι στεγανά ακόμα και σε μεγάλα βάθη, ενώ ταυτόχρονα να διαθέτουν επαρκή φωτισμό. Όλα τα κριτήρια που θέσαμε ικανοποιήθηκαν από τα φώτα Lumen Subsea Light for ROV/AUV της Blue Robotics (Σχήμα 3.9). Τα συγκεκριμένα φώτα [15] ήταν 1500 lumen και τεχνολογίας LED, εκτός αυτού παρείχαν θερμοκρασία χρώματος 5700K τιμή πολύ κοντινή στα 5500K που παρέχει ο ήλιος κατά τις μεσημεριανές ώρες (λευκό φως). Ένα ακόμα πολύ σημαντικό χαρακτηριστικό ήταν η αντοχή τους στις πιέσεις, έτσι μπόρεσαν να τοποθετηθούν έως και 500 μέτρα κάτω από την στάθμη της επιφάνειας. Εκτός αυτού μας έδωσαν την δυνατότητα για τοποθέτηση περισσότερων από ένα φώτα (σε σειρά ή παράλληλα) με πολύ απλή συνδεσμολογία, μιας και για τις αποστολές χρειάστηκαν αρκετά στον αριθμό φώτα (παρότι η δέσμη φωτός που παράγουν είναι 135°) ώστε να δημιουργήσουν ικανοποιητική δέσμη φωτός για να επιτευχθεί η ζητούμενη ορατότητα για τον αισθητήρα. Κάθε φως είχε ονομαστική ισχύ 15 Watt σε πλήρη λειτουργία, γεγονός που κάνει υποδεικνύει ότι μεγάλος αριθμός από φώτα θα ήταν απαγορευτικός για μεγάλα χρονικά διαστήματα. Για το συγκεκριμένο μοντέλο μέσα από πειραματικές διατάξεις κατέστη δυνατή την εύρεση της βέλτιστης λύσης ανάμεσα στην κατανάλωση

ενέργειας και του εκπεμπόμενου φωτός (μιας και θεωρητικά η σχέση των δύο μεγεθών είναι ανάλογη). Στο σύστημα από τα φώτα τοποθετήθηκε ο ψηφιακός διακόπτης MOSWITCH-SPDT (Σχήμα 3.9) για να ανοίγουν και να κλείνουν με ασφάλεια. Συνεπώς, το συγκεκριμένο μοντέλο αποτέλεσε ιδανική λύση για πηγή φωτισμού στο υποθαλάσσιο περιβάλλον καθώς συνδύασε την αντοχή, την φωτεινότητα (ένταση), αλλά και την κατανάλωση ενέργειας.



Σχήμα 3.9: Αριστερά: Lumen Subsea Light for ROV/AUV. Δεξιά: ο διακόπτης MOSWITCH-SPDT, Πηγή : <https://bluerobotics.com>

Το τελευταίο ηλεκτρονικό στοιχείο που αξιολογήθηκε ήταν ο εξωτερικός σκληρός δίσκος για την αποθήκευση των δεδομένων. Επιλέχθηκε ο Crucial P3 (SSD) με χωρητικότητα 2TB ο οποίος τοποθετήθηκε εντός του περιβλήματος της εταιρίας Lemorele (Σχήμα 3.10). Η αιτία που αναδείχθηκε ο συγκεκριμένος σκληρός ως βέλτιστη επιλογή ήταν λόγω του μικρού μεγέθους του (11×3.6 cm). Εκτός αυτού εξαιτίας της τεχνολογίας SSD παρείχε την δυνατότητα για πολύ υψηλές ταχύτητες εγγραφής. Σε αυτό το σημείο πρέπει να σχολιαστεί ότι η θύρα USB του μικροϋπολογιστή που επιλέχθηκε δεν παρείχε την δυνατότητα μεταφοράς τόσο μεγάλου όγκου δεδομένων όσο μπορούσε να καταγράψει ο εξωτερικός σκληρός. Παρόλα αυτά επιλέχθηκε η συγκεκριμένη τεχνολογία στον σκληρό δίσκο ώστε να υπάρξουν οι βέλτιστες δυνατές αποδόσεις. Με αυτό τον τρόπο εξασφαλίστηκε ότι θα χρησιμοποιηθούν πλήρως οι δυνατότητες της θύρας USB, σε αντίθεση με την χρήση εξωτερικού σκληρού τεχνολογίας HDD. Όσον αφορά την χωρητικότητα, ανάλογα τις ανάγκες του κάθε πειράματος αυτή ενδέχεται να χρειαστεί να μεταβληθεί, παρόλα αυτά όπως φάνηκε και στις δοκιμές που θα παρουσιαστούν στην συνέχεια, το μεγαλύτερο ζήτημα που περιόριζε την χρονική διάρκεια των πειραμάτων ήταν η κατανάλωση ενέργειας, καθώς η τοποθέτηση επιπρόσθετης εξωτερικής μνήμης θα αποτελούσε μια αρκετά απλή διαδικασία (αν και σχετικά κοστοβόρα).



Σχήμα 3.10: Αριστερά: ο εξωτερικός σκληρός Crucial P3. Δεξιά: το περίβλημα σκληρού της εταιρίας Lemorele

3.2.5 Υποβρύχιο περίβλημα

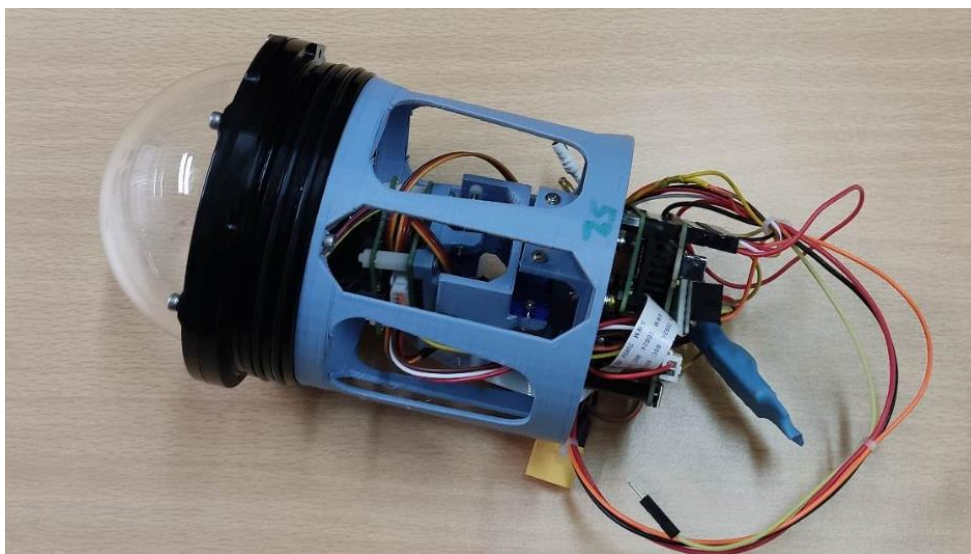
Το υποβρύχιο περίβλημα ήταν το τελευταίο κομμάτι του υλιστικού που αξιολογήθηκε, διότι έπρεπε όλα τα επιμέρους τμήματα να μπορούν να εδραστούν σε αυτό. Η ορθή επιλογή του αποτέλεσε πολύ σημαντικό τμήμα για την διασφάλιση του υλισμικού. Το ΥΠ χωρίστηκε σε τρία βασικά τμήματα. Το κύριο μέρος φαίνεται στο Σχήμα 3.11, εντός του οποίου τοποθετήθηκε το πρόσθιο τμήμα και στις δύο άκρες του το παράθυρο θέασης και η υποδοχή της ηλεκτρικής συνδεσμολογίας. Το κύριο μέρος αποτελούταν από έναν ενιαίο μεταλλικό κύλινδρο (αλουμινίου). Στην συνέχεια ορίστηκε το τμήμα του παραθύρου θέασης (Σχήμα 3.12) του μετρητικού οργάνου και τέλος το πρόσθιο τμήμα (Σχήμα 3.12) πάνω στο οποίο εδράστηκε η κάμερα, ο μικροϋπολογιστής και γενικότερα τα περισσότερα ηλεκτρονικά εξαρτήματα. Σε αυτό το στάδιο πρέπει να αναφερθεί ότι τελικά έγινε κατασκευή δύο διαφορετικών ΥΠ, στο ένα εδράστηκαν τα όργανα και στο άλλο οι ΠΕ, μιας και για μεγάλες χρονικά αποστολές δεν αρκούσε η χωρητικότητα του πρώτου για την τοποθέτηση ικανοποιητικού αριθμού μπαταριών.

Το κύριο μέρος που επιλέχθηκε για τα όργανα [15] ήταν 30 εκατοστά σε συνολικό μήκος και είχε εσωτερική διάμετρο 10 εκατοστών. Το υλικό κατασκευής του ήταν το 6061-T6 αλουμίνιο και η ονομαστική του αντοχή ήταν μέχρι βάθος 950 μέτρων. Για τις μπαταρίες επιλέχθηκε το ίδιο υλικό με διαφορετικές διαστάσεις. Το συνολικό του μήκος ήταν 40 εκατοστά ενώ η εσωτερική διάμετρος 7.5 εκατοστά. Οι διαστάσεις διαφοροποιήθηκαν μιας και οι μπαταρίες δεν έχουν ίδιο όγκο με τα μετρητικά όργανα. Τα κύρια μέρη (σωλήνες αλουμινίου) έχουν δύο «κενά» τμήματα στις δύο άκρες τους όπως φαίνεται και στο Σχήμα 3.11. Στο ένα τμήμα τοποθετήθηκε το ΠΘ μαζί με το πρόσθιο τμήμα ενώ στο άλλο τοποθετήθηκε ειδική υποδοχή (καπάκι) για να γίνει η ηλεκτρονική συνδεσμολογία (Σχήμα 3.11).

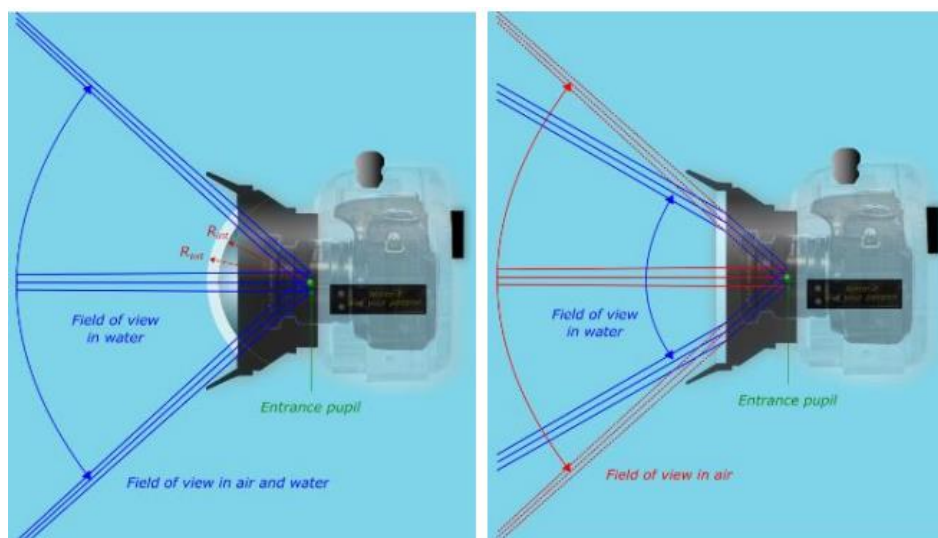


Σχήμα 3.11: Αριστερά: το κύριο μέρος του υποβρυχίου περιβλήματος. Δεξιά: ειδική υποδοχή ηλεκτρονικής συνδεσμολογίας, Πηγή : <https://bluerobotics.com/>

Το επόμενο σκέλος που αξιολογήθηκε ήταν το ΠΘ του μετρητικού οργάνου. Αρχικά, για λόγους κόστους επιλέχθηκε το υλικό του να είναι ακρυλικό αφού η άλλη επιλογή, το γυαλί, ήταν πιο ακριβό και βαρύ, αλλά δεν χαρασσόταν. Η πιο σημαντική επιλογή ήταν η γεωμετρία του ΠΘ. Οι επιλογές που αξιολογήθηκαν ήταν να είναι επίπεδο ή θολωτό. Το σημαντικότερο κριτήριο σε αυτό το στάδιο ήταν πως η συγκεκριμένη γεωμετρία θα επηρέαζε το οπτικό πεδίο της κάμερας, μιας και ζητούμενο σε υποθαλάσσιες εφαρμογές είναι η επίτευξη της μέγιστης δυνατής τιμής του. Βάσει της [14], το οπτικό πεδίο της κάμερας (φακού) μένει σταθερό σε περίπτωση χρήσης θολωτού παραθύρου θέασης ενώ μειώνεται αισθητά σε περίπτωση χρήσης επίπεδου, (Σχήμα 3.13) βάσει ενός συντελεστή (που είναι κοντά στην τιμή του δείκτη διάθλασης). Μια ακόμα σημαντική διαφορά αποτέλεσε το κόστος όπου το επίπεδο παράθυρο θέασης ήταν πιο οικονομικό σε σχέση με το θολωτό. Παρόλα αυτά κρίθηκε ότι η μείωση οπτικού πεδίου θα αποτελούσε σημαντικό επιβαρυντικό παράγοντα και συνεπώς επιλέχθηκε θολωτό παράθυρο θέασης.



Σχήμα 3.12: Παρουσίαση του θολωτού παραθύρου θέασης μαζί με το πρόσθιο πάνω στο οποίο εδράζονται τα περισσότερα μέρη του υλισμικού



Σχήμα 3.13: Αριστερά: ο τρόπος που επηρεάζεται το οπτικό πεδίο (FOV) με χρήση θολωτού παραθύρου θέασης . Δεξιά: ο τρόπος που επηρεάζεται το οπτικό πεδίο (FOV) με χρήση επιπέδου παραθύρου θέασης, Πηγή : [14]

Το πρόσθιο τμήμα (Σχήμα 3.12) κατασκευάστηκε με στόχο να εδράσει το σύνολο των αισθητήρων, του μετρητικού οργάνου, του μικροϋπολογιστή και των μεταξύ τους ηλεκτρονικών με ασφαλή τρόπο. Η κατασκευή έγινε με την μέθοδο τρισδιάστατης εκτύπωσης από την εταιρία Ploatech η οποία εξειδικεύεται σε υποθαλάσσιες εφαρμογές και είχε μεγάλη συνεισφορά στην υλοποίηση του υλικοτεχνικού σκέλους της συγκεκριμένης εργασίας. Όπως φαίνεται στο Σχήμα 3.12 στο δεξιό μέρος του πρόσθιου τμήματος εδράζεται ο μικροϋπολογιστής, το Witty Pi, ο αισθητήρας Power Sense Module καθώς και η συνδεσμολογία της κάμερας, της τροφοδοσίας και των σερβοκινητήρων. Εντός, της κατασκευής φαίνεται η κάμερα καθώς και οι σερβοκινητήρες που είναι υπεύθυνοι για την κίνηση της. Γενικότερα, η κατασκευή της συγκεκριμένης βάσης έκανε δυνατή την ορθή τοποθέτηση όλου του εξοπλισμού εντός του ΥΠ. Τα τρία τμήματα που αναλύθηκαν σε αυτό το κεφάλαιο ενώθηκαν και ασφαλίστηκαν (για να αποφευχθούν εισροές νερού) υλοποιώντας το τελικό σύστημα της αυτόνομης θαλάσσιας κάμερας (Σχήμα 3.14), σε συνδυασμό και με το υποβρύχιο ΥΠ που περιλάμβανε της μπαταρίες.



Σχήμα 3.14: Τελικό σύστημα αυτόνομης θαλάσσιας κάμερα.

3.3 Συνδυασμός τεχνικών στοιχείων και λογισμικού

Αφού ολοκληρώθηκε η μεμονωμένη αξιολόγηση όλων των στοιχείων που αποτέλεσαν το σύστημα της υποθαλάσσιας κάμερας, σημαντικό είναι να παρουσιαστεί η συνδεσμολογία και οι αλληλεπιδράσεις που παρουσίασαν μεταξύ τους. Η συγκεκριμένη ανάλυση είναι ουσιώδης καθώς στις περισσότερες περιπτώσεις το πιο σύνθετο ζήτημα είναι η αλληλεπίδραση και ο περιορισμός μεταξύ των επιμέρους τμημάτων. Γνωρίζοντας την συμπεριφορά τους η επίλυση πολλών ζητημάτων επιτεύχθηκε άμεσα. Εκτός από αυτό αυτό το κεφάλαιο θα γίνει επεξήγηση του λογισμικού που υλοποιήθηκε και οι εφαρμογές που θέλαμε να επιτύχουμε βάσει αυτού. Συνολικά το συγκεκριμένο κεφάλαιο αποσκοπεί στο να επεξηγήσει στον αναγνώστη τις βασικές λειτουργίες του συστήματος σε επίπεδο υλιστικού και λογισμικού.

3.3.1 Συνδεσμολογία και αλληλεπίδραση τεχνικών στοιχείων

Η συνδεσμολογία ενός συστήματος όπως αυτό που κατασκευάστηκε αποτελεί ένα αρκετά σύνθετο ζήτημα καθώς δεν υπήρχαν περιθώρια σφαλμάτων. Παρατυπίες σε επίπεδο υλιστικού συνήθως οδηγούν σε μεγάλες ζημίες που αυξάνουν το κόστος εκθετικά, για αυτό και όλες οι συνδέσεις έγιναν με ιδιαίτερη προσοχή. Το βασικό

εργαλείο για την παρουσίαση της συνδεσμολογίας αποτελεί το Σχήμα 3.15 που ακολουθεί και αναδεικνύει το σύνολο της συνδεσμολογίας, όλων των μεμονωμένων τμημάτων του συστήματος.

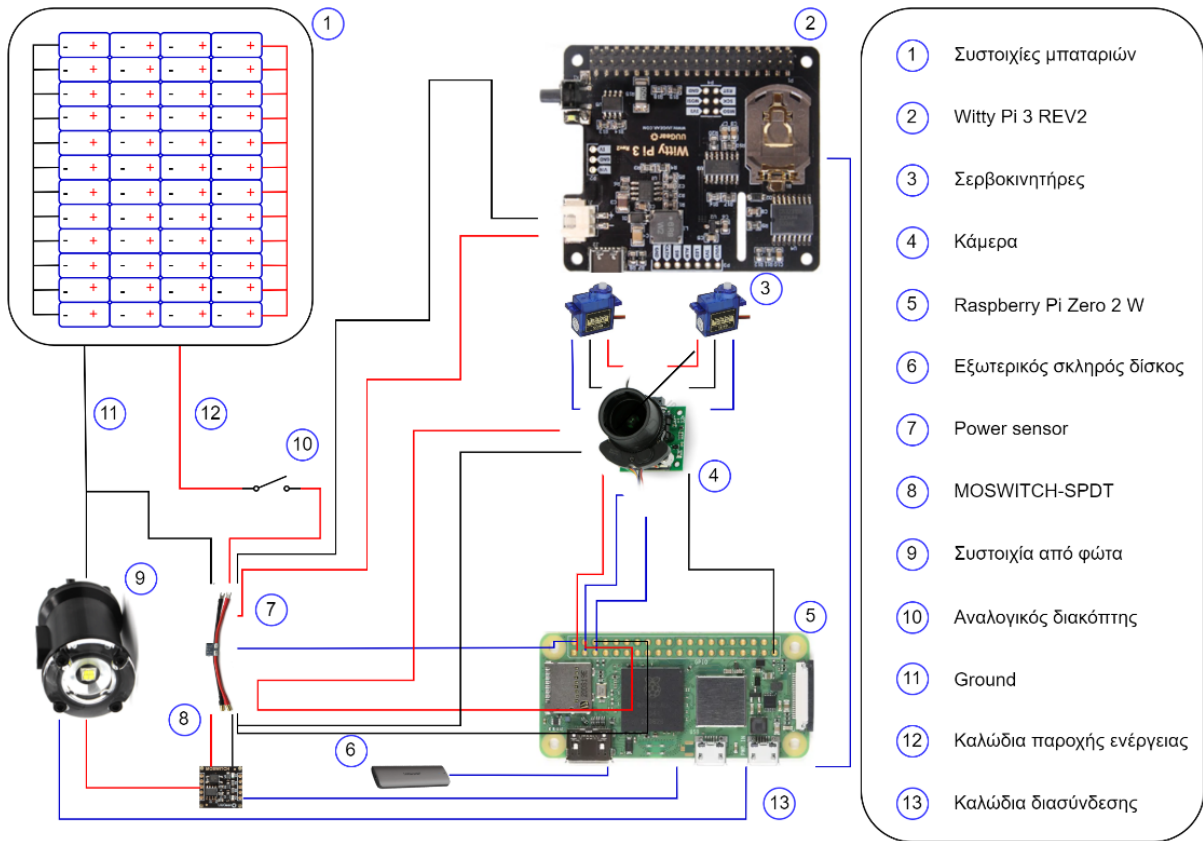
Η «αφετηρία» του συστήματος ήταν οι μπαταρίες (ΠΕ) οι οποίες αποτελούνταν από τέσσερις κυψέλες ιόντων λίθου σε σειρά (οι οποίες πρακτικά δημιουργούσαν μια μπαταρία) και δώδεκα τέτοιες τετράδες τοποθετημένες παράλληλα. Όταν γίνεται τοποθέτηση κυψέλων σε σειρά αυτό που συμβαίνει είναι ότι προστίθεται η τάση της κάθε κυψέλης (3.7 V), έτσι τελικά η απόδοση της συστοιχίας μπαταριών είναι 3.7×4 V, δηλαδή 14.8 V. Αντίθετα, όταν οι μπαταρίες είναι τοποθετημένες παράλληλα αυξάνεται η χωρητικότητα χωρίς να επηρεάζεται η τάση. Πριν συνεχίσει η ανάλυση είναι σημαντικό να σχολιαστεί ότι στα αρχικά πειράματα χρησιμοποιήθηκαν μπαταρίες πολυμερών λιθίου, όμως η τελική πηγή ενέργειας που παρουσιάζεται αποτελούταν από μπαταρίες ιόντων λιθίου, κυρίως εξαιτίας του μεγαλύτερου ενεργειακού αποθηκευτικού χώρου για ίδια φυσικά μεγέθη, αλλά και εξαιτίας της μεγαλύτερης διάρκειας ζωής τους. Παρατηρώντας το Σχήμα 3.15 φαίνεται ότι από την συστοιχία μπαταριών «ξεκινούσαν» δύο καλώδια, ένα θετικό (παροχής ενέργειας) και ένα αρνητικό (ground). Το θετικό συνδεόταν με έναν αναλογικό διακόπτη ώστε να έχουμε την δυνατότητα να διακόπτουμε την ροή ενέργειας προς το σύστημα εφόσον χρειαζόταν, κυρίως για τη διεξαγωγή πειραμάτων. Στην συνέχεια η ροή ενέργειας (θετικό καλώδιο) έφτανε στο power sensor module το οποίο μετρούσε τάση (V) ρεύμα (A).

Από το power sensor ξεκινούσε καλώδιο διασύνδεσης που συνδεόταν στο pin του μικροϋπολογιστή (Σχήμα 3.15) και μετέφερε την πληροφορία για τις ενεργειακές μετρήσεις. Γενικά, το power sensor είχε επιφορτιστεί με την διανομή ενέργειας στα περισσότερα τμήματα του συστήματος όπως το Witty Pi 3 REV2, το Raspberry Pi Zero 2W, η κάμερα και το MOSWITCH-SPDT. Εκτός αυτού ενδιαφέρον παρουσίασέ η τοποθέτηση των καλωδίων στα pins γενικής χρήσης εισόδου-εξόδου του μικροϋπολογιστή (Σχήμα 3.16). Όπως γίνεται αντιληπτό από το σχήμα η τροφοδοσία του μικροϋπολογιστή επετεύχθη χρησιμοποιώντας τα pins με θέσεις 2, 3 (η αρίθμηση των pins γίνεται βάσει του σχήματος 3.16) παρέχοντας στο Raspberry τάση 5 V και ground. Ο μικροϋπολογιστής συνδέθηκε με όλα τα συστήματα με καλώδια σύνδεσης μιας και αποτέλεσε τον πυρήνα όλων των εντολών προς τα υπόλοιπα μέρη του συστήματος. Η πλακέτα της κάμερας συνδέθηκε με τον μικροϋπολογιστή με δύο καλώδια στα GPIO pin 3, 5. Με αυτό τον τρόπο επετεύχθη η εξαγωγή των δεδομένων της κάμερας και η εφαρμογή εντολών από του μικροκινητήρων που ήταν υπεύθυνοι για την κίνηση, το οπτικό ζουμ και την εστίαση. Αντίστοιχα και το Witty είχε δύο

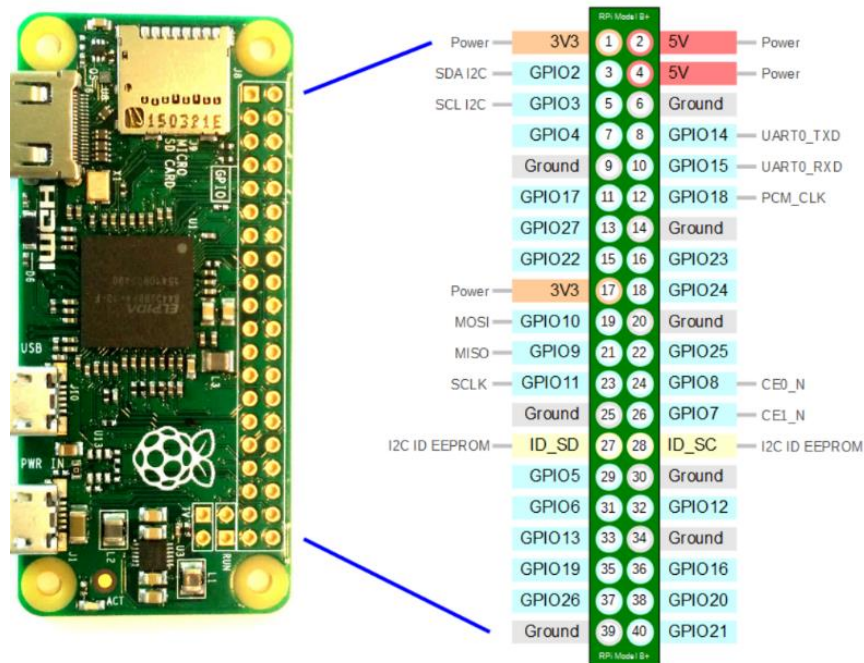
τρόπους σύνδεσης με τον μικροϋπολογιστή, αρχικά με ένα καλώδιο αλλά και με την σύνδεση των μεταξύ τους pin. Με αυτό τον τρόπο του Witty είχε την δυνατότητα να ενεργοποιήσει και να απενεργοποιήσει το σύστημα, ενώ ταυτόχρονα να εξάγει δεδομένα όπως η τάση και ο χρόνος. Πέρα από αυτό ο μικροϋπολογιστής ήταν συνδεδεμένος με τον εξωτερικό σκληρό δίσκο στον οποίο αποθηκευόντουσαν τα δεδομένα ενώ ταυτόχρονα του παρέχει και ενέργεια για τις ανάγκες λειτουργίας του. Τέλος υπήρξαν δύο ακόμα καλώδια σύνδεσης με τα φώτα (για να τα ελέγχει) και στο MOSWITCH-SPDT με χρήση του οποίου γινόταν η εκκίνηση και ο τερματισμός της λειτουργιών από τα φώτα.

Τα τελευταία στοιχεία συνδεσμολογίας που δεν σχολιάστηκαν, είναι πως παρέχεται ενέργεια στα φώτα και η σύνδεση της πλακέτας της κάμερας με τους σερβοκινητήρες. Η ενέργεια «έφτανε» στα φώτα μέσα από το MOSWITCH-SPDT, όταν διέκοπτε την παροχή ενέργειας αυτά απενεργοποιούνταν για αυτό και λειτούργησε ως διακόπτης. Οι σερβοκινητήρες δέχονται τρία καλώδια από την πλακέτα της κάμερας, τα δύο αφορούσαν την παροχής ενέργειας (θετικό και αρνητικό) ενώ το τρίτο την «επικοινωνία» μεταξύ των συστημάτων ώστε τελικά η σύνδεση να φτάσει μέχρι τον μικροϋπολογιστή, όπου σε επίπεδο λειτουργικού πια θα διαχειριζόντουσαν οι λοιπές διαδικασίες.

