Boat Recognition and Automated Harbor Management System

by W.D.L.S.Weerasekara
198779P

Master of Science in Information Technology

Faculty of Information Technology

University of Moratuwa

July 2022

Boat Recognition and Automated Harbor Management System

W.D.L.S.Weerasekara

198779P

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfilment of the requirements of Degree of Master of Science in Information Technology

Declaration

I declare that this thesis is my work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student:	Signature of Student
W.D.L.S.Weerasekara	….………..……………………
	Date:

Supervised by:

Name of Supervisor	Signature of Supervisor
Dr. S.C.Premaratne	….………..……………………
	Date:

Dedication

This research is dedicated to all the energetic people who dedicate their lives for the upliftment of the fisheries industry.

Acknowledgement

I would like to acknowledge and express my appreciation to those efforts and support in one way or other contributed to the successful completion of this research.

First, I am deeply grateful for the supervision throughout the one year of my research and the helps received from my supervisor Dr. S. C. Premaratne, Department of Information Technology, Faculty of Information Technology, University of Moratuwa. I have learned so much from our project discussions. His willing to motivate me to contributed tremendously to my job.

Besides I am grateful to all academic staff from the Faculty of Information Technology, who shared their vast knowledge throughout these two years by providing me a good environment which influenced a lot to achieve this goal.

It is with great pleasure to thank the University of Moratuwa, Sri Lanka, for all its efforts and facilities that it has taken to contribute towards this postgraduate program.

Specially, I would like to thank my colleagues for their support to complete this research in many ways.

I am as ever, especially indebted to my husband, my son and parents for their love and support throughout my life to improve my career.

Abstract

Fisheries industry is a vital sector of Sri Lanka's economy since it is an island surrounded by a vast ocean. Over thousands of fishing vessels are departing to the ocean within a day from harbors all around the island. All the departing and arriving fishing vessels should have gone though an ample security check by the harbor authorities one by one. But with the COVID 19 pandemic situation and the social distancing procedure, harbor authorities are facing difficulties to detect and recognize fishing vessels by getting on the boats as before the pandemic situation. Also, currently harbors are using a manual, paper-based system for recording the information on boat departures and arrivals. This leads to the inefficiency of harbor management process, delays in rescue missions and failures of security missions. To solve these problems, this paper introduces a Boat Recognition and Automated Harbor Management System (BRAHMS) which is based on YOLO(You Only Look Once) v5 algorithm. A web-based solution is provided to manage fishing boat tracking information as one deliverable of the project. Also, YOLO based desktop applicationto recognize boats through the registered number is given as another outcome. Final deliverable is a backend reporting solution to send boat tracking information according to daily, weekly, monthly or yearly preschedule intervals. In this system, I have implemented a novel deskewing method for the slanted license plate recognition process. The deskewing process is aimed for three main approaches as auto deskewing, manual deskewing and a hybrid deskewing which uses both auto and manual processes together.

Keywords: Fishing Vessels, Fisheries industry, YOLOv5, Boat recognition, License plate recognition, image processing, license plate deskewing

Contents

Chapter 1

Introduction

1.1 Prolegomena

This chapter presents a basic understanding of the system, the background motivation for the research idea, aim and objectives of the project. Scope of the project and the problem statementalso described in this chapter.

1.2 Background and Motivation

Being a nation in an island, fisheries sector plays an significant part in Sri Lanka's economy.The fishery industry contains of three sub sectors as coastal, offshore/deep-sea and inland. Thisfisheries sector provides over 250000 employments to fishermen and over 100000 supporting facilities [4]. And it funds 1.2% to the country's Gross Domestic Production (GDP).

Over thousands of fishing vessels are lounged to the ocean within a day for fishing purpose. These fishing vessels are range in size from small Dinghies to large Multiday Fishing Vessels. Those are lounging to the sea through a harbor where the departure and arrival of fishing vessels are going through an ample security check by the harbor authorities one by one.

But with the COVID 19 pandemic, harbor authorities are facing difficulties detecting and recognizing fishing vessels arriving and departing the harbor due to social distancing and virus spreading issue. So, authorities are unable to get on the boat and performthe recognition process as they have been done before the pandemic situation. Therefore, a solution must be given to detect and recognize the boats with maintaining the social distancing.

The boat detection and classification system, specially designed for Sri Lanka Navy, gives a solution to the above problem. With this system, fishing boats are detecting from faraway and recognize with their identities. Detecting and classification is done through image processing.

Recognizing the boat by the shape is done using YOLO which the name implies "You Only Look Once". Handling images with YOLO is simple and direct. The input image splits into a lattice of cells, and cells are accountable for calculating bounding boxes that enclosean object. Instead of choosing specific parts of an image, YOLO algorithm predicts classes andbounding boxes in one go.

1.3 License Plate Recognition

All the fishing boats are registered in the Fisheries Department and have been offered a unique number plate for each fishing vessel. This license plate is appeared in outside view of the vessel.

As mentioned before this license plate is unique to a boat. So, a boat can be recognized from far away by detecting the license plate without violating the social distancing regulations duringCOVID 19 pandemic situation.

Figure 0-1.1: Unique License Plate of Fishing Vessels

Reading license plate is another challenge that can be solved using same YOLO algorithm. Before recognizing the letters and numbers in the license registration plate using YOLO, following twoapproaches should be done [6].

1. Crop the plate – then crop the detected license plate from the image
2. Detecting the license plate – detect the license plate in the boat and draw a rectangle around it.

After executing above two steps, then the characters and numbers should be recognized by applying character recognition in YOLO algorithm.

So the boat detection and classification system will differ from other existing boat recognition systems because this system is a combination of two detection methods: detecting via the vessel's shape and detecting via the unique license plate. So, both togetherwill produce a robust and straightforward system which delivers more accurate results.

Monitoring and tracking processes of the fishing vessels arriving and departing in the harbor are currently executed by a manual, paper-based system. This process is a time consuming and inefficient approach which may delay the whole system. Therefore, from this system, existing inefficient manual system will be converting to an efficient and accurate automated harbor management process.

1.4 Aim and Objectives

1.4.1 Research Aim

Aim of this research is to design, implement and evaluate a boat recognition and classificationsystem for Sri Lanka Navy and Coast Guard which would enable the harbor authorities to automate the harbor management process towards fishing vessels.

1.4.2 Research Objectives

- To research on how latest technologies can be used for the real time object recognitionprocess.
- To research on the technologies can be used for the character recognition for real timelicense plate reading.
- To design and develop a web-based solution for managing fishing boat trackinginformation.
- To design and develop a YOLO based desktop application to recognize boats via registered number.
- To design and develop a backend reporting solution to send boat tracking informationaccording to preschedule intervals (daily, weekly, monthly, etc.)
- To test and evaluate the system.

1.5 Scope of Research

This research aims to mitigate the difficulties occur in the existing harbor management systemsby automating the harbor management procedure of fishing vessel arrival and departures. The system will be able to recognize the fishing vessels by recognition techniques and able to provide full details related to given fishing vessel as well.

1.6 Problem Statement

Over thousands of fishing vessels are departing from harbors around the island for fishing purpose and thousands are arriving back each day. In out of fishing vessels in a harbor are managed by Sri Lanka Navy. They get into the boats and give a comprehensive security checkat every arrival and departure time at the harbor.

Monitoring and tracking processes of the fishing vessels arriving and departing in the harbor are currently executed by a manual system rather than a systematic approach. Recordsare maintained in paper documents. At the departure of a fishing vessel, the owner needs to submit several documents and the Navy authorities should record some important data relevantto the boat. Also, Navy authorities are recording some other important data too.

Any fishing vessel may consist of permanent crew as well as temporary members. Vessel crew must get verified their National Identity card or passport before the departure. A fishing boat may have two destinations. They are headed to the international sea or local sea area for fishing purposes. So, if the vessels heading towards the international sea, they need to submittheir passport to the Navy authorities for the verification. Otherwise NIC is enough.

Before the departure, authorities also record some data regarding the fishing vessel. They use these data for different situations such as rescue missions and security purposes. For an example, if there is a report of a cyclone in a particular are of the sea, they need to check out the possible vessels drove in that area. There are over 3500 fishermen all around the island and over 1500 fishermen are registered using vessel registration numbers at Fishing Vessel Monitoring System. So, when a vessel is in the sea, their locations will be captured time to timeusing GPS. These spatial data also need to record with the vessel registration number. But withthe paper-based system such data do not record clearly.

In a cyclone situation fisherman may gone missing or boats may capsize. So, the Navy shouldbe able to find out which vessels are in danger, how many fishermen have died, how many have survived, where they have gone fishing, which area they located now, etc.

Those type of information need to retrieve and analyze quickly in a disaster situation. Such important information cannot be retrieved from a manual – paper-based system easily. So, during such a situation, the Navy authority must get those data mostly from the Fisheries Department, which is a time-consuming process. So, they are having the need of an automatedsystem where the important information of vessels can be derived rapidly.

Also, when there is an illegal action reported in the sea such as human trafficking, drug trafficking and illegal fishing, they must access the important data of departed vessels in thatarea. In a manual system, such illegal activities are not recorded against fishing vesselinformation directly. If there is such an illegal activity reported, they can tally the vessel information with the incident and take necessary actions immediately. Navy uses that information for security purposes.

Time to time, such illegal actions are reported in the sea. For an example, Indian Navy informsthat Sri Lankan boat is responsible for a human trafficking or drug trafficking in the sea. So, with the existing manual system, Sri Lanka Navy cannot find information such asthe time of the departure of the mentioned boat and from which harbor it departed with the paper-based system. So, this may delay the investigation procedure. Currently, they must inform each harbor around the country to go through their records and search for the fishing vessel. Also, if one harbor noted down the vessel registration number incorrectly, the whole investigation process delays for a longer time or even fails completely. This search task is a one click away process if they have an automated system. So, this information is important for rescue missions as well as security missions.

With considering all the matters described above, automated harbor management system will be a worthy solution.

1.7 Structure of the thesis

This thesis has grouped in to eight Chapters. Chapter one (1) present the problem and the motivation of this thesis. Chapter Two (2) include the Literature Review Where the piece of papers that already discussed about the analytics, neural network, machine learning, image/video processing. Chapter Three (3) contains the details and capacities of the specific technologies and their usability to obtain the desired objectives. Chapter Four (4) elaborated the proper approach to the research problem to follow the path towards goal to fulfill the desired objectives. Chapter Five (5) discussed about the top-level design and the analysis to provide an outline structure of the research. Chapter Six (6) illustrates and discussed the in-detail explanation of the implementation of each step of the previously elaborated design. Further it revealed all algorithms, tools, models, libraries etc. Chapter Seven (7) followed the evaluation of the implemented system and its results. Chapter Eight (8) finally discussed the conclusion and the future works to be done to improve the findings and the observations.

Chapter 2

Literature Review - Watercraft Recognition

2.1 Introduction

The previous chapter presented the basic idea of the project with the background motivation. This chapter provides a literature survey of the existing systems which related to image recognitionparadigm of watercrafts.

Shipping industry is developing rapidly with the technology. Much researches are lounged related to automated port management.

2.2 Intelligent ship image/video detection and classification with RDCNN

One application of this area is intelligent ship image/video detection and classification with improved Regressive Deep Convolutional Neural Network [1]. They are presenting a RDCNNnetwork to attain the ship image/ video detection and classification task. They have experimented on seven types of ships with the detection and classification task with a self-builddataset. This method is able to detect ships in real time approach and could solve the problemof low recognition rate. Although they are able to develop an accurate and efficient method, this can be executed on small datasets.

2.3 Ship Classification using Image Dataset

Another research related to this area is Ship Classification using Image Dataset [2]. In this research they have developed an algorithm to classify ships based on RGB color scheme of the image. This algorithm is consisting of three sets of classification approaches which are bag offeatures, convolutional neural networks (CNNs), and SVM along with some preprocessing. The input is a ship image in RGB color format, and the output is the category of the given shipbelongs. Although this algorithm provides better performance with CNN, other two methods are underperforming due to lack of datasets.

2.4 N-YOLO Approach

N-YOLO approach is another research done for ship detection [3]. N-YOLO is a method that uses a noise level classifier, a synthetic aperture radar target potential area extraction module, and a detection module based on YOLOv5. The image of the possible target area is combined with the original photograph in this method. Then it produces an image with a bright target and less noise. Then the image is sent to YOLOv5 for recognition purpose. Although this method provides a better detection performance, it partially damages the information of ship edge.

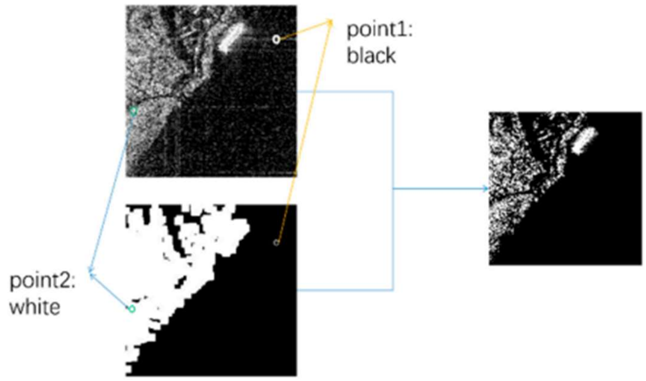


Figure 0-1.1: N- YOLO Approach

2.5 A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector

This research has presented a real-time license plate recognition using Convolutional Neural Networks (CNN) using YOLO object detection technology [15]. They have trained a network for automatic license plate recognition for each and every stage of the recognition process but for the character recognition stage have used two separate CNNs for letter recognition and number recognition. A public dataset has been used for the training purpose, including approximately 4500 annotated images from 150 vehicles. The positive side of this research is that it is able to achieve 93.53% accuracy. But for the downside, the output delays due to the bottleneck occur in character segmentation and recognition stages.