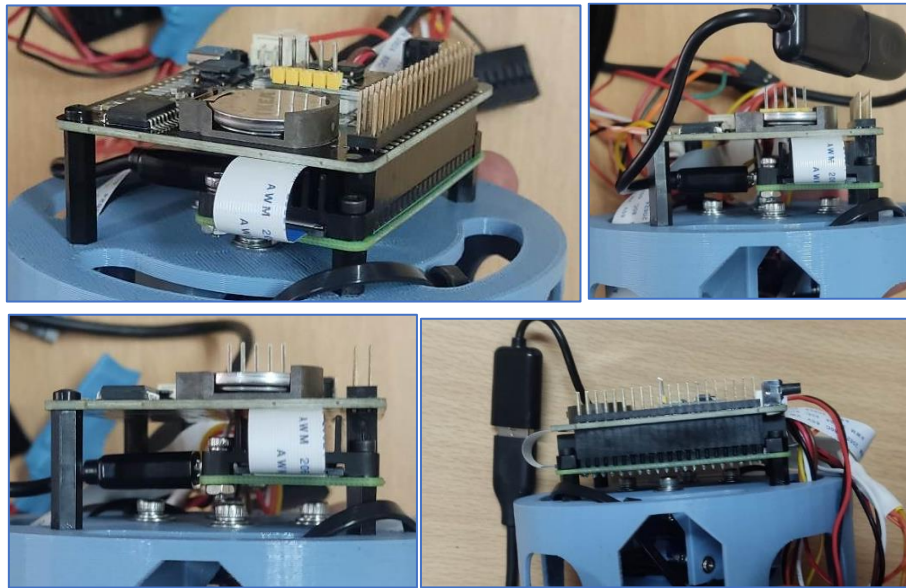
Όπως φαίνεται βάσει όσων σχολιάστηκαν τα διάφορα συστήματα αλληλοεπιδρούν μεταξύ τους σύμφωνα με τη συνδεσμολογία που υλοποιήθηκε. Εξαιτίας αυτού βλάβες σε ένα τμήμα του συστήματος θα οδηγούσαν σε ζητήματα και σε διαφορετικά τμήματα. Παραδείγματος χάριν, αν εμφάνιζε βλάβη το power sensor, τότε η ενεργειακή ροή σε όλα τα συστήματα που συνδέονται με αυτό θα διαταρασσόταν με μεγάλη πιθανότητα να σταματούσε η λειτουργία τους. Συνολικά δεν αντιμετωπίστηκαν τέτοια ζητήματα και το τελικό σύστημα λειτούργησε βάσει των ζητούμενων προδιαγραφών.



Σχήμα 3.15: Διάγραμμα διασυνδέσεων.



Σχήμα 3.16: Διάγραμμα pins γενικής χρήσης εισόδου εξόδου (GPIO) του μικροϋπολογιστή.
 Πηγή : [<https://raspberrypi.stackexchange.com/questions/83610/gpio-pinout-orientation-raspberrypi-zero-w>]



Σχήμα 3.17: Παρουσίαση του τρόπου σύνδεσης του Raspberry Pi Zero 2 W με το Witty Pi 3 REV2

3.3.2 Λογισμικό λειτουργίας συστήματος

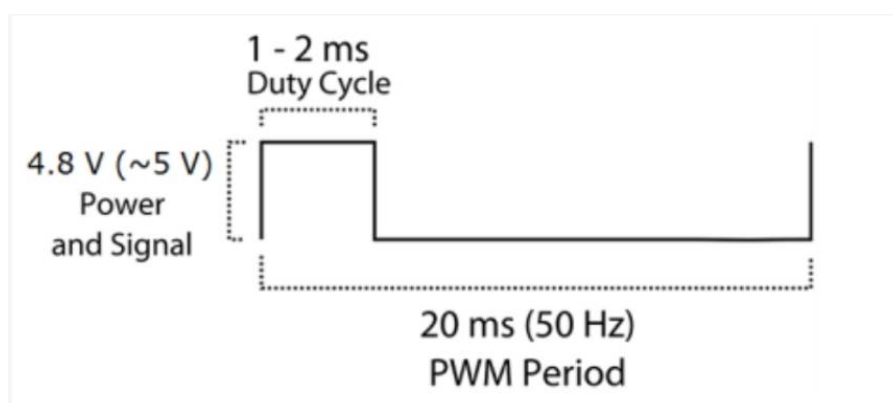
Πέραν του κατασκευαστικού μέρους του συστήματος που υλοποιήθηκε, πρόκληση αποτέλεσε και η δημιουργία του απαραίτητου λογισμικού, διότι όλοι οι κώδικες έπρεπε να υλοποιούνται από τον μικροϋπολογιστή, ο οποίος πέρα από τις βασικές του αδυναμίες σε υπολογιστική ισχύ, είχε και δικό του λειτουργικό σύστημα διαφορετικό από αυτό που έχουν οι συνήθεις υπολογιστές (αν και μοιάζει αρκετά με τα ubuntu, μιας και είναι βασισμένα στο ίδιο distribution). Παρόλα αυτά κατέστη δυνατό να υλοποιηθεί το λογισμικό για τις βασικές λειτουργίες. Οι πιο σύνθετες εργασίες επιλέχθηκε να περατωθούν σε κανονικό υπολογιστή ώστε η επεξεργασία των δεδομένων να γίνεται με τον ταχύτερο δυνατό τρόπο.

Πριν ξεκινήσει η υλοποίηση λογισμικού έπρεπε να εγκατασταθεί λειτουργικό σύστημα (os) στο μικροϋπολογιστή που επιλέχθηκε. Τα Raspberry Pi έχουν δικό τους λειτουργικό σύστημα με ονομασία Raspian 32 ή 64 bit. Η εγκατάσταση του λειτουργικού έγινε στην micro SD κάρτα η οποία στην συνέχεια τοποθετήθηκε στην υποδοχή του Raspberry. Όστε να επιτύχουμε πρόσβαση στον μικροϋπολογιστή χρησιμοποιήθηκε οθόνη, ποντίκι και πληκτρολόγιο σε αυτά τα αρχικά στάδια. Το επόμενο βασικό βήμα ήταν να «στήσουμε» ένα τοπικό δίκτυο χρησιμοποιώντας ένα access point στο οποίο συνδέεται τόσο το Raspberry, με στατική διεύθυνση IP, όσο και οι υπολογιστές με τους οποίους θα το χειριζόμασταν. Έτσι καταφέραμε να κάνουμε

την διασύνδεση των δύο υπολογιστών χρησιμοποιώντας το πρωτόκολλο SSH, το οποίο δίνει την δυνατότητα σε δύο υπολογιστές να επικοινωνούν. Με αυτό τον τρόπο δημιουργήσαμε ένα δίκτυο που τοποθετώντας στην γραμμή εντολών του υπολογιστή τη στατική IP Raspberry, το όνομα του χρήστη αλλά και έναν κωδικό που βάλουμε, μπορέσαμε να συνδεθούμε στον μικροϋπολογιστή. Αυτό είχε αξία για δύο λόγους, αρχικά εξάλειψε την ανάγκη για περιφερειακά στοιχεία όπως οθόνη ποντίκι κ.α., και επιπλέον παρείχε την δυνατότητα να χειριστούμε το Raspberry εξ αποστάσεως, το οποίο ήταν καίριο μιας και τοποθετήθηκε εντός του προστατευτικού περιβλήματος.

Αφού ολοκληρώθηκε η προαναφερθείσα διαδικασία ξεκίνησε η υλοποίηση λογισμικού για όλα τα τμήματα του υλιστικού. Αρχικά, επιλέχθηκε να μελετηθούν οι τέσσερις μικροκινητήρες της κάμερας, καθώς ο κατασκευαστής παρείχε μεγάλο μέρος του λογισμικού χρειάστηκε στο αποθετήριο κώδικα (<https://github.com/ArduCAM>). Προκειμένου να χρησιμοποιηθεί ο κώδικας που παρέχεται κρίθηκε αναγκαίο να εκτελεστούν κάποια θεμελιώδη βήματα. Αρχικά, ήταν απαραίτητο να ενεργοποιηθεί το πρωτόκολλο I2C (bus) το οποίο επιτρέπει σε πολλές περιφερειακές συσκευές να συνδεθούν στο Raspberry με διαφορετικές διευθύνσεις, που ήταν αναγκαίο για την λειτουργία του συστήματος κάμερας-μικροκινητήρων. Στην συνέχεια χρησιμοποιήθηκε το script Focuser.py του github στο οποίο υπήρχαν οι κατάλληλες διευθύνσεις ώστε να επικοινωνήσουμε μέσω του I2C πρωτοκόλλου με το module που έλεγχε τους κινητήρες της κάμερας. Ο τρόπος που επιτεύχθηκε αυτό ήταν μέσω του συγκεκριμένου πρόγραμματος όπου έγινε σωστή αντιστοίχιση τιμών (values) οι οποίες αντιστοιχούσα στις διάφορες διευθύνσεις για να ορίσουν την λειτουργία των κινητήρων (μέσα από ένα σύνολο συναρτήσεων όπως η read, write κ.α). Χρησιμοποιώντας την κλάση Focuser μπορέσαμε να καθορίσουμε την θέση των διάφορων κινητήρων. Συγκεκριμένα, φτιάξαμε κώδικες για την αρχικοποίηση των θέσεων των κινητήρων. Ο τελικός κώδικας είχε εφαρμογή προγράμματος διεπαφή ώστε να μπορούμε σε πραγματικό χρόνο να μεταβάλλουμε τις θέσεις των κινητήρων. Σημαντικό σχόλιο σε αυτό το σημείο είναι ότι η αρχικοποίηση των θέσεων των κινητήρων μπορούσε να μεταβληθεί από τους κώδικες μέσω της διεπαφής ώστε να επιλέγονται οι βέλτιστες τιμές σε κάθε περίπτωση, ούτως ώστε κατά το άνοιγμα σε δυσμενείς συνθήκες όπως το θαλάσσιο περιβάλλον το σύστημα να μπορούσε να παράξει τα καλύτερα δυνατά αποτελέσματα (σε περίπτωση που η σύνδεση μέσω SSH δεν είναι δυνατή). Ο τρόπος που ελέγχουμε τα αποτελέσματα των κινητήρων ήταν κάνοντας streaming της κάμερας στον δικό μας υπολογιστή με την χρήση του gstreamer.

Το επόμενο τμήμα του λογισμικού που υλοποιήθηκε αφορούσε τον έλεγχο στα φώτα. Στο μοντέλο που επιλέχθηκε, μπορούσαμε να ρυθμίσουμε την ένταση τους μέσω ενός κατάλληλου παλμού PWM (Pulse width modulation) (Σχήμα 3.18). Τα φώτα είχαν τρία καλώδια, δυο για την τάση (θετικό και αρνητικό) και ένα τρίτο για την διάδοση παλμών (σήμα εισόδου). Στέλνοντας μέσω ενός pin GPIO του μικροϋπολογιστή και της κατάλληλης βιβλιοθήκης (RPi.GPIO) της python, παλμούς συχνότητας 50Hz (20ms) και duty cycle (ενεργό κομμάτι του παλμού) από 1 έως 2 ms, δηλαδή 5-10% της συνολικής διάρκειας ενός παλμού, μπορέσαμε να ελέγξουμε την ένταση (φωτεινότητα) της φωτεινής πηγής. Στον πίνακα που ακολουθεί παρουσιάζεται η αντιστοίχιση των τιμών του duty cycle (% του κάθε παλμού), το ποσοστό που αντιστοιχεί στην ένταση (της φωτεινής πηγής) καθώς και το ρεύμα που μετρήσαμε με χρήση τροφοδοτικού. Όπως γίνεται αντιληπτό με την αύξηση του μεγέθους του duty cycle, αυξάνεται και το ρεύμα που μετρήσαμε με αποτέλεσμα να αυξάνεται και η κατανάλωση ρεύματος μιας και η τάση ήταν κοινή. Αν το duty cycle είναι από 5-5.5% της συνολικής διάρκειας του παλμού, τα φώτα είναι σβηστά, ενώ για μεγαλύτερα ποσοστά οι λαμπτήρες Led παρήγαγαν φως.



Σχήμα 3.18: Παράδειγμα ενός παλμού PWM σαν αυτούς που υλοποιήθηκαν για τα φώτα.

| Duty Cycle % | Χρήση % | Ρεύμα (A) |
|--------------|---------|-----------|
| 5 | 0 | 0 |
| 5.5 | 10 | 0.04 |
| 6 | 20 | 0.09 |
| 6.5 | 30 | 0.16 |
| 7 | 40 | 0.46 |
| 7.5 | 50 | 0.84 |
| 8 | 60 | 1.42 |
| 8.5 | 70 | 2.35 |
| 9 | 80 | 3.72 |
| 9.5 | 90 | 3.87 |
| 10 | 100 | 3.94 |

Πίνακας 3.2: Στατιστικά στοιχεία για τα φώτα

Ένα ακόμα σκέλος που υλοποιήθηκε ήταν το λογισμικό του μετρητή power sensor. Σε αυτό το τμήμα του λογισμικού δεν εφαρμόστηκε κάποια σύνθετη μεθοδολογία προγραμματιστικά, διότι απλά χρησιμοποιήσαμε τον driver ads1x15 από τον οποίο και εξάγαμε τις τιμές ρεύματος και τάσης (<https://github.com/RobTillaart/ADS1X15>), ενώ χρησιμοποιήσαμε και την βιβλιοθήκη gpiozero για να εξάγουμε την θερμοκρασία του επεξεργαστή. Όλη αυτή η πληροφορία αποθηκεύτηκε σε αρχείο, ώστε να επιτύχουμε εποπτικό έλεγχο της λειτουργίας του συστήματος. Εκτός αυτού όπως και στα προηγούμενα παραδείγματα ο κώδικας υλοποιήθηκε σε python.

Για τα δεδομένα της κάμερας επιλέχθηκε η βιβλιοθήκη libcamera [15], η οποία υποστηρίζει σύνθετα συστήματα καμερών. Μέσα από την συγκεκριμένη βιβλιοθήκη μπορέσαμε να εξάγουμε εικόνα με αρκετά απλή μεθοδολογία. Οι κώδικες που υλοποιήθηκαν ήταν ένα launcher.sh (Σχήμα 3.19), εντός του οποίου τοποθετήθηκε η βιβλιοθήκη και οι επιλογές: της μορφής της εικόνας (raw), ο χρόνος λήψης της κάμερα, το όνομα των αρχείων, η τοποθεσία που θα λάμβανε χώρο η διαδικασία, καθώς και ένα configure file (Σχήμα 3.19) εντός του οποίου αναγραφόταν πόσες εικόνες θα υλοποιούνταν ανά δευτερόλεπτο, οι διαστάσεις τους, αλλά και το γεγονός ότι τις θέλαμε σε μορφή ξεχωριστών εικόνων και όχι βίντεο.



```
GNU nano 5.4 launcher.sh
#!/bin/sh
#launcher.sh
NOW=`date +%F_%H%M` ;
YEAR=`date +%y`
MONTH=`date +%m`
DAY=`date +%d`
HOURL=`date +%H`
MINUTE=`date +%M`

cd /
cd media/pi/fourth
mkdir $YEAR
mkdir $YEAR/$MONTH
mkdir $YEAR/$MONTH/$DAY
mkdir $YEAR/$MONTH/$DAY/$HOURL
#mkdir $YEAR/$MONTH/$DAY/$HOURL/$MINUTE
cd $YEAR/$MONTH/$DAY/$HOURL

#cd /
#cd media/pi/third
libcamera-raw -c ~/camstereo.conf -o ptz50frame%08d.raw -t 180000 -v 2
cd /
```

```
GNU nano 5.4
mode=2028:1520
framerate=5
segment=1
```

Σχήμα 3.19: Αριστερά: Ο τρόπος που αποθηκεύονται οι εικόνες. Δεξιά: Το configure αρχείο.

Πέραν τούτου, στο πλαίσιο του λογισμικού μέσα από το interface (Σχήμα 3.20) του Witty έγινε μια σειρά πολύ σημαντικών διαδικασιών. Αρχικά, ορίστηκαν οι τιμές low voltage threshold αλλά και recovery voltage threshold. Το πρώτο κατώφλι ορίστηκε σε τιμή 12.6 V ενώ το δεύτερο 14 V. Η σημασία τους ήταν μεγάλη, διότι όταν ο

αισθητήρας θα μετρούσε μια τιμή μικρότερη από 12.6 V θα απενεργοποιούσε τον μικροϋπολογιστή για να μην προκληθούν ζημιές στις μπαταρίες. Από την άλλη το recovery threshold αποτέλεσε τιμή η οποία λαμβάνεται υπόψιν μόνο αφού ενεργοποιηθεί το low voltage threshold και σε περίπτωση που το Voltage που μετράει ο αισθητήρας ήταν λιγότερο από την προκαθορισμένη τιμή δεν ενεργοποιούσε τον μικροϋπολογιστή. Δηλαδή, στην περίπτωση μας όταν η τάση έπεφτε κάτω από 12.6 V το σύστημα απενεργοποιούνταν και αν η τάση παρέμενε κάτω από 14 V το σύστημα δεν θα επανενεργοποιούνταν. Εκτός αυτού υλοποιήθηκε πρόγραμμα (Σχήμα 3.21) το οποίο καθόρισε τις ώρες λειτουργίας του συστήματος. Το συγκεκριμένο πρόγραμμα είχε αρχή τις 12 το βράδυ στις 13/06/2023 και λήξη στις 15/10/2023 12 το βράδυ, δηλαδή το πείραμα θα λάμβανε χώρα τέσσερις μήνες και δύο μέρες. Εκτός αυτού με την εντολή ON ορίστηκε ένα διάστημα 8 λεπτών στο τέλος των οποίων το σύστημα θα απενεργοποιούνταν και με την εντολή OFF το διάστημα που θα παρέμενε κλειστό μέχρι να κάνει την επόμενη επανεκκίνηση.

```
Witty Pi - Realtime Clock + Power Management for Raspberry Pi
< Version 3.51 > by Dun Cat B.V. (UUGear)
=====
>>> Current temperature: 34.0°C / 93.2°F
>>> Your system time is: Fri 02 Jun 2023 17:36:03 EEST
>>> Your RTC time is: Fri 02 Jun 2023 17:36:03 EEST
>>> Vin=14.78V, Vout=5.07V, Iout=0.27A
Now you can:
 1. Write system time to RTC
 2. Write RTC time to system
 3. Synchronize time
 4. Schedule next shutdown
 5. Schedule next startup
 6. Choose schedule script
 7. Set low voltage threshold [12.6V]
 8. Set recovery voltage threshold [14V]
 9. View/change other settings...
10. Reset data...
11. Exit
What do you want to do? (1-11) |
```

Σχήμα 3.20: Interface του Witty.

```
BEGIN 2023-06-13 00:00:00
END 2023-10-15 00:00:00
ON M8 # keep ON state for 8 minutes
OFF H8 M14 # keep OFF state for 8 hour 14 minutes
```

Σχήμα 3.21: Πρόγραμμα Witty.

Αφού έχουν ολοκληρωθεί οι απαραίτητοι κώδικες για την λειτουργία του συστήματος, ήταν πολύ σημαντικό αυτά να μπορούν να εναρμονιστούν με το πρόγραμμα του witty και να μην χρειάζονται τον χρήστη να τα ενεργοποιήσει χειροκίνητα. Αυτό επιτεύχθηκε μέσα από το αρχείο crontab το οποίο δημιουργήθηκε για την περάτωση επαναλαμβανόμενων διαδικασιών. Σε αυτό το αρχείο (Σχήμα 3.22)

ορίσαμε πότε θέλουμε να ξεκινήσει ο κάθε κώδικας. Στην περίπτωση μας αποφασίσαμε να ξεκινάει με το άνοιγμα του μικροϋπολογιστή και μετά από ένα διάστημα ώστε η εκκίνηση να γινόταν απρόσκοπτα. Για το power sensor αλλά και για την κίνηση των μικροκινήτρων επιλέχθηκε η αρχή τους να γίνεται σχεδόν αμέσως μετά την εκκίνηση μιας και δεν παρατηρήθηκε να την επηρεάζουν αρνητικά τον μικροϋπολογιστή ενώ για τα φώτα και την κάμερα δώσαμε ένα εύλογο χρονικό διάστημα.

Κλείνοντας, το συγκεκριμένο υποκεφάλαιο υλοποιήθηκε κώδικας ο οποίος παρότι δεν αφορά άμεσα το σύστημα κρίθηκε άξιος αναφοράς. Ο συγκεκριμένος κώδικας λάμβανε σαν δεδομένα εισαγωγής τα 12 bit αρχεία εικόνων που αποθηκεύαμε τα μετέτρεπε σε 8 bit και στην συνέχεια τα αποθήκευε είτε σε jpg format, είτε σε png format ανάλογα την επιλογή του χρήστη. Σε αυτό το σημείο το λογισμικό που χρησιμοποιήθηκε έως αυτό το σκέλος της συγκεκριμένης εργασίας έχει αναλυθεί ενώ στην συνέχεια θα ακολουθήσουν τα πειράματα που έγιναν πριν τις πραγματικές αποστολές.

```
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
@reboot sleep 120 && /home/pi/launcher.sh >/home/pi/logs/cronlog 2>&1
@reboot sleep 180 && python /home/pi/Light/light2.py >/home/pi/logs/cronlogl 2>&1
@reboot sleep 60 && python /home/pi/MIPI_Camera/Jetson/JetsonNano_PTZ/finalinit.py >/home/pi/Logs/cronlogMot 2>&1
@reboot sleep 20 && python /home/pi/powersensor/ads.py >/home/pi/Logs/cronlogPower 2>&1
```

Σχήμα 3.22: Αρχείο crontab.

3.4 Πειραματικές διατάξεις, σχεδιασμός και έλεγχος.

Έχοντας παρουσιάσει όλα τα στοιχεία του υλισμικού και λογισμικού του συστήματος που υλοποιήθηκε, βασικό ζητούμενο ήταν να γίνουν εργαστηριακά έλεγχοι που θα έδειχναν αν η συνολική λειτουργία του ήταν η ζητούμενη. Γενικά παρουσιάστηκαν κάποια βασικά σκέλη που έπρεπε να μελετηθούν. Αρχικά, υπήρξε το ζήτημα της αυτονομίας όπου σε αυτό εντάσσονται οι μπαταρίες, τα φώτα και η συνολική ενεργειακή ανάγκη του συστήματος. Σε συνδυασμό με το ζήτημα της

αυτονομίας έπρεπε να τεθεί και ο σχεδιασμός μιας αποστολής. Ο συγκεκριμένος σχεδιασμός αφορούσε το χρονικό διάστημα που το σύστημα θα είναι ενεργοποιημένο, τι δεδομένα παρήγαγε κ.α. Τέλος, υπήρξαν οι έλεγχοι που έπρεπε να γίνουν για να επιβεβαιωθεί ότι ο σχεδιασμός μπορούσε να υλοποιηθεί με αξιοπιστία και ασφάλεια για τον εξοπλισμό. Συνολικά το συγκεκριμένο κεφάλαιο αποσκοπεί στο να παρουσιάσει στον αναγνώστη την διαδικασία υλοποίησης μιας αποστολής, η οποία απαιτεί αναλυτική προετοιμασία ενώ ταυτοχρόνως οδηγεί σε αλλαγές στο υλισμικό και στο λογισμικό που έχει υλοποιηθεί μέχρι τότε.

3.4.1 Αυτονομία και δυνατότητες συστήματος.

Ως ένα ηλεκτρονικό σύστημα το βασικότερο στοιχείο που του παρέχει αυτονομία είναι το μέγεθος των αποθηκών ενέργειας, σε συνάρτηση με την ενέργεια που καταναλώνει. Για αυτό το πρώτο βήμα ήταν να μελετηθεί η κατανάλωση που είχαν τα φώτα. Αυτό για να επιτευχθεί, αρχικά δημιουργήθηκε μια κλίμακα βάσει του ποσοστού χρήσης τους, από 0% (δηλαδή τα φώτα είναι κλειστά) έως 100% όπου τα φώτα είχαν την μέγιστη ισχύ, το οποίο ισοδυναμούσε με την μέγιστη κατανάλωση ενέργειας. Βάση του κατασκευαστή κάθε φως έχει μέγιστη ονομαστική ισχύ τα 15 Watt συνεπώς για τα τέσσερα φωτά που τοποθετήθηκαν η μέγιστη ισχύς ήταν 60 Watt, τιμή η οποία ήταν ενεργειακά απαγορευτική για το σύστημα το οποίο είχε κατασκευαστεί. Η μέγιστη τιμή που μπορούσε το σύστημα να παρέχει, ανάλογα τις μπαταρίες που θα επιλέχθηκαν (3 ή 4 κυψέλες σε σειρά), υπολογίστηκε χρησιμοποιώντας μπαταρίες και μετρώντας μια σειρά παραμέτρων όπως το αρχικό, το τελικό Voltage, τις mAh που καταναλώθηκαν και τον χρόνο που χρειάστηκε για να γίνει αυτό. Ο πίνακας 3.3 παρουσιάζει με ροζ χρώμα τα μετρημένα στοιχεία ενώ με γκρι τα στοιχεία που υπολογίστηκαν. Οι Wh υπολογίστηκαν ως εξής : $Wh = \frac{mAh \times Voltage}{1000}$ (11.1). Αντίστοιχα Wh ανά λεπτό υπολογίστηκαν διαιρώντας τις Wh με το χρόνο λειτουργίας. Παρατηρώντας τον πίνακα γίνεται άμεσα αντιληπτό ότι η μέγιστη τιμή Wh που μπορεί να υποστηρίξει το σύστημα για μπαταρίες με 3 κυψέλες είναι ≈ 22.3 . Αυτό φαίνεται ότι για μεγάλες τιμές χρήσης (%) από τα φώτα δεν αλλάζει η τιμή (δηλαδή δεν αυξάνεται η φωτεινότητα που μπορούν να παρέχουν). Συνεπώς επιλέχθηκαν μπαταρίες με 4 κυψέλες οι οποίες μπορούσαν να παρέχουν μεγαλύτερες τιμές ισχύς.

Σημαντικό ζητούμενο ήταν να βρεθεί πως γίνεται η «αντιστοίχιση» της χρήσης με την ισχύ για τις μπαταρίες ιόντων λιθίου (4S) που επιλέχθηκαν μετά από το προηγούμενο πείραμα. Για να επιτευχθεί αυτό υλοποιήθηκαν δύο πειράματα με την χρήση τροφοδοτικού το οποίο παρέχει τιμές ηλεκτρικού ρεύματος (Ampere) καθώς και

Voltage, από τα οποία με μια απλή πράξη πολλαπλασιασμού μπορεί να βρεθεί η ισχύς (Watt). Έγιναν δύο διαφορετικές μετρήσεις και οι δύο στο άνοιγμα του συστήματος όπως θα γίνει και στην αποστολή, η διαφορά τους είναι ότι στην μια περίπτωση (Σχήμα 3.23) η μέτρηση έγινε αμέσως μόλις άνοιξαν τα φώτα ενώ στη άλλη (Σχήμα 3.24) η μέτρηση έγινε μετά από ένα διάστημα δύο λεπτών. Η αιτία που έγινε η δεύτερη μέτρηση είναι διότι παρατηρήθηκε έντονη διακύμανση στις τιμές που αναγραφόντουσαν από το τροφοδοτικό αν δεν δινόταν χρόνος στο σύστημα να σταθεροποιηθεί. Κατά την εφαρμογή του συστήματος στο πεδίο επειδή τα φώτα θα άνοιγαν για ένα χρονικό διάστημα από 1-10 λεπτά ανάλογα τις ανάγκες κρίθηκε ότι πρέπει να υπάρχει και μια ενδεικτική τιμή για την συμπεριφορά τους μετά από ένα χρονικό διάστημα. Οι μετρήσεις προέκυψαν μεταβάλλοντας την χρήση από τα φώτα κατά 10% (δηλαδή αλλάζοντας το duty cycle όπως σχολιάστηκε στο κεφάλαιο του λογισμικού) και υπολογίζοντας τα Watt ανά περίπτωση. Παρατηρώντας τα διαγράμματα που προέκυψαν φαίνεται ότι η μεθοδολογία που ακολουθήθηκε ήταν όντως ορθή μιας και η ισχύς που υπολογίστηκε για τα φώτα πλησιάζει πολύ την ονομαστική τιμή του κατασκευαστή στα 60 Watt.

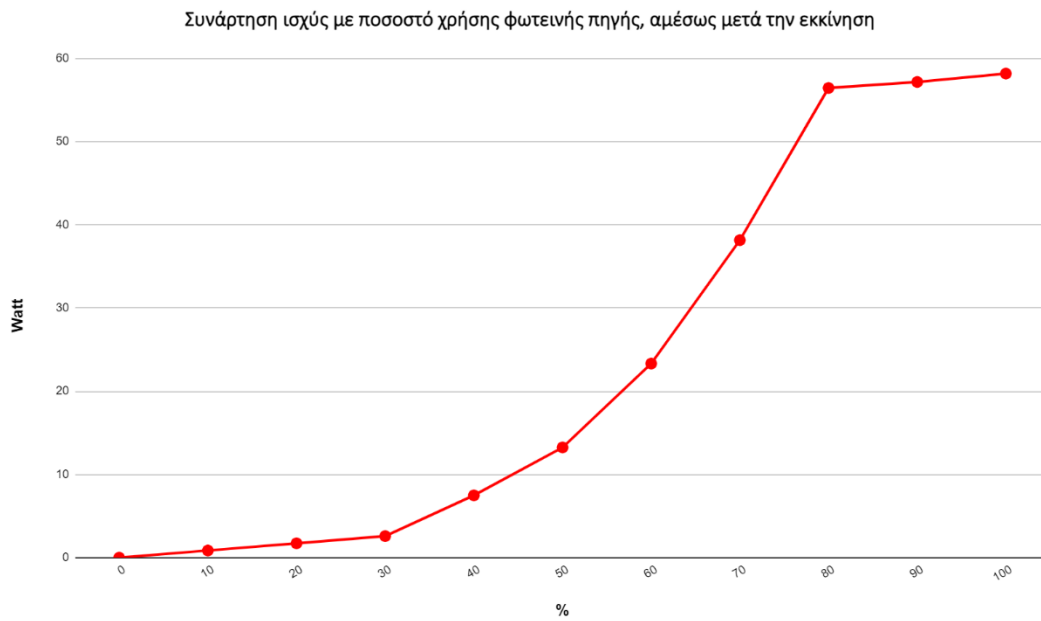
Παρατηρώντας τα γραφήματα εξάχθηκαν κάποια πολύ ενδιαφέροντα συμπεράσματα, όπως το γεγονός ότι όσο δίνεται χρόνος στο σύστημα να λειτουργήσει η ισχύς τελικά μειώνεται. Παρατηρήθηκε ακόμη μια «σταθεροποίηση» του ρεύματος που αντλούσαν τα φώτα. Εκτός αυτού φαίνεται ότι το ποσοστό χρήση της φωτεινής πηγής βάσει του και Πίνακα 3.3 ήταν λίγο πάνω από 50% συνεπώς αν είχαν επιλεχθεί μπαταρίες με τρεις μόνο κυψέλες σε σειρά δεν θα είχαμε την δυνατότητα να ξεπεράσουμε αυτό το όριο. Από το συγκεκριμένο πείραμα μας δόθηκε η δυνατότητα πρόβλεψης της ισχύς για διαφορετικά ποσοστά χρήσης από τα φώτα καθώς και συγχρόνως το μέγιστο όριο που μπορούμε να χρησιμοποιήσουμε όσον αφορά το ενεργειακό σκέλος μιας και οι μπαταρίες που είχαν τοποθετηθεί παρουσίαζαν πεπερασμένα όρια αποθήκευσης. Εκτός αυτού βάσει του συγκεκριμένου πειράματος μπόρεσε να γίνει ένας πρώιμος υπολογισμός της απαιτούμενης ενέργειας που θα απαιτούταν ανάλογα με την φύση του πειράματος ώστε γινόταν η επιλογή των βέλτιστων μπαταριών (η υλοποίηση των μπαταριών που παρουσιάστηκαν σε προηγούμενο κεφάλαιο έγινε βάση των αυτών των στοιχείων).

-Test 1- Μπαταρία Lipo 2200 mAh 11.1V 3S -χωρίς χρήση κάμερα-

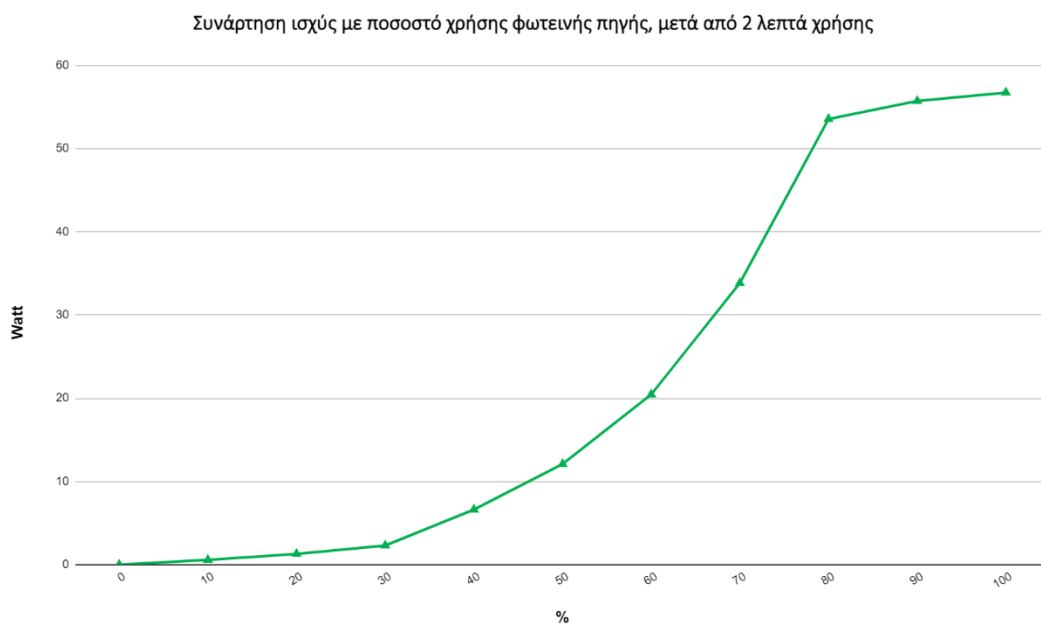
| | | | | | |
|--|------|------|------|------|------|
| <i>Αρχικά V</i> | 12.2 | 12.2 | 12.3 | 12.3 | 12.3 |
| <i>Τελικά V</i> | 11 | 10.8 | 10.8 | 10.8 | 10.8 |
| <i>mAh</i> <i>(που καταναλώθηκαν)</i> | 1915 | 2008 | 2000 | 2000 | 2000 |
| <i>Wh</i> | 21.3 | 22.3 | 22.2 | 22.2 | 22.2 |

| | | | | | |
|--|------|------|------|------|------|
| Χρόνος (min) | 63 | 70 | 80 | 80 | 115 |
| | 0.34 | 0.32 | 0.28 | 0.28 | 0.19 |
| Wh/min | | | | | |
| Ποσοστό (%) χρήσης από τα φώτα | 100 | 100 | 80 | 80 | 75 |
| Αριθμός από φώτα που χρησιμοποιήθηκαν | 4 | 4 | 4 | 4 | 4 |

Πίνακας 3.3: Βασικά στοιχεία που υπολογίστηκαν για την ισχύ που απαιτεί η φωτεινή πηγή.



Σχήμα 3.23: Παρουσίαση συνάρτηση ισχύς με ποσοστό χρήσης φωτεινής πηγής, αμέσως μετά την εκκίνηση.



Σχήμα 3.24: Παρουσίαση συνάρτηση ισχύς με ποσοστό χρήσης φωτεινής πηγής, μετά από 2 λεπτά χρήσης.

Πέραν από τις πειραματικές διατάξεις για τις μπαταρίες έπρεπε να γίνουν και υπολογισμοί για το μέγεθος των παραγόμενων δεδομένων που μπορούσαν να συλλεχθούν από την κάμερα. Βασικότατο κριτήριο όσον αφορά τα δεδομένα ήταν το format στο οποίο αυτά θα υλοποιούνταν. Το πρώτο format που αξιολογήθηκε ήταν το png, εξαιτίας του lossless compression, το οποίο σημαίνει ότι η συμπίεση των εικόνων γίνεται χωρίς να χαθεί πληροφορία. Επίσης, τα jpg αποτέλεσαν μια άλλη αξιολογημένη επιλογή αλλά απορρίφθηκαν λόγω του ότι η συμπίεση τους οδηγεί στο να χαθεί πληροφορία. Τόσο τα jpg όσο και τα png παρουσίασαν ένα πολύ σημαντικό ζήτημα κατά την υλοποίησή τους, το οποίο αφορούσε την «πίεση» που τόσο η δημιουργία τους όσο και η συμπίεση τους ασκούσαν στον πολύ αδύναμο επεξεργαστή του μικροϋπολογιστή που επιλέχθηκε. Εξαιτίας αυτού ο αριθμός των fps (λήψεις ανά δευτερόλεπτο) σε ικανοποιητική ανάλυση που μπορούσε να επιτύχει το σύστημα ήταν πάρα πολύ μικρός (περίπου 2-3). Για αυτό το λόγο επιλέχθηκε να υλοποιηθούν τα δεδομένα σε μορφή raw (όπως φαίνεται και από τον κώδικα που παρουσιάστηκε στο προηγούμενο κεφάλαιο), όπου η απαιτούμενη υπολογιστική ισχύ ήταν αισθητά μικρότερη αλλά το μέγεθος των αρχείων αρκετά μεγαλύτερο (αν στα άλλα format γίνει συμπίεση).

Στον πίνακα 3.4 που ακολουθεί φαίνεται η διαφοροποίηση της λήψης png (χωρίς compression) format σε αντίθεση με την raw λήψη (12 bit) για κοινές διαστάσεις εικόνας. Ο τρόπος υπολογισμού του μεγέθους των αρχείων ανά λήψη ήταν : για την png
$$\frac{\text{Οριζόντια ανάλυση} \times \text{Κατακόρυφη ανάλυση} \times \text{Αριθμός καναλιών}}{1024 \times 1024}$$
 ενώ για την raw
$$\frac{\text{Stride} \times \text{Κατακόρυφη ανάλυση} \times \text{Αριθμός καναλιών}}{1024 \times 1024}$$
. Δύο εύλογα ερωτήματα που προκύπτουν είναι αρχικά τι είναι το 1024×1024 στον παρονομαστή, το οποίο προκύπτει λόγω της μετατροπής από bit σε byte και τι είναι το stride στις raw εικόνες. Το Stride προκύπτει από την μετατροπή της 12 bit εικόνας σε 8 bit. Η αιτία που τα MB ανά sec στην περίπτωση της Raw είναι πράσινο ενώ της png κόκκινο οφείλεται στο ότι το σύστημα βάσει της θύρας μεταφοράς δεδομένων που έχει (micro USB) είχε την δυνατότητα μεταφορά μόλις 25-30 MB/sec, γεγονός που μας οδήγησε στο συμπέρασμα ότι πέντε png εικόνες δεν θα μπορούσε να αποθηκευτούν εντός ενός δευτερολέπτου χωρίς να γίνει συμπίεση. Διαδικασία που όμως ήταν απαγορευτική στο υπολογιστικό σύστημα που επιλέχθηκε. Συνεπώς, έγινε αντιληπτό ότι η μοναδική ικανοποιητική επιλογή ήταν οι raw εικόνες. Τέλος όσον αφορά την ανάλυση βασικό ζητούμενο όπως σχολιάστηκε και στην αρχή του κεφαλαίου ήταν ο εντοπισμός των φουσαλίδων. Αυτό όμως συνεπάγεται την ανάγκη για όσο το δυνατόν περισσότερες λήψεις ανά δευτερόλεπτο

(με ικανοποιητική ανάλυση ώστε να μπορούν να ανιχνευθούν) έτσι κρίθηκε ότι η ανάλυση 2028×1520 ήταν πολύ ικανοποιητική (τουλάχιστον σε αρχικό στάδιο).

| | Raw 12 bit | Png |
|---------------------|------------|------|
| Οριζόντια ανάλυση | 2028 | 2028 |
| Κατακόρυφη Ανάλυση | 1520 | 520 |
| Αριθμός Καναλιών | 1 | 3 |
| MB/λήψη | 4.46 | 8.82 |
| Λήψεις/δευτερόλεπτο | 5 | 5 |
| MB/δευτερόλεπτο | 22.27 | 44 |

Πίνακας 3.3: Υπολογισμένα μεγέθη για διαφορετικά format αποθήκευσης λήψεων.

Συνεχίζοντας την ανάλυση στο ίδιο μοτίβο η επόμενη βασική επιλογή ήταν να επιλεγεί ο χρόνος που ο μικροϋπολογιστής θα βρισκόταν σε λειτουργία. Για να γίνει αυτό έπρεπε να ληφθεί υπόψιν η κατανάλωση (η οποία θα αξιολογηθεί στην συνέχεια), το μέγεθος του αποθηκευτικού χώρου (το οποίο επίσης θα αξιολογηθεί στην συνέχεια) καθώς και χαρακτηριστικά των αντικειμένων που θα αποτυπωθούν σε συνδυασμό με τις ικανότητες του μικροϋπολογιστή. Με συνεχή πειραματισμό βρέθηκε ότι έπρεπε να ξεκινήσει η λειτουργία από τα φώτα και την κάμερα τουλάχιστον δύο με τρία λεπτά μετά το άνοιγμα του μικροϋπολογιστή, καθώς παρατηρήθηκαν δύο βασικά προβλήματα. Αρχικά τα φώτα τρεμόπαιζαν σε περίπτωση που δεν είχε περάσει αυτό το χρονικό διάστημα από την εκκίνηση του υπολογιστή και επίσης η ενεργειακή κατανάλωση παρουσίαζε μια αλλοπρόσαλλη συμπεριφορά (η οποία ήταν δύσκολο να εξηγηθεί). Σαν ασφαλή τιμή επιλέχθηκε τα φώτα και η κάμερα να ανοίγουν τρία λεπτά μετά την εκκίνηση, ενώ ο σένσορας και οι σερβοκινητήρες μόλις 20 δευτερόλεπτα και ένα λεπτό αντίστοιχα, καθώς παρατηρήθηκε ότι δεν επηρέαζαν σε τίποτα το σύστημα.

Σε αυτό το στάδιο αποφασίστηκε ότι η αυτονομία του συστήματος έπρεπε να είναι ένα διάστημα τουλάχιστον τεσσάρων μηνών. Ωστε να επιτευχθεί αυτό υλοποιήθηκε πείραμα όπου η κάμερα και τα φώτα θα λειτουργούσαν δύο λεπτά ενώ το σύστημα συνολικά έμενε σε λειτουργία για οκτώ. Τρία λεπτά η εκκίνηση, δύο λεπτά η συλλογή δεδομένων και τρία λεπτά για να εξασφαλίσουμε ότι όλα τα προγράμματα είχαν υλοποιηθεί και τερματιστεί σωστά, μιας και η κατανάλωση του μικροϋπολογιστή ήταν αμελητέα σε σχέση με τα άλλα στοιχεία του συστήματος. Σε αυτό το σημείο αποφασίστηκε να εξαχθούν οι μετρήσεις που συνέλεξε το power sensor module, για δύο τέτοια διαστήματα ώστε να υπολογιστεί μια ρεαλιστική κατανάλωση ενέργειας του συστήματος. Στον πίνακα 3.4 παρουσιάζονται αυτά τα στοιχεία. Μεγάλο ενδιαφέρον παρουσίασε ότι η διαφορά στα σύνολα των Wh αλλά και των mAh στα δύο χρονικά διαστήματα ήταν μικρότερη από 1% (η μια από την άλλη), συνεπώς κρίθηκε ότι σε αυτό το στάδιο τα δεδομένα ήταν αρκετά αξιόπιστα. Επίσης τα Watt που προέκυψαν

για τα ανοιχτά φώτα (16.40-16,82) λαμβάνοντας υπόψιν το Σχήμα 3.24 αντιστοιχούν σε τιμές χρήσης από 55-60% ενώ πράγματι το πείραμα έγινε με 50% χρήση της φωτεινής πηγής. Η αιτία της μικρής μετατόπισης προς τα πάνω οφειλόταν κυρίως ότι στο ίδιο χρονικό διάστημα λειτουργούν όλα τα όργανα και οι μετρητές του συστήματος.

Για να βρεθεί το διάστημα που οι μπαταρίες μπορούν να παρέχουν ενέργεια στο σύστημα με ασφάλεια ακολουθήθηκε η εξής μεθοδολογία: Οι μπαταρίες που επιλέχθηκαν είχαν 14.8 Volt (ονομαστική τιμή τάση, 4 κυψέλες των 3.7 Volt) και ήταν 36000 mAh. Από αυτά τα δύο στοιχεία μπορούν να υπολογιστούν οι $Wh = \frac{mAh \times Volt}{1000} = 532.8$. Από τον πίνακα παίρνουμε την τιμή που προέκυψε για τις Wh ανά κύκλο λειτουργίας ως 0.5766. Σε αυτό το σημείο πρέπει σχολιαστεί ότι στις συγκεκριμένες μπαταρίες όσο βρίσκονταν σε λειτουργία, η τάση τους άρχιζε να φθίνει, με συνέπεια η ισχύς (Watt) να μειώνεται. Αυτό οδηγεί στην μείωση των Wh όσο η λειτουργία της μπαταρίας πλησιάζει προς το τέλος των δυνατοτήτων της. Για αυτό και οι κύκλοι λειτουργίας του συστήματος που προέκυψαν σε αυτό το στάδιο, ελέγχθηκαν και με ένα τελικό πείραμα που θα παρουσιαστεί στην συνέχεια. Για λόγους ασφαλείας επιλέχθηκε να χρησιμοποιηθεί το 80% των Wh που παρείχε η μπαταρία δηλαδή 426.24. Διαιρώντας το συγκεκριμένο αριθμό με τις Wh ανά κύκλο λειτουργία προέκυψε ότι το σύστημα μπορούσε να υποστηρίξει 739 κύκλους λειτουργίας.

| Volt | Ampere | Watt (Volt×Amp) | Wh $\frac{\Sigma Watt}{120}$ | mAh $\frac{(\Sigma Ampere) \times 1000}{120}$ |
|-------|--------|--------------------|---------------------------------|--|
| 13.82 | 0.1 | 1.33 | 0.0115 | 42.17 |
| 13.84 | 0.005 | 0.07 | 0.0006 | |
| 13.83 | 0.005 | 0.07 | 0.0006 | |
| 13.83 | 0.01 | 0.14 | 0.0012 | |
| 13.83 | 0.015 | 0.21 | 0.0018 | |
| 13.83 | 0.01 | 0.14 | 0.0012 | |
| 13.67 | 1.23 | 16.82 | 0.1401 | |
| 13.67 | 1.22 | 16.68 | 0.1390 | |
| 13.67 | 1.22 | 16.68 | 0.1390 | |
| 13.67 | 1.22 | 16.68 | 0.1390 | |
| 13.82 | 0.005 | 0.07 | 0.0006 | |
| 13.82 | 0.005 | 0.07 | 0.0006 | |
| 13.82 | 0.005 | 0.07 | 0.0006 | |
| 13.83 | 0.005 | 0.07 | 0.0006 | |
| 13.82 | 0.005 | 0.07 | 0.0006 | |
| | | 69.19 | 0.5766 | |
| 13.81 | 0.0014 | 0.19 | 0.0016 | |
| 13.82 | 0.02 | 0.28 | 0.0023 | |

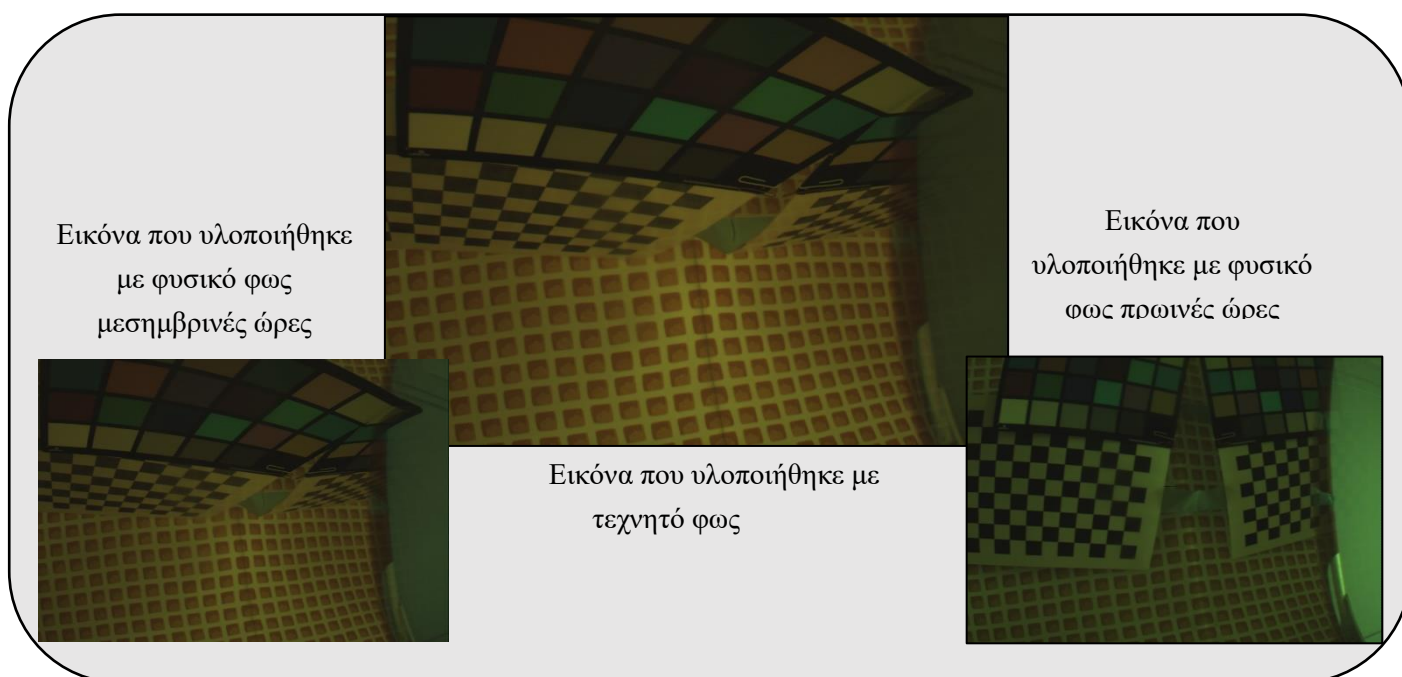
| | | | |
|-------|------|-------|--------|
| 13.82 | 0.02 | 0.28 | 0.0023 |
| 13.82 | 0.02 | 0.28 | 0.0023 |
| 13.82 | 0.02 | 0.28 | 0.0023 |
| 13.82 | 0.01 | 0.14 | 0.0012 |
| 13.66 | 1.23 | 16.80 | 0.1400 |
| 13.66 | 1.23 | 16.80 | 0.1400 |
| 13.66 | 1.21 | 16.53 | 0.1377 |
| 13.66 | 1.2 | 16.40 | 0.1366 |
| 13.82 | 0.01 | 0.14 | 0.0012 |
| 13.82 | 0.01 | 0.14 | 0.0012 |
| 13.82 | 0.01 | 0.14 | 0.0012 |
| 13.82 | 0.01 | 0.14 | 0.0012 |
| 13.82 | 0.01 | 0.14 | 0.0012 |
| | | 68.65 | 0.5721 |

Πίνακας 3.4: Έλεγχος της ενεργειακής συμπεριφοράς του συστήματος μέσα από το log του powersensor module.

Αφού υπολογίστηκαν όλα αυτά τα στοιχεία για την ενεργειακή επάρκεια πρέπει να σχολιαστεί ότι για χρήση 739 κύκλων όπου η κάμερα θα λειτουργούσε 2 λεπτά απαιτούνταν 1.97 TB μνήμης. Όμως ο εξωτερικός σκληρός δίσκος ήταν μόλις 2 TB, έτσι για λόγους ασφαλείας επιλέχθηκε να ελεγχθεί η δυνατότητα του συστήματος για 650 κύκλους (όπου θα απαιτούνταν μόλις (1.66 TB)). Σε αυτό το σημείο υλοποιήθηκε το τελικό πείραμα όσον αφορά την επάρκεια της ενέργειας και του αποθηκευτικού χώρου. Στο πλαίσιο του πειράματος περατώθηκε μια «ψευδοαποστολή» 650 κύκλων εντός μερικών ημερών. Το σύστημα άνοιξε με την χρήση του Witty για τον προκαθορισμένο χρόνο των 8 λεπτών και έκλεινε για δύο λεπτά επαναλαμβάνοντας την διαδικασία αυτή 650 φορές που ήταν και ο τελικός στόχος. Πράγματι ο εξωτερικός σκληρός γέμισε στο μέγεθος που αναμενόταν από τις προδιαγραφές και το Voltage των μπαταριών ήταν πάνω από τα 13 Volt που ήταν η τιμή ασφαλείας (ώστε οι μπαταρίες να μπορούν να επαναφορτιστούν και να ξαναχρησιμοποιηθούν). Συνεπώς κρίθηκε ότι το σύστημα ήταν έτοιμο για πραγματική αποστολή. Σημαντικό είναι να σημειωθεί ότι εύκολα μπορούσαμε να υπολογίσουμε πόσους κύκλους θα άντεχε το σύστημα σε μελλοντική αποστολή με μεγαλύτερο ποσοστό χρήσης από τα φώτα, το οποίο σε πραγματική αποστολή θα είναι αναγκαίο. Αυτό επιλέχθηκε την στιγμή των ελέγχων εντός του νερού μιας και μέσα από το streaming των εικόνων είχαμε την δυνατότητα να ελέγξουμε αν η φωτεινότητα θα ήταν πράγματι επαρκής. Για αυτό το λόγο υπολογίστηκε ότι για 70% ποσοστό χρήσης από τα φώτα το οποίο ήταν και το μέγιστο που ήμασταν διατεθειμένοι να θέσουμε, οι κύκλοι που άντεχε το σύστημα ήταν 350. Για λόγους ασφαλείας ακολουθήσαμε και μια δεύτερη μεθοδολογία ελέγχου όπου αθροίσαμε το ρεύμα (A) που προέκυψε από το power sensor και βρήκαμε πόσους κύκλους μπορούσε να κάνει βάσει της ονομαστικής χωρητικότητας σε (A) των

μπαταριών. Η συγκεκριμένη τιμή ήταν πολύ κοντινή σε αυτή που υπολογίστηκε με την άλλη μεθοδολογία (π.χ για 70% υπολογίστηκαν 345 κύκλοι έναντι 350).

Πέρα από τα πειράματα που αναλύθηκαν, ενδιαφέρον παρουσίασε η αξιολόγηση της λειτουργίας της κάμερας σε περιβάλλον που προσομοιάζε τις συνθήκες εντός της θάλασσας. Για να υλοποιηθεί αυτό χρησιμοποιήθηκε η πισίνα του εργαστηρίου σε συνδυασμό με το σκέπαστρο της ώστε να υπάρχει πλήρης έλλειψη φωτός όπως θα συμβαίνει και σε μεγάλα βάθη στο θαλάσσιο περιβάλλον. Στο σχήμα 3.25 φαίνονται τρεις εικόνες, δύο από τις οποίες λήφθηκαν με φυσικό φως διαφορετικές ώρες της μέρας και μια η λήφθηκε με τεχνητό φως (βάλαμε σε πολύ χαμηλή τιμή λειτουργίας τα φώτα γιατί αλλιώς δημιουργούσαν αντανακλάσεις). Βλέποντας τις γίνεται αντιληπτό ότι το αποτέλεσμα με το τεχνητό φως είναι πολύ ικανοποιητικό (αν και η απόσταση που λήφθηκαν ήταν περί του ενός μέτρου ενώ στην πραγματικότητα θα υπάρχουν και μεγαλύτερες αποστάσεις). Ο λόγος που το συγκεκριμένο πείραμα δεν ήταν πλήρως επαρκές για να αποφανθούμε ότι η συγκεκριμένη τιμή φωτεινότητας (με την οποία τελεστικέ το πείραμα) ήταν αρκετή για την θάλασσα είχε να κάνει με παράγοντες όπως η οπισθοσκέδαση που θα υπάρχει έντονη στο θαλάσσιο περιβάλλον, η αλατότητα του νερού, η διαφορά γεωμετρίας αλλά και γενικότερα της σκηνής κ.α.



Σχήμα 3.25: Πειραματικές εικόνες που λήφθηκαν (BGR).

Ένα ζήτημα που παρατηρήθηκε κατά τη διάρκεια των πειραμάτων αφορούσε τους σερβοκινητήρες που κινούσαν την κάμερα σε δύο άξονες. Ο τρόπος λειτουργίας τους ήταν: ότι μετατοπίζονταν με βάση μοίρες (από 0° έως 180°) αλλά το θεωρητικό τους κέντρο (90°) εξαρτιόταν από τον τρόπο που εγκαταστάθηκαν στο σύστημα. Ως πραγματικό κέντρο τους (το οποίο διέφερε από το θεωρητικό) λογίστηκε όταν για κάποιες συγκεκριμένες μοίρες (στον κάθε άξονα) η κάμερα βρισκόταν στην κεντρική της θέση σε σχέση με το dome (Σχήμα 3.26). Παρότι εν δυνάμει κάθε κινητήρας μπορούσε να κινηθεί από 0-180°, στην πραγματικότητα δεν ισχύει αυτό καθώς μετά από κάποιες μοίρες η κάμερα ερχόταν σε επαφή με το περίβλημα (Σχήμα 3.26), με συνέπεια να παρουσιαζόταν η πιθανότητα να γίνουν ζημιές στους κινητήρες. Για αυτό έγινε «χαρτογράφηση» των θέσεων που μπορούσε να πάρει η κάμερα χωρίς να «βρίσκει» στο περίβλημα. Η συγκεκριμένη διαδικασία έγινε βρίσκοντας τις ακραίες δυνατές θέσεις σε κάθε άξονα και στην συνέχεια «ελέγχοντας» τις ενδιάμεσες θέσεις που προκύπτουν ανάμεσα στις ακραίες τιμές. Τελικά προέκυψε ο χάρτης που φαίνεται στο Σχήμα 3.27 όπου οι θέσεις με το πράσινο χρώμα είναι αυτές που με ασφάλεια μπορεί να μετακινηθεί η κάμερα. Ένα ζήτημα που εμφανίστηκε σε αυτό το σημείο είναι ο τρόπος που θα μετακινείται η κάμερα από την μια θέση στην άλλη διότι υπάρχουν παραδείγματα όπου αν δεν μετακινηθεί με σωστό τρόπο η κάμερα τελικά θα «χτυπήσει» στο dome. Ένα τέτοιο παράδειγμα είναι αν η θέση του Servo 1 έχει τιμή 90° ενώ του Servo 2 100° (θέση 90°,100°) και στόχος είναι η νέα θέση να είναι η (80°,90°), αν πρώτα κινηθεί ο Servo 2 η κάμερα θα έρθει σε επαφή με το dome μιας και θα πάει στην θέση 100,100° αρχικά. Για αυτό το λόγο δημιουργήθηκε αλγόριθμος βάσει του οποίου: αν το σύστημα είχε τιμή 60° ή 100° για το σέρβο 1 πρώτα θα κινηθεί ο σέρβο 1 και μετά ο 2, ενώ για όλες τις άλλες τιμές (πρακτικά τις τρεις κεντρικές του διαγράμματος, για 70°, 80° και 90° στον σέρβο 1) πρώτα θα κινηθεί ο σερβοκινητήρας 2. Με αυτό τον τρόπο εξασφαλίστηκε, όσο το δυνατόν πιο απλά, ότι δεν θα έρθει σε επαφή η κάμερα με το περίβλημα της.