2.6 Summary

This chapter presented a literature survey of the existing systems related to watercraft recognition. Next chapter describes the environment information of the technological setup.

Chapter 3

Specific Technologies Used

3.1 Introduction

Previous chapter presented a literature survey of the existing systems related to watercraft recognition. This chapter describes the important technological environmental information in implementing the system.

3.2 Image Processing

There is a plethora of scenarios in which software image processing is required. They can be categorized into the following groups in general. [25].

1. Checking for presence
2. Object detection and localization
3. Measurement
4. Identification and verification

In many circumstances, the image taken by the camera is not managed straight within the program. As an alternative, the image is pre-processed to improve it for the intended job [25]. Noise reduction and brightness and contrast improvement are examples of preprocessing. Some of these procedures can be performed directly by the camera, reducing host CPU burden [26]. The camera must be calibrated to a physical environment in order to be used as a measuring device [26]. There are two types of camera calibration [27].

1. Geometric calibration
2. Color calibration

Geometric calibration is used to correct lens distortion [27]. We can also figure out the link among the camera's natural units, pixels, and real-world units like millimeters and inches [27]. Color calibration ensures that colors are reproduced accurately [27].

The higher the image feature and the better the outcomes of the image handling within the program, the better the preprocessing. When it comes to locating parts, it's common to use matching [26]. This entails searching for areas that are alike to or identical to a predetermined template. This template might be an image or a geometric design with edges and geometric characteristics [26].

Measurement using image processing is most commonly used in alignment or inspection applications [25]. Edge detection algorithms are used in most measurement procedures [28]. An edge in a picture is a region where the image intensity changes dramatically. It analyzes the image's gray levels and, as a result, recognizes forms, measures distances, and calculates geometry [28].

3.3 YOLOv5

YOLO is a state-of-the-art machine learning model for object detection. There are different versions of YOLO and by the time the most advanced YOLO version is YOLOv5 [16]. Since YOLO is an object detection model it takes in an image or batch of images as the input. After processing this image, it puts bounding boxes on it which expresses that one object detected in each bounding box [16]. These bounding boxes are its output. It also recognizes what object it is inside each bounding box. So, YOLO is a machine that puts such bounding boxes on on pictures. But YOLO does not directly draw or render these boxes onto the images. Instead for each bounding box it provides with several values as follows.

1. Xmin
2. Xmax
3. Ymin
4. Ymax
5. Label
6. Confidence

For one bounding box it provides an X min and X max, a Y min and Y max to represent where this bounding box is. A fifth value stands for the label or the ID of the object. Final value is the confidence and it expresses how sure it is with this entire result [16].

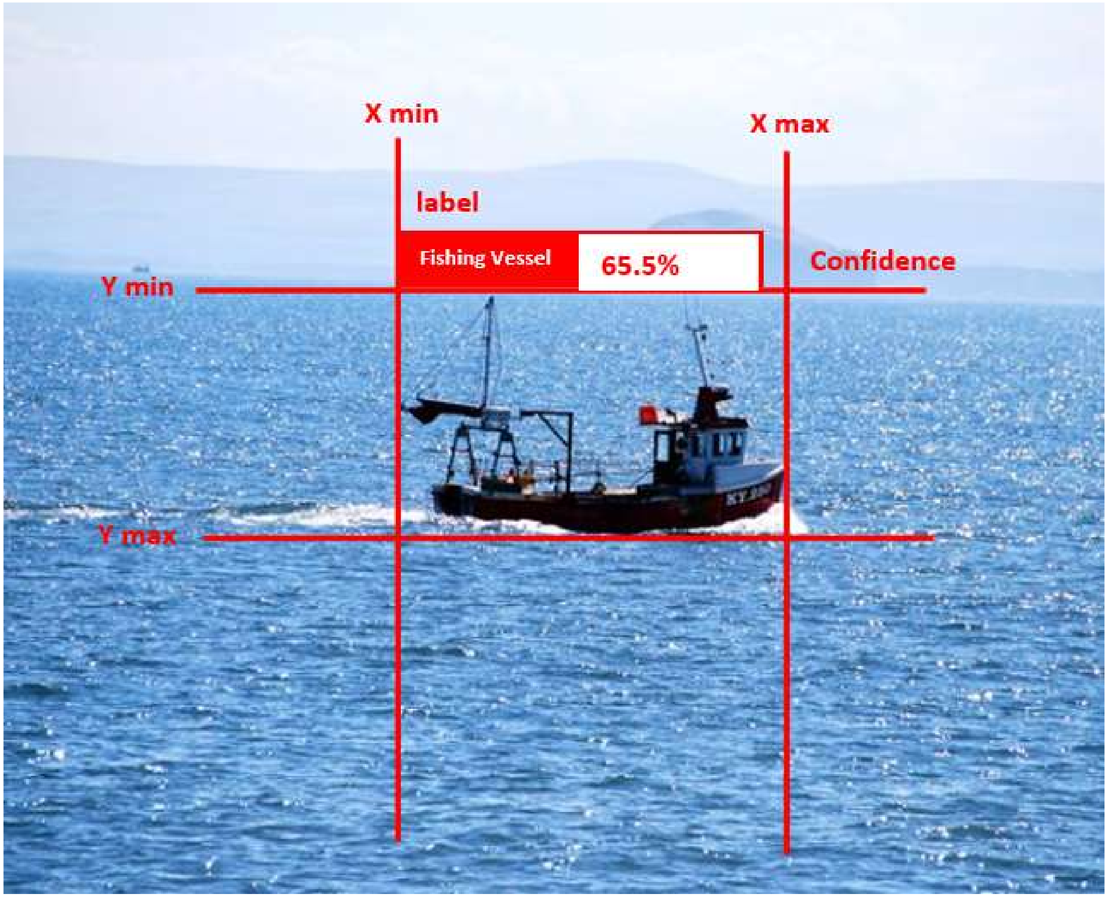


Figure 0-1.1: Values of the bounding box in YOLOv5

When assuming YOLO as a function its output should be a series of vectors [17]. The users can directly manipulate the output factors, or the user can choose to output the results to data frame or csv file. Also, the user can use the raw data for some other programming tasks or can finetune the model as per the user's perspective [17].

3.4 OpenCV

OpenCV is used for the image recognition process. It is an open-source package that is a cross platform for Python. This library widely taken for video, motion detection and image recognition purposes.

It is created to support standard computer vision and machine learning algorithms. OpenCV is a cross platform library and it is an open-source software which the user can have the access to the source code and if the user wants to change the OpenCV and add new features to OpenCV, he can just download the source code and make changes to it as preferred [19]. The most effective fact about OpenCV is that the user does not have to write everything by himself as most of the algorithms are standard algorithms and the code is readily readable. OpenCV is capable of image analyzing and processing

[18]. It can extract frames from a real time video and can run algorithms on top of those frames to manipulate information.

OpenCV can be utilized for many applications as follows [19].

- Face detection processes
- Object detection processes
- Tracking human actions for security purposes
- Tracking object movements
- 3D modeling apps
- Recognizing similar images
- Image editing processes
- Etc.

3.5 Optical Character Recognition (OCR)

The contents of an image can be deduced just by looking at it. We can read the words on the image because we view it as text. Computers, on the other hand, do not work in the same way. To comprehend that, they require something more comprehensive and well-organized. This is when OCR (Optical Character Recognition) comes into play. This technique is highly effective for recognizing car plates from a camera or handwritten documents that need to be transformed into multiple copies [20].

OCR is the process of detecting text content on images and translating that content into text that a computer can understand [22]. Humans are constantly interested in detecting characters and understanding text. A computer, on the other hand, merely sees a series of dots or pixels [21]. As a result, the photos are scanned first, and the text, graphics, and features are turned into a bitmap, which is just a matrix of black and white dots [21]. The image is then preprocessed, with the brightness and contrast of the image being increased to improve the method' accuracy [21]. Now, the image split findings aid shape the extraction process by identifying areas of interest, such as where the images or text are located. Text-filled areas can be further divided into lines, words, and characters. The OCR engine or software can then compare and match the characters using various AI techniques. The wording in the image we've been given is the result [20].

The process may not be 100 percent accurate, and some items that were not scanned correctly may require human intervention. A dictionary or even Natural Language Processing can be used to fix errors [22].

OCR is very commonly used in airports to automate the processes and extract information. Other uses include data input automation, carbon emission detection and recognition, and more [22].

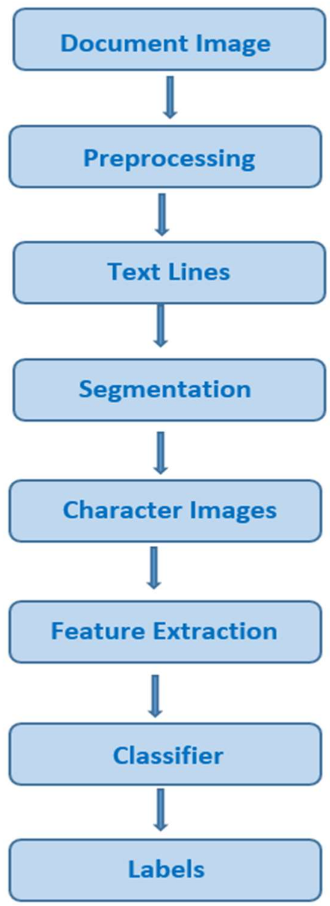


Figure 3.2: OCR Functionality

Sorting out the characters in an image and linking them to a set of templates is one method for managing OCR. Following that, the software can turn the data gathered into editable and searchable data. Automatic number plate recognition is a popular application of OCR.

3.6 TensorFlow

TensorFlow is a machine learning framework used to train the model. Tesseract is used in the license plate recognition phase and its responsibility is to convert the image to the text.

3.7 PyTesseract

Python Tesseract or PyTesseract is a wrapper for Google's Tesseract OCR engine [23]. This is a completely open source and being advanced and upheld by Google [23]. PyTesseract is able to detect and recognize the text in an image [24]. Tesseract is an open-source OCR engine used for Optical Character Recognition [14].

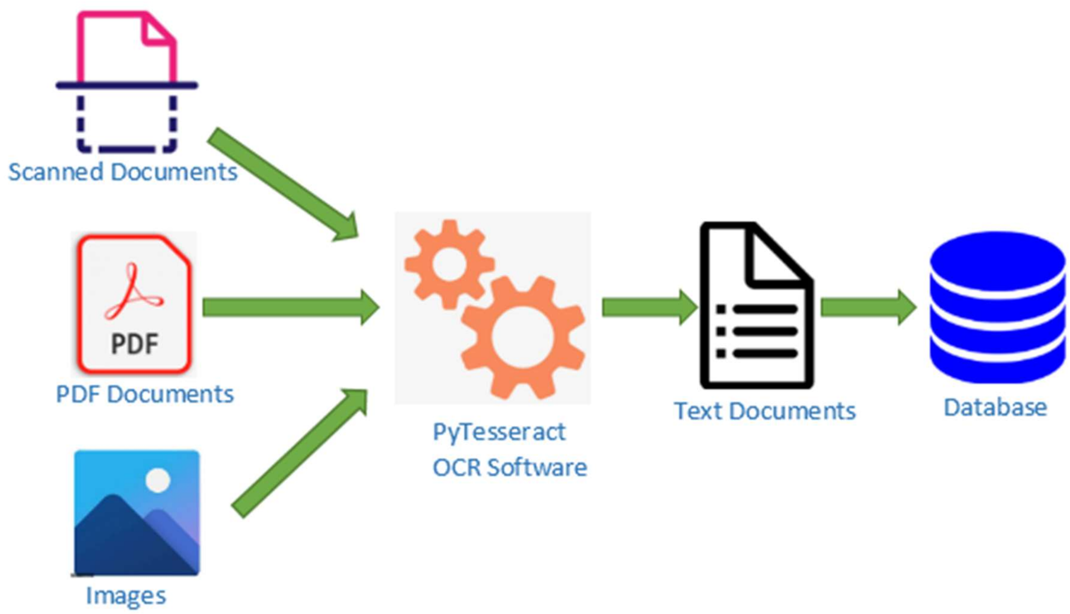


Figure 3.3: PyTesseract Functionality

3.8 makesence.ai

For labeling of the dataset an online tool has been used called makesence.ai.

3.9 Other Calibrations

The implementation is mainly executed using Python3 libraries. YOLOv5 has been used for the object detection process. YOLO stands for You Only Look Once and it is a CNN based object detector which delivers high accurate results [13]. COCO dataset has been used to trainthe YOLOv5 model [13].

High resources such as Graphical Processing Unit (GPU) integrated with additional drivers will be needed if the system is implemented locally. GPU is a silicon based microprocessor whichhas a potential of accelerate the machine learning task. At the moment, the system is implemented using Google Colaboratory. Google Colab is a Jupyter notebook stored in GoogleDrive. Colab joins the notebook to a cloud-based runtime. Therefore, Python codes can beexecuted deprived of any essential setup on the local device. Extra code cells are implemented using the same runtime, causing in a rich and collaborating coding practice which can use anyof the functionality that Python deals.

3.10 Summary

This chapter described the important technological environmental information in implementing the system. Next chapter will present the research methodology used up to the interim level of the project.