Σχήμα 3.26: Αριστερά: Η κάμερα έρχεται σε επαφή με το περίβλημα της. Δεξιά: Η κάμερα βρίσκεται στην σχετική κεντρική της θέση.

| PTZ 50 | Servo 1 (°) | | | | | | | |
|-------------|-------------|----|----|----|----|-----|-----|-----|
| | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120 |
| Servo 2 (°) | 50 | x | x | x | x | x | x | x |
| | 60 | x | x | ✓ | x | x | x | x |
| | 70 | x | ✓ | ✓ | ✓ | x | x | x |
| | 80 | ✓ | ✓ | ✓ | ✓ | ✓ | x | x |
| | 90 | ✓ | ✓ | ✓ | ✓ | ✓ | x | x |
| | 100 | ✓ | ✓ | ✓ | ✓ | ✓ | x | x |
| | 110 | ✓ | ✓ | ✓ | ✓ | ✓ | x | x |
| | 120 | x | ✓ | ✓ | ✓ | ✓ | x | x |

Σχήμα 3.27: Χάρτης «ασφαλών» θέσεων της κάμερας.

Σε αυτό το στάδιο ελέγχθηκαν τα τελευταία ζητήματα του συστήματος. Αρχικά, αν η σύνδεση μπορούσε να γίνει μέσω του access point (Σχήμα 3.28) εντός του νερού, διότι το σήμα «εξασθενεί» πολύ γρήγορα στο συγκεκριμένο μέσο. Πράγματι μέσα σε αποστάσεις μικρότερες από μισό μέτρο η σύνδεση λειτούργησε πολύ ικανοποιητικά και μπορούσαμε να δούμε στιγμιότυπα ώστε να ελέγξουμε την θέση της κάμερας, το οπτικό ζουμ αλλά και την εστίαση. Έγινε έλεγχος ότι η εστίαση (στο άπειρο) και το οπτικό ζουμ λειτούργησαν κανονικά με τον κώδικα που υλοποιήθηκε (Σχήμα 3.29), κάτι το οποίο όπως φαίνεται πράγματι συνέβη. Εκτός αυτού έγινε ένας τελικός έλεγχος σε όλες τις συνδεσμολογίες μπαταριών με τις κάμερες. Τέλος, έγινε ο έλεγχος στεγανότητας όπου δημιουργήσαμε κενό αέρος εντός των υποβρυχίων περιβλημάτων και παρατηρήσαμε αν υπήρχαν αλλαγές στην πίεση. Αφού υλοποιήθηκαν όλα αυτά το σύστημα ήταν πράγματι έτοιμο να ανταπεξέλθει σε πραγματική αποστολή.



Σχήμα 3.28: Σημείο σύνδεσης που κατασκευάστηκε



Σχήμα 3.29: Παραδείγματα με σωστό και λάθος εστίαση με διαφορετικά ζουμ

3.5 Διαδικασίες επί του πεδίου

Σε αυτό το υποκεφάλαιο θα παρουσιαστούν οι διαδικασίες που ακολουθήθηκαν για την εξαγωγή των δεδομένων και το ερευνητικό ταξίδι διεξάχθηκε. Θα επεξηγηθεί η συνεργασία μας με ένα σύνολο ερευνητών στην προσπάθεια να επιτευχθούν τα βέλτιστα αποτελέσματα. Επίσης, θα τονιστεί η ανάγκη εξοπλισμού και οικονομικής υποστήριξης ώστε να υλοποιηθούν εργασίες που θα παρουσιάσουν ουσιαστική συνεισφορά στον κλάδο τους. Το συγκεκριμένο τμήμα δεν θα μπορούσε να επιτευχθεί σε καμιά περίπτωση χωρίς τις παροχές του εργαστηρίου τηλεπισκόπησης της ΣΑΤΜ. Τα ταξίδια που θα σχολιαστούν πραγματοποιήθηκαν στην Μήλο και την Σαντορίνη (συγκεκριμένα στο υποθαλάσσιο ηφαίστειο του Κολούμπο).

Το ταξίδι που έγινε πρώτα ήταν στην Μήλο και αποτέλεσε μια πολύ καλή ευκαιρία για ελέγχους στο σύστημα, μιας και οι αποστολές αφορούσαν αρκετά ρηγά νερά άρα εξάχθηκαν συμπεράσματα χωρίς να χρειαστεί περεταίρω εξοπλισμός. Εκτός αυτού σε αυτό το ταξίδι την τελευταία μέρα συλλέξαμε και με την χρήση μικρού ROV τα δεδομένα που παρουσιάζονται στην παρούσα διπλωματική εργασία. Την πρώτη μέρα του ταξιδιού τοποθετήθηκε το σύστημα εντός του νερού και έγινε προσπάθεια

αλληλεπίδρασης μέσω SSH. Κατά αυτή την διαδικασία παρατηρήθηκε ότι οι αντένες του πομπού μας θα έπρεπε να βρίσκονται πολύ κοντά στο προστατευτικό περίβλημα του συστήματος ώστε να επιτευχθεί σύνδεση. Συνεχίσαμε κάνοντας ελέγχους για τα φώτα, δοκιμάζοντας σε τι ποσοστό φωτεινότητας παρείχαν ικανοποιητική φωτεινότητα αλλά και το πόση ενέργεια καταλάωναν, μιας και αυτή η διαδικασία είχε γίνει πειραματικά μόνο στην πισίνα. Την επόμενη μέρα έγιναν έλεγχοι της ποιότητας των εικόνων που μας παρείχε το σύστημα ραδιομετρικά και γεωμετρικά. Τελικά, την τελευταία μέρα χρησιμοποιήθηκε το ROV του εργαστηρίου για να συλλεγούν τα δεδομένα που παρουσιάζονται στο κεφάλαιο 4.

Η πιο ουσιαστική αποστολή (κυρίως για το μέλλον), από την οποία θα γίνει περισυλλογή δεδομένων περίπου στα μέσα του Νοέμβρη (και τα οποία επίσης θα αναλυθούν, αλλά όχι στο πλαίσιο της συγκεκριμένης εργασίας), έγινε στο Κολούμπο στις 13/06/2023. Σε αυτή την αποστολή η οποία υλοποιήθηκε στα πλαίσια του ερευνητικού έργου Santory χρησιμοποιήθηκε το ωκεανογραφικό ερευνητικό σκάφος ΑΙΓΑΙΟ του ΕΛ.ΚΕ.Θ.Ε. (Σχήμα 3.30) το οποίο εξοπλίστηκε με μεγάλο ROV (Σχήμα 3.31). Χρησιμοποιώντας το ROV είχαμε την δυνατότητα τοποθέτησης όλων των εξοπλισμών του ερευνητικού έργου αλλά και του δικό μας συστήματος σε μεγάλα βάθη (περί των 500 μέτρων), που υφίστανται στην περιοχή μελέτης. Από προηγούμενες ερευνητικές αποστολές στην περιοχή, υπήρχε χάρτης αναγλύφου καθώς και γνώση της ενεργής περιοχής, η οποία παρουσίαζε πολλές καμινάδες, που παρήγαγαν μεγάλο αριθμό φυσαλίδων. Το σύστημα ενσωματώθηκε από την ερευνητική μας ομάδα (Σχήμα 3.32) στο ROV και η αποστολή ξεκίνησε ώστε να συλλεχθούν δεδομένα. Στόχος, ήταν αρχικά να συλλεγεί κάποιος αριθμός δεδομένων για επεξεργασία μετά το πέρας της αποστολής αλλά και να γίνει τοποθέτηση των οργάνων για μελλοντική αξιολόγηση.



Σχήμα 3.30: Το ερευνητικό καράβι του ΕΛΚΕΘΕ.

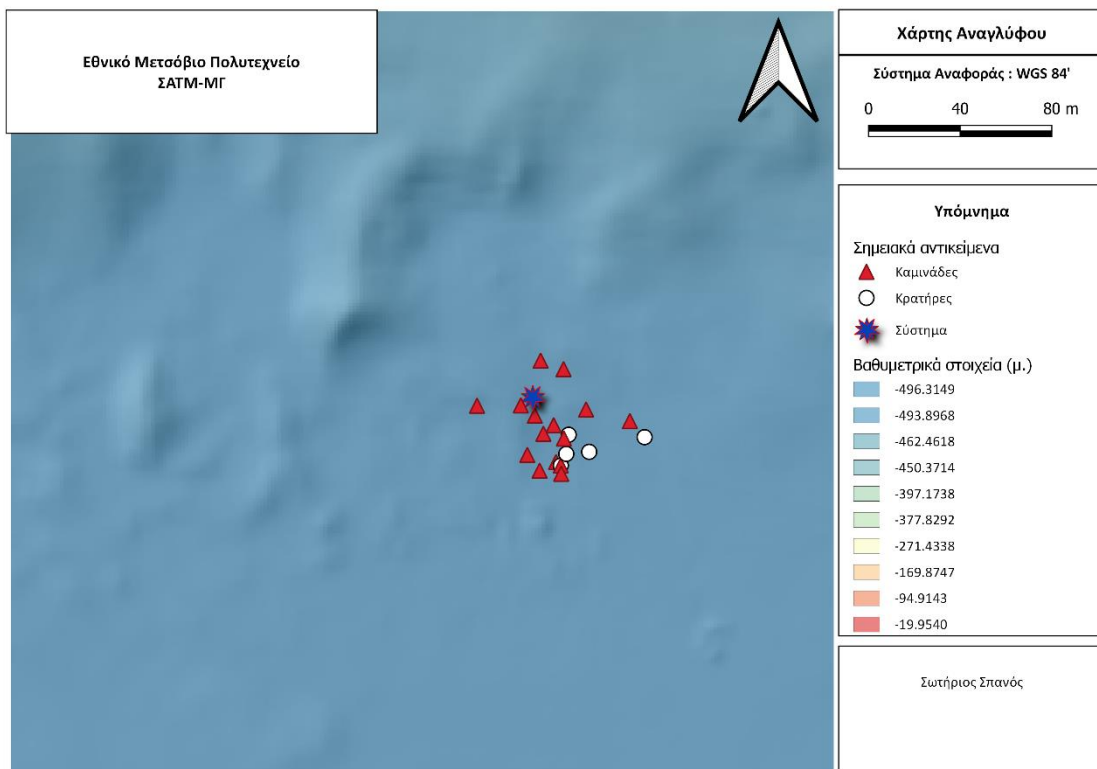


Σχήμα 3.31: Το ROV που χρησιμοποιήθηκε.



Σχήμα 3.32: Αριστερά: Υλοποίηση της βάσης του συστήματος μαζί με τον κύριο Καράντζαλο. Δεξιά: Τοποθέτηση των συστημάτων με τον επιστημονικό υπεύθυνο.

Για να επιτευχθεί η τελική τοποθέτηση των συστημάτων έπρεπε να γίνει ενδελεχής χαρτογράφηση της περιοχής ώστε το σύστημα να τοποθετούνταν σε όσο το δυνατόν πιο ενεργή περιοχή, ιδανικά δίπλα σε μια μεγάλη καμινάδα. Στο Σχήμα 3.33 φαίνεται η περιοχή που επιλέχθηκε καθώς και η θέση του συστήματος βάσει των συντεταγμένων εντός της ηφαιστειογενούς περιοχής. Μεγάλη δυσκολία είχε η πρόσβαση στο σύστημα μας, μιας και έπρεπε το ROV να έρθει σε πολύ κοντινή θέση στην τοποθετημένη κάμερα (Σχήμα 3.34) ώστε να υπάρξει η απαραίτητη επικοινωνία μέσω SSH, παρόλα αυτά με επιδέξιους χειρισμούς και αρκετή υπομονή οι οδηγοί του ROV κατάφεραν να μας παρέχουν τα μέσα ώστε να αλληλοεπιδράσουμε με το σύστημα και να ελέγξουμε την εστίαση, τα φώτα αλλά και να στρέψουμε την κάμερα προς την κατεύθυνση των φουσαλίδων.



Σχήμα 3.33: Υποτυπώδης χάρτης σημείων ενδιαφέροντος.



Σχήμα 3.34: Προσπάθεια επικοινωνίας με το σύστημα μας σε βάθος 490 μέτρων

Συνολικά, το συγκεκριμένο ερευνητικό ταξίδι μας έμαθε πολλά τόσο για τις δυνάμεις όσο και για τις αδυναμίες του συστήματος μας, οι οποίες αφορούσαν σε μεγάλο βαθμό τον τρόπο επικοινωνίας με τον μικροϋπολογιστή. Η αποστολή παρουσίασε μεγάλη επιτυχία καθώς το σύστημα πράγματι λειτούργησε και σύλλεξε δεδομένα, τόσο πάνω στο ROV όσο και κατά την τοποθέτηση του σε βάθος 490 μέτρων. Όταν συλλέξουμε τα μελλοντικά δεδομένα θα μπορέσουμε να εξάγουμε και άλλα συμπεράσματα κυρίως ως προς την ανθεκτικότητα του συστήματος σε πολύ θερμά και όξινα περιβάλλοντα, αλλά και ως προς την περιοδικότητα των φυσαλίδων από την πηγή τους και του ρυθμού που αυτές αυξάνονται ή μειώνονται.

Κεφάλαιο 4

Μεθοδολογίες που εφαρμόστηκαν στα δεδομένα

Αυτό το κεφάλαιο αποτελεί συνέχεια των αποστολών που έγιναν στο πεδίο και παρουσιάζει τα δεδομένα που συλλέχθηκαν. Ο διαχωρισμός του θα γίνει σε τρία τμήματα, αρχικά θα παρουσιαστούν τα δεδομένα και θα αναλυθούν οι διαδικασίες που έγιναν ώστε αυτά να είναι αξιοποιήσιμα. Στο επόμενο τμήμα θα αναλυθεί το νευρωνικό δίκτυο που υλοποιήθηκε, καθώς και οι παράμετροι που επιλέχθηκαν, ώστε να επιτευχθεί η ζητούμενη ανίχνευση των φυσαλίδων. Τελικά, θα παρουσιαστούν τα αποτελέσματα του μοντέλου που κατασκευάστηκε και θα αναλυθούν οι διάφορες μετρικές αξιολόγησης. Το συγκεκριμένο κεφάλαιο έχει πολύ μεγάλη αξία καθώς αρχικά στην υπάρχουσα βιβλιογραφία (η οποία παρουσιάστηκε αναλυτικά στο δεύτερο κεφάλαιο) δεν υπάρχουν σετ δεδομένων για υλοποίηση νευρωνικών δικτύων ανίχνευσης φυσαλίδων. Το μόνο που εμφανίζεται στην βιβλιογραφία είναι αναλύσεις με συνθετικά δεδομένα ή και πραγματικά δεδομένα τα οποία όμως δεν παρέχονται στον αναγνώστη ώστε να μπορέσει και αυτός να τα αξιοποιήσει σε κάποια δικιά του αρχιτεκτονική.

4.1 Εισαγωγή

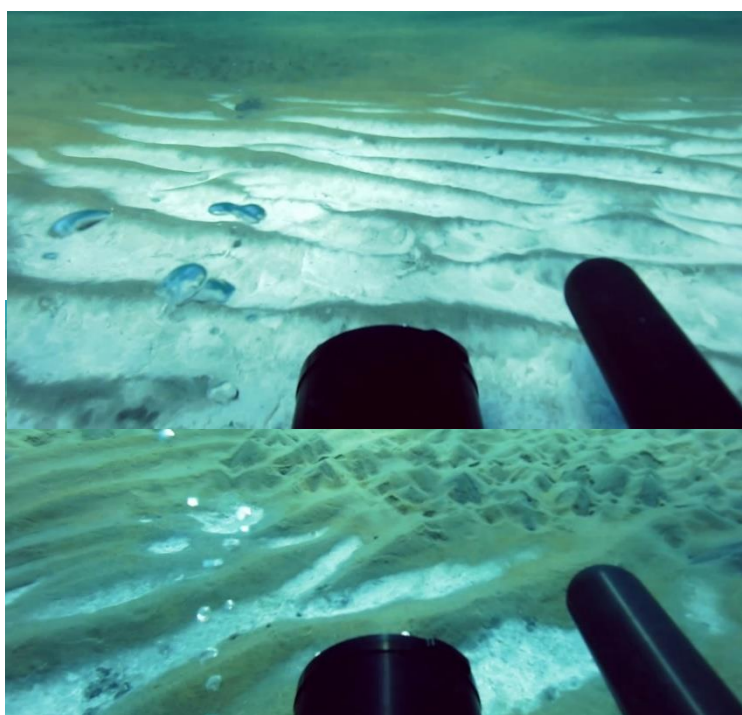
Η συλλογή των υποθαλάσσιων δεδομένων, όπως φάνηκε και στο κεφάλαιο 3 ήταν ένα αρκετά σύνθετο ζήτημα, για αυτό σε τέτοιες συνθήκες ο στόχος του μελετητή

οφείλει να είναι η εξαγωγή της μέγιστης πληροφορίας που αυτά μπορούν να παρέχουν. Έτσι σε αυτό το κεφάλαιο έγιναν και κάποιες επεξεργασίες που δεν ήταν απαραίτητες για το νευρωνικό δίκτυο που υλοποιήθηκε, παρόλα αυτά δίνουν ακόμα μεγαλύτερη αξία στο σετ δεδομένων που κατασκευάστηκε. Καθώς με αυτόν τον τρόπο μελλοντικά άλλοι ερευνητές μπορούν να εφαρμόσουν δικές του αρχιτεκτονικές και να «πάρουν» ίσως καλύτερα ή διαφορετικά αποτελέσματά από αυτά που θα παρουσιαστούν. Το βαθύ συνελκτικό νευρωνικό δίκτυο που επιλέχθηκε για την παρούσα εργασία είναι το Faster R-CNN που παρουσιάστηκε και στο δεύτερο κεφάλαιο. Η συγκεκριμένη επιλογή βασίστηκε κυρίως στα γενικότερα καλά αποτελέσματα της συγκεκριμένης αρχιτεκτονικής στην ανίχνευση αντικειμένων, στο σχετικά μικρό της μέγεθος (σε αντίθεση με αρχιτεκτονικές όπως το YOLO), στο ότι είναι γρήγορος ανιχνευτής δύο σταδίων και στα anchors που μας παρέχουν την δυνατότητα ανίχνευσης αντικειμένων διαφορετικών κλιμάκων. Τα αποτελέσματα που προέκυψαν από το συγκεκριμένο νευρωνικό δίκτυο μπορούν να χρησιμοποιηθούν από ένα σύνολο επιστημονικών ειδικοτήτων όπως είναι η γεωλογία, η βιολογία και άλλες πολλές, για να εξάγουν συμπεράσματα όπως ο ρυθμός μεταβολής των φυσαλίδων (όπου πιθανώς εμφανίζεται κάποια συσχέτιση με την ραδιενέργεια σε υποθαλάσσια ηφαίστεια), η μέτρηση του άνθρακα που προκύπτει από μια πηγή, τα αέρια που δημιουργούνται από κάποιους υποθαλάσσιους οργανισμούς αλλά και άλλες εφαρμογές. Όπως θα φανεί και στο τέλος του κεφαλαίου η αρχιτεκτονική που χρησιμοποιήθηκε οδήγησε στην δημιουργία ενός μοντέλου με πάρα πολύ καλές μετρικές αλλά και αρκετή στιβαρότητα (και γενίκευση) μιας και χρησιμοποιήθηκαν και δεδομένα διαφορετικά από αυτά της αποστολής, τα οποία βρέθηκαν μέσω του διαδικτύου.

Η δομή του συγκεκριμένου κεφαλαίου είναι η ακόλουθη. Στο κεφάλαιο 4.2 θα γίνει η παρουσίαση των δεδομένων και όλες οι διαδικασίες που έγιναν ώστε να δημιουργηθεί το τελικό σετ δεδομένων που αποτελεί και ένα από τα σημαντικότερα επιτεύγματα της συγκεκριμένης ερευνητικής εργασίας. Στο κεφάλαιο 4.3 θα σχολιαστεί η αρχιτεκτονική που χρησιμοποιήθηκε και θα παρουσιαστούν οι διάφοροι παράμετροι που επηρέασαν το αποτέλεσμα του μοντέλου. Στο κεφάλαιο 4.4 θα παρουσιαστούν τα αποτελέσματα μέσα από διάφορες μετρικές αλλά και inference (παρουσίαση της αρχικής εικόνας με την ετικετοποίηση που έγινε χειροκίνητα και της εικόνας που προέκυψε από το μοντέλο). Τέλος, θα σχολιαστούν πιθανά ζητήματα που παρουσιάστηκαν και πιθανές μελλοντικές αλλαγές που θα οδηγούσαν σε καλύτερα αποτελέσματα.

4.2 Επεξεργασία των δεδομένων

Από το σύστημα που παρουσιάστηκε στο κεφάλαιο τρία προέκυψε ένας αρκετά μεγάλος αριθμός δεδομένων ο οποίος δεν ήταν δυνατόν να επεξεργαστεί και να αναλυθεί στην πληρότητα του, για αυτό το λόγο τέθηκε εξ αρχής το ερώτημα σε ποια και σε πόσα δεδομένα θα γίνουν οι διάφορες επεξεργασίες που θα αναλυθούν. Σε πρώτο στάδιο η μετατροπή των δεδομένων από raw σε png φορμάτ έγινε επί του συνόλου των δεδομένων για να γίνει έλεγχος αρχικά αν το σύστημα παρήγαγε τα δεδομένα που αναμενόταν. Στο σχήμα 4.1 που ακολουθεί φαίνονται κάποια ενδεικτικά δεδομένα που προέκυψαν. Σε αυτό το σημείο οφείλει να σχολιαστεί ότι το σύστημα μας τοποθετήθηκε πάνω σε ROV μαζί με άλλα όργανα του εργαστηρίου για αυτές τις πρώτες αποστολές. Αυτό συνέβη για να γίνουν και έλεγχοι επί του πεδίου, της στεγανότητας του συστήματος, της συμπεριφοράς των μπαταριών μέσα στους διάφορους κύκλους λειτουργίας τους και γενικότερα για να υπάρξει πλήρης σιγουριά για την απόδοση του πριν γίνει η τελική εναπόθεση του σε τοποθεσία για μεγάλο χρονικό διάστημα. Ένα ενδιαφέρον στοιχείο για την εικόνες που παρουσιάζονται είναι ότι έχουν τοποθετηθεί στο διάγραμμα βάσει της απόστασης που είχε το μετρητικό όργανο από της φυσαλίδες με την πιο πάνω εικόνα να είναι η πιο μακρινή και η πιο κάτω η πιο κοντινή. Αφού παρατηρήθηκε ότι εμφανίζονταν οι φυσαλίδες σε περίπου τρεις κατηγορίες για κοντινές, μέσες και μακρινές αποστάσεις επιλέχθηκε να γίνει ένας εποπτικός έλεγχος του συνόλου των δεδομένων ώστε να βρεθεί σε τι ποσοστά ήταν αυτές οι λήψεις. Οι μακρινές λήψεις ήταν οι επικρατέστερες με 45% οι μέσες ήταν αμέσως μετά με 40% και οι κοντινές με 15%.



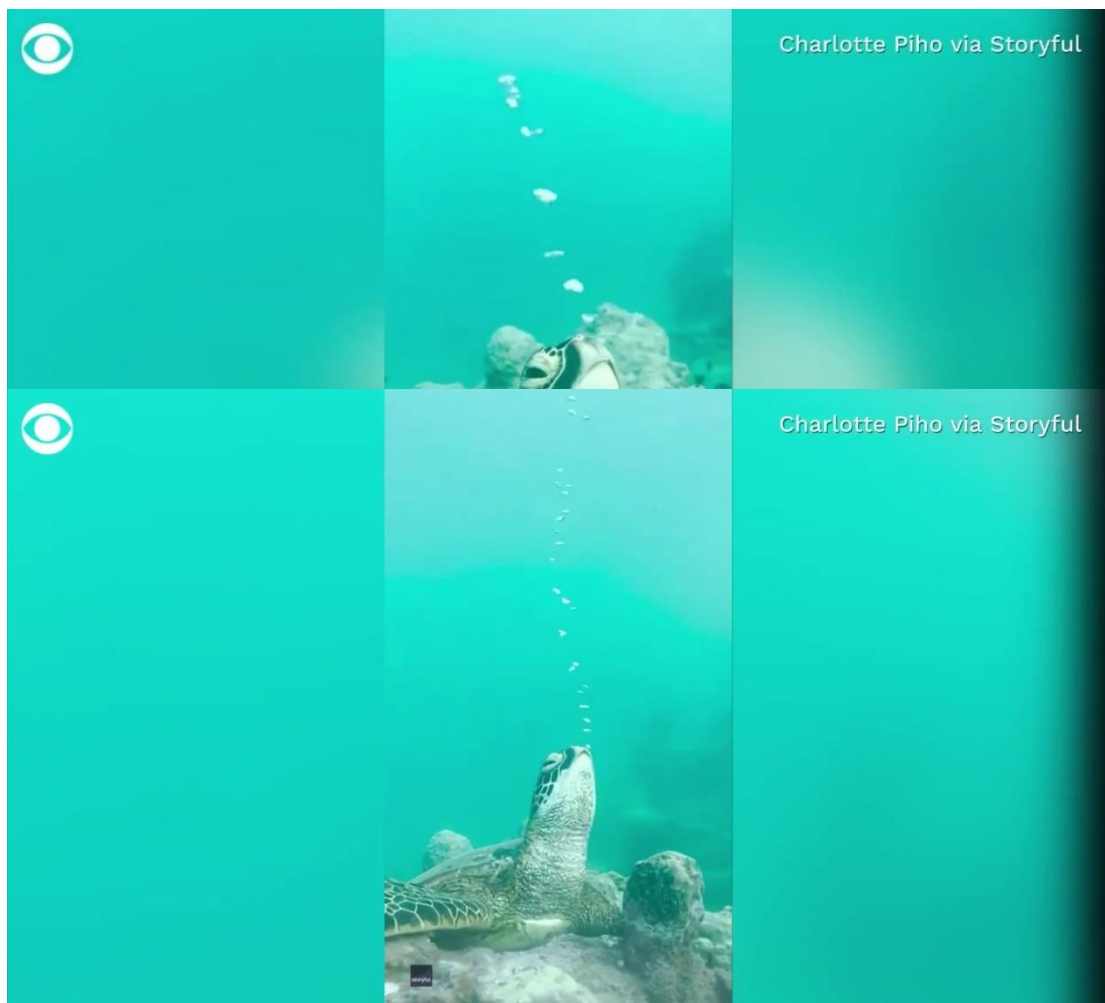


Σχήμα 4.1: Εικόνες που συλλέχθηκαν στο πλαίσιο των αποστολών. Πάνω : Οι φυσαλίδες βρίσκονται μακριά από το μετρητικό όργανο. Μέση: Οι φυσαλίδες βρίσκονται σε μέση απόσταση. Κάτω : Οι φυσαλίδες βρίσκονται πολύ κοντά.

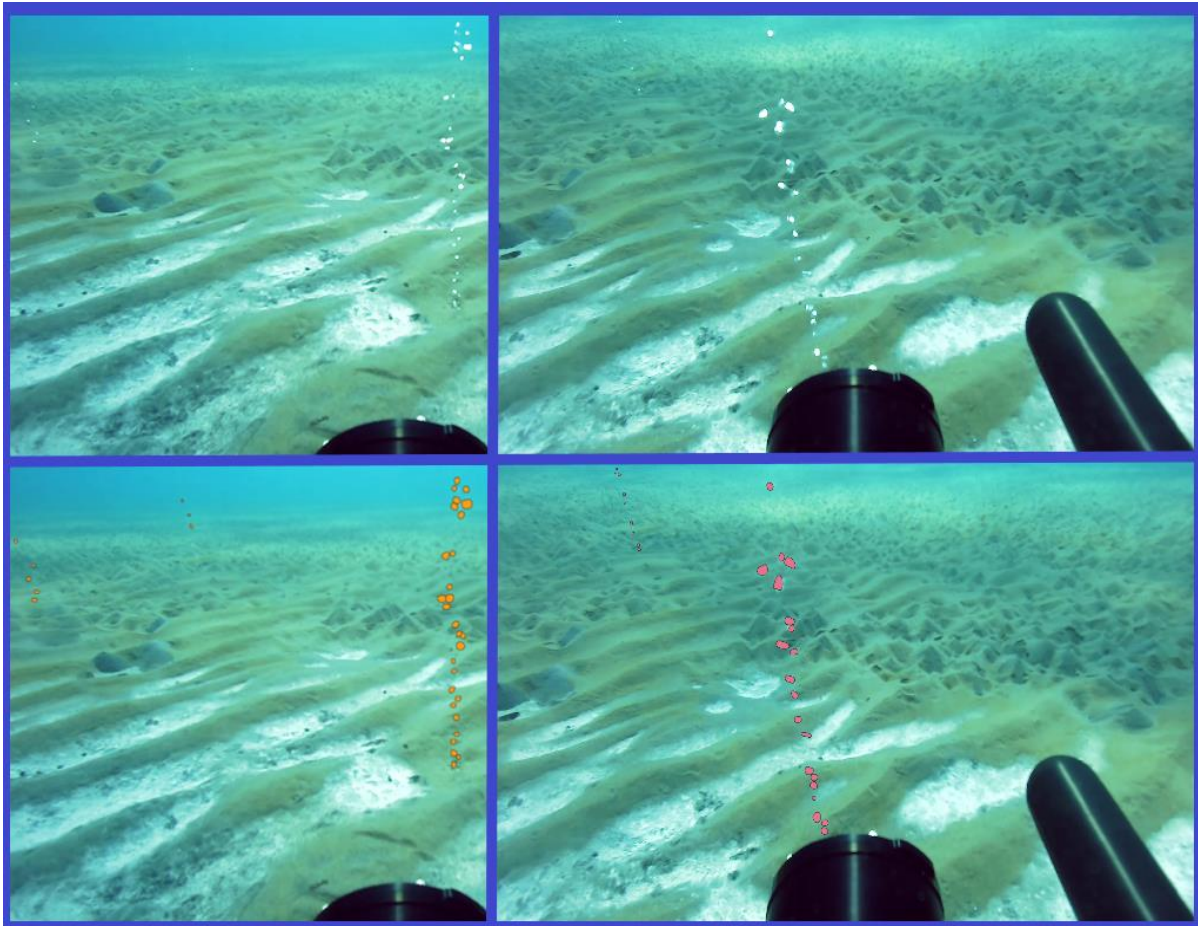
Σε αυτό το στάδιο έπρεπε να ληφθεί μια βασική απόφαση η οποία ήταν σε τι δεδομένα θέλουμε να είναι καλύτερο (πιο αποτελεσματικό) το νευρωνικό και ποια απόσταση είναι η πιο ρεαλιστική (όσον αφορά τα δεδομένα που θα θέλουμε να εξάγουμε και στο μέλλον), μετά από σκέψη και παρατηρώντας τις εικόνες κρίθηκε ότι κατά την εναπόθεση του συστήματος το πιο πιθανό είναι οι φυσαλίδες να βρίσκονται σε μέσες και μεγάλες αποστάσεις από το όργανο. Για αυτό το λόγο αποφασίστηκε να ακολουθηθούν τα ποσοστά των δεδομένων που αναφέρθηκαν και στο τελικό dataset. Για να είναι η διαδικασία με όσο το δυνατόν λιγότερο bias δημιουργήθηκαν 3 φάκελοι ανάλογα τις αποστάσεις που είχαν από το όργανο και με μια συνάρτηση τυχαιότητας επιλέχθηκαν τα δεδομένα που θα επεξεργαστούν. Συνολικά κρίθηκε ότι ένας αριθμός κοντά στις 300 λήψεις θα ήταν ικανοποιητικός για τις πρώτες δοκιμές του νευρωνικού δικτύου και σε περίπτωση που τα αποτελέσματα δεν ήταν τα ζητούμενα μπορεί να χρησιμοποιούνταν και περισσότερα δεδομένα. Η διαδικασία της ετικετοποίησης που ακολουθεί για την προετοιμασία των δεδομένων ήταν πολύ χρονοβόρα για αυτό στόχος ήταν να επιτευχθούν τα βέλτιστα δυνατά αποτελέσματα με τις λιγότερες δυνατές λήψεις. Εκτός αυτού κρίθηκε ότι για να αυξηθεί η στιβαρότητα του μοντέλου που θα υλοποιηθεί καλό θα ήταν να χρησιμοποιηθούν και μερικές εικόνες που βρέθηκαν από το διαδίκτυο. Στο σχήμα 4.2 που ακολουθεί φαίνονται κάποιες από αυτές. Συνολικά δημιουργήθηκε ένα σύνολο 359 εικόνων οι οποίες θα πρέπει να επεξεργαστούν (70 μέσω διαδικτύου και 289 μέσα από τις αποστολές).

Αφού πάρθηκαν οι επιλογές που σχολιάστηκαν, ξεκίνησε η διαδικασία της ετικετοποίησης των δεδομένων. Το εργαλείο που επιλέχθηκε να χρησιμοποιηθεί σε αρχικό στάδιο είναι QGIS ένα open source σύστημα γεωγραφικών πληροφοριών (ΣΓΠ). Το συγκεκριμένο σύστημα [16] αποτελεί πολύ χρήσιμο εργαλείο για επεξεργασία raster και vector δεδομένων καθώς παρέχει πληθώρα λειτουργιών αλλά

και δυνατότητες οπτικοποιήσεων μέσα από χάρτες. Για να γίνει η ετικετοποίηση τοποθετήθηκαν εντός του προγράμματος οι εικόνες που έπρεπε να επεξεργαστούν και κατασκευάστηκαν αρχεία .shp το οποίο αποτελεί ένα εύκολο φορμάτ στην διαχείριση vector δεδομένων. Τα ονόματα των εικόνων αλλά και των αρχείων vector που τους αντιστοιχούν έπρεπε να είναι τα ίδια ώστε μετά το νευρωνικό δίκτυο να μπορεί να τα συσχετίσει ως εικόνες και annotations. Έτσι, ξεκίνησε η διαδικασία της κατασκευής μάσκων (παρότι στην υλοποίηση που έγινε χρησιμοποιήθηκαν κουτιά περικλεισης, bounding boxes,) καθώς οι μάσκες παρέχουν πολύ μεγάλες δυνατότητες για υλοποίηση πολλών διαφορετικών αρχιτεκτονικών όπως είναι και το segmentation. Κάθε φυσαλίδα αντιμετωπίστηκε σαν ξεχωριστό αντικείμενο (πολύγωνο) έχοντας το δικό της id. Στο Σχήμα 4.3 που ακολουθεί φαίνονται κάποια παραδείγματα των αρχικών εικόνων και πως είναι μετά τις ψηφιοποιήσεις. Όπως φαίνεται έχουν γίνει όσο το δυνατόν καλύτερες μάσκες των φυσαλίδων ώστε στην συνέχεια να κατασκευαστούν ικανοποιητικά bounding boxes. Σε αυτό το σημείο τέθηκαν κάποια ζητήματα μιας και η ψηφιοποίηση πρέπει να έχει ομοιομορφία. Στις εικόνες παρουσιάστηκαν κάποια σημεία όπου οι φυσαλίδες ήταν πολύ κοντά ή μια στην άλλη και έπρεπε να επιλεγεί αν θα ψηφιοποιηθούν 1 ή 2 μάσκες. Όπως φαίνεται και στο Σχήμα 4.4 επιλέχθηκε να γίνουν σε όλες τις περιπτώσεις δύο μάσκες.



Σχήμα 4.2: Εικόνες που λήφθηκαν μέσω του διαδικτύου. Πάνω : Οι φυσαλίδες βρίσκονται κοντά στο μετρητικό όργανο. Κάτω : Οι φυσαλίδες βρίσκονται μακριά. Πηγή: <https://www.youtube.com/watch?v=anxOxKsNEC4>



Σχήμα 4.3: Παραδείγματα ψηφιοποίησης.



Σχήμα 4.4: Περιπτώσεις όπου φυσαλίδες εφάπτονταν αλλά πάρθηκαν ως δύο διαφορετικά αντικείμενα.

Έχοντας ολοκληρώσει το στάδιο των ψηφιοποιήσεων έπρεπε να δημιουργηθεί η σωστή μορφοποίηση στα δεδομένα για να είναι συμβατά με το νευρωνικό δίκτυο που θα χρησιμοποιηθεί. Το φορμάτ του συγκεκριμένου δικτύου είναι το COCO το οποίο είναι τύπου .json και αποτελεί ίσως το πιο συχνό σε αντικείμενα που αφορούν το detection και το segmentation. Ένα παράδειγμα του συγκεκριμένου φορμάτ φαίνεται στο Σχήμα 4.5 που ακολουθεί. Αρχικά, υπάρχουν κάποια στοιχεία για τις άδειες που δεν έχουν σημασία. Στην συνέχεια σε ένα επίπεδο τοποθετούνται οι κλάσεις με τα id τους στην περίπτωση μας οι φυσαλίδες (“bubbles”), στην συνέχεια υπάρχει ένα επίπεδο που περιλαμβάνει όλες τις εικόνες με ένα id το όνομα του αρχείου τους και τις διαστάσεις τους. Τέλος βρίσκονται οι ετικέτες όπου έχουν ένα id που είναι αύξων αριθμός μέσα σε μια εικόνα, δηλαδή αν μια εικόνα έχει 20 φυσαλίδες τα id των αντικειμένων θα είναι από 0 έως 19, εκτός αυτού έχουν το id της εικόνα στην οποία βρίσκονται (για να γίνει η αντιστοίχιση) αλλά και την κατηγορία που είναι (κάθε ετικέτα αντιστοιχεί σε ένα αντικείμενο, δηλαδή μια φυσαλίδα). Στην συνέχεια υπάρχει η πληροφορία για το bounding box, όπου οι πληροφορίες που παρέχονται είναι το xmin, ymin, το πλάτος και το ύψος του και εκτός αυτού αναγράφεται το εμβαδόν. Κλείνοντας, υπάρχει η πληροφορία για το iscrowd (αν είναι 1) το οποίο υποδηλώνει αν τα αντικείμενα είναι επικαλυπτόμενα (το νευρωνικό δίκτυο που χρησιμοποιήθηκε δεν τα λαμβάνει υπόψιν) και το segmentation το οποίο δεν αφορά την δικιά μας περίπτωση.

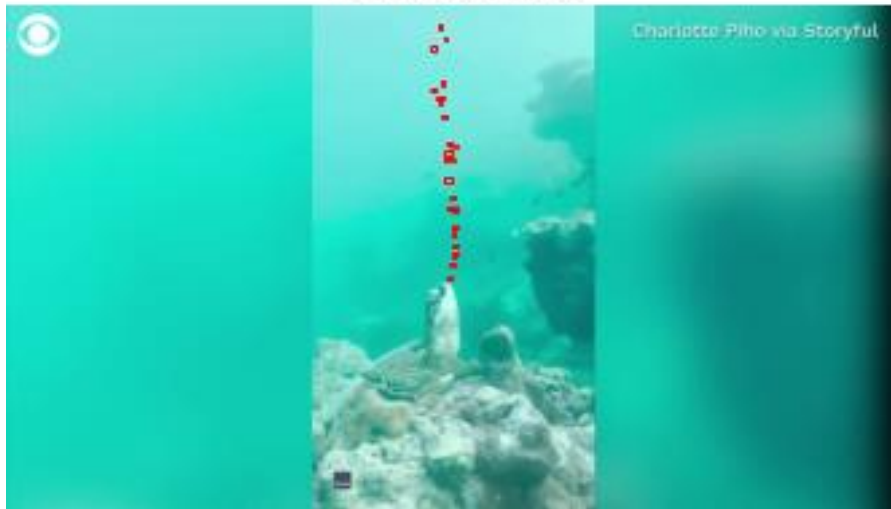
```
"categories": [
  {
    "id": 1,
    "name": "bubbles",
    "supercategory": ""
  }
],
"images": [
  {
    "id": 0,
    "file_name": "frame_mm_1274.png",
    "width": 1920,
    "height": 1080
  },
  {
    "id": 1,
    "file_name": "frame_mm_1280.png",
    "width": 1920,
    "height": 1080
  },
  {
    "id": 2,
    "file_name": "frame_mm_1285.png",
    "width": 1920,
    "height": 1080
  },
  {
    "id": 3,
    "file_name": "frame_mm_1291.png",
    "width": 1920,
    "height": 1080
  },
  {
    "id": 4,
    "file_name": "frame_mm_1299.png",
    "width": 1920,
    "height": 1080
  }
],
"annotations": [
  {
    "id": 0,
    "image_id": 0,
    "category_id": 1,
    "bbox": [
      336,
      26,
      4,
      4
    ],
    "area": 16,
    "iscrowd": 0,
    "segmentation": []
  },
  {
    "id": 1,
    "image_id": 0,
    "category_id": 1,
    "bbox": [
      332,
      36,
      4,
      4
    ],
    "area": 16,
    "iscrowd": 0,
    "segmentation": []
  },
  {
    "id": 2,
    "image_id": 0,
    "category_id": 1,
    "bbox": [
      341,
      36,
      8,
      9
    ],
    "area": 72,
    "iscrowd": 0,
    "segmentation": []
  }
]
```

Σχήμα 4.5: Ενδεικτικό παράδειγμα του COCO format.

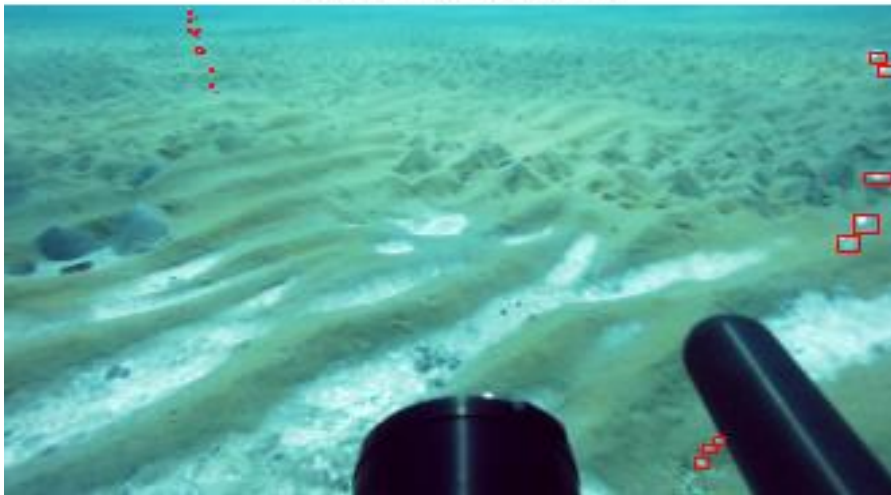
Για να επιτευχθεί το φορματ που παρουσιάστηκε έπρεπε σε αρχικό στάδιο να βρεθεί ένας τρόπος να παρατεθούν οι συντεταγμένες κάθε αντικείμενου σε ένα αρχείο, αυτό επιτεύχθηκε κάνοντας όλα τα .shp (διανυσματικά αρχεία με τις φυσαλίδες) σε .geojson φορματ (Σχήμα 4.6) στο οποίο φαινόντουσαν οι συντεταγμένες κάθε σημείου της μάσκας για κάθε φυσαλίδα. Προφανώς όπως φάνηκε και στο COCO φορματ δεν χρειαζόντουσαν όλα τα σημεία κάθε μάσκας έτσι για κάθε αντικείμενο πάρθηκαν οι τέσσερις ακραίες συντεταγμένες (xmin, xmax, ymin, ymax) και από αυτές βρέθηκε το ύψος το πλάτος και κατασκευάστηκαν τα απαραίτητα BB. Από τα 359 .geojson αρχεία τελικά προέκυψαν 2 .json με COCO φορματ όπου το ένα αφορούσε το κομμάτι της εκπαίδευσης του νευρωνικού δικτύου και είχε το 80% των δεδομένων και το άλλο τον έλεγχο του τελικού μοντέλου με 20% των δεδομένων. Αφού υλοποιήθηκαν τα συγκεκριμένα αρχεία έγινε έλεγχος ότι πράγματι η αντιστοίχιση των εικόνων με τα bounding boxes γίνεται σωστή και συνεπώς ότι τα αρχεία .json υλοποιήθηκαν σωστά. Αυτό έγινε με δύο τρόπους, αρχικά απλά πλοτάροντας με την rython μερικές ενδεικτικές εικόνες Σχήμα 4.7 και στην συνέχεια με το cvat. Το cvat αποτελεί εργαλείο ετικετοποίησης το οποίο δέχεται μόνο «σωστό» COCO φορματ συνεπώς εφόσον τα δεδομένα μπορούσαν να τοποθετηθούν σε αυτό και να οπτικοποιηθούν η διαδικασία είχε γίνει σωστά. Στο Σχήμα 4.8 φαίνονται κάποια παραδείγματα μέσα στην εφαρμογή cvat όπου πράγματι οι ετικέτες έχουν αντιστοιχηθεί στις σωστές εικόνες. Σε αυτό το σημείο αφού έγιναν όλες αυτές οι διαδικασίες έχει κατασκευαστεί ένα πλήρες σετ εικόνων που μπορεί να χρησιμοποιηθεί σε ένα σύνολο διαδικασιών και είναι ευθυγραμμισμένο με της απαιτήσεις του νευρωνικού δικτύου που θα υλοποιηθεί.

```
{
  "type": "FeatureCollection",
  "name": "frame_m_1543",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
  "features": [
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 272.281928757580829, -3.179517810589246 ],
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 304.192441355321932, -111.609578627270409 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 313.661598609883185, -123.158666377053208 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 342.807993049500908, -213.03557775836869 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 341.795395308406228, -227.018353605550118 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 369.026064291898763, -271.080039096427527 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 367.466116420482592, -282.65649435272644 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 508.326672458710334, -25.566138140736648 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 569.958297132906637, -13.716007819277062 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 662.569939183297492, -0.223827106853085 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 669.274978279735365, -10.9518986615369 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 666.538227628128311, -38.264661164194528 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 667.523457862706664, -95.722741094689766 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 673.735881841855303, -116.48099478131124 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 676.513683753236478, -131.683644656809633 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 660.804735013020259, -51.934730668972918 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 943.593179843591543, -20.051958577747719 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1450.056255430038163, -1.386946133785958 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1470.581885317092883, -36.088944396166511 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1497.347306689821058, -53.111533449163893 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1481.14574283296719, -61.48599043082223 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1415.764769765398569, -82.887380538651342 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1395.84122502169771, -92.466007819276896 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1530.078844483035937, -192.466876629007572 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1370.827324066006668, -207.464270199815502 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1358.484578627257633, -246.025086808962277 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1363.903344917440563, -315.319613379659074 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1370.663119026910181, -345.670178105984064 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1360.865516941560111, -372.87347958296067 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1341.489357080776472, -396.02639009555844 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1316.475456125086112, -468.112402258894633 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1332.622284969568909, -468.550282363151178 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1322.140529973913317, -498.490334491735837 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1304.926368375303355, -501.582862728052078 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1305.665291051237546, -517.674956559502903 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1300.520199826215958, -551.172784535176334 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1291.297350130299492, -568.113271068625409 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1286.836446568179554, -584.670162510849651 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1289.381624674174191, -597.122827975662972 ]
    { "type": "Feature", "properties": { "id": 1 }, "geometry": { "type": "MultiPolygon", "coordinates": [ [ [ [ 1276.491529105103837, -616.143245004333721 ]
  ]
}
```

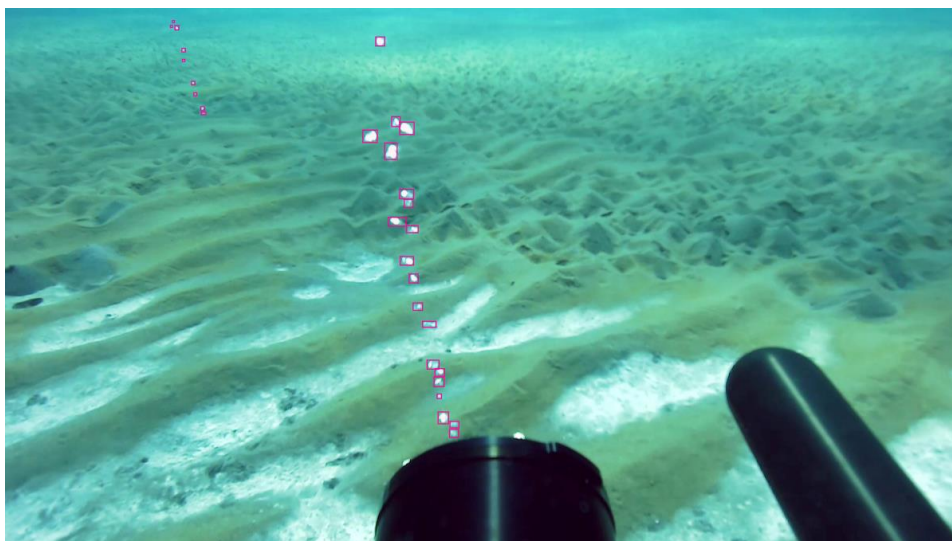
Σχήμα 4.6: Ενδεικτικό παράδειγμα .geojson αρχείου.
frame_t_95.png

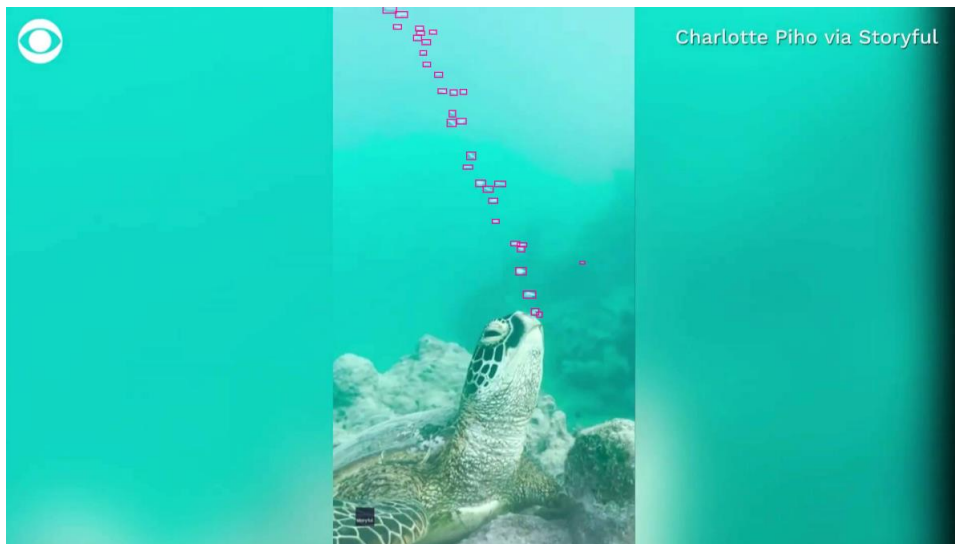


frame_mm_1392.png



Σχήμα 4.7: Παραδείγματα εικόνων με τα bounding boxes με οπτικοποίηση μέσα από ρυθμό.





Σχήμα 4.8: Παραδείγματα εικόνων με τα bounding boxes με οπτικοποίηση με το cvat.

4.3 Υλοποίηση μοντέλου

Αφού έχει ολοκληρωθεί η υλοποίηση του σετ δεδομένων αρχίζει η διαδικασία της εφαρμογής των μεθόδων για την εξαγωγή του ζητούμενου μοντέλου. Σε αυτό το σημείο είναι σημαντικό να βρεθεί ένα περιβάλλον εργασιών (framework) ώστε να υλοποιηθούν όλες οι διαδικασίες από την προσαρμογή των δεδομένων εκπαίδευσης, στην εκπαίδευση του μοντέλου και τελικά στην αξιολόγηση του μέσα από ένα ειδικό σετ δεδομένων. Ακολούθως, θα πρέπει να γίνουν επιλογές για τις διάφορες παραμέτρους του νευρωνικού μας δικτύου (οι οποίες θα αναλυθούν) ώστε να παράξουμε το βέλτιστο δυνατό μοντέλο. Τέλος, θα δούμε τα αποτελέσματα που προέκυψαν και θα σχολιαστεί πως αλλαγές των παραμέτρων βελτίωσαν ή όχι το μοντέλο.

Το πλαίσιο στο οποίο θα γίνει αυτή η εφαρμογή είναι το MMDetection [17] το οποίο αποτελεί «Μια “εργαλειοθήκη” για ανίχνευση αντικειμένων που παρέχει ένα μεγάλο αριθμό *object detection*, *instance segmentation* και *panoptic segmentation* μεθόδων, καθώς και όλα τα αναγκαία εργαλεία για την υλοποίησή τους». Το συγκεκριμένο πλαίσιο εργασίας παρέχει ένα εύρος δικών του βιβλιοθηκών όπως είναι η MMCV, η MMEval και η MMEngine. Οι βιβλιοθήκη MMCV περιλαμβάνει εργαλεία για:

- Την επεξεργασία εικόνων και βίντεο.
- Την οπτικοποίηση εικόνων και ετικετών.
- Τον μετασχηματισμό εικόνων.
- Διάφορες αρχιτεκτονικές CNN

- Υψηλού επιπέδου εφαρμογή συχνών CUDA ops.

Αντίστοιχα η MMEval συνεισφέρει κυρίως στο κομμάτι της αξιολόγησης των μοντέλων με ένα πολύ μεγάλο αριθμό μετρικών. Η βιβλιοθήκη MMEngine παρέχει τα εργαλεία για την εκπαίδευση και διαχείριση μοντέλων. Σε αυτό το σημείο είναι πολύ σημαντικό να σχολιαστεί ότι όλες οι βιβλιοθήκες βασίζονται στην Python και τα τμήματα που αφορούν την βαθιά μάθηση στην Pytorch, η οποία αποτελεί μια από τις πιο διαδεδομένες βιβλιοθήκες για τον κλάδο της (μαζί με την tensorflow).

Ο βέλτιστος τρόπος για να παρατεθεί η εκπαίδευση στον αναγνώστη είναι μέσα από το αρχείο διαμόρφωσης του για τις διάφορες παραμέτρους του νευρωνικού στο οποίο παρέχεται πολύ μεγάλο αριθμός πληροφορίας, που αφορά τις βιβλιοθήκες που χρησιμοποιήθηκαν (dependencies), για τα στοιχεία του υπολογιστή στον οποίο έγινε η εκπαίδευση αλλά και για τις παραμέτρους του μοντέλου. Τα dependencies που χρησιμοποιήθηκαν για την εκπαίδευση ήταν:

- MMCV 2.0.0
- MMDetection 3.0.0
- MMEngine 0.7.4
- PyTorch 1.12.0
- CUDA toolkit 11.6.0

Το υλισμικό του ηλεκτρονικού υπολογιστή στον οποίο έγινε η εκπαίδευση ήταν (βασικότερα στοιχεία):

- CPU: Intel Core i5 – 13600KF 2.6GHz
- GPU: NVIDIA RTX 3070 (8 GB GDDR6 VRAM, 5888 CUDA cores)
- RAM: 2x16 GB Kingston Fury 5200 MHz DDR5 CL 40
- Motherboard: ASRock Z690 Phantom Gaming 4/D5 ATX
- Hard Drive: Adata Legend SSD 1 TB M.2 NVMe PCI Express 4.0

Σε αυτό το σημείο θα γίνει η παρουσίαση του αρχείου που περιλαμβάνει όλες τις παραμέτρους του μοντέλου. Ο τρόπος που θα γίνει ο αρχικός σχολιασμός είναι με ένα μικρό κείμενο δίπλα από την κάθε γραμμή του αρχείου (κάποιες γραμμές που δεν έχουν ιδιαίτερη σημασία έχουν παραληφθεί για λόγους χωρικούς). Το συγκεκριμένο αρχείο όσον αφορά το κομμάτι των παραμέτρων δεν έχει πολλές αλλαγές από αυτές που είχε το default του MMDetection (όσες χρειαζόντουσαν για να έρθει σε πέρας το μοντέλο μας). Κατά το στάδιο της εκπαίδευσης υλοποιήσαμε 3 μοντέλα (πέρα από τα ενδιάμεσα

δοκιμαστικά) με την αρχιτεκτονική Faster R-CNN, στο δεύτερο μοντέλο κάναμε μικρές αλλαγές σε λίγες τιμές για να βελτιώσουμε ένα λάθος που θα σχολιαστεί στην συνέχεια και ένα τρίτο μοντέλο στοχεύοντας στην βελτιστοποίηση των αποτελεσμάτων. Έτσι, στην συνέχεια του κεφαλαία θα παρατεθούν τι αλλαγές έγιναν σε κάθε μοντέλο και η αιτία που επιλέχθηκαν. Ενώ στο κεφάλαιο 4.4 που ακολουθεί θα δούμε οπτικά και μετρητικά αποτελέσματα και από τα τρία μοντέλα και θα γίνουν οι συγκρίσεις ανάμεσα τους.