Chapter 4

Novel Approach on Boat Detection and Harbor Management

4.1 Introduction

The last chapter discussed the technologies which are adapted for the novel approach of Boat Recognition and Automated Harbor Management System. This chapter focuses on the data collection and utilizing the gathered data for training and validation purposes. After that those trained data will be used to recognize the fishing vessels in the open sea. Then after the recognition the system will retrieve information related to the detected fishing vessels.

4.2 Input

For this research, I have captured images and video footage of fishing vessels under around 75 percent of training data and around 25 percent of validating data. Then the trained images are used for the license plate recognition process.

4.3 Output

The main output of this research is the information of the detected fishing vessels. This output is used by the harbor authorities for their daily functionalities and security purposes.

4.4 Process

The process of this system consists of four major modules as registration, boat recognition, search and report module. All the fishing vessels and fishing vessel owners should be registered with the system with reference to the related port location. Respective information is stored in the database. Fishing vessel detection and recognition is executed using special methods and tools. The detected license plate numbers of the fishing vessels are matched with the stored license plate numbers in the database and retrieve requested information with reference to the given fishing vessel. In the search module, any information can be retrieved as boat name wise, boat owner wise, port wise, etc. In the boat information report module, the retrieved information is displayed as a report and circulates among authorized parties.

4.5 Users

The users of the implemented system are Sri Lanka Navy, harbor authorities, defense authorities and researchers interested in this research area.

4.6 Features

The most important module of this research is the boat detection process. Unlike regular license plates of the other vehicles, fishing vessels' license plates are placed as skewed and cannot be detected by ordinary license plate recognition algorithms. Therefore, I have researched and developed a method for detecting the slanted license plates of the fishing vessels. In this method, first the cropped license plate should go through a deskewing process and there it will deskew the image. Then the detecting algorithm can read the fishing vessel's license plate. The findings of this research will be beneficial for all the future researchers in their research studies.

4.7 Summary

This chapter discussed the novel approach used for the boat detection and harbor management system and the inputs, outputs, process, users and the unique features of the research. It also discussed how the findings of the research will be beneficial for the future researchers. Next chapter will be described on the research design of the discussed approach of the system.

Chapter 5

The Research Design for the Boat Recognition and Automated Harbor Management System

5.1 Introduction

The previous chapter discussed the novel approach used for the boat detection and harbor management system and the inputs, outputs, process, users and the unique features of the research. This chapter focuses on the research design of the system with respect to four main modules and overall design and database structure.

5.2 High level Architecture of Research Design

As discussed in the problem statement above, the harbor authorities are using a paper based, manual system for the fishing vessel management purpose. That leads to lots of inconsistencies, redundancies and so many delays on certain tasks. Therefore, an automated harbor management system will be a worthy solution.

The system consists of four main modules as follows.

1. Boat Registration Module
2. Boat Recognition Module
3. Boat Information Search Module
4. Boat Information Report Module

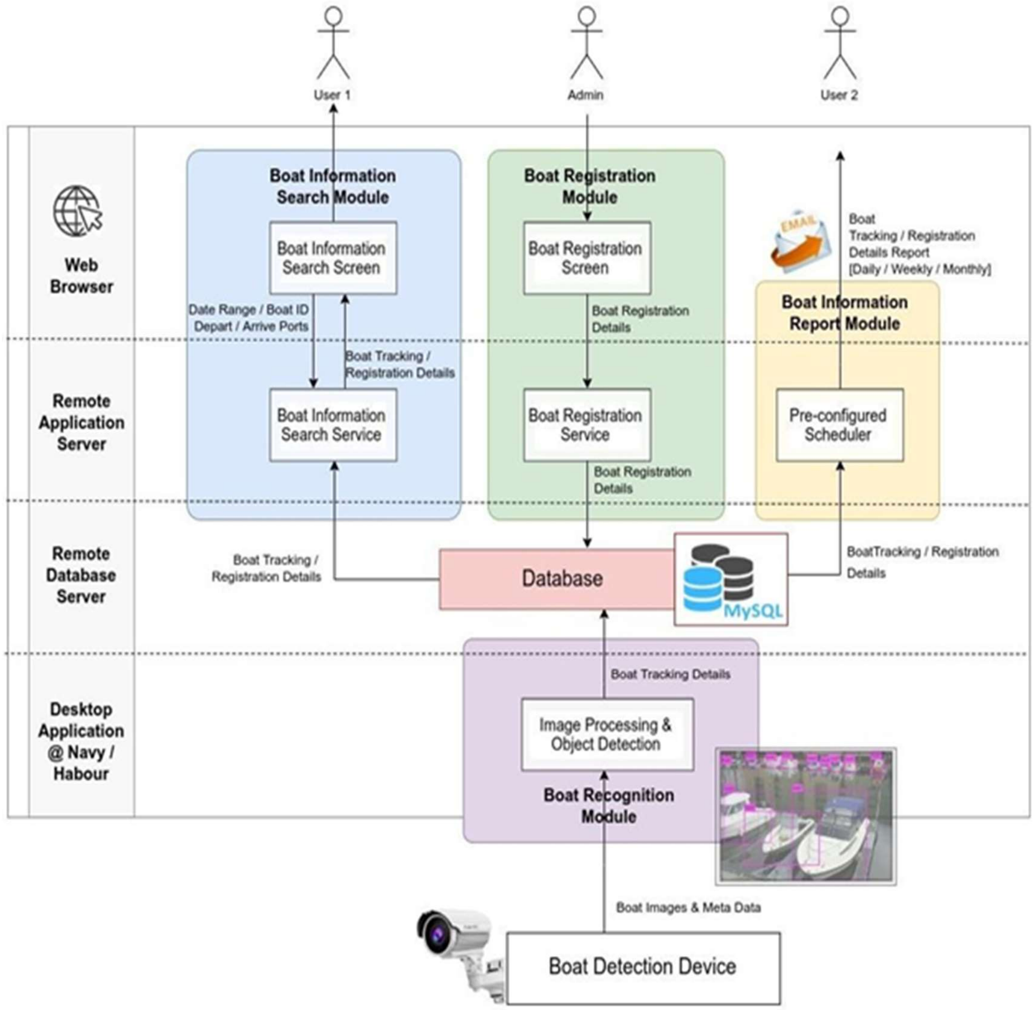


Figure 5.1: Module Diagram

5.3 Boat Registration Module

In this module, initial information relevant to the boat and the boat owner should be entered. The information entered in the registration phase are as follows.

5.4 Boat Recognition Module

Due to the COVID 19 pandemic situation, to maintain social distancing harbor authorities needed an automated fishing boat detection and classification method. Implemented system is a robust system which gives accurate and fast results. The boat detection and classification procedure has been automated into a straightforward manner.

Boat recognition process was previously designed to consist of two main approaches as recognizing boats by the shape and recognizing boats by the license plate. But as the research development continues, it is convinced that the recognition of the fishing vessel from the license plate is enough for the recognition process by using strong methods and algorithms for the detection and recognition process.

Recognizing the boat is done using the YOLO algorithm which the name implies “You Only Look Once”. All the fishing vessels have a unique vessel registration number which is placed two sides of the boat, back and the top of the cabin. Using the YOLO algorithm, this license plate will be detected and cropped then read it using character recognition.

With this information, arrival time of the boat and tracking details should be entered into the central remote database of the system.

Among many improvements of R-CNN such as Fast R-CNN, Faster R-CNN and Mask R- CNN, YOLO takes different approach.

YOLO really looks at the image at one time as the name implies “You Only Look Once”. YOLO splits the image into a lattice of 13*13 cells. Each of these cells is accountable for predicting five boundary boxes. A boundary box describes the square that encloses an object.

Each bounding box is described using five aspects [5].

1. Center of bounding box (bxby)
2. Width (bw)
3. Height (bh)
4. Class of the object (c)
5. Probability of the object (pc)

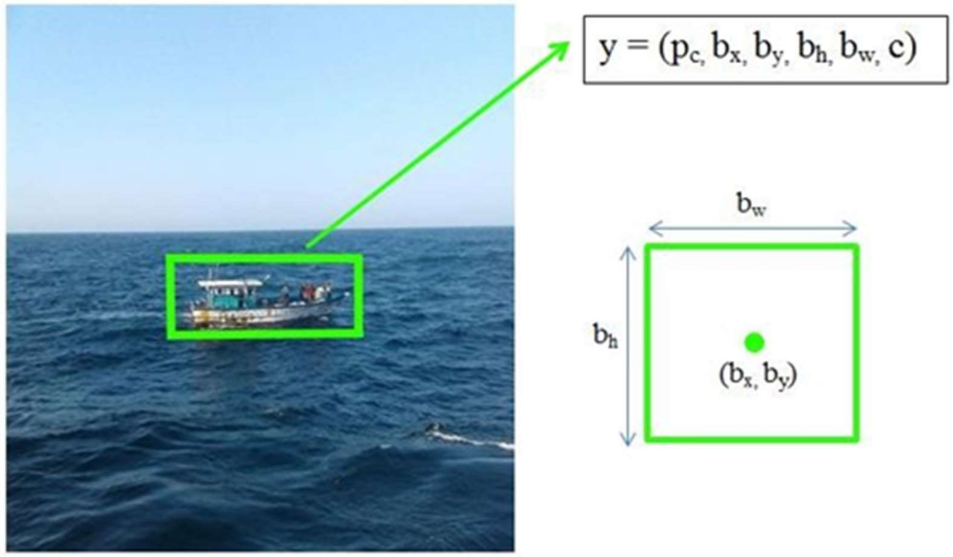


Figure 5.2: Specifying the bounding box using YOLO

YOLO will also generate a confidence score, which indicates how convinced it is that the forecasted boundary box truly encircles some object. So it's not just guessing whether or not a set of boundary boxes exists; it's also predicting what class each of the bounding boxes is in.

Higher the confident score, fatter the box that is drawn. For each bounding box, it predicts two things.

1. Whether or not bounding box exists

2. Class of that bounding box

With YOLO algorithm, the image is spilt into a grid of cells. The grid consists of 13*13 cells as mentioned above. Each of these cells is in charge of estimating the positions of five boundary boxes. The rectangle that encloses an object is known as a boundary box. The majority of these cells are empty. That’s why the Probability of the object (pc) is needed.

Then the bounding boxes with small object prospect should be detached and highest shared area should be bounded together to form one bounding box. This process is called as non-max suppression [5].

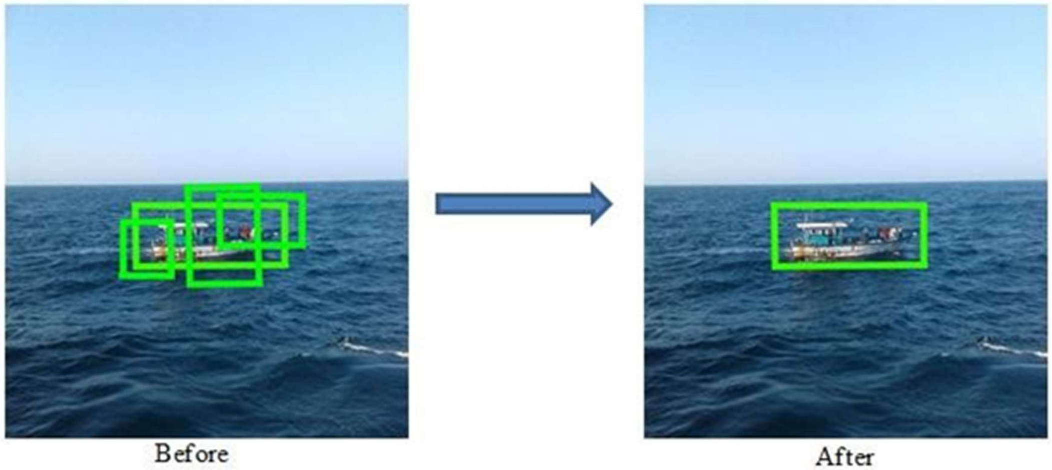


Figure 5.3 Applying Non-Max Suppression

This works similar to a classifier that provides a probability scattering over all the probable classes that the system has been trained on. The confident score and the class prediction are joined into one final mark which represents the boundary box is containing one special type of object. Based on a threshold value, it can ensure a certain object is inside the boundary box.

Recognizing from license plate has two steps.

1. Detect license plate

2. Extract the text in the license plate number

To recognize the regions of interest of license plate within an image, YOLO v5 is used. Then OpenCV and Tesseract to go in pre-process the image and extract the license plate number from the image itself. The process can be breakdown into following steps [9].

1. Detect the license plate within the image by YOLO

2. Crop the bounded area

3. Run bunch of the preprocessing techniques on the image

    · Grayscale the cropped image

    · Blow it up for easy to Tesseract to read

    · Threshold the image

    · Dilate the image and make the contours easier to find

4. Find The contours

5. Filter to get the actual letters and numbers of the license plate number

6. Segment them

7. Run more preprocessing on

8. Run through Tesseract in order to extract the individual characters

9. Append them all together

5.5 Preprocessing the images

First bounding box coordinates and the sub-image region should be extracted using YOLO. Original cropped license plate image is small and must enlarge the image up to 3x from original size. For that purpose, cv2.resize() command is used. Then transform the picture into grayscale and smooth it by applying Gaussian blur effect to the image. Then image is inverted as black background and white characters. Then apply Otsu's method which performs automatic image thresholding [8]. Then find the contours of the letters. To make contours clearer, the image should be dilated. This step is performed using OpenCV. After that all the rectangular shape contours will be filtered out and arrange them into left to right order. In this case, unwanted contours may be selected such as dashes and other irrelevant patterns in the license plate. To select only the license plate characters, must use some fixed parameters such as width and height of the contours. By that, unwanted parameters can be eliminated. Now the individual characters can be filtered out and need to sector each character image. Then invert the image again into black characters in white background. At this condition, Tesseract can perform more accurately. The generated image should blur into medium level and then it is ready to be read by Tesseract. At the end, Tesseract can read the image and extract the license plate number of the fishing vessel [9].