Αρχείο παραμέτρων:

Βασικές παράμετροι του μοντέλου

model = dict(
 type='FasterRCNN', Όνομα του ανιχνευτή

Βασικές παράμετροι της «ραχοκοκαλιάς» (backbone)

backbone=dict(
 type='ResNet', Το είδος του backbone
 depth=50, Το βάθος του backbone
 num_stages=4, Ο αριθμός των σταδίων
 out_indices=(0, 1, 2, 3), Η αρίθμηση των χαρτών χαρακτηριστικών που
 προκύπτουν σε κάθε στάδιο
 frozen_stages=1, Τα βάρη του πρώτου σταδίου είναι παγωμένα

Το αρχείο παραμέτρων των κανονικοποιημένων σταδίων

norm_cfg=dict(type='BN', requires_grad=True),
 norm_eval=True, Η διαχείριση των στατιστικών (παγωμένα ή μη)
 style='pytorch', Το στυλ του backbone
 init_cfg=dict(type='Pretrained', checkpoint='torchvision://resnet50'), Αρχικοποίηση

Βασικές παράμετροι του «λαιμού» (neck)

neck=dict(
 type='FPN', Ο λαιμός του backbone είναι FPN
 in_channels=[256, 512, 1024, 2048], Τα κανάλια εισαγωγής
 out_channels=256, Τα κανάλια εξαγωγής
 num_outs=5), Ο αριθμός των εξαγόμενων κλιμάκων

Βασικές παράμετροι του Region Proposal Network (Head)

rpn_head=dict(
 type='RPNHead', Ο τύπος του RPN
 in_channels=256, Τα κανάλια εισαγωγής για κάθε «χάρτη
 χαρακτηριστικών»
 feat_channels=256, Τα κανάλια χαρακτηριστικών των συνελκτικών επιπέδων
 του head

Βασικές παράμετροι των αγκυρώσεων (Anchors)

anchor_generator=dict(
 type='AnchorGenerator', Μέθοδος «παραγωγής» αγκυρώσεων
 scales=[8], Κλίμακα αγκυρώσεων
 ratios=[0.5, 1.0, 2.0], Οι αναλογίες των αγκυρώσεων σε πλάτος και μήκος
 strides=[4, 8, 16, 32, 64]), Τα Strides της γεννήτριας των αγκυρώσεων

Βασικές παράμετροι για την κωδικοποίηση και αποκωδικοποίηση των bb κατά της εκπαίδευση και τον έλεγχο

bbox_coder=dict(
 type='DeltaXYWHBBoxCoder', Τύπος αποκωδικοποιητή

target_means=[0.0, 0.0, 0.0, 0.0], Χρησιμοποιούνται για την κωδικοποίηση και την αποκωδικοποίηση
target_stds=[1.0, 1.0, 1.0, 1.0]), Η επιλεγμένη τυπική διακύμανση

Βασικές παράμετροι της συνάρτησης κόστους (όσον αφορά την ταξινόμηση)

loss_cls=dict(
type='CrossEntropyLoss', use_sigmoid=True, loss_weight=1.0), Το πρώτο τμήμα του τύπου της συνάρτησης κόστους, στην συνέχεια γίνεται χρήση σιγμοειδούς συνάρτησης και τέλος είναι το βάρος της συνάρτησης
loss_bbox=dict(type=L1Loss, loss_weight=1.0)), Το τμήμα παραμέτρων της συνάρτησης κόστους της παλινδρόμησης, καθώς και η επιλογή σιγμοειδούς συνάρτησης και βαρών ίσων με μονάδα

Βασικές παράμετροι του RoI (head)

roi_head=dict(
type='StandardRoIHead', Ο τύπος του RoI head (τυπικός)
bbox_roi_extractor=dict(
type='SingleRoIExtractor', Ο τύπος του εξαγωγέα
roi_layer=dict(type='RoIAlign', output_size=7, sampling_ratio=0), Το αρχείο παραμέτρων του RoI επιπέδου (max pooling, ο τύπος του επιπέδου, το μέγεθος του χάρτη χαρακτηριστικών, ο τρόπος δειγματοληψίας, 0 σημαίνει προσαρμοστικός)
out_channels=256, Τα κανάλια των εξαγόμενων χαρακτηριστικών
featmap_strides=[4, 8, 16, 32]), Τα strides των πολυδιάστατων «χαρτών» χαρακτηριστικών

Βασικές παράμετροι των bb στο RoI (head)

bbox_head=dict(
type='Shared2FCBBoxHead', Τύπος
in_channels=256, Τα κανάλια εισαγωγής
fc_out_channels=1024, Τα κανάλια εξαγωγής
roi_feat_size=7, Το μέγεθος των RoI χαρακτηριστικών
num_classes=1, Ο αριθμός των κλάσεων για ταξινόμηση

Βασικές παράμετροι για τον κωδικοποιητή των bb στο δεύτερο στάδιο του αλγορίθμου

bbox_coder=dict(
type='DeltaXYWHBBoxCoder', Ο τύπος του κωδικοποιητή
target_means=[0.0, 0.0, 0.0, 0.0], Χρήση για κωδικοποίηση και αποκωδικοποίηση των bb
target_stds=[0.1, 0.1, 0.2, 0.2]), Τυπική διακύμανση
reg_class_agnostic=False, Αν η παλινδρόμηση δεν ξέρει την κλάση (ανιχνεύει χωρίς να λαμβάνει υπόψιν την κλάση)

Βασικές παράμετροι συνάρτησης κόστους για το στάδιο της ταξινόμησης

loss_cls=dict(
type='CrossEntropyLoss', use_sigmoid=False, loss_weight=1.0), Τύπος, χρήση σιγμοειδούς, βάρη

Βασικές παράμετροι συνάρτησης κόστους για το στάδιο της παλινδρόμησης

loss_bbox=dict(type=L1Loss, loss_weight=1.0)), Τύπος, βάρη

Βασικές παράμετροι για την εκπαίδευση (υπερπαράμετρων)

train_cfg=dict(
rpn=dict(
assigner=dict(
type='MaxIoUAssigner', Τύπος
pos_iou_thr=0.7, Για IoU >= 0.7 έχουμε θετικά δείγματα
neg_iou_thr=0.3, Για IoU < 0.3 έχουμε αρνητικά δείγματα
min_pos_iou=0.3, Για τιμή IoU 0.3 έχουμε την μικρότερη τιμή που λογίζεται ως θετικό δείγμα
match_low_quality=True, Αν είναι να γίνει η αντιστοίχιση ως low quality (0.3-0.7)

ignore_iof_thr=-1) IoF κατώφλι για να μην δίνουμε σημασία στα bb

Βασικές παράμετροι για δειγματοληψία

sampler=dict(
 type='RandomSampler', Τύπος
 num=256, Αριθμός δειγμάτων
 pos_fraction=0.5, Η αναλογία θετικών δειγμάτων στα συνολικά
 neg_pos_ub=-1, Το ανώτερο όριο αρνητικών δειγμάτων
 add_gt_as_proposals=False), Η εισαγωγή GT σαν προτάσεις μετά την δειγματοληψία
allowed_border=-1, Τα όρια που επιτρέπονται μετά από padding σε έγκυρες
αγκυρώσεις
pos_weight=-1, Βάρη θετικών δειγμάτων κατά την εκπαίδευση
debug=False), Αν γίνεται χρήση του debug mode

Βασικές παράμετροι για τις προτάσεις κατά την εκπαίδευση (RPN)

rpn_proposal=dict(
 nms_pre=2000, Ο αριθμός των κουτιών πριν γίνει η NMS
 max_per_img=1000, Ο μέγιστος αριθμός μετά
 nms=dict(type='nms', iou_threshold=0.7), Ο τύπος και το κατώφλι που χρησιμοποιούνται κατά
την NMS
 min_bbox_size=0), Το ελάχιστο μέγεθος των bb

Βασικές παράμετροι για τα RoI (διαφορετικό από αυτό στο RPN)

rcnn=dict(
 assigner=dict(
 type='MaxIoUAssigner', Τύπος assigner
 pos_iou_thr=0.5, Για IoU >= 0.5 έχουμε θετικά δείγματα
 min_pos_iou=0.5, Για τιμή IoU 0.5 έχουμε την μικρότερη τιμή που λογίζεται ως
θετικό δείγμα
 match_low_quality=False, Αν είναι να γίνει η αντιστοίχιση ως low quality
 ignore_iof_thr=-1), IoF κατώφλι για να μην δίνουμε σημασία στα bb

Βασικές παράμετροι του assigner για το δεύτερο στάδιο

match_low_quality=False, Αν είναι να γίνει η αντιστοίχιση ως low quality
ignore_iof_thr=-1), IoF κατώφλι για να μην δίνουμε σημασία στα bb

Βασικές παράμετροι για δειγματοληψία

sampler=dict(
 type='RandomSampler', Τύπος
 num=512, Αριθμός δειγμάτων
 pos_fraction=0.25, Η αναλογία θετικών δειγμάτων στα συνολικά
 neg_pos_ub=-1, Το ανώτερο όριο αρνητικών δειγμάτων
 add_gt_as_proposals=True), Η εισαγωγή GT σαν προτάσεις μετά την δειγματοληψία
pos_weight=-1, Βάρη θετικών δειγμάτων κατά την εκπαίδευση
debug=False)), Αν γίνεται χρήση του debug mode

Βασικές παράμετροι για τον έλεγχο των υπερπαραμέτρων

test_cfg=dict(
 rpn=dict(
 nms_pre=1000, Ο αριθμός των κουτιών πριν γίνει η NMS
 max_per_img=1000, Ο μέγιστος αριθμός μετά
 nms=dict(type='nms', iou_threshold=0.7), Τύπος, κατώφλι (κατά την NMS)
 min_bbox_size=0), Ελάχιστο μέγεθος bb

Βασικές παράμετροι για να δημιουργήσουν προτάσεις κατά τον έλεγχο (RPN)

max_per_img=1000, Ο μέγιστος αριθμός μετά
nms=dict(type='nms', iou_threshold=0.7), Τύπος, κατώφλι (κατά την NMS)
min_bbox_size=0), Ελάχιστο μέγεθος bb

Βασικές παράμετροι για τα RoI

rcnn=dict(
 score_thr=0.05, Κατώφλι για τον διαχωρισμό των bb
 nms=dict(type='nms', iou_threshold=0.5), Τύπος, κατώφλι (κατά την NMS)
 max_per_img=100))) Μέγιστος αριθμός ανιχνεύσεων κατά εικόνα

Πληροφορίες για το σετ δεδομένων

dataset_type = 'CocoDataset' Ο τύπος του σετ δεδομένων
data_root = 'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/' Η θέση του σετ
δεδομένων στον ηλεκτρονικό υπολογιστή
backend_args = **None**

Πληροφορίες για την μεθοδολογία εκπαίδευσης

```
train_pipeline = [  
    dict(type='LoadImageFromFile', backend_args=None),    «Τραβάμε» την εικόνα από τον φάκελο  
    dict(type='LoadAnnotations', with_bbox=True),        Εισάγουμε τις ετικέτες  
    dict(type='Resize', scale=(1920, 1080), keep_ratio=True), Αλλαγή διαστάσεων των εικόνων αν δεν  
    είναι 1920 με 1080  
    dict(type='RandomFlip', prob=0.5),                    Τυχαία περιστροφή (50%)  
    dict(type='PackDetInputs')  
]
```

Πληροφορίες για την μεθοδολογία ελέγχου

```
test_pipeline = [  
    dict(type='LoadImageFromFile', backend_args=None),    «Τραβάμε» την εικόνα από τον φάκελο  
    dict(type='Resize', scale=(1920, 1080), keep_ratio=True), Αλλαγή διαστάσεων των εικόνων αν δεν  
    είναι 1920 με 1080  
    dict(type='LoadAnnotations', with_bbox=True),        Εισάγουμε τις ετικέτες  
    dict(  
        type='PackDetInputs',  
        meta_keys=('img_id', 'img_path', 'ori_shape', 'img_shape',  
                  'scale_factor'))                            Στοιχεία εικόνων  
]
```

Πληροφορίες για τον εισαγωγέα δεδομένων εκπαίδευσης

```
train_dataloader = dict(  
    batch_size=1,                                            Μέγεθος batch  
    num_workers=2,  
    persistent_workers=True,  
    sampler=dict(type='DefaultSampler', shuffle=True),    Χρήση δειγματολήπτη  
    batch_sampler=dict(type='AspectRatioBatchSampler'),  
    dataset=dict(  
        type='CocoDataset',                                Τύπος ετικετών  
        data_root=  
        'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Train/', Θέση root  
        ann_file=  
        'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Train/annotations1/annotatio  
        ns1.json',                                            Θέση αρχείου ετικετών  
        data_prefix=dict(  
            img=  
            'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Train/images1/'  
        ),  
    ),
```

Πληροφορίες για την μεθοδολογία φίλτρου

```
filter_cfg=dict(filter_empty_gt=True, min_size=32),  
pipeline=[  
    dict(type='LoadImageFromFile', backend_args=None),    «Τραβάμε» την εικόνα από τον φάκελο  
    dict(type='LoadAnnotations', with_bbox=True),        Εισάγουμε τις ετικέτες  
    dict(type='Resize', scale=(1920, 1080), keep_ratio=True), Αλλαγή διαστάσεων των εικόνων  
    αν δεν είναι 1920 με 1080  
    dict(type='RandomFlip', prob=0.5),                    Τυχαία περιστροφή (50%)  
    dict(type='PackDetInputs')  
],  
backend_args=None,
```

metainfo=**dict**(classes='bubbles', palette=[(255, 20, 20)])) Πληροφορίες

Πληροφορίες για τον εισαγωγή δεδομένων validation

```
val_dataloader = dict(
    batch_size=1,                Μέγεθος batch
    num_workers=2,
    persistent_workers=True,
    drop_last=False,
    sampler=dict(type='DefaultSampler', shuffle=False), Χρήση δειγματολήπτη
    dataset=dict(
        type='CocoDataset',      Τύπος ετικετών
        data_root=
        'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test', Θέση root
        ann_file=
        'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test/annotations2/annotatio
        ns2.json', Θέση αρχείου ετικετών
        data_prefix=dict(
            img=
            'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test/images2'
        ),
        test_mode=True,        Διαδικασία ελέγχου
        pipeline=[
            dict(type='LoadImageFromFile', backend_args=None), «Τραβάμε» την εικόνα από τον φάκελο
            dict(type='Resize', scale=(1920, 1080), keep_ratio=True), Αλλαγή διαστάσεων των εικόνων
            αν δεν είναι 1920 με 1080
            dict(type='LoadAnnotations', with_bbox=True),      Εισάγουμε τις ετικέτες
            dict(
                type='PackDetInputs',
                meta_keys=('img_id', 'img_path', 'ori_shape', 'img_shape',
                    'scale_factor')) Στοιχεία εικόνων
        ],
        backend_args=None,
        metainfo=dict(classes='bubbles', palette=[(255, 20, 20)])) Στοιχεία κλάσης
```

Πληροφορίες για τον εισαγωγή δεδομένων ελέγχου

```
test_dataloader = dict(
    batch_size=1,                Μέγεθος batch
    num_workers=2,
    persistent_workers=True,
    drop_last=False,
    sampler=dict(type='DefaultSampler', shuffle=False), Χρήση δειγματολήπτη
    dataset=dict(
        type='CocoDataset',      Τύπος ετικετών
        data_root=
        'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test', Θέση root
        ann_file=
        'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test/annotations2/annotatio
        ns2.json', Θέση αρχείου ετικετών
        data_prefix=dict(
            img=
            'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test/images2'
        ),
        test_mode=True,        Διαδικασία ελέγχου
        pipeline=[
            dict(type='LoadImageFromFile', backend_args=None), «Τραβάμε» την εικόνα από τον φάκελο
            dict(type='Resize', scale=(1920, 1080), keep_ratio=True), Αλλαγή διαστάσεων των εικόνων
            αν δεν είναι 1920 με 1080
            dict(type='LoadAnnotations', with_bbox=True),      Εισάγουμε τις ετικέτες
```

```

dict(
    type='PackDetInputs',
    meta_keys=('img_id', 'img_path', 'ori_shape', 'img_shape',
               'scale_factor')) Στοιχεία εικόνων
],
backend_args=None,
metainfo=dict(classes='bubbles', palette=[(255, 20, 20)])) Στοιχεία κλάσης

```

Πληροφορίες για τις μετρικές του validation

```

val_evaluator = dict(
    type='CocoMetric',           Τύπος
    ann_file=
'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test/annotations2/annotatio
ns2.json',                     Θέση αρχείου ετικετών
    metric='bbox',              Μετρική
    format_only=False,
    backend_args=None)

```

Πληροφορίες για τις μετρικές του ελέγχου

```

test_evaluator = dict(
    type='CocoMetric',           Τύπος
    ann_file=
'C:/Thesis/Datasets/Training_full_dataset_Full/Training_full_dataset_Full/Test/annotations2/annotatio
ns2.json',                     Θέση αρχείου ετικετών
    metric='bbox',              Μετρική
    format_only=False,
    backend_args=None)

```

train_cfg = dict(type='EpochBasedTrainLoop', max_epochs=50 val_interval=1) Τύπος, Μέγιστος αριθμός εποχών και κάθε πότε θα γίνεται έλεγχος -Εκπαίδευση-
val_cfg = dict(type='ValLoop') Τύπος -Validation-
test_cfg = dict(type='TestLoop') Τύπος -Έλεγχος-

Πληροφορίες για την εκμάθηση (ρυθμός)

```

param_scheduler = [
    dict(
        type='LinearLR', start_factor=0.0001, by_epoch=False, begin=0, Ρυθμός εκπαίδευσης
        end=1000),
    dict(
        type='MultiStepLR',
        begin=0,
        end=50,
        by_epoch=True,
        milestones=[16, 22],
        gamma=0.1)
]

```

Πληροφορίες για τον βελτιστοποιητή

```

optim_wrapper = dict(
    type='OptimWrapper',
    optimizer=dict(type='SGD', lr=0.002, momentum=0.9, weight_decay=0.0001))

```

Πληροφορίες για την αυτοματοποιημένη διαχείριση ρυθμού εκμάθησης

```

auto_scale_lr = dict(enable=False, base_batch_size=16)
default_scope = 'mmdet'

```

Πληροφορίες για τα Hooks

```

default_hooks = dict(
    timer=dict(type='IterTimerHook'),

```

```
logger=dict(type='LoggerHook', interval=50),
param_scheduler=dict(type='ParamSchedulerHook'),
checkpoint=dict(type='CheckpointHook', interval=1),
sampler_seed=dict(type='DistSamplerSeedHook'),
visualization=dict(type='DetVisualizationHook'))
```

Πληροφορίες για το περιβάλλον

```
env_cfg = dict(
    cudnn_benchmark=False,
    mp_cfg=dict(mp_start_method='fork', opencv_num_threads=0),
    dist_cfg=dict(backend='nccl')
)
vis_backends = [dict(type='LocalVisBackend')]
```

Πληροφορίες για την οπτικοποίηση

```
visualizer = dict(
    type='DetLocalVisualizer',
    vis_backends=[dict(type='LocalVisBackend')],
    name='visualizer')
```

Πληροφορίες για την καταγραφή των logs

```
log_processor = dict(type='LogProcessor', window_size=50, by_epoch=True)
log_level = 'INFO'
```

```
load_from = None
resume = False
metainfo = dict(classes='bubbles', palette=[(255, 20, 20)])
```

Πληροφορίες για τον launcher

```
launcher = 'none'
```

Πληροφορίες για τον φάκελο εργασίας

```
work_dir = './work_dirs/faster-rcnn_r50_fpn_2x_coco'
```

Έχοντας το συγκεκριμένο σεντ παραμέτρων κατά την υλοποίηση του πρώτου μοντέλου μας παρατηρήσαμε ότι παρότι είχε αρκετά ικανοποιητικά αποτελέσματα από την άποψη των μετρικών, υπήρχε αδυναμία στην ανίχνευση μικρών σε μέγεθος φουσαλίδων. Λόγω της φύσης του συγκεκριμένου αλγορίθμου στην ανίχνευση διαφορετικών μεγεθών αντικειμένων μεγάλο μέρος παίζουν οι αγκυρώσεις και τα μεγέθη τους για αυτό και επιλέχθηκε να γίνει και μια δεύτερη υλοποίηση αλλάζοντας κάποια πράγματα στις αγκυρώσεις. Οι αλλαγές που έγιναν φαίνονται στον πίνακα που θα ακολουθήσει, όπου φαίνονται συνολικά οι αλλαγές ανάμεσα στις δοκιμές που έγιναν. Πράγματι όπως και θα φανεί στο επόμενο κεφάλαιο οι αλλαγές που κάναμε και μικρύνουμε το μέγεθος των anchors μας έδωσε πολύ καλύτερα αποτελέσματα και όσον αφορά το κομμάτι της ποιοτικής αξιολόγησης (οπτικοποιήσεις) αλλά και όσον αφορά τις μετρικές εκτοξεύοντας τον AP σε τιμές μεγαλύτερες του 90%. Παρόλα αυτά επιλέχθηκε να γίνει και μια τρίτη δοκιμή στην προσπάθεια να επιτύχουμε όσο δυνατόν καλύτερες μετρικές, η αλλαγή που έγινε είναι αύξηση του output size των RoI features όσο μας το επέτρεπε η υπολογιστική ισχύς του συστήματος, καθώς έτσι απεικονίζεται όσο το δυνατόν περισσότερη πληροφορία για τον detector του fast-RCNN. Κάποια σημαντικά σχόλια που πρέπει να γίνουν σε αυτό το κεφάλαιο είναι η επιλογή των 100 εποχών κατά την εκπαίδευση και στα τρία μοντέλα αλλά και το batch size. Το batch

size επιλέχθηκε να είναι 1 μιας και η κάρτα γραφικών που χρησιμοποιήθηκε κατά την εκπαίδευση δεν είχε αρκετή RAM για να τοποθετηθεί άλλη μεγαλύτερη τιμή. Όσον αφορά τις εποχές η τιμή των τελικών εποχών προέκυψε εκπαιδύοντας το μοντέλο για μεγαλύτερο αριθμό εποχών και παρατηρώντας ότι η συνάρτηση κόστους συνέκλινε για περίπου 25 εποχές αλλά και το AP δεν είχε κάποια βελτίωση μετά από αυτό το σημείο. Παρόλα αυτά το μοντέλο ένα εκπαιδεύτηκε για 100 εποχές ενώ τα 2,3 για 50 παρότι όπως θα παρουσιαστεί και στο επόμενο κεφάλαιο χρησιμοποιήθηκαν προγενέστερα checkpoints (για να αποφευχθεί το overfitting). Τόσο οι loss function όσο και τα AP αλλά και ο ποιοτικός τρόπος αξιολόγησής θα παρουσιαστούν αναλυτικά στο επόμενο κεφάλαιο με τα αποτελέσματα. Κλείνοντας, τα πρώτα δύο μοντέλα ήθελαν 4 λεπτά ανά εποχή ενώ το τρίτο μοντέλο χρειαζόταν 6.

Κάποια σημαντικά στοιχεία που οφείλουν να σχολιαστούν όσον αφορά τα μοντέλα που υλοποιήσαμε είναι ότι έχουν για backbone το ResNet 50 με pretrained βάρη. Εκτός αυτού ως neck χρησιμοποιείται ένα feature pyramid network (τα οποία δεν ήταν στην αρχική έκδοση του faster rcnn), το learning rate ότι είναι adaptive δηλαδή αρχικά έχει μικρές τιμές που αυξάνονται γραμμικά για να κάνει warm up στην συνέχεια μειώνεται για να επιτύχει την βέλτιστη τιμή σύγκλισης. Σημαντικά να σχολιαστεί είναι και η NMS η οποία χρησιμοποιήθηκε για να αποφευχθούν οι επιλογές που εμφανίζουν μεγάλες επικαλύψεις, στο RPN επιλέχθηκε να χρησιμοποιηθούν κατώφλια για IoU 0.7 και 0.3 όπου για τιμές μεγαλύτερες από 0.3 στο IoU τα δείγματα θεωρήθηκαν ως θετικά ενώ για μικρότερες από 0.3 αρνητικά, ενώ στα RoI σαν κατώφλι επιλέχθηκε η τιμή 0.5 IoU.

| Parameters/Models | Model 1 | Model 2 | Model 3 |
|-------------------|---------|---------|---------|
| Scales (anchors) | 8 | 2,4,8 | 2,4,8 |
| Output_size (ROI) | 7 | 7 | 14 |
| Epochs | 50 | 40 | 40 |

Πίνακας 4.1: Οι σημαντικότερες παράμετροι που μεταβλήθηκαν ανάμεσα στα μοντέλα

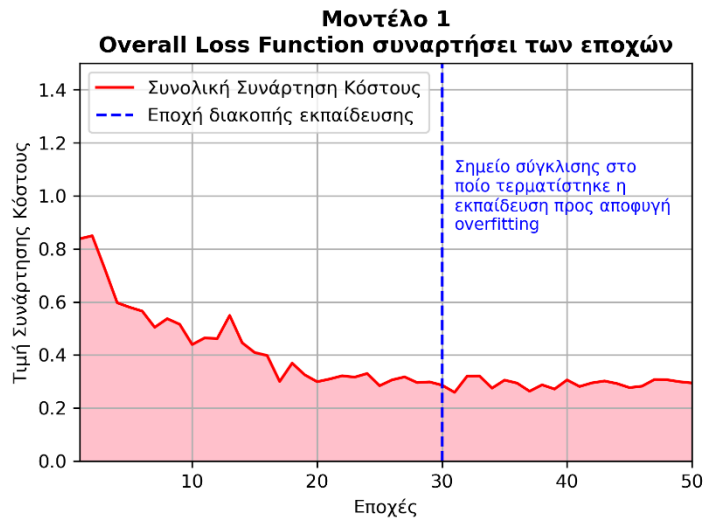
Συνολικά εφαρμόστηκε η μεθοδολογία που φαίνεται στο αρχείο παραμέτρων και οι αλλαγές που έγιναν ήταν λίγες και στοχευμένες για να βελτιωθούν οι δείκτες ποσοτικής αξιολόγησης του μοντέλου. Πέρα από το πρώτο μοντέλο το οποίο είχε πολύ ασθενείς μετρικές κυρίως λόγω της αδυναμίας στην ανίχνευση των μικρών σε μέγεθος φυσαλίδων, τα υπόλοιπα μοντέλα που παρουσιάζονται έχουν ιδιαίτερα καλούς δείκτες ποιοτικής και ποσοτικής αξιολογήσεις οι οποίοι παρουσιάζονται και στην συνέχεια. Ένα σχόλιο πριν ολοκληρωθεί το συγκεκριμένο υποκεφάλαιο είναι ότι προφανώς κατά την διαδικασία της υλοποίησης δεν έγιναν μόνο τρεις δοκιμές εκπαίδευσης και αυτά τα τρία μοντέλα προέκυψαν μετά από σειρά δοκιμών μέχρι να βρεθούν οι παράμετροι που θα έδιναν καλό αποτέλεσμα. Απλά τα τρία συγκεκριμένα μοντέλα αποτελούν τα πιο

σημαντικά βήματα που έγιναν, από το μοντέλο 1 που έβγαζε πολύ μέτρια αποτελέσματα στο μοντέλο 2 που οι μετρικές τους ήταν πάρα πολύ ικανοποιητικές στο μοντέλο 3 που έγινε προσπάθεια βελτιστοποίησης των παραμέτρων του μοντέλου 2.

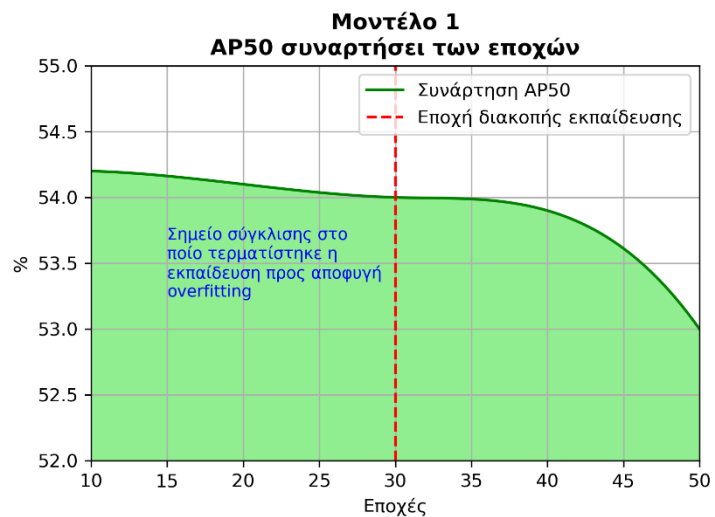
4.4 Αποτελέσματα και αξιολόγηση

Έχοντας ολοκληρώσει το στάδιο της υλοποιήσεως των τριών μοντέλων πρέπει να εφαρμοστεί μια βασική μεθοδολογία. Βάση αυτής βρίσκεται σε ποια εποχή πρέπει να σταματήσει η εκπαίδευση ώστε να αποφευχθεί το *overfitting*, καθώς αλλιώς το μοντέλο θα βγάζει καλά αποτελέσματα μόνο για σετ δεδομένων πολύ όμοια με αυτά που εκπαιδεύτηκε. Η συγκεκριμένη διαδικασία γίνεται λαμβάνοντας υπόψιν ως κριτήρια τότε συγκλίνει η *loss function* αλλά και τότε σταματάει να μειώνεται το *Average Precision*. Αφού παρουσιαστεί αυτή η διαδικασία θα γίνει αξιολόγηση των μοντέλων. Η αξιολόγηση θα χωριστεί σε δύο βασικά τμήματα, το ποσοτικό (το οποίο αφορά τις μετρικές αξιολόγησης) αλλά και το ποιοτικό, όπου θα παρατεθούν διάφορα παραδείγματα όπου τα μοντέλα έχουν επιτύχει ή αποτύχει.

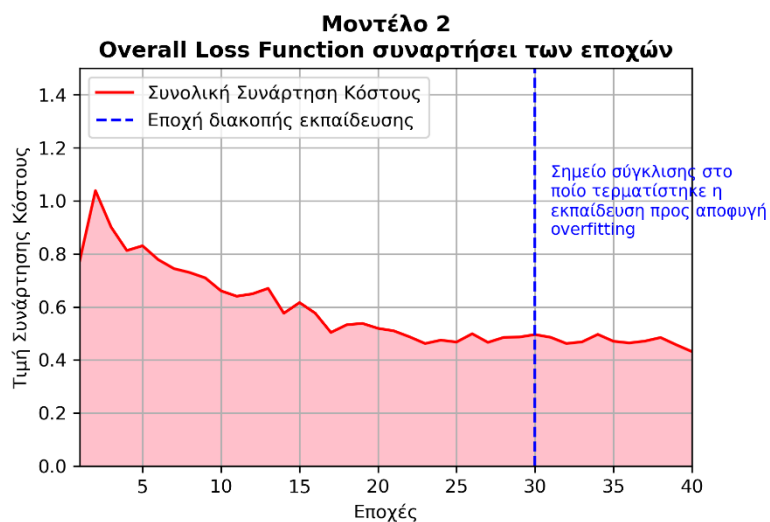
Παρατηρώντας τα Σχήματα που ακολουθούν φαίνεται ότι και στα τρία μοντέλα το συνολικό *loss function* έχει την τάση να σταθεροποιείται μετά την εποχή 20. Συνεχίζοντας την εκπαίδευση δεν φαίνεται να προκύπτει ιδιαίτερη μείωση της μετρικής. Εκτός αυτού το ίδιο ισχύει και με τα AP50 όπου φαίνεται ότι η βελτίωση του μοντέλου μετά την 20 εποχή φθίνει αρκετά. Στην πραγματικότητα στα περισσότερα μοντέλα που υλοποιήθηκαν παρατηρήθηκε ότι μετά το σημείο σύγκλισης του *loss function* οι μετρικές αξιολόγησης σταματούσαν να βελτιώνονται και σε κάποιες περιπτώσεις γινόντουσαν ακόμα και χειρότερες. Βάσει αυτού επιλέχθηκε η διαδικασία της μάθησης σε όλα τα μοντέλα να σταματήσει στο *checkpoint* της εποχής 30, ώστε να είμαστε σίγουροι ότι το μοντέλο έχει εκπαιδευτεί κατά το μέγιστο δυνατό αριθμό εποχών χωρίς όμως να υπάρξει κίνδυνος *overfitting*. Ένα αρκετά σημαντικό στοιχείο που πρέπει να σχολιαστεί είναι ότι στο διάγραμμα παρουσιάζεται το συνολικό *loss function* το οποίο αποτελείται από δυο βασικά τμήματα: της ταξινόμησης και της παλινδρόμησης. Στην περίπτωση μας, όσον αφορά την ταξινόμηση, επειδή έχουμε ένα δυαδικό πρόβλημα γίνεται η σύγκριση των αντικειμένων της κλάσης μας με αυτά του *background*. Το κομμάτι του *regression* αφορά την τοπικοποίηση των αντικειμένων, το οποίο ειδικά στις περιπτώσεις ανίχνευσης πολύ μικρών αντικειμένων αποτελεί ίσως το πιο δύσκολο κομμάτι στο να αντιμετωπιστεί ορθά.



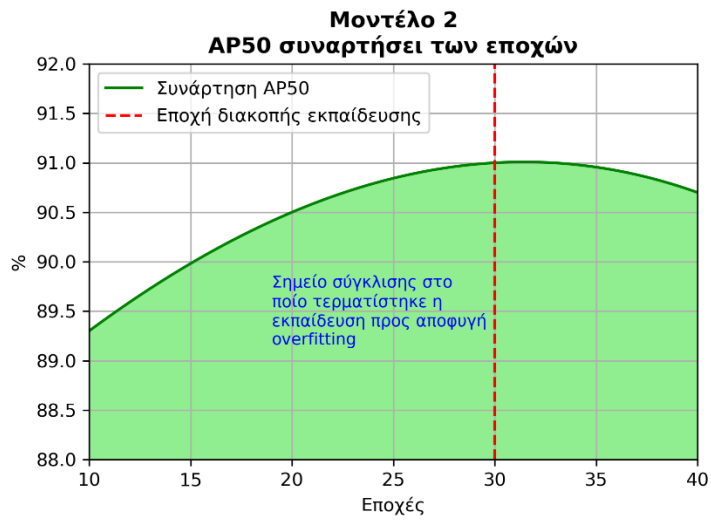
Σχήμα 4.8: Συνάρτηση του συνολικού loss function του μοντέλου 1 με τις εποχές



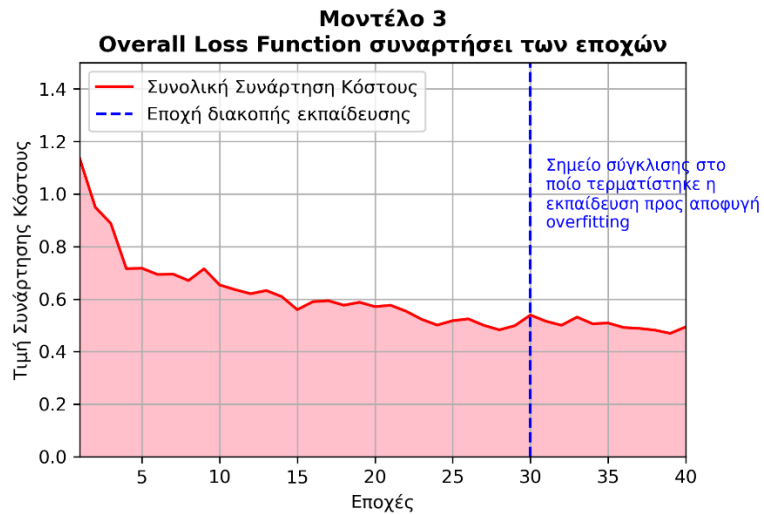
Σχήμα 4.9: Συνάρτηση του AP50 του μοντέλου 1 με τις εποχές



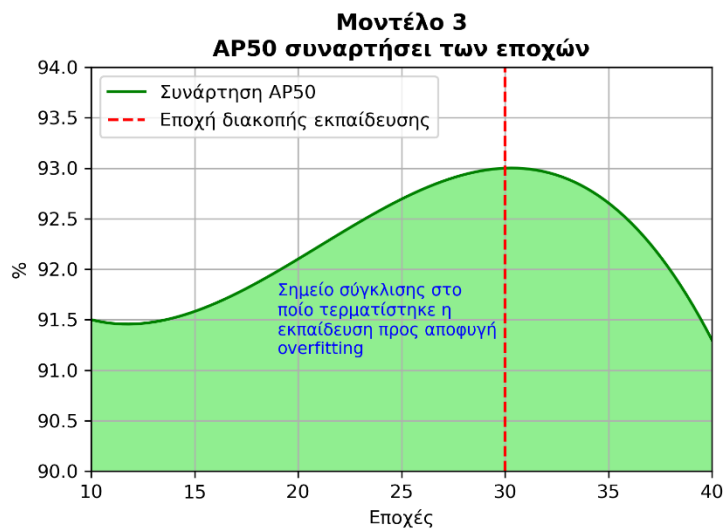
Σχήμα 4.10: Συνάρτηση του συνολικού loss function του μοντέλου 2 με τις εποχές



Σχήμα 4.11: Συνάρτηση του AP50 του μοντέλου 2 με τις εποχές



Σχήμα 4.12: Συνάρτηση του συνολικού loss function του μοντέλου 3 με τις εποχές



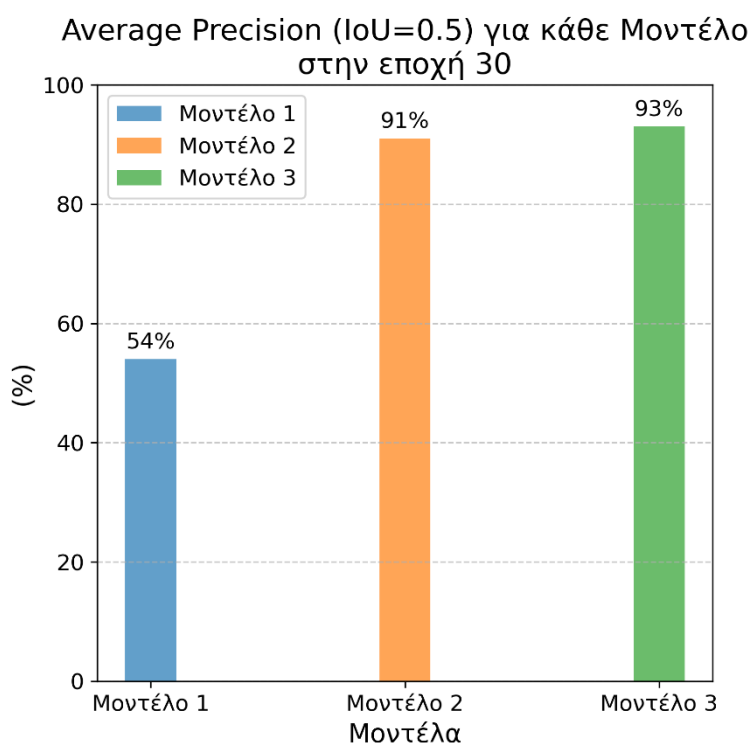
4.4.1 Ποσοτική αξιολόγηση μοντέλων

Για να γίνει η ποσοτική αξιολόγηση των μοντέλων έπρεπε να χρησιμοποιηθούν κάποιες μετρικές. Επειδή, το πρόβλημα της ανίχνευσής που προσπαθήσαμε να λύσουμε ήταν δυαδικό, δεν θα χρησιμοποιείται ο όρος mAP μιας και έχουμε μια κλάση, για αυτό το λόγο από εδώ και πέρα θα αναγράφεται απλά ως AP (αφού το mAP αποτελεί τον μέσο όρο των AP των διάφορων κλάσεων). Οι δύο βασικές μετρικές που χρησιμοποιήθηκαν ήταν το Precision και το Recall (αναλυτικά κεφάλαιο 2). Το precision βοήθησε στην κατανόηση της ακρίβειας με την οποία είχε βρει το μοντέλο τα δείγματα, δηλαδή όσο πλησιάζει την μονάδα σημαίνει ότι οι προβλέψεις του μοντέλου είναι σωστές (δεν έκανε λανθασμένες περιττές προβλέψεις). Το recall δείχνει πόσα δείγματα έχει βρει το μοντέλο ασχέτως αν έχει κάνει κάποιες παραπάνω λάθος προβλέψεις.

Οι ακριβείς μετρικές που χρησιμοποιήθηκαν ήταν οι: AP (IoU=0.5), AR50 (IoU=0.5) και AP (IoU 0.40:0.70). Σαν μέγεθος το AP (IoU=0.40:0.70) προέκυψε ως: $AP@0.40:0.70 = \frac{AP40+AP45+AP50+AP55+AP60+AP65+AP70}{7}$, δηλαδή συνυπολόγισε τα average precision για τα διαφορετικά IoU. Στην περίπτωση του δικού μας προβλήματος τα AP με μεγάλα IoU ιδιαίτερα πάνω από 70 δεν παρουσίασαν μεγάλη αξία, διότι τα προς ανίχνευση αντικείμενα ήταν μικρού μεγέθους (μικρότερα και από 50 εικονοστοιχεία). Αυτό συνεπαγόταν ότι αν το bb την πρόβλεψης παρουσίαζε μετατόπιση ενός εικονοστοιχίου σε έναν άξονα, σε σχέση με το bb στα δεδομένα αληθείας, η IoU τους έφθινε σημαντικά. Ενδεικτικά, ένα bb (μιας φυσαλίδας) με διαστάσεις $6 \times 6 = 36$ εικονοστοιχεία, με μετατοπισμένη πρόβλεψη κατά 1 μονάχα εικονοστοιχείο σε κάθε άξονα, οδηγεί το τελικό $IoU = \frac{25}{47} < 0.6$ και συνεπώς θα λογιζόταν σαν λάθος πρόβλεψη, σε όλες τις μετρικές για IoU 0.6 και άνω. Για αυτό και επιλέχθηκε να χρησιμοποιηθεί μια παραλλαγή της κλασσικής μετρικής AP50:95 η AP40:70, ώστε να ήταν πιο αντιπροσωπευτική στα δεδομένα μας. Αφού λάμβανε υπόψιν και τιμές για μικρότερα IoU χωρίς να συνυπολόγιζε για τα μεγάλα το οποίο θα οδηγούσε σε παραπλανητικά αποτελέσματα.

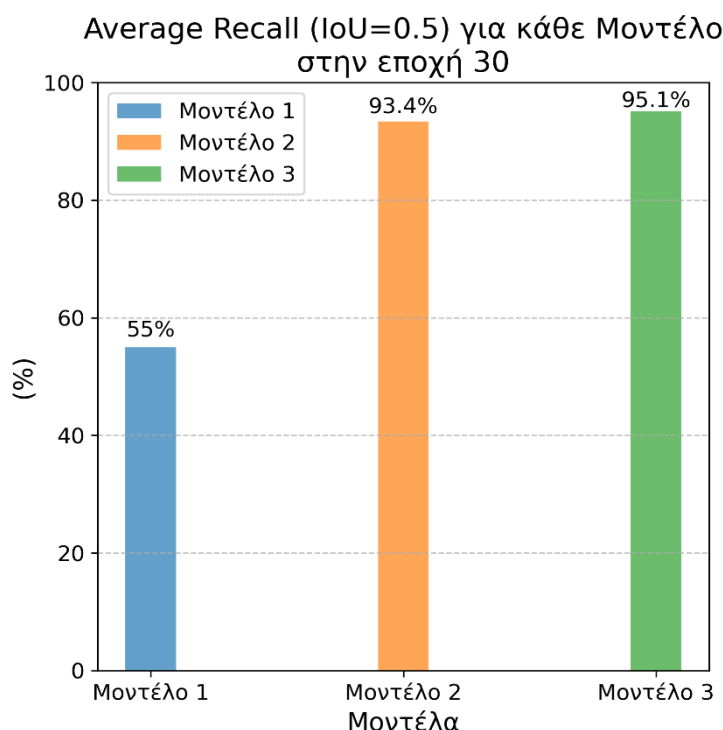
Η πρώτη μετρική που θα παρουσιαστεί είναι η Average Precision (IoU=0.5) (Σχήμα 4.14), βάσει της οποίας φαίνεται ότι τα πρώτο μοντέλο δεν είχε καταφέρει να βρει τις φυσαλίδες με μεγάλη ακρίβεια, καθώς λόγω του μεγάλου μεγέθους των αγκιστρώσεων που είχε παρουσιάζει μεγάλη αδυναμία στις μικρές σε μέγεθος

φουσαλίδες. Σε αντίθεση με το πρώτο μοντέλο τα επόμενα δύο παρουσίασαν τιμές στην μετρική μεγαλύτερες του 90% οι οποίες μας έδειξαν ότι το μοντέλο ανίχνευε τα δείγματα με αρκετά μεγάλη ακρίβεια. Εκτός αυτού η συγκεκριμένη μετρική είναι εξαιρετική αφού στόχος ήταν η ανίχνευση πολύ μικρών αντικειμένων, αλλά και αντικειμένων διαφορετικών κλιμάκων. Η υψηλή τιμή της μετρικής οδήγησε στο συμπέρασμα ότι είχαν επιτευχθεί πολύ καλές επικαλύψεις ανάμεσα στα bb της πρόβλεψης με τα δεδομένα αληθείας, συνεπώς υπήρξε μεγάλη επιτυχία στην τοπικοποίηση των φουσαλίδων. Η αιτία που οι συγκεκριμένες τιμές δεν ήταν ακόμα υψηλότερες οφειλόταν (στα Μοντέλα 2,3) κυρίως με ζητήματα στις ετικετοποιήσεις. Στο επόμενο κεφάλαιο που παρουσιάζεται η ποιοτική αξιολόγηση γίνεται αντιληπτό ότι σε αρκετές εικόνες κατά την διαδικασία της ετικετοποίησης είχαν παραληφθεί κάποιες μικρές φουσαλίδες ή κάποιες που ήταν πολύ δύσκολες να αναγνωριστούν εξαιτίας του φόντου. Παρόλα αυτά το μοντέλο κατάφερε να ανιχνεύσει και αυτές, οι οποίες όμως στις μετρικές λογίστηκαν ως False Positive. Αφού το Precision προκύπτει ως: $precision = \frac{True\ Positives}{False\ Positives + True\ Positives}$ η μετρική αποδυναμώθηκε. Συνολικά, η συγκεκριμένη μετρική έδειξε ότι τα δύο μοντέλα (2,3) που υλοποιήθηκαν παρουσίασαν πολύ καλά στοιχεία τόσο στην τοπικοποίηση όσο και στην ακρίβεια της ανίχνευσης.



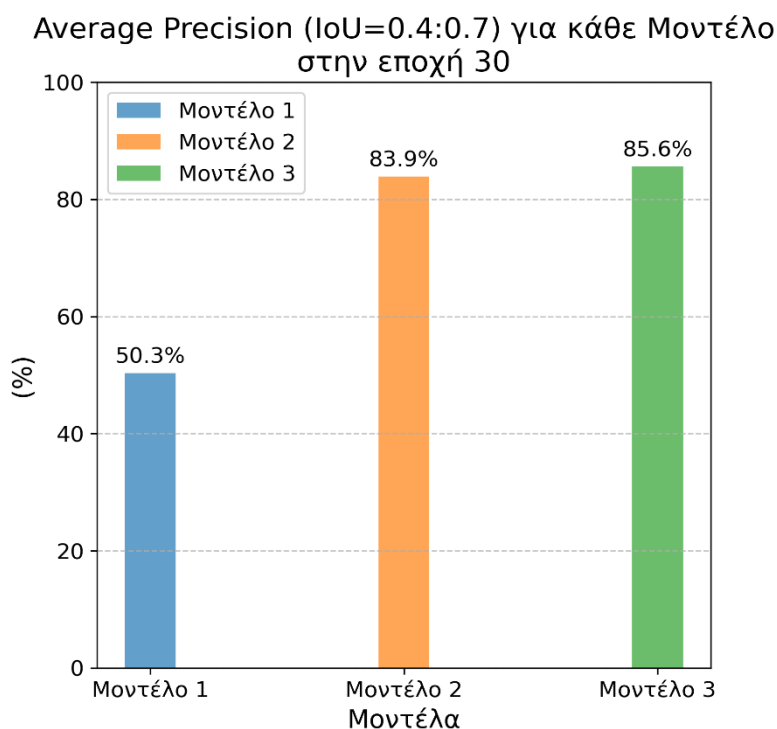
Σχήμα 4.14: Συγκριτικά αποτελέσματα των 3 μοντέλων, για την μετρική AP (IoU=0.5) μετά από 30 εποχές

Η δεύτερη μετρική που χρησιμοποιήθηκε ήταν το Average Recall (IoU=0.5) (Σχήμα 4.15) το οποίο ανέδειξε την αδυναμία του πρώτου μοντέλου στην ανίχνευση των φυσαλίδων (κυρίως μικρού μεγέθους). Η τιμή 55% υποδεικνύει ότι το συγκεκριμένο μοντέλο κατάφερε να βρει μόλις το 55% των φυσαλίδων. Αυτό οφείλεται στην αδυναμία του συγκεκριμένου μοντέλου να ανιχνεύσει φυσαλίδες μικρού μεγέθους, λόγω των αγκυρώσεων που επιλέχθηκαν. Αντίθετα τα άλλα δύο μοντέλα κατάφεραν να βρουν το 93.4% και το 95.1% αντίστοιχα. Όπως και στις τιμές του AP που παρουσιάστηκαν η συγκεκριμένη τιμή για IoU=0.5 (κατώφλι) έκαναν αντιληπτή την ποιοτική τοπικοποίηση του μοντέλου μας. Εκτός αυτού χάρις στην συγκεκριμένη μετρική αντιληφθήκαμε ότι ειδικά το τρίτο μοντέλο κατάφερε να εντοπίσει το 95.1% των συνολικών φυσαλίδων. Η συγκεκριμένη τιμή είναι αρκετά εντυπωσιακή καθώς τα αντικείμενα που ανιχνεύθηκαν ήταν μικρού μεγέθους αλλά και μεταβαλλόμενης γεωμετρίας και ραδιομετρίας. Συνολικά από την συγκεκριμένη μετρητική αντιληφθήκαμε ότι το τελικό μοντέλο είχε ελάχιστα False Negatives (αφού $recall = \frac{True\ Positives}{False\ Negatives + True\ Positives}$) και ότι κατάφερε να ανιχνεύσει με πολύ υψηλά ποσοστά το σύνολο των φυσαλίδων.



Σχήμα 4.15: Συγκριτικά αποτελέσματα των 3 μοντέλων, για την μετρική Average Recall (IoU=0.5) μετά από 30 εποχές

Κλείνοντας, το τελευταίο στοιχείο ποσοτικής αξιολόγησης που αξιολογήθηκε ήταν η μετρική AP (IoU 0.4:0.7) (Σχήμα 4.16). Η αξία της συγκεκριμένης μετρικής ήταν μεγάλη, καθώς εξαιτίας της δομής της αναδεικνύει τα κατώφλια IoU στα οποία αρχίζει η αδυναμία του μοντέλου, δηλαδή ειδικά για μεγάλα αντικείμενα δίνει πολύ πληροφορία όσον αφορά την τοπικοποίηση. Το πρώτο μας μοντέλο όπως και στις άλλες μετρικές δεν είχε ικανοποιητικά αποτελέσματα έτσι παραλείπεται. Τα μοντέλα 2,3 παρουσίασαν 83.9% και 85.6% αντίστοιχα. Από τον τρόπο υπολογισμού της μετρικής σε συνδυασμό με την τιμή της μετρικής AP (IoU=0.5) γίνεται εμφανές ότι τα AP για τις τιμές τους IoU μεγαλύτερες από 0.5 παρουσιάζουν χειρότερες τιμές. Αυτό ήταν αναμενόμενο όπως σχολιάστηκαν και πρωτύτερα εξαιτίας του μεγέθους των αντικειμένων. Παρόλα αυτά η συγκεκριμένη μετρική ενισχύει ακόμα την πεποίθηση ότι η τοπικοποίηση των αντικειμένων μας έγινε με πολύ μεγάλη ακρίβεια, καθώς παρότι ήταν μικρά αντικείμενα τα AP διατήρησαν πολύ υψηλές τιμές ακόμα και για αρκετά μεγάλα IoU (αναλογικά με το μέγεθος των δεδομένων προς ανίχνευση). Συμπερασματικά από την συγκεκριμένη μετρική παρατηρήσαμε ότι εμφανώς όσο αυξήθηκε το κατώφλι IoU το AP μειώθηκε, κάτι που όμως ήταν αναμενόμενο αφού αντικείμενα που παρότι έχουν ανιχνευθεί από το μοντέλο δεν θεωρήθηκαν ως True Positive αφού δεν ικανοποιούσαν το κριτήριο της IoU.



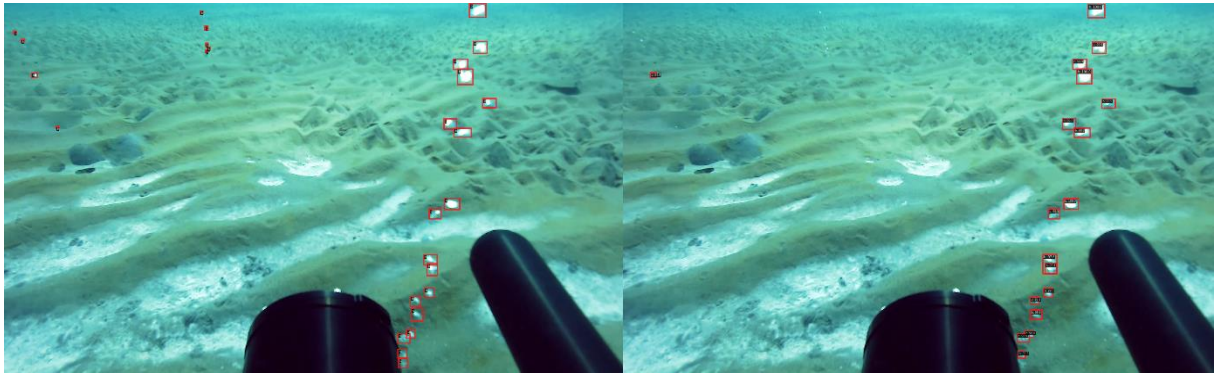
Σχήμα 4.16: Συγκριτικά αποτέλεσμα των 3 μοντέλων, για την μετρική Average Recall (IoU=0.4:0.7) μετά από 30 εποχές

Συνολικά, οι μετρικές των μοντέλων μας ανέδειξαν τις αδυναμίες του πρώτου μοντέλου καθώς και τα προτερήματα των άλλων δύο. Το βέλτιστο μοντέλο (3) παρουσίασε εξαιρετική ακρίβεια στην ανίχνευση των αντικειμένων, χωρίς να κάνει περιττές λάθος προβλέψεις. Εκτός αυτού είχε την δυνατότητα εύρεσης του 95.1% των συνολικών φυσαλίδων του σετ ελέγχου, το οποίο ήταν αρκετά ελπιδοφόρο. Λαμβάνοντας υπόψιν τα αποτελέσματα των της ποσοτικής αξιολόγησης το βέλτιστο μοντέλο που υλοποιήθηκε κρίθηκε ως αρκετά επιτυχημένο αν και μελλοντικά υπάρχουν βελτιώσεις που θα μπορούσαν να γίνουν και θα παρουσιαστούν στο επόμενο κεφάλαιο.

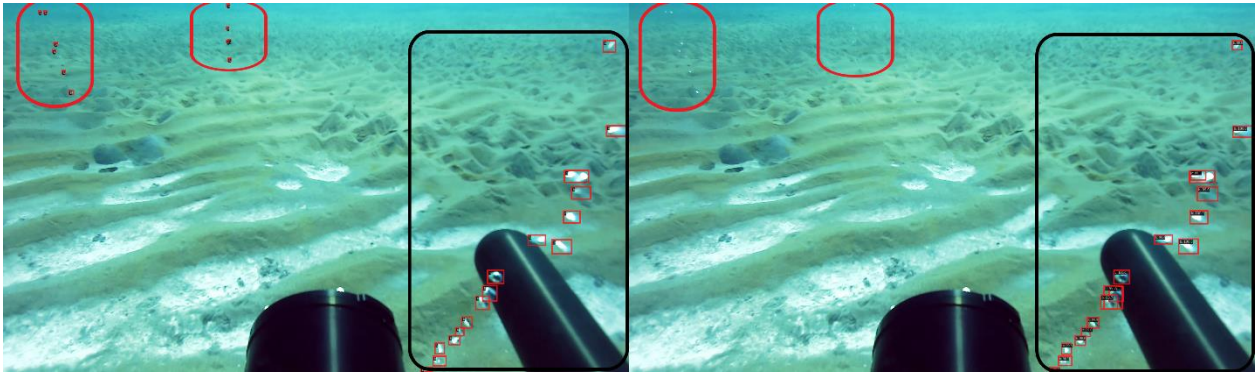
4.4.2 Ποιοτική αξιολόγηση μοντέλων

Όσον αφορά την ποιοτική αξιολόγηση αυτή θα χωριστεί σε τμήματα. Αρχικά θα παρουσιαστούν μερικά παραδείγματα και από τα τρία μοντέλα σε αντιδιαστολή με την ετικετοποίηση που έγινε αρχικά και θα σχολιαστούν. Στην συνέχεια θα αναλυθούν εκτενέστερα κάποια στοιχεία του τελικού μοντέλου (3) και θα παρουσιαστούν και αποτελέσματα του μοντέλου σε δεδομένα που δεν έχουν σχέση με αυτά της εκπαίδευσης και του ελέγχου.

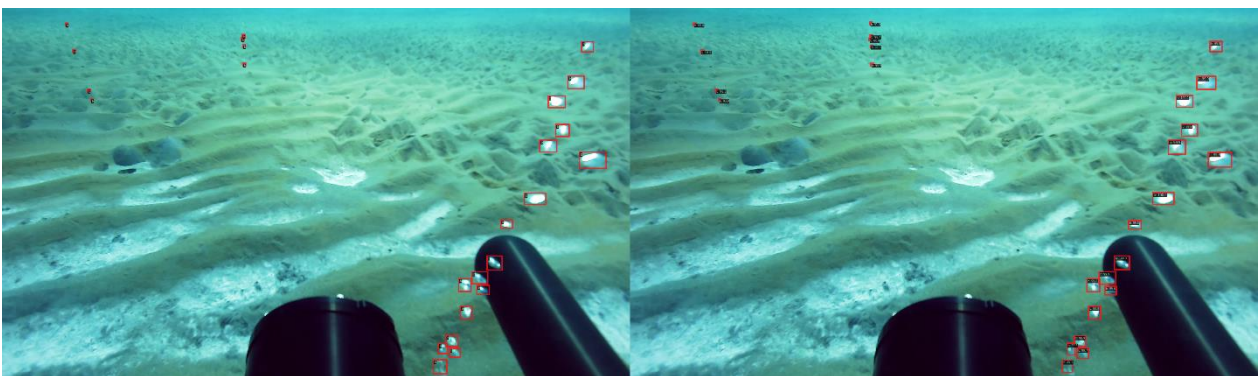
Ξεκινώντας από το μοντέλο 1, παρουσίασε ικανοποιητικά αποτελέσματα σε φυσαλίδες μεσαίου μεγέθους (Σχήμα 4.17), αλλά εμφάνισε ζητήματα στα αντικείμενα μικρότερου μεγέθους, εξαιτίας της λανθασμένης εκτίμησης που έγινε στα anchors (Σχήμα 4.18). Αντίθετα τα μοντέλα 2 και 3 τα είχαν εξαιρετικά αποτελέσματα (Σχήματα 4.19, 4.20) γεγονός που κατέστη σχεδόν αδύνατον να παρατηρηθούν οι μεταξύ τους διαφορές με γυμνό μάτι (αυτό ήταν αναμενόμενο μιας και παρουσίασαν πολύ κοντινές τιμές στις μετρικές). Τα ζητήματα που παρουσίασαν τα συγκεκριμένα μοντέλα αφορούσαν περιπτώσεις όπου οι φυσαλίδες ήταν σε πυκνές συστάδες. Αυτές οι περιπτώσεις ήταν δύσκολο να αξιολογηθούν ακόμα και κατά την ετικετοποίηση, ενδεικτικά παραδείγματα φαίνονται στα Σχήματα 4:21, 4:22. Σε κάποιες από αυτές τις περιπτώσεις τα αποτελέσματα των μοντέλων ήταν πιο αντιπροσωπευτικά από αυτό που έγινε κατά την ετικετοποίηση. Εκτός αυτού το μεγαλύτερο ζήτημα που παρουσίασαν στο μοντέλο 2 αλλά επιδιορθώθηκε με το 3, ήταν ότι σε κάποιες περιπτώσεις έγιναν σφάλματα όταν οι φυσαλίδες ήταν πολύ κοντά χρωματικά στο φόντο (Σχήματα 4:23, 4:24), το οποίο ήταν αρκετά λογικό, διότι σε αυτές τις περιπτώσεις ακόμα και ένας άνθρωπος για να τις αναγνωρίσει έπρεπε να σκεφτεί αρκετά σύνθετα και να βασιστεί κυρίως σε χωρικά χαρακτηριστικά όπως στην συσχέτιση τους με την υπόλοιπη ροή των φυσαλίδων.