5.6 Boat Information Search Module

In this module, users are able to search boat tracking details and boat registration details by entering data such as Date/Time range, arrival port information, arrival time, departure time etc.

5.7 Boat Information Report Module

In this module, boat tracking information report is emailed to predefined set of e-mail addresses. This is done using a preconfigured scheduler. The report can be generated for a specific time interval such as daily, monthly, weekly or annually.

5.8 Database Structure

A centralized remote database is used to share the data with all four modules. Data updated to the remote database from Boat Registration Module and Boat Recognition Module together. Data consumed from remote database by Boat Information Search Module and Boat Information Report Module.

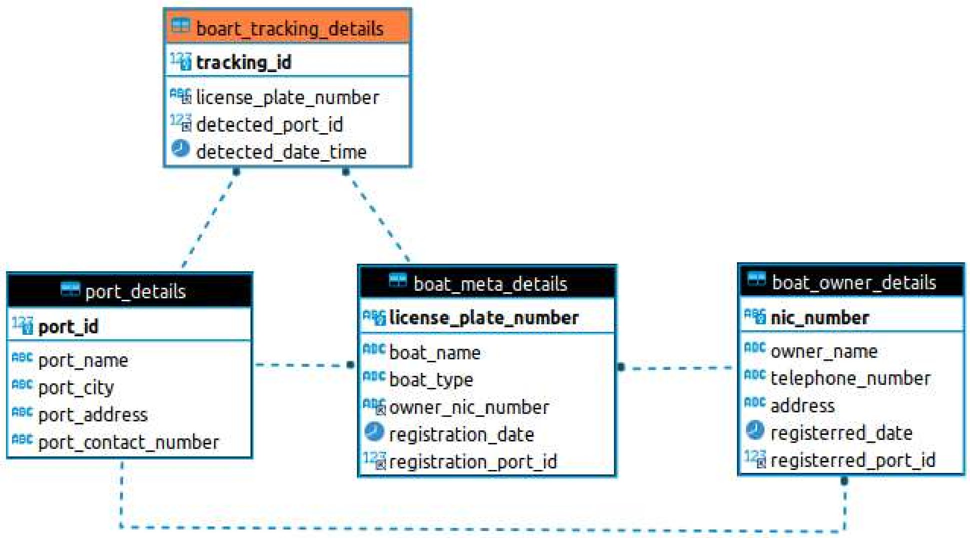


Figure 5.4 Database Structure

5.9 Overall Design Facts

Suitable detection devices (eg: camera) should be used to collect boat recognition details. Details can be entered using a desktop application. This application is installed into a computer which is located in the environment where the recognition process occurs. The information gathered from this desktop application is fed to the centralized remote database for further access.

Among these modules, Boat Information Search Module and Boat Registration Module are using a web-based application. Two user interfaces are used separately for two module screens. These user interfaces are connected to the centralized remote database.

So, a user from admin level should update boat registration information to the remote database by using boat registration screen through boat registration service. Also, a front-end user can view and update boat tracking details and boat registration details using the boat information search screen. This can be done through boat information search service.

In the meantime, a boat related information report is generated and forwarded as an e-mail to relevant users. The report is auto generated using boat tracking information and boat registration information from the above mentioned centralized remote database. This is achieved by a preconfigured background scheduler which is run in the remote application server.

Besides these modules, the user registration and login module handle the application's authorization and authentication.

Basically, following are the deliverables and outputs of the system.

1. Web based solution for managing fishing boat tracking information.

2. YOLO based desktop application to recognize boats through shape and registered number.

3. Backend reporting solution to send boat tracking information according to preschedule intervals (daily, weekly, monthly, etc.)

5.10 Preparation of Data Set

When building a computer vision model and deploying that model the system goes through a series of paces. First we collect the data, organize and label it. Then preprocess those images, augment that data, train a model, deploy and display the final results. Image preprocessing are the phases taken to transform an image in its raw format into a form that the model is ready to use for training and implication [10]. The image dataset used for this system consists of images with different dimensions, various contrast levels, oriented and wrong in dissimilar methods. Therefore, image pre-processing techniques should be used in order to correctly format all the input images. This can be considered as the data cleaning. When considering image pre-processing there's a series of different steps that can be used. The data set used for this system is a set of fishing vessel images. They have been called out and labeled before.

There are multiple types of preprocessing options available. Auto orient, static crop, resize, grayscale, auto adjust contrast, tile, modify classes and filter null are some of them [10]. Since my dataset consists of images of different sizes, the resize option needs to be used in order to convert them into the same size. Resize is changing the dimensions of the image such that the model can use it. The key thing as it relates to resize is that the resize decision used for training is the same one that is used in production. All preprocessing steps apply to the training set, validation set and testing test. So, these images need to be re-sized to be the similar dimensions, so images are stretched to be the same size. As mentioned, there are other preprocessing options as well. Auto orient option is removing the EXIF (Exchangeable Image File Format) data that is attached to the images such as image captured date/time, GPS location etc. [11]. EXIF data can control the orientation of how the model views the image relative to how they are stored on disk in memory. Grayscale in preprocessing is converting the image into black and white. Tiling is another preprocessing option which is used to break the image into smaller components if the image is too big by itself. Static crop means only taking a portion of the image. It can choose a side of an image. By auto adjusting contrast the model is able to learn the edges more easily. Modify classes can be applied to labels and only labeled data can be kept by filter null option. By applying preprocessing options to the dataset, versioned dataset of a series of images that are all formatted the same way can be obtained. This formatting applied to all the training, validation and testing image data sets for the model to learn and use.

5.11 Summary

This chapter discussed the research design of the system with respect to four main modules and overall design and database structure. It also described the preprocessing of data, the methods of preparing data and the overall functionalities of the research design. The next chapter will be focused on the detailed explanation on the implementation of each four modules of the system.

Chapter 6

Implementation

6.1 Introduction

The previous chapter discussed the high-level design architecture with the database design and component level research design. It also described the preprocessing of data, the methods of preparing data and the overall functionalities of the research design. chapter will be focused on the detailed explanation on the implementation of each four modules of the system.

6.2 Module Components

The system consists of four main modules as follows.

1. Boat Registration Module
2. Boat Recognition Module
3. Boat Information Search Module
4. Boat Information Report Module

6.3 Data Handling

6.3.1 Dataset Creation

For this research, I generated my own fishing vessel dataset by manually capturing over hundred images and video frames from the Negombo boatyard. It is recommended that users train with over 1500 images per class and over 10,000 instances per class to achieve a robust YOLOv5 model. It is also advised to include up to 10% background images in order to reduce false-positive errors. Because my dataset is so small, I have used transfer learning techniques to narrow the training process.

In order to train the custom model, labeled data is needed. Before labeling the data, first images of fishing vessels were captured from a boat yard.

Figure 6.1: Captured images of fishing vessels as input data

6.3.2 Labeling Dataset

To train models, first need to label and annotate the image dataset. Inside the training data, need to separate images and labels. Train and validation data sets must be kept inside images folder disjointedly. The images and labels names should be as it is.

Before training the dataset need to classify them with specific labels. First should insert the label name, then draw bounding boxes around the sequence and select the relevant label.

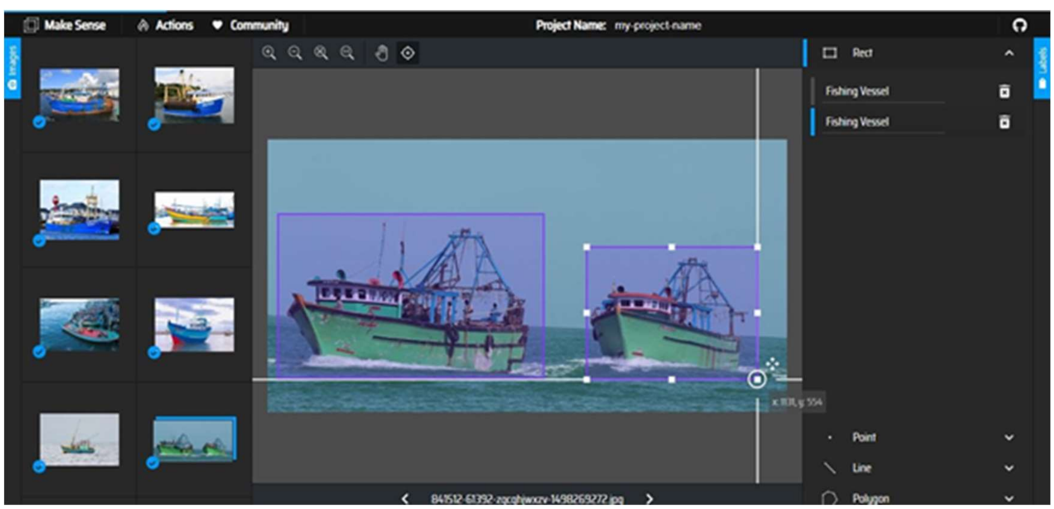


Figure 6.2: Labeling the Dataset

After finishing labeling the images, then should export those labeled images in YOLO format inside the labels folder. Validation dataset should also have gone through the same labeling procedure.

For the training purpose, Google Colab has been used. Labeled dataset then uploaded to this platform and specify the class name. Trained the dataset with three epoch times. Epoch time is the number of iterations the training dataset passed through [12].

Each labeled image consists of certain information of the label which included various numbers as follows.

0 0.533923 0.480826 0.843658 0.766962

First entry denoted here as “0” suggests the location number of the class annotated. Since “fishing vessel” is the first and only class label entry it represents the number as 0 because indexing is starts from 0. Next number “0.533923” is the center point x of the labeled object. Next to that, is the center point of y (“0.480826”) of the labeled object. Next coordination 0.843658 is the object width and the last coordinate 0.766962 is the object height. These entries always range from 0 to 1. These values are normalized such that object’s canter x value is divided by image width and object’s center y value is divided by image height. In order to follow YOLO guidelines and standards, every image needs to have a corresponding text file with the name same as the image file.

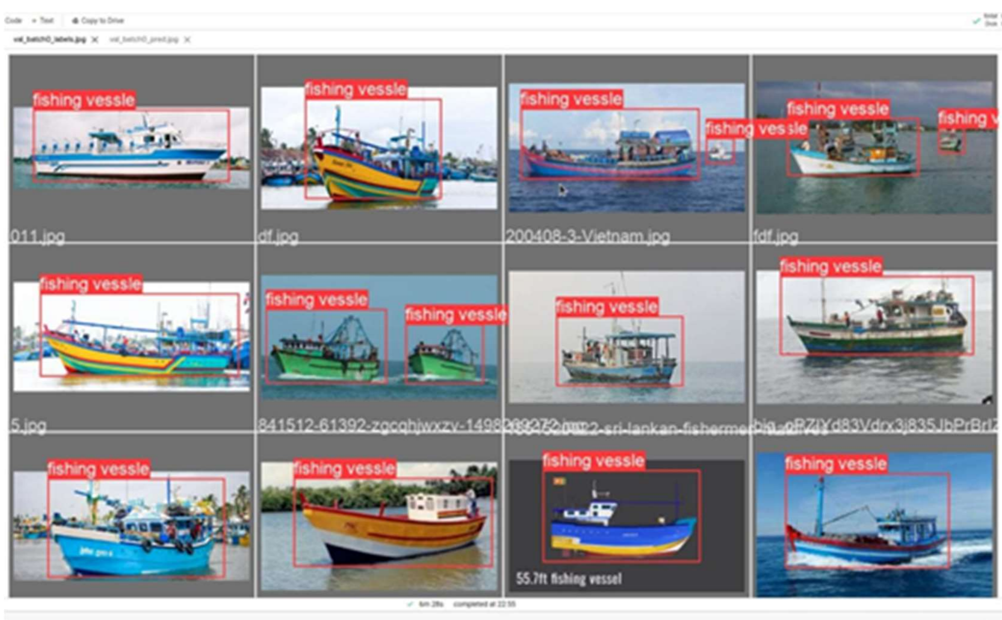


Figure 6.3: Labeled dataset

Labeled images are directly saved in YOLO format. One of the advantages in this YOLO format is we don’t need to resize any image in the dataset and we can annotate images of any size. YOLO takes care of image size when training.

After training the labeled images, it generates a predicted dataset as follows with the confidence level. Predicted bounding boxes as appeared here.

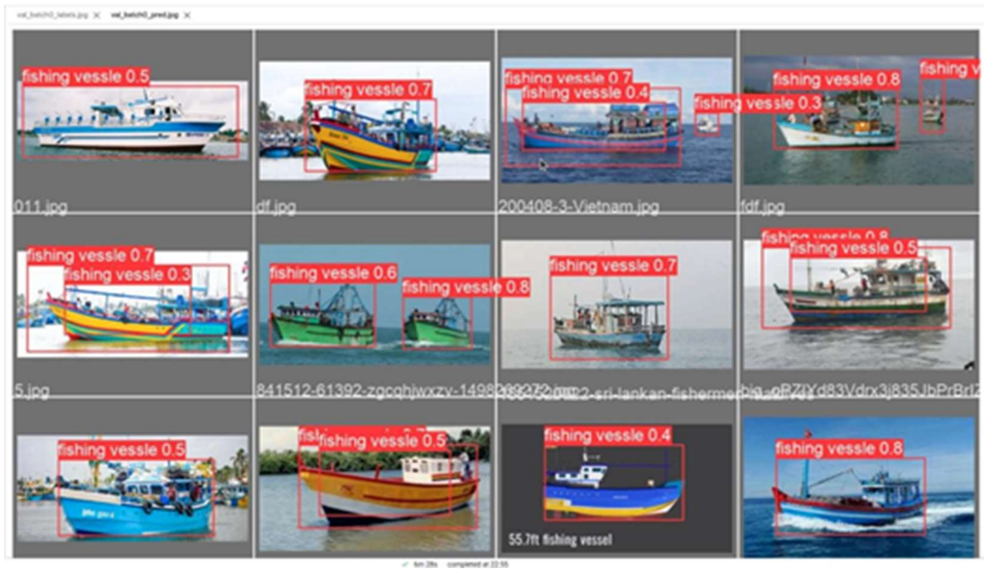


Figure 6.4: Predicted Dataset

6.4 Capturing License Plate

In the boat recognition phase, the recognition is mainly processed through identifying the license plate of the fishing vessel. The code for capturing license plate area is as follows.

```
python3 detect.py --weights last.pt --img 640 --conf 0.25 --source
data/videos/20220512_115353.mp4 --save-crop
```

Here, "detect.py" is a file which comes along with YOLOv5. It is the runnable file to access the input data. The input data can be from a real-time video stream or from a live CCTV footage. "--weights last.pt" is an argument which acts as the intelligence in the detecting process. "--img 640" is the default image size in pixels. "--conf 0.25" is the confidence level which is used to detect the labels. "--save-crop" argument used to save the cropped images of the license plates in the device. But this was done only for demonstration purposes. This argument has to be removed when deploying the system in a real environment since these saved images can consume a lot of space inside the device.

6.5 Training

For different approaches different training mechanisms can be used. First, I have downloaded the YOLOv5 and installed all the requirements. YOLOv5 is an architecture used for object detection purposes.

```
git clone https://github.com/ultralytics/yolov5  # clone
cd yolov5
pip install -r requirements.txt  # install
```

Figure 6.5: YOLOv5 installation command

The required libraries such as PyTesseract and OpenCV which needed to run YOLOv5 will be downloaded and installed in the user device automatically. To train the dataset, a GPU or Graphical Processing Unit is needed for large data sets. Since such a facility is not available with local computers, Google Colab can be used for that purpose.

Then the data is uploaded to YOLOv5 for the training process. Simplest way to upload is, compress the image files into a zip file and upload. After that coco128.yaml file should be downloaded and modified. The path of stored train and validation images, number of classes and the class names should be edited inside the file.

```
train : ../train_data/images/train/  # train images (relative to 'path') 128 images
val: ../train_data/images/val/  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
nc: 1  # number of classes
names: ['Fishing Vessel']  # class names
```

Then unzipped the train data set. The command for unzipping the train data set is as follows.

```
!unzip -q ../train_data.zip -d ../
```

Then upload the edited file after renaming it to the yolov5/data folder. By this setting up the training environment is completed and next remains the actual training process of the image dataset.

The Python command for training the dataset is as follows.

!python train.py --img 640 --batch 16 --epochs 3 --data custom_data.yaml --weights yolov5s.pt --cache

“--img 640” is the image size. When the command runs for the first time the weights are downloaded. Weights will be saved at the runs/train/exp2/weights/ folder. If the GPU is getting out of the memory, the batch size needs to be reduced. Training process takes some time and it depends on the size of the dataset. To successfully function the model, you need to increase the epoch amount. I had to use a minimum of 60 epochs to get a satisfactory accuracy. At 60 epochs I was able to get a 97% accuracy.

6.6 License Plate recognition

The most important component of the system is to identify the fishing vessel by its license plate. Recognition of the license plate is done using PyTesseract library.

6.6.1 Image to Text Reader

Input file is the fishing vessel image which was given previously for this class. Wait time is allocated 10ms for the demonstration purpose. But in the actual system wait time is not given for the sake of the efficiency of the system.

```
def imageToTextReader(image, waitTime=10, filterText=True):
    cv2.imshow('ImageToTextReader', image)
    cv2.waitKey(100)

    img2char = pytesseract.image_to_string(image)
    print('Detected text : ', img2char)

    if(filterText):
        img2char =  re.sub('[^A-Za-z0-9]+', '', img2char)
        print('Filterred text : ', img2char)

    #cv2.putText(image, img2char, (10, 30),cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255), 2)
    cv2.imshow('ImageToTextReader', putTextOnImage(image,img2char))
    cv2.waitKey(waitTime)

    return img2char
```

Figure 6.6: Image to Text Reader Code Segment

Here the retrieved text will be filtered further to remove new lines, un-recognized symbols which may lead to system malfunctions. “filterText” parameter should always keep true to ignore symbols which are not letters and numbers. [^A-Za-z0-9]. So it will remove all the unwanted characters which are not playing an important role in the recognition process.

Text will be detected by the “img2char” command. It converts the image into a string value.

img2char = pytesseract.image_to_string(image)

To display the retrieved text on the image, “putTextOnImage” method is utilized. The text will appear on top of the image. But this is utilized only for the demonstration purpose. When the system is processing in real time in order to reduce the execution time and increase the efficiency, this code segment can be omitted.

This method of converting an image to a text only works for perfectly aligned standard license plates. But if the image is slanted, skewed, blurred or handwritten, the above method cannot be successful in the license plate recognition process.

6.6.2 Deskewing the License Plate

The registration number or the license plate of the fishing vessel is not aligned horizontally when we look at the fishing vessel from the front. It is skewed a little.

Figure 6.7: Skewed License Plate of the Fishing Vessel

In order to read the license plate, first need to deskew the image horizontally. Because when the text is in the angle shape, the text cannot be detected. This is the most challenging component of the research.

Deskewing of the license plate comes with three approaches for different scenarios.

1. Auto Deskewing
2. Manual Deskewing

3. Auto and Manual Deskewing

6.6.2.1 Auto Deskewing

This is the first approach I have tried for the deskewing process of the license plate. In this approach first a minimum area of the rectangle is drawn around the region of interest. The bounding rectangle is drawn around the minimum area. In my research the minimum area rectangle is drawn around the skewed license plate. Then the skewed rectangle should bring on to the normal bounding rectangle as follows.

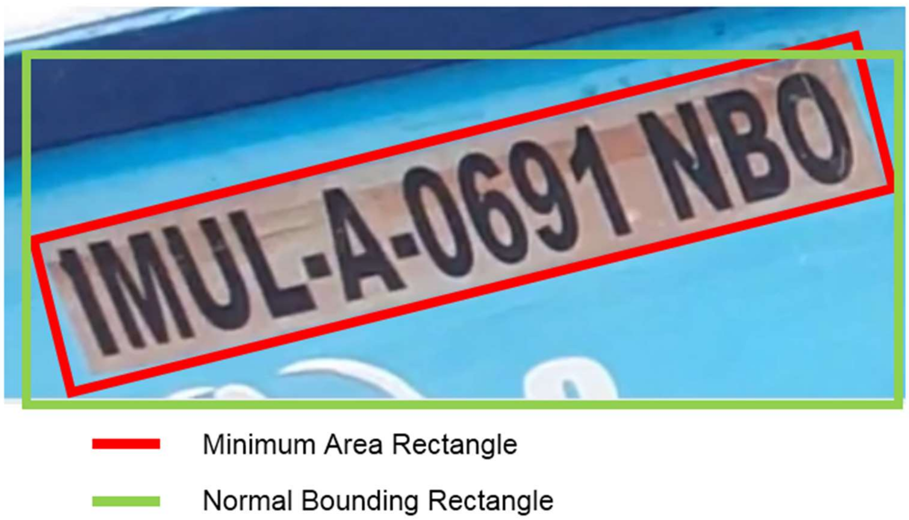


Figure 6.8: Minimum Area Rectangle and Normal Bounding Rectangle

This process is executed using OpenCV. Minimum area rotated rectangle can be detected using the following function of OpenCV.

cv2.minAreaRect()

It will return the following information of the minimum area rotated rectangle.

(center(x, y), (width, height), angle of rotation)

To find the four points of the rectangle it uses cv2.boxPoints() function and return the four corner points. Based on that, it will easily draw the rectangle around the skewed license plate as displayed above. The four points are numbered clockwise from 0 to 3. The rotating angle is derived from cv2.minAreaRect() function and it results the angle between the horizontal line and the skewed line which joins the starting and end point of the rotated rectangle.

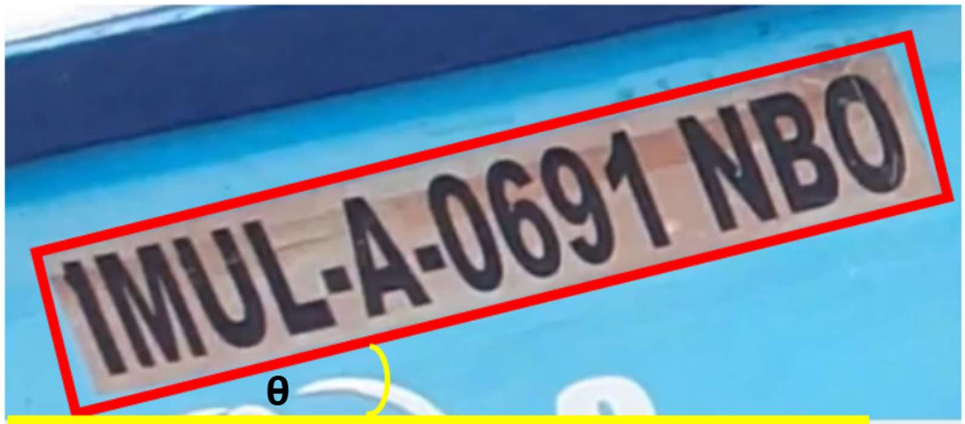


Figure 6.9: Angle of Rotation

By this function it finds out the angle of rotation and deskew the slanted license plate into the horizontal line. Now the PyTessarcat is able to convert the image into a text. The downside of the auto deskew method is that, if the cropped license plate is not clear enough or cropped license plate area is consistent with other background images such as part of the boat name, the function confuses which object is to be rotated. Therefore, it will not result the accurate angle to be rotated horizontally. To use the auto deskew, the area of the cropped license plate should be clear. But that is not always practical with Sri Lankan fishing vessels. Therefore, in this research I have used another advanced method for the deskew process called manual deskew.

6.6.2.2 Manual Deskewing

Cropped image of the license plate should be deskewed in order to accurately detect the text. As mentioned above, in some cases for the system, it is difficult to find out the angle of rotation of the skewed license plate if the cropped area is consistent with other details like part of the boat body, extra paintings etc. So, in this method, the image will be rotated in both directions one degree by degree up to 20 degrees until the system is capable of correctly spot the text of the registration plate. In each of the rotation it will go through the text detection procedure. Once the text is detected, it will be checked for successful match against the registered license plates that is retrieved from the database (boat_meta_data). These registered boat license plates were added to the data base at the boat registration module.

So, once the correct match found it will stop rotating further. The recognized license plate text will be saved in the boat_tracking_info with the time and port details. Then it will continue to next frame of the input source to detect further license plates and recognize its contents.

```
def manualDeskew(image,waitTime=200, maxAngle=20):
    height, width = image.shape[:2]
    center_img = (width / 2, height / 2)

    for ang in range(0, maxAngle):

        plateIdentified = rotateImageAndIdentifyPlate(image,-ang,center_img,width,height,waitTime,'Clockwise')
        if(plateIdentified):
            print("!!!! BREAKING THE ROTATION LOOP SINCE PLATE IS RECOGNIZED !!!!!")
            break

        rotateImageAndIdentifyPlate(image,ang,center_img,width,height,waitTime,'Anti - Clockwise')
        if(plateIdentified):
            print("!!!! BREAKING THE ROTATION LOOP SINCE PLATE IS RECOGNIZED !!!!!")
            break
```

Figure 6.10: Manual Deskewing code segment

Image is rotated by rotateImageAndIdentifyPlate() method and the maximum angle for rotation is given as 20 degrees at maxAngle parameter. Once the license plate is identified the rotation stops.

Although this process produces high detection ratio and high accurate results, it consumes more time because the image needs to be examined in each and every degree value until it aligns horizontally.

6.6.2.3 Auto and Manual Deskewing

This method is a hybrid version of both auto deskewing and manual deskewing. Here, If the license plate is not recognized by Auto Deskew, then proceed with Manual Deskew. This is a balanced method when compared to previous execution modes since it provides accurate results and speedy results.