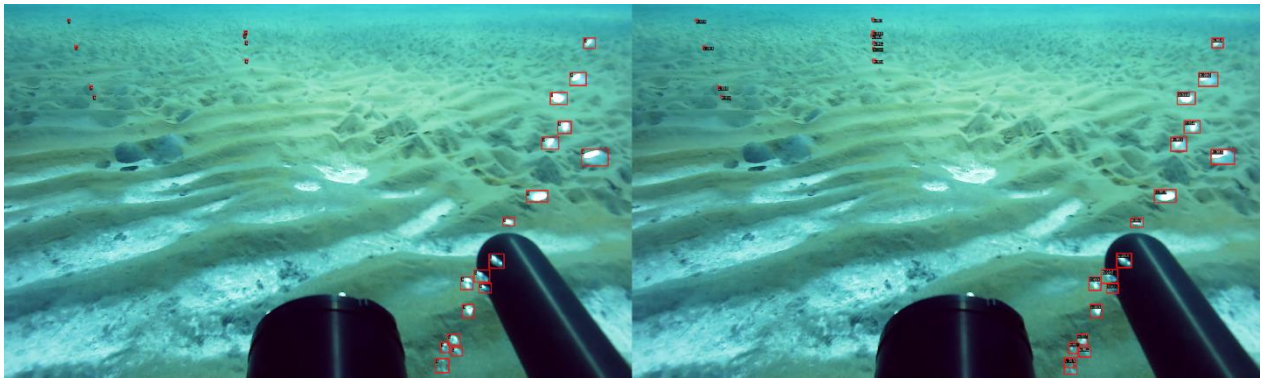
Σχήμα 4.17: Αριστερά: Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (1). Περίπτωση όπου το μοντέλο έχει ανιχνεύσει πολύ ικανοποιητικά συστάδα φυσαλίδων μεσαίου μεγέθους



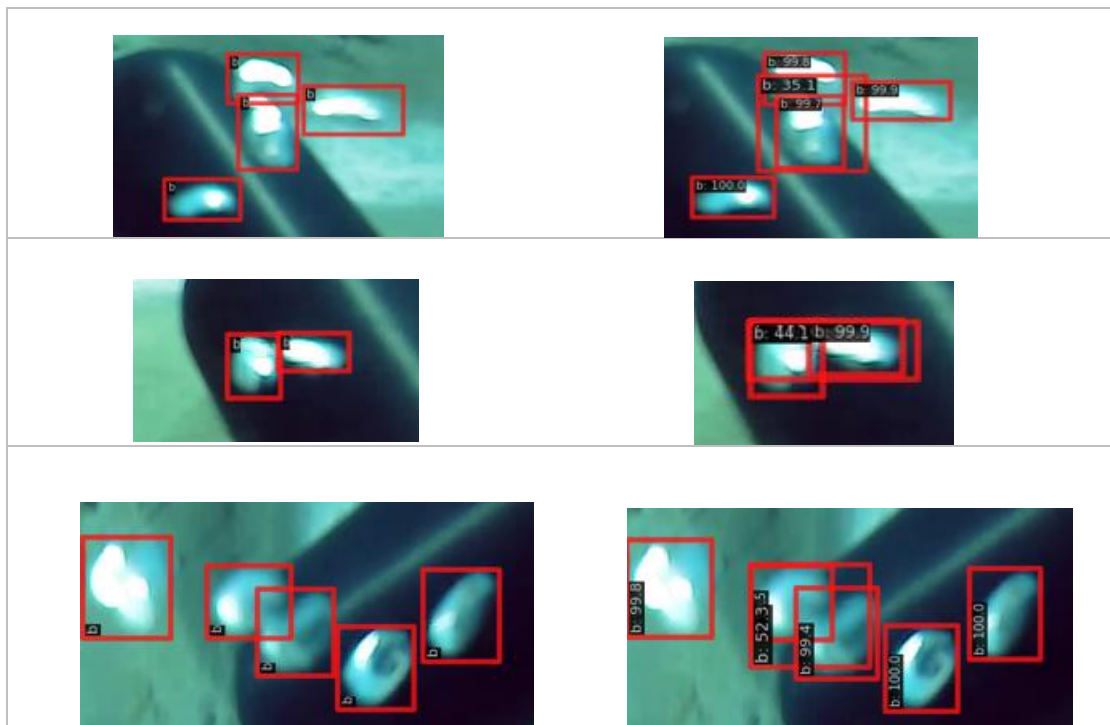
Σχήμα 4.18: Αριστερά: Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (1). Περίπτωση όπου το μοντέλο έχει ανιχνεύσει πολύ ικανοποιητικά συστάδα φυσαλίδων μεσαίου μεγέθους (μαύρο παραλληλόγραμο), αλλά έχει αποτύχει πλήρως στις φυσαλίδες μικρού μεγέθους (κόκκινα σχήματα)



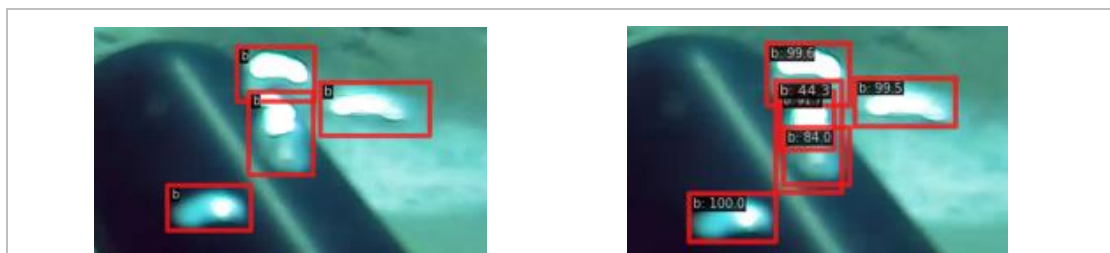
Σχήμα 4.19: Αριστερά: Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (2). Περίπτωση ιδανικών αποτελεσμάτων

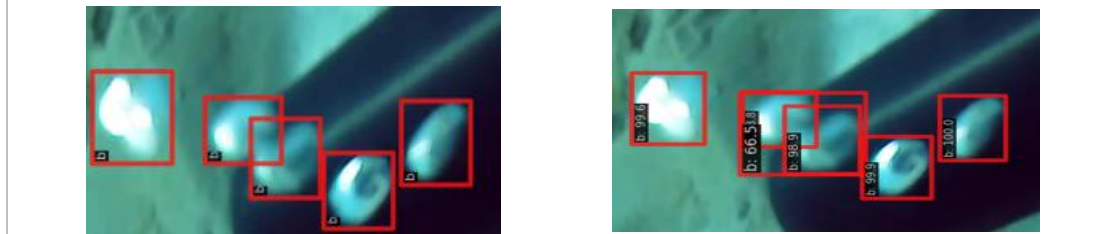
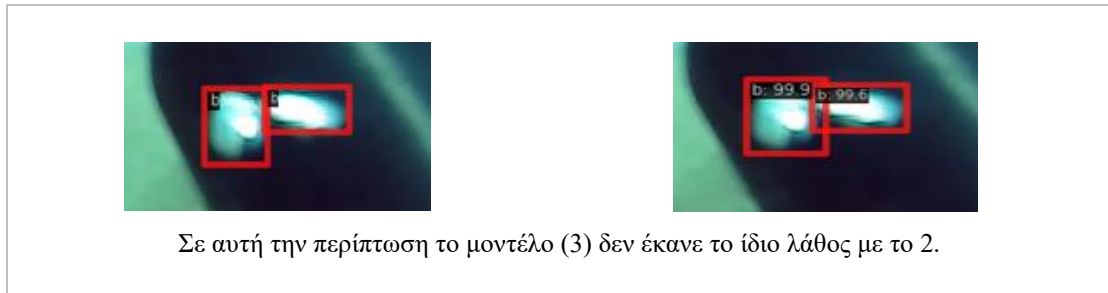


Σχήμα 4.20: Αριστερά: : Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (3).
Περίπτωση ιδανικών αποτελεσμάτων

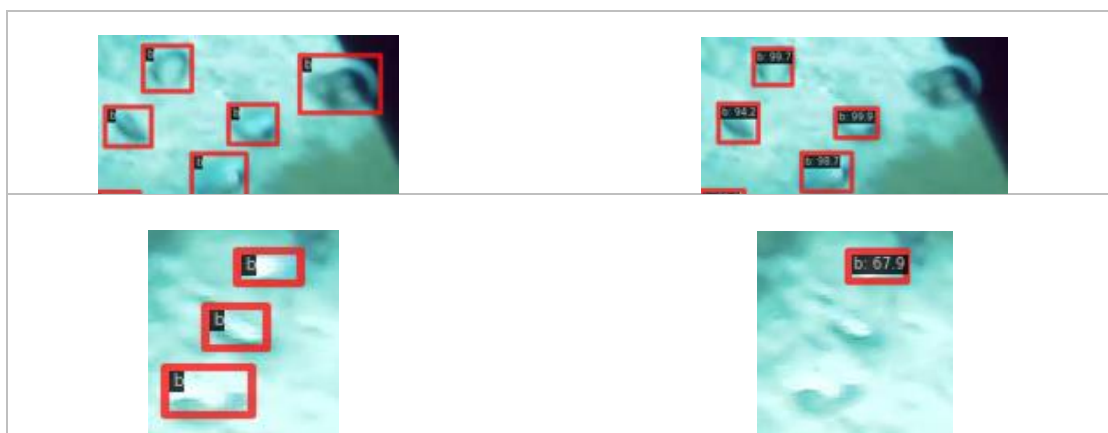


Σχήμα 4.21: Αριστερά: : Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (2).
Συστάδες όπου το μοντέλο έκανε λάθος ανιχνεύσεις





Σχήμα 4.22: Αριστερά: : Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (3).
Συστάδες όπου το μοντέλο έκανε λάθος ανιχνεύσεις

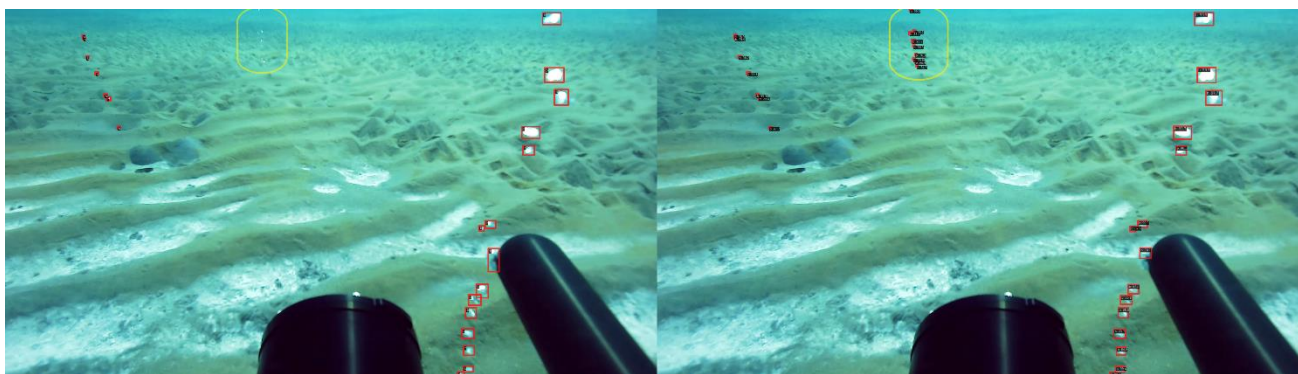


Σχήμα 4.23: Αριστερά: : Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (2).
Λάθη στις ανιχνεύσεις που προέκυψαν κυρίως λόγω φόντου

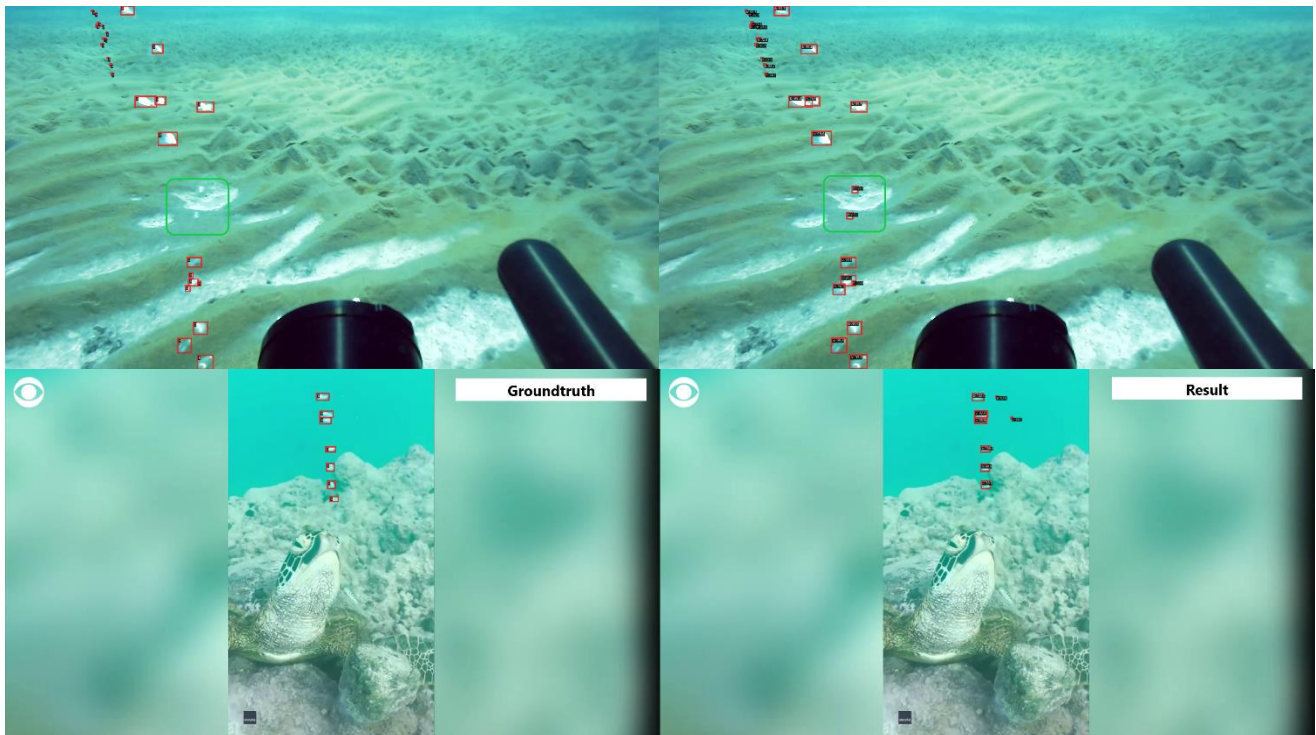


Σχήμα 4.24: Αριστερά: : Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου (3).
Λάθη που διορθώθηκαν σε σχέση με το μοντέλο 2

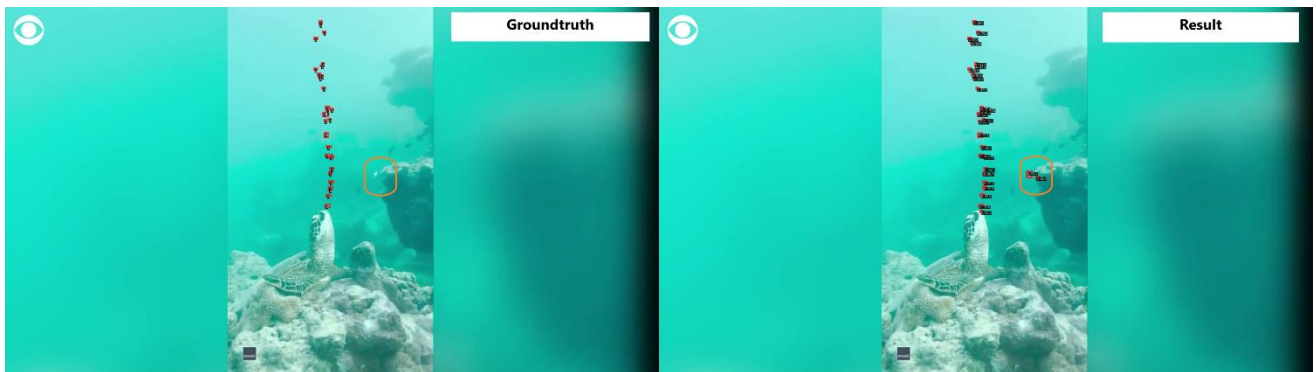
Εμβαθύνοντας στο μοντέλο 3 στο Σχήμα 4.25 παρουσιάζεται ένα πολύ χαρακτηριστικό παράδειγμα όπου παρότι μια ολόκληρη συστάδα φυσαλίδων δεν έγινε αντιληπτή κατά την ετικετοποίηση το μοντέλο κατάφερε να την εντοπίσει. Ένα ακόμα χαρακτηριστικό παράδειγμα γίνεται αντιληπτό στο Σχήμα 4.26 όπου παρότι οι φυσαλίδες δεν είχαν παρατηρηθεί κατά την ετικετοποίηση (καθώς έχουν πολύ κοντινό χρώμα με το φόντο) το μοντέλο κατάφερε να τις εντοπίσει με αρκετά υψηλά επίπεδα σιγουριάς (85%, 64%, 68% και 88% αντίστοιχα). Ακολούθως παρουσιάζεται στο Σχήμα 4.27 ένα από τα ζητήματα του μοντέλου όπου οφειλόταν στην υπερευαισθησία που παρουσίασε. Αυτό οδήγησε στο να προκύψουν λάθη και να ανιχνεύσει μικρά αντικείμενα του φόντου για φυσαλίδες. Αυτό το φαινόμενο ήταν πολύ σπάνιο και παρουσιάστηκε μονάχα σε μικρά μεγέθους αντικείμενα τα οποία εκ πρώτης όψεως έμοιαζαν πράγματι με φυσαλίδες. Τέλος, με χρήση του μοντέλου κάναμε προσπάθεια ανίχνευσης φυσαλίδων από δεδομένα πολύ διαφορετικά από αυτά της εκπαίδευσης. Τα αποτελέσματα παρουσιάζονται στο Σχήμα 4:28 όπου φαίνεται ότι το μοντέλο μας κατάφερε να επιτύχει πολύ ικανοποιητική ανίχνευση των αντικειμένων. Τα συγκεκριμένα αποτελέσματα κάνουν αντιληπτή την στιβαρότητα που παρουσίασε το υλοποιημένο μοντέλο. Μοναδικά λάθη που παρουσιάστηκαν αφορούσαν τις φυσαλίδες μεγάλου μεγέθους το οποίο ήταν αναμενόμενο, αφού τα anchors που χρησιμοποιήθηκαν κατά την εκπαίδευση του μοντέλου ήταν πολύ μικρά για ανίχνευση μεγάλων αντικειμένων.



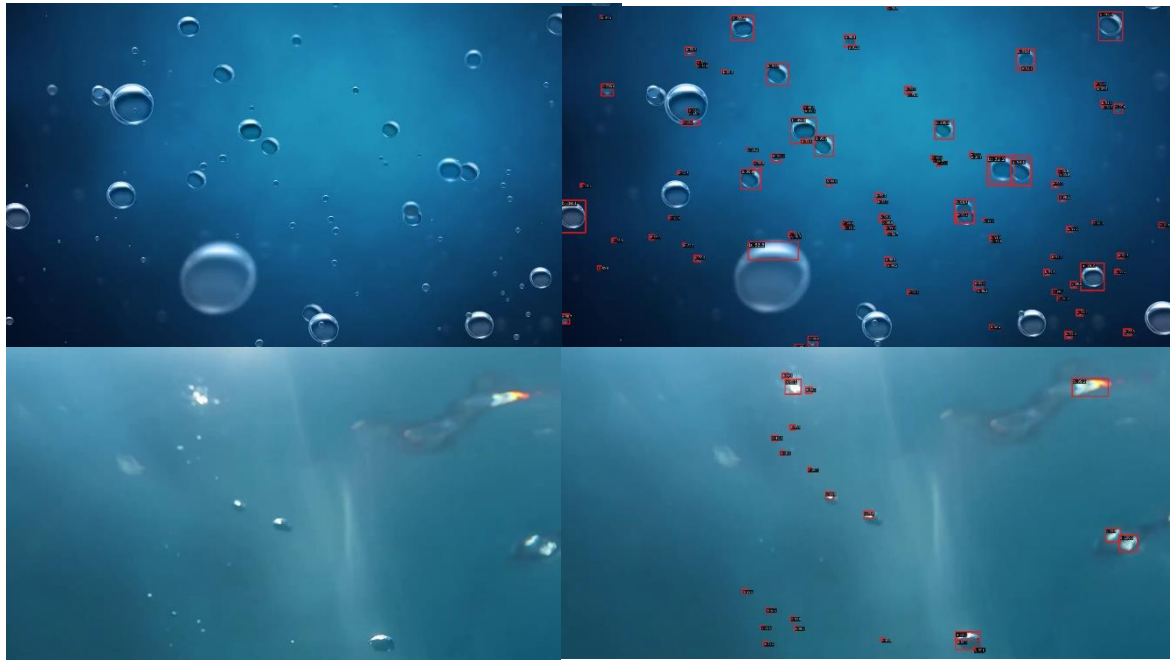
Σχήμα 4.25: Αριστερά: Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου. Περίπτωση όπου το μοντέλο έχει εντοπίσει συστάδα που δεν παρατήρηθηκε κατή την εκπαίδευση (κίτρινο σχήμα)



Σχήμα 4.26: Αριστερά: Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου. Περίπτωση όπου το μοντέλο έχει ανιχνεύσει φυσαλίδες που είναι αρκετά δύσκολο να παρατηρηθούν λόγω του φόντου (πράσινο παραλληλόγραμο).



Σχήμα 4.27: Αριστερά: Η εικόνα με τα δεδομένα αληθείας. Δεξιά: Το αποτέλεσμα του μοντέλου. Περίπτωση όπου το μοντέλο θεώρησε ως φυσαλίδες διαφορετικά αντικείμενα (πορτοκαλί σχήμα)



Σχήμα 4.28: Αριστερά: Οι αρχικές εικόνες. Δεξιά: Το αποτέλεσμα του μοντέλου.
Δεδομένα που δεν παρουσίαζαν κοινά χαρακτηριστικά με αυτά την εκπαίδευσης (πάνω συνθετικά,
κάτω πολύ κοντά στην επιφάνεια)

Κεφάλαιο 5

Συμπεράσματα και μελλοντικοί στόχοι

Λόγω της διττής φύσης της παρούσας διπλωματικής εργασίας τα συμπεράσματα και οι μελλοντικοί στόχοι θα χωριστούν σε δύο διαφορετικά τμήματα. Το πρώτο θα αφορά το υλισμικό σκέλος και τις αποστολές ενώ το δεύτερο θα παρουσιάζει στοιχεία για το μοντέλο που υλοποιήθηκε.

Όσον αφορά το υλισμικό σκέλος κρίνεται ότι η υλοποίηση που κάναμε ήταν εντός των ζητούμενων προδιαγραφών τόσο στο κομμάτι της αντοχής και της συμπεριφοράς της όσο και ως προς τα δεδομένα που συνέλεξε. Παρόλα αυτά θα μπορούσαν να γίνουν κάποιες αλλαγές στο σύστημα που αφορούν κυρίως την μείωση του κόστους του, ώστε να είναι ακόμα πιο προσιτό σε περισσότερες επιστημονικές ομάδες. Το βέλτιστο θα ήταν να υπάρχει το παρόν σύστημα σαν βάση υλοποίησης και κάθε ερευνητική ομάδα να κάνει τις προσαρμογές που θέλει κατά περίπτωση. Στο υλισμικό κομμάτι ενδιαφέρον θα παρουσίαζε και η προσαρμογή του συστήματος ώστε να δέχεται διαφορετικές (ή και πάνω από μια) κάμερες που θα μπορούσαν να συλλέγουν και πολυφασματικά δεδομένα. Κάποιοι ακόμα στόχοι της ομάδας μας θα ήταν να κάνουμε αλλαγές στον μικροϋπολογιστή του συστήματος ώστε να μπορούμε να τρέξουμε κάποια μοντέλα όπως αυτό που υλοποιήθηκε σε πραγματικό χρόνο και να κάνουμε το σύστημα μας πιο έξυπνο. Δηλαδή, θα μπορούσαμε να υλοποιήσουμε αλγόριθμο ώστε αφού κάνει την ανίχνευση των φυσαλίδων, οι σερβοκινητήρες να κινούν την κάμερα στο σημείο όπου θα έχει την μεγαλύτερη συγκέντρωση φυσαλίδων.

Μεγάλο ενδιαφέρον παρουσιάζουν και τα ευρήματα μας όσον αφορούν το μοντέλο που υλοποιήσαμε. Συνολικά πετύχαμε σε πολύ καλά επίπεδα την ανίχνευση των φυσαλίδων και το σετ δεδομένων που υλοποιήσαμε φαίνεται ότι έχει αρκετές δυνατότητες. Μελλοντικοί στόχοι θα ήταν αρχικά να εμπλουτίσουμε και με άλλες αποστολές το σετ δεδομένων μας με δείγματα υπό διαφορετικές συνθήκες ώστε να αυξήσουμε την στιβαρότητα του μοντέλου, αλλά και τις δυνατότητες ανίχνευσης. Εκτός αυτού μεγάλο ενδιαφέρον θα παρουσίαζε η υλοποίηση μοντέλων με διαφορετικές αρχιτεκτονικές όπως είναι το YOLO ή και με άλλες μεθοδολογίες εντελώς όπως είναι το instance segmentation, ώστε να συγκρίνουμε και τα αποτελέσματα και να βρούμε την αρχιτεκτονική που παρέχει τα καλύτερα δυνατά αποτελέσματα με την μικρότερη δυνατή υπολογιστή ισχύ. Τέλος, ένας ακόμα μελλοντικός στόχος είναι

να «ποσοτικοποιήσουμε» τις ανιχνευμένες φυσαλίδες ώστε να μπορούμε να παρατηρήσουμε μεταβολές που υπάρχουν ανάμεσα σε συνεχόμενες χρονικές στιγμές.

Αναφορές

- [1] Mouy, X., Black, M., Cox, K., Qualley, J., Mireault, C., Dosso, S., & Juanes, F. (2020). FishCam: A low-cost open source autonomous camera for aquatic research. *HardwareX*, 8, e00110
- [2] Humbert, J. W., Onthank, K. L., & Williams, K. (2023). The Open-source Camera Trap for Organism Presence and Underwater Surveillance (OCTOPUS). *HardwareX*, 13, e00394.
- [3] Almero, V. J. D., Falconit, M. G. B., Alejandrino, J. D., Concepcion, R. S., Vicerra, R. R. P., Sybingco, E., ... & Dadios, E. P. (2021, November). Development of a raspberry pi-based underwater camera system for inland freshwater aquaculture. In 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM) (pp. 1-6). IEEE.
- [4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.
- [5] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science*, 199, 1066-1073.
- [6] Kim, Y., & Park, H. (2021). Deep learning-based automated and universal bubble detection and mask extraction in complex two-phase flows. *Scientific reports*, 11(1), 8940.
- [7] Bhattarai, P., Krupiński, S., Unnithan, V., Maurelli, F., Secciani, N., Franchi, M., ... & Ridolfi, A. (2021, September). A deep learning approach for underwater bubble detection. In *OCEANS 2021: San Diego–Porto* (pp. 1-5). IEEE.
- [8] Al-Lashi, R. S., Gunn, S. R., & Czernski, H. (2016). Automated processing of oceanic bubble images for measuring bubble size distributions underneath breaking waves. *Journal of Atmospheric and Oceanic Technology*, 33(8), 1701-1714
- [9] <https://www.arducam.com/> Κατασκευάστρια εταιρία κάμερας.

- [10] <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
Κατασκευάστρια εταιρία Jetson-Nano
- [11] <https://www.raspberrypi.com> Κατασκευάστρια εταιρία μικροϋπολογιστών
Raspberry Pi
- [12] Chen, W., Liang, J., Yang, Z., & Li, G. (2019). A review of lithium-ion battery for electric vehicle applications and beyond. *Energy Procedia*, 158, 4363-4368.
- [13] Moore, S. W., & Schneider, P. J. (2001). A review of cell equalization methods for lithium ion and lithium polymer battery systems. *SAE Technical Paper*, 2001, 01-0959.
- [14] Menna, F., Nocerino, E., & Remondino, F. (2017). Flat versus hemispherical dome ports in underwater photogrammetry. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 481.
- [15] <https://libcamera.org/> Σελίδα της βιβλιοθήκης libcamera
- [16] <https://www.qgis.org/en/site/about/index.html> Σελίδα του προγράμματος QGIS
- [17] <https://mmdetection.readthedocs.io/en/latest/overview.html>
- [18] Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., & Zou, H. (2018). Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 145, 3-22.
- [19] Jaffe, J. S. (2014). Underwater optical imaging: the past, the present, and the prospects. *IEEE Journal of Oceanic Engineering*, 40(3), 683-700.