6.7 Preprocessing

At the preprocessing stage of the input images first the brightness and contrast levels are controlled. It removes the background. Here everything will be changed to white where pixel is not black. The code segment is as follows.

```
img[img > 40] = 255 # change everything to white where pixel is not black
cv2.imwrite('my_img2.jpeg', img)
cv2.imshow('non black ', img)
```

Figure 6.11: Preprocessing - Removing background

Then the cropped license plate is converted to grayscale.

```
gray_image = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
cv2.imshow('gray_image ', gray_image)

gray_image[gray_image >  72] = 255
cv2.imwrite('my_img2.jpeg', gray_image)
```

Figure 6.12: Preprocessing - Grayscale

Then the grayscale image is inverted. So, it converts into a negative image using bitwise not. I have implemented this preprocessing step because generally detection is successful if the background is black and the text is white. The code segment for bitwise not image is as follows.

```
gray_image = cv2.bitwise_not(gray_image)
cv2.imshow('Gray Image 1', gray_image)
```

Figure 6.13: Preprocessing - Bitwise not

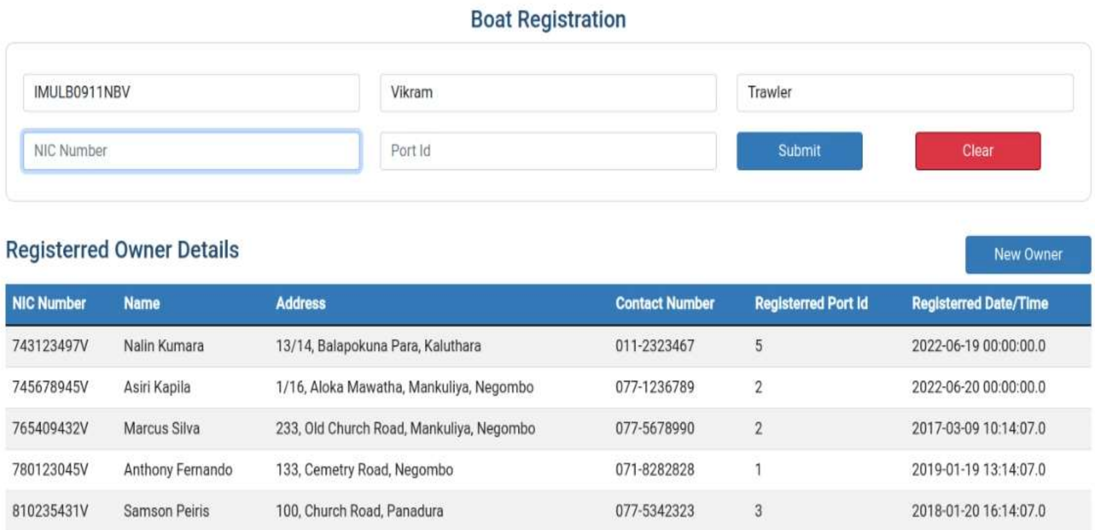
At this point it will detect the angle of the cropped license plate and then rotate the image horizontally as discussed above.

6.8 Boat Registration Module

Boat details and boat owner details are inserted to the database from the following User Interface.

Figure 6.14: Boat Registration UI

License plate number, Boat name, Boat type, NIC number of the boat owner and the Port ID should insert from this UI. When trying to enter the NIC number of the boat owner, it will automatically display all the owners already registered with the system as follows.



NIC Number	Name	Address	Contact Number	Registerred Port Id	Registerred Date/Time
743123497V	Nalin Kumara	13/14, Balapokuna Para, Kaluthara	011-2323467	5	2022-06-19 00:00:00.0
745678945V	Asiri Kapila	1/16, Aloka Mawatha, Mankuliya, Negombo	077-1236789	2	2022-06-20 00:00:00.0
765409432V	Marcus Silva	233, Old Church Road, Mankuliya, Negombo	077-5678990	2	2017-03-09 10:14:07.0
780123045V	Anthony Fernando	133, Cemetry Road, Negombo	071-8282828	1	2019-01-19 13:14:07.0
810235431V	Samson Peiris	100, Church Road, Panadura	077-5342323	3	2018-01-20 16:14:07.0

Figure 6.15: Registered Owner Details

If the respective boat owner is a not in the existing list, then with the New Owner function, we can add a new owner as follows.

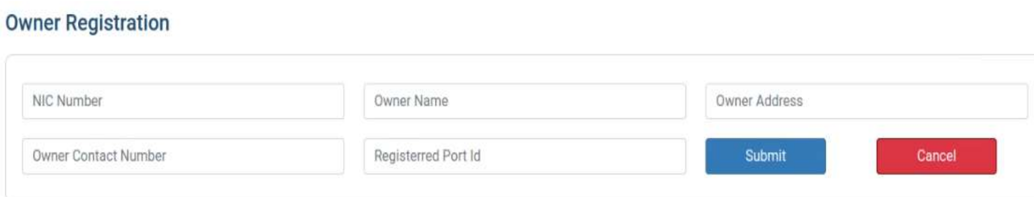


Figure 6.16: New Owner Registration

New owners should submit following details.

- NIC Number
- Owner Name
- Owner Address
- Owner Contact Number
- Registered Port ID

When trying to add the Port ID details, it will automatically display all the registered ports as follows and the user must select the relevant port from the appeared list.

Registerred Port Details

New Port

Port ID	Port Name	Port City	Address	Port Contact Number
1	Negombo South	Negombo	20A, Negombo South, Negmbo	031-2445454
2	Mankuliya	Negombo	100, Church Road, Mankuliya, Negombo	031-2655111
3	Panaduraya North	Panadura	35, Beach Road, Panadura	011-2323232
4	Panaduraya South	Panadura	111, Kudapaduwa, Panadura	011-4321111
5	Kaluthara North	Kaluthara	101, Beach Road, Kaluthara South, Kaluthara	011-0222033
6	Batticaloa North	Batticaloa	20A, St.Mary Road, Batticaloa North, Batticaloa	065-7898745

Figure 6.17: Registered Port Details

If the necessary port is not available with the give list, then the user must add the port details from the New Port function as follows.

Figure 6.18: New Port Registration

6.9 Boat Information Search Module

The information regarding the departed fishing vessels can be retrieved using this module. Information can be retrieved using following labels.

- Boat name
- Boat type
- License plate number
- Port name
- Port city
- Owner name
- Date period

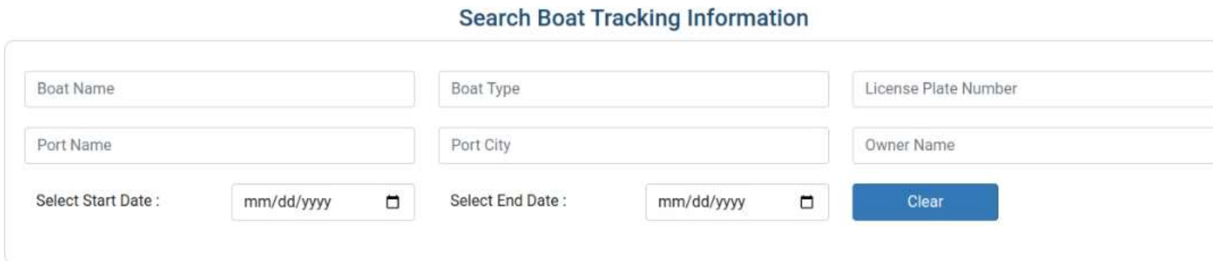
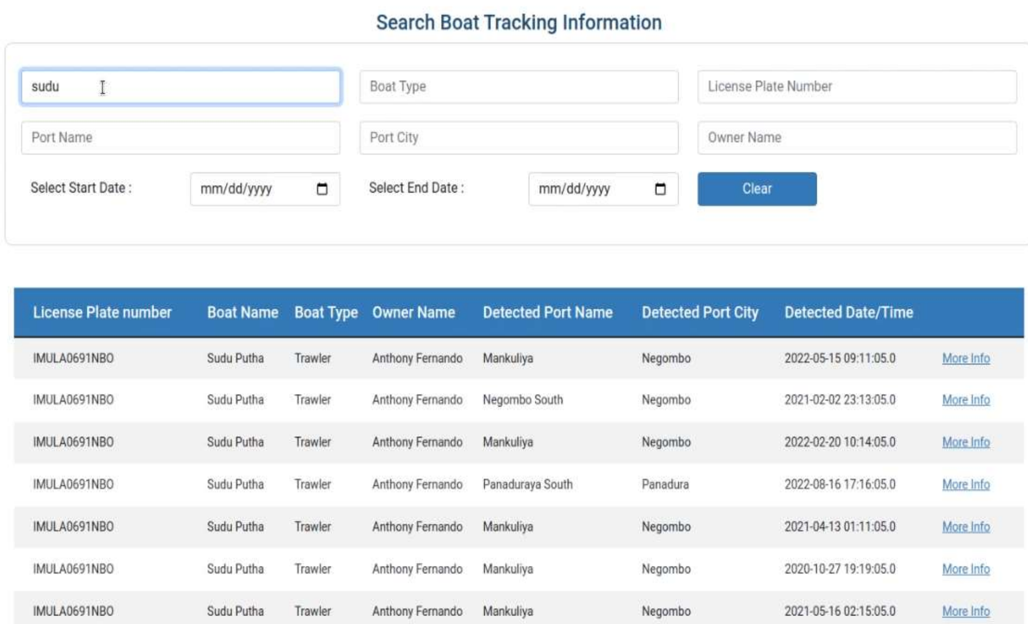


Figure 6.19: Search Boat Tracking Information UI

Search results will be displayed under the information search table when the user start inserting the relevant label as follows.



License Plate number	Boat Name	Boat Type	Owner Name	Detected Port Name	Detected Port City	Detected Date/Time	
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2022-05-15 09:11:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Negombo South	Negombo	2021-02-02 23:13:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2022-02-20 10:14:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Panaduraya South	Panadura	2022-08-16 17:16:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2021-04-13 01:11:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2020-10-27 19:19:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2021-05-16 02:15:05.0	More Info

Figure 6.20: Search Results

Information can be filtered out according to the preference of the user. If the user wants for more details of a selected row, he can click on the More Info link and get the descriptive details of the fishing vessels as follows.

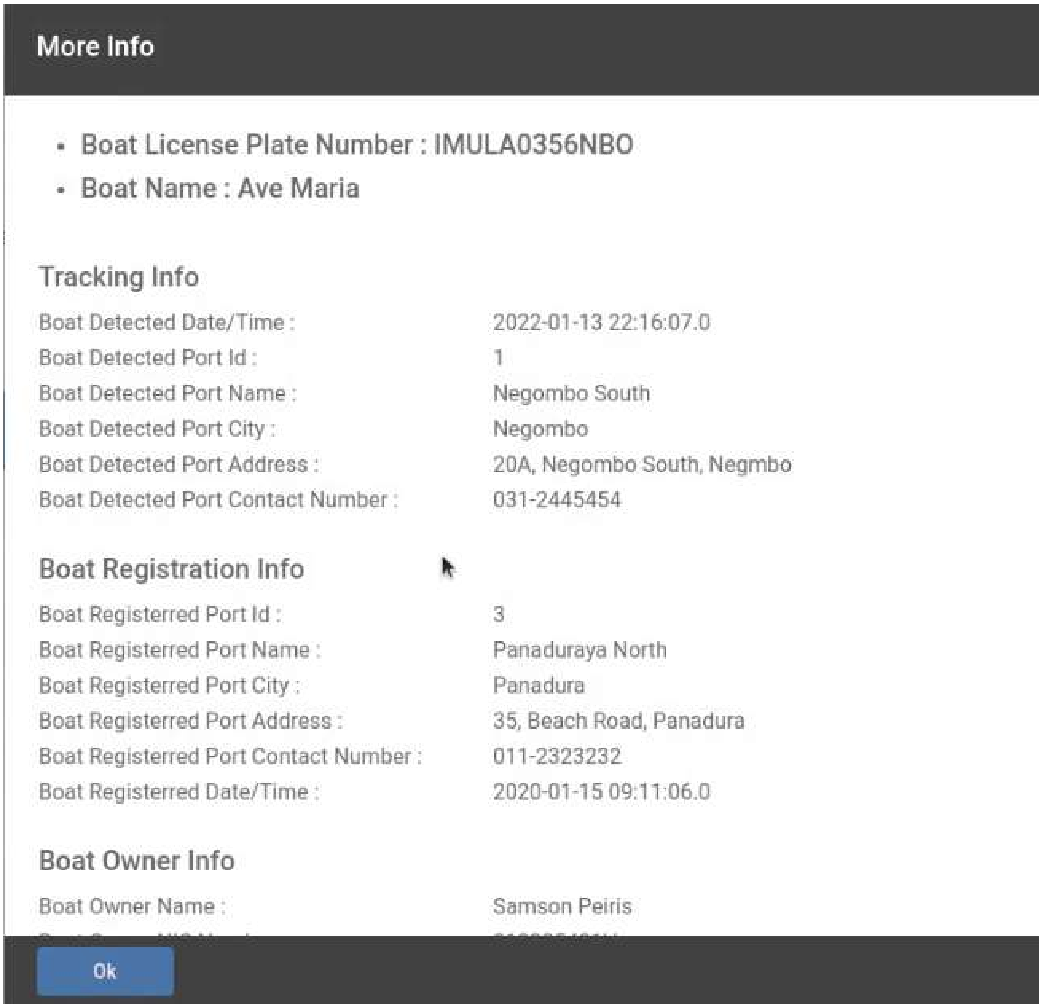


Figure 6.21: Descriptive details of the fishing vessel

6.10 Boat Information Report Module

In this module, boat tracking information report is e-mailed to predefined set of e-mail addresses. This is done using a preconfigured scheduler. The report can be generated for a specific time interval such as daily, monthly, weekly or annually.

6.11 Summary

This chapter discussed the detail collaboration of step-by-step implementation as module components and the high-level architecture. Source codes and screenshots are provided for the further clarifications of the implementation. Next chapter focuses on the evaluation process of the resulted outcomes of the system.

Chapter 7

Evaluation

7.1 Introduction

Earlier chapter discussed the implementation of the system with its four main components separately. Also discussed the expected outcomes and challenges faced during the implementation. This chapter focuses on the implemented system's results and the system's evaluation process.

7.2 Measuring the Accuracy of the Model

With better effectiveness, better performance may be automatically generated. To measure the performance of the system, a Confusion matrix can be used. The confusion matrix is a simple methodology to represent how many expected categories or classes were productively anticipated vs how many were incorrectly guessed [29]. It is used to determine the number of classes that were precisely forecasted as their real class using the outputs of a predictive model with a class outcome. There are basically four combinations of predicted and actual values in the confusion matrix as follows [29].

- True Positive (TP)
- False Positive (FP)
- False Negative (FN)
- True Negative (TN)

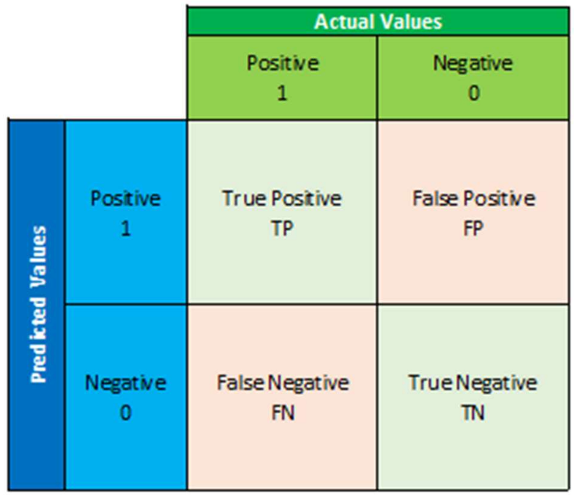


Figure 7.1: Confusion Matrix

Predictions are either positive or negative. Actual classes are positive or negative. True Positive is when predicted positive and the outcome is true. True Negative is when predicted negative and the outcome is true. False Positive is when predicted positive and the outcome is false. Lastly, False Negative is when predicted negative and the outcome also negative. TP and TN classes are correctly classified samples. But FP and FN classes are generating errors. In FP class it is predicted as positive but actually false. Also, in FN it is predicted as negative but actually it is false. So, FP classes result Type 1 error and FN classes result Type 2 error.

Confusion matrix can be used to display how many samples were predicted accurately from each label [29]. It will assess the performance of the classifier. To assess the performance of the model there are a lot of Key Performance Indicators (KPI).

7.2.1 Classification Accuracy

Classification Accuracy is how many values are predicted as correct in the model. When considering all the samples, it will result the accurate predictions.

Classification Accuracy = (TP + TN) / (TP + TN + FP + FN)

7.2.2 Error Rate

Error Rate is the misclassification rate compared to all the samples given and is calculated as follows. Simply it will output how many files we messed up.

Error Rate = (FP + FN) / (TP + TN + FP + FN)

7.2.3 Precision

Precision is the ratio of True Positive samples and total true predictions. It will output how many selected items are relevant in the model. Precision should be as high as possible.

Precision = TP / (TP + FP)

7.2.4 Recall

Recall is the ratio of True Positive samples and total actual true values. It will output how many relevant items are selected in the model. Recall also should be as high as possible.

Recall = TP / (TP + FN)

Following is the confusion matrix which generated for this system.

Figure 7.2: Resulted Confusion Matrix

Following are the some of the other results generated while the training process.

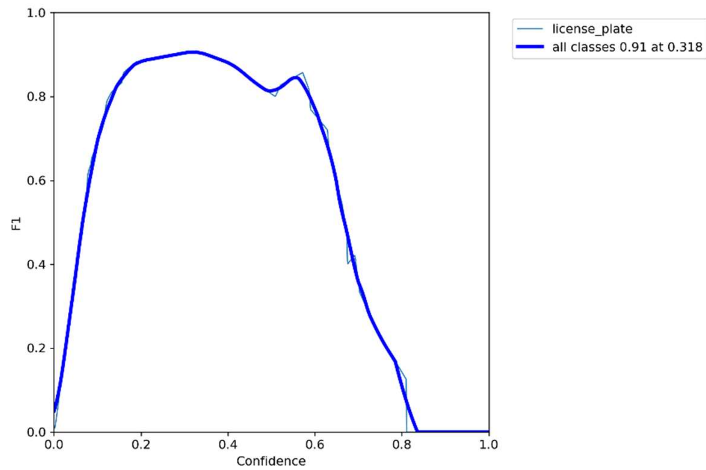


Figure 7.3: F1 Curve

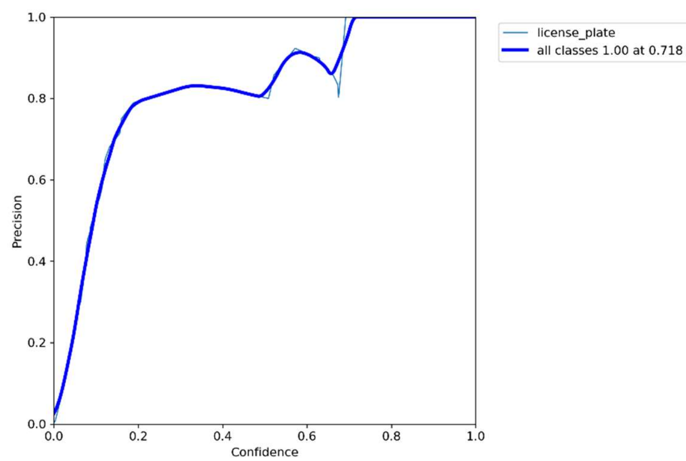


Figure 7.4: P curve

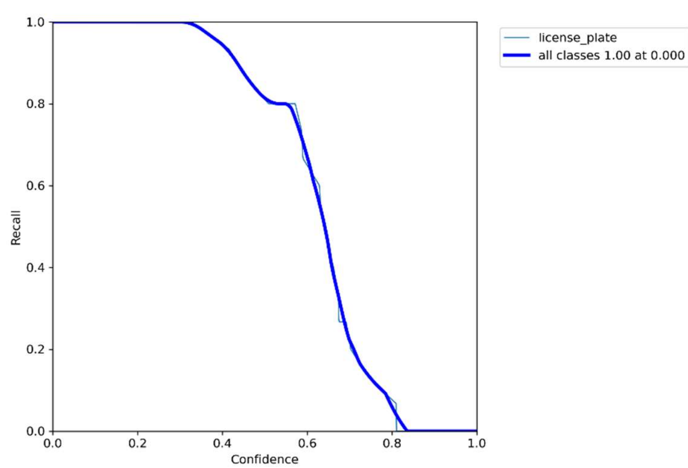


Figure 7.5: R Curve

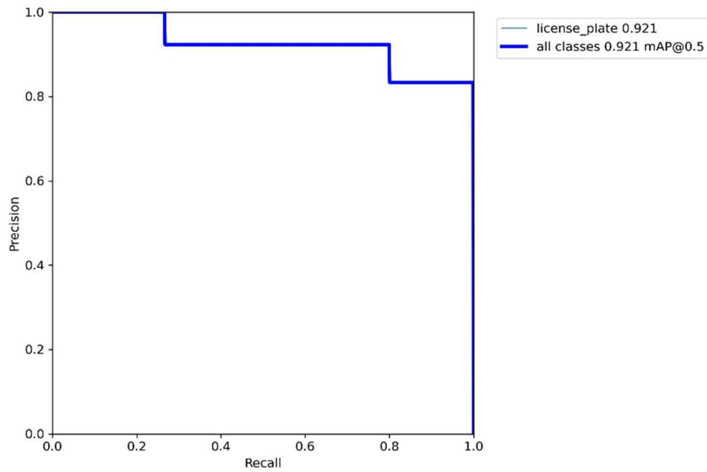


Figure 7.6: PR Curve

The system is able to reach 95% accuracy level with reference to the log file of training data.

```
     Epoch   gpu_mem       box       obj       cls    labels
img_size
      0/99     3.73G    0.1202   0.03141         0         7
640: 100% 3/3 [00:02<00:00,  1.24it/s]
               Class     Images     Labels          P          R
mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  3.17it/s]
                 all         11         15   0.000606      0.133
0.000364   9.16e-05

     Epoch   gpu_mem       box       obj       cls    labels
img_size
      1/99     3.73G    0.1209   0.03224         0        11
640: 100% 3/3 [00:00<00:00,  4.65it/s]
               Class     Images     Labels          P          R
mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  4.94it/s]
                 all         11         15   0.000303     0.0667
0.000203   2.03e-05

     Epoch   gpu_mem       box       obj       cls    labels
img_size
      2/99     3.73G    0.1157   0.03267         0         9
640: 100% 3/3 [00:00<00:00,  4.63it/s]
               Class     Images     Labels          P          R
mAP@.5 mAP@.5:.95: 100% 1/1 [00:00<00:00,  5.42it/s]
                 all         11         15   0.000606      0.133
0.000394   0.000112
```

Figure 7.7: Accuracy Level as per Log File

7.3 Summary

This chapter focused on the evaluation procedure of the implementation outcomes of the system. Next chapter is focused on the conclusion and the future works of the research area of study.

Chapter 8

Conclusion and Future Work

8.1 Introduction

This chapter focuses on the summery of the overall research study and how I was able to provide the solutions for the problem state. Also, this chapter discusses the limitations and future works of the research study.

8.2 Overview of the Research

Harbor authorities are currently using a manual, paper-based system for recording the information on fishing vessel management like boat departures, arrivals and other tasks. That leads to lots of inconsistencies, redundancies and so many delays on certain tasks. Therefore, an automated harbor management system will be a worthy solution. Boat Recognition and Harbor Management System (BRAHMS) is consists of four main modules as Boat Registration Module, Boat Recognition Module, Boat Information Search Module and Boat Information Report Module.

Boat recognition process executed by license plate recognition techniques. Here The input image is splits into a grid of cells and cells are responsible for predicting bounding boxes which encloses an object. System is trained with boat images and, once a boat appeared it will detect by the shape automatically. Each bounding box is described using five aspects. In the Boat Registration Module, initial information relevant to the boat should be entered. YOLOv5 is also going to yield a confident score that tells as how certain it is, that the predicting boundary box is encloses some object. Then the bounding boxes with low object probability should be detached and highest shared area should be bounded together to form one bounding box. The confident score and the class prediction are combined into one final score which represents the boundary box is containing one special type of object.

All the fishing vessels have a unique vessel registration number which is placed two sides of the boat, back and the top of the cabin. Using YOLO algorithm, this license plate will be detected and cropped then read it using character recognition. Recognizing from license plate has two steps. To recognize the regions of interest of license plate within an image, YOLO v5 is used. Then OpenCV and Tesseract to go in pre-process the image and extract the license plate number from the image itself.

Next module is Boat Information Search Module and it's a web-based application. In this module, user should be able to search boat tracking details and boat registration details by entering data such as Date/Time range, arrival port information, arrival time, departure time etc.

In the boat information report module, boat tracking information report is e-mailed to predefined set of e-mail addresses. This is done using a preconfigured scheduler. The report can be generated for a specific time interval such as daily, monthly, weekly or annually.

In this system, I have implemented a novel deskewing method for the slanted license plate recognition process.

8.3 Limitations

The quality of the input images are directly depend on the quality of the capturing device. For the current system I have used a 4K resolution camera. But for high accuracy, a capturing device with high FPS could be used for the research. Also, satisfactory data set was not found therefore has to manually capture the input images and implement the system.

With reference to the lack of GPU and the hardware limitation of the device, there can be limitations of thread pool execution.

8.4 Future Work

In the future, the research paradigm is planned to widen by using many other boat types and ships in all the sizes for the research rather than deploying the system on fishing vessels. Deep learning techniques are planned to use for that purpose as well. Preprocessing techniques also will be enhanced with more steps. So, more accurate and efficient system will be presented in the future.

In this system distorted license plates are not able to recognize. There can be damaged license plates and plates with un-formalized hand writings. So in the future, I'm planning to enhance the system to read those types of license plates using deep learning algorithms too.

Since I could not get fishing vessel images which are located in far way in the open sea, I could not demonstrate the license plate recognition for such boats. I'm planning to generate that recognition procedure with two approaches. Since the boat is located in far, direct license plate recognition is difficult. So first need to detect the boat by its shape, then zoom it and crop the license plate. Then can proceed with the normal recognition process.

8.5 Summary

This chapter described the overall conclusion of the research study, the limitations encountered during the research process and the expected future implementations for the research.

Reference

[1] Zhijian Huang, Bowen Sui, Jiayi Wen and Guohe Jiang, "An Intelligent Ship Image/Video Detection and Classification Method with Improved Regressive Deep Convolutional Neural Network", Hindawi Complexity, Volume 2020, Article ID 1520872, 2020.

[2] Okan Atalar, Burak Bartan, "Ship Classification using Image Dataset".

[3] Gang Tang, Yichao Zhuge, Christophe Claramunt, Shaoyang Men, "N-YOLO: A SAR Ship Detection Using Noise-Classifying and Complete-Target Extraction", Remote Sens, 2021

[4] The National Aquatic Resources Research and Development Agency (NARA), "FisheriesIndustry Outlook", 2018

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once:
Unified, Real-Time Object Detection", 2016

[6] Budi Setiyono, "Number plate recognition on vehicle using YOLO - Darknet", et al 2021 J. Phys.: Conf. Ser. 1821 012049, 2021

[7] S. M. Chittajallu, N. Lakshmi Deepthi Mandalaneni, D. Parasa and S. Bano, "Classification of Binary Fracture Using CNN," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-5, doi: 10.1109/GCAT47503.2019.8978468.

[8] Liu, D. and Yu, J., 2009, August. Otsu method and K-means. In 2009 Ninth International Conference on Hybrid Intelligent Systems (Vol. 1, pp. 344-349). IEEE.

[9] Sajjad, K.M., 2010. Automatic license plate recognition using python and opencv. Department of Computer Science and Engineering MES College of Engineering.

[10] Reddy, B.R., Suresh, A., Mani, M.R. and Kumar, V.V., 2009. Classification of textures based on features extracted from preprocessing images on random windows. International journal of advanced Science and technology, 9.

[11] "Exif - Wikipedia", En.wikipedia.org, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Exif. [Accessed: 06- Dec- 2021].

[12] "Epoch in Machine Learning: A Simple Introduction (2021)", Jigsaw Academy, 2021. [Online]. Available: https://www.jigsawacademy.com/blogs/ai-ml/epoch-in-machine-learning.[Accessed: 06- Dec- 2021].

[13] Huang, R., Pedoeem, J. and Chen, C., 2018, December. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.

[14] Smith, R., 2007, September. An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 629-633). IEEE.

[15] Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Gonçalves, G.R., Schwartz, W.R. and Menotti, D., 2018, July. A robust real-time automatic license plate recognition based on the YOLO detector. In 2018 international joint conference on neural networks (ijcnn) (pp. 1-10). IEEE.

[16] Ting, L., Baijun, Z., Yongsheng, Z. and Shun, Y., 2021, July. Ship detection algorithm based on improved YOLO V5. In *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)* (pp. 483-487). IEEE.

[17] Tao, J., Wang, H., Zhang, X., Li, X. and Yang, H., 2017, October. An object detection system based on YOLO in traffic scene. In *2017 6th International Conference on Computer Science and Network Technology (ICCSNT)* (pp. 315-319). IEEE.

[18] Culjak, I., Abram, D., Pribanic, T., Dzapo, H. and Cifrek, M., 2012, May. A brief introduction to OpenCV. In *2012 proceedings of the 35th international convention MIPRO* (pp. 1725-1730). IEEE.

[19] Xie, G. and Lu, W., 2013. Image edge detection based on opencv. *International Journal of Electronics and Electrical Engineering*, *1*(2), pp.104-106.

[20] Mithe, R., Indalkar, S. and Divekar, N., 2013. Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, *2*(1), pp.72-75.

[21] Chaudhuri, A., Mandaviya, K., Badelia, P. and Ghosh, S.K., 2017. Optical character recognition systems. In *Optical Character Recognition Systems for Different Languages with Soft Computing* (pp. 9-41). Springer, Cham.

[22] Charles, P.K., Harish, V., Swathi, M. and Deepthi, C.H., 2012. A review on the various techniques used for optical character recognition. *International Journal of Engineering Research and Applications*, *2*(1), pp.659-662.

[23] Saoji, S., Arora, A., Singh, R., Mangal, A. and Eqbal, A., TEXT RECOGNITION AND DETECTION FROM IMAGES USING PYTESSERACT.

[24] Chadha, A., Kashyap, S., Gupta, M. and Kumar, V., 2020. License Plate Recognition System using OpenCV & PyTesseract. *CSI Journal of*, p.31.

[25]] Xu, M., Li, C., Zhang, S. and Le Callet, P., 2020. State-of-the-art in 360 video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, *14*(1), pp.5-26.

[26] Wiley, V. and Lucas, T., 2018. Computer vision and image processing: a paper review. *International Journal of Artificial Intelligence Research*, *2*(1), pp.29-36.

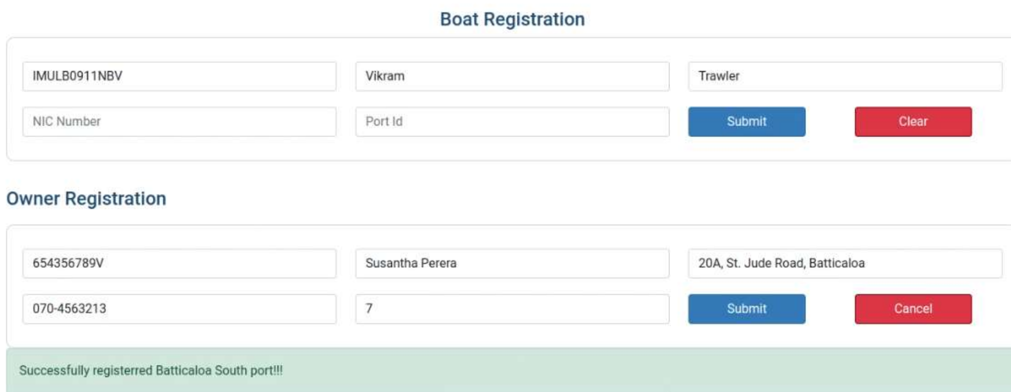
[27] An, G.H., Lee, S., Seo, M.W., Yun, K., Cheong, W.S. and Kang, S.J., 2018. Charuco board-based omnidirectional camera calibration method. *Electronics*, *7*(12), p.421.

[28] Zhu, T., Lou, Y., Zhou, Y., Zhang, J., Huang, J., Li, Y., Luo, H., Wen, S., Zhu, S., Gong, Q. and Qiu, M., 2019. Generalized spatial differentiation from the spin hall effect of light and its application in image processing of edge detection. *Physical Review Applied*, *11*(3), p.034043.

[29] Visa, S., Ramsay, B., Ralescu, A.L. and Van Der Knaap, E., 2011. Confusion matrix-based feature selection. *MAICS*, *710*(1), pp.120-127.

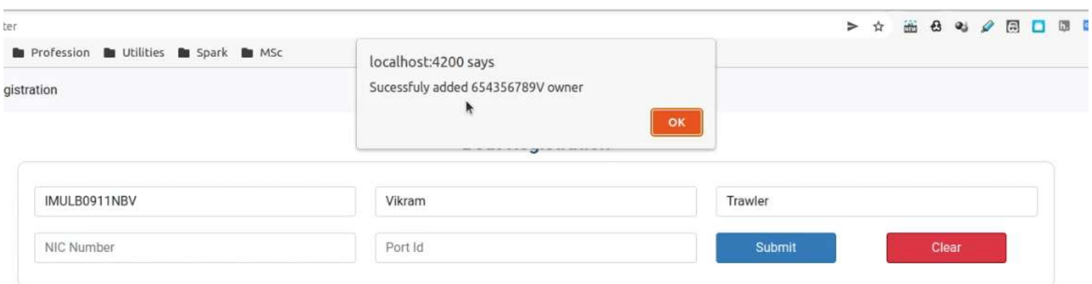
Appendix – A

Boat Registration UI



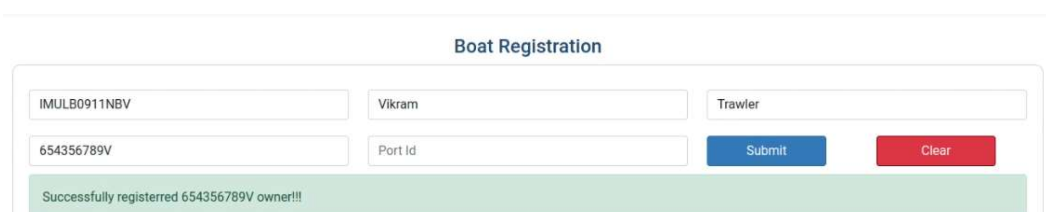
Appendix A: 1 Boat Registration UI

Boat Owner Registration



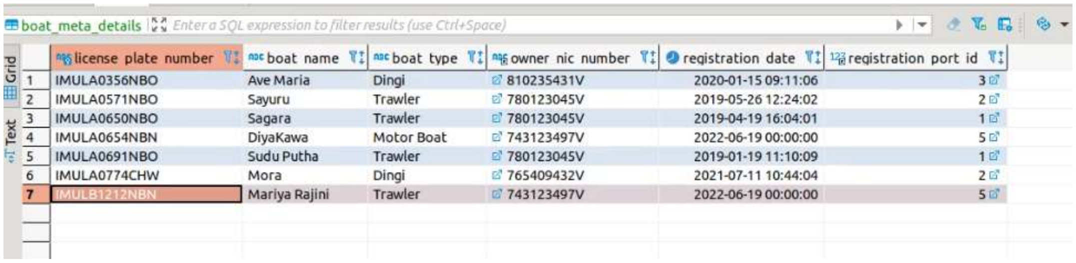
Appendix A: 2 Boat Owner Registration

Owner details amended to the boat details



Appendix A: 3 Owner details amended to the boat details

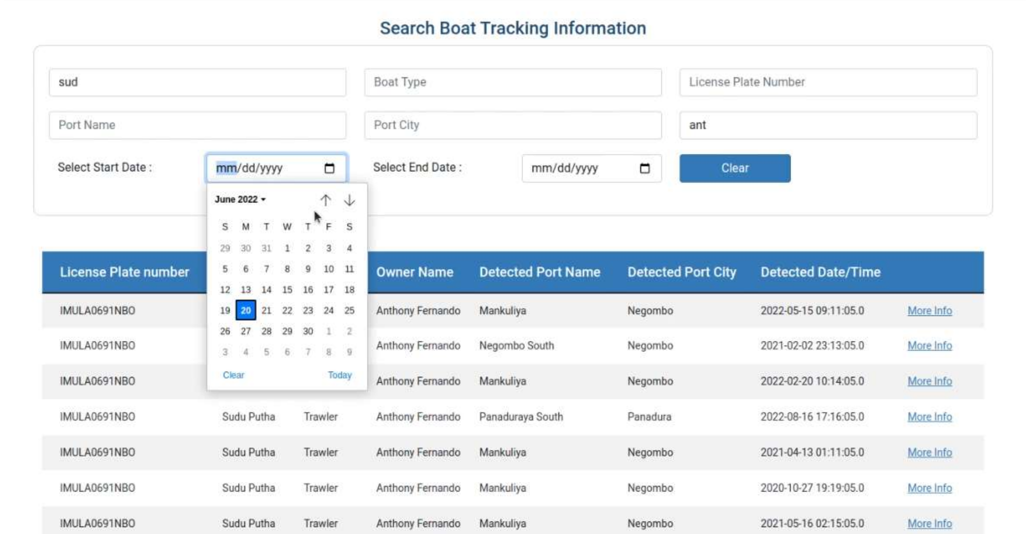
Boat Meta Details



	license plate number	boat name	boat type	owner nic number	registration date	registration port id
1	IMULA0356NBO	Ave Maria	Dingi	810235431V	2020-01-15 09:11:06	3
2	IMULA0571NBO	Sayuru	Trawler	780123045V	2019-05-26 12:24:02	2
3	IMULA0650NBO	Sagara	Trawler	780123045V	2019-04-19 16:04:01	1
4	IMULA0654NBN	DiyaKawa	Motor Boat	743123497V	2022-06-19 00:00:00	5
5	IMULA0691NBO	Sudu Putha	Trawler	780123045V	2019-01-19 11:10:09	1
6	IMULA0774CHW	Mora	Dingi	765409432V	2021-07-11 10:44:04	2
7	IMULB1212NBN	Mariya Rajini	Trawler	743123497V	2022-06-19 00:00:00	5

Appendix A: 4 Boat Meta Details

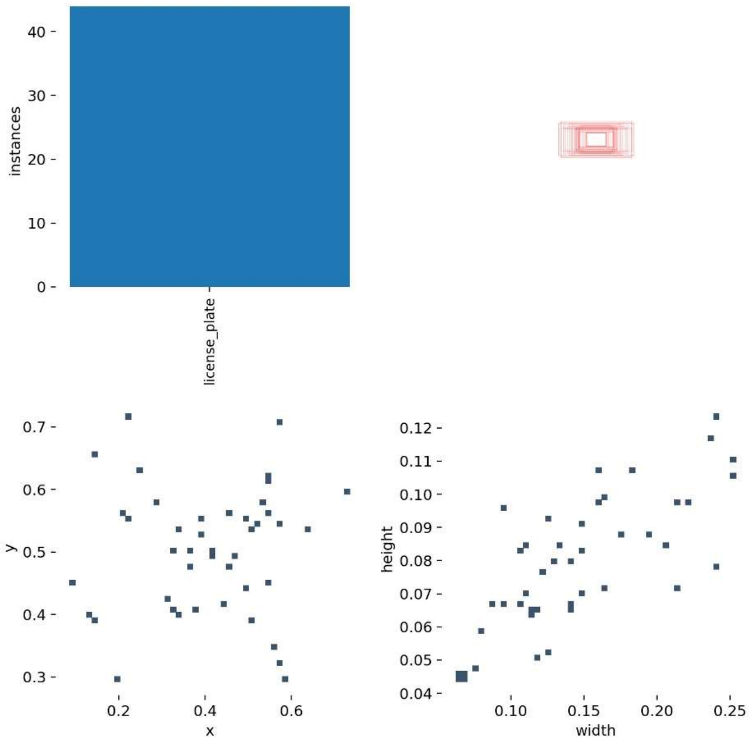
Searching information by filtering date



License Plate number			Owner Name	Detected Port Name	Detected Port City	Detected Date/Time	
IMULA0691NBO			Anthony Fernando	Mankuliya	Negombo	2022-05-15 09:11:05.0	More Info
IMULA0691NBO			Anthony Fernando	Negombo South	Negombo	2021-02-02 23:13:05.0	More Info
IMULA0691NBO			Anthony Fernando	Mankuliya	Negombo	2022-02-20 10:14:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Panaduraya South	Panadura	2022-08-16 17:16:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2021-04-13 01:11:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2020-10-27 19:19:05.0	More Info
IMULA0691NBO	Sudu Putha	Trawler	Anthony Fernando	Mankuliya	Negombo	2021-05-16 02:15:05.0	More Info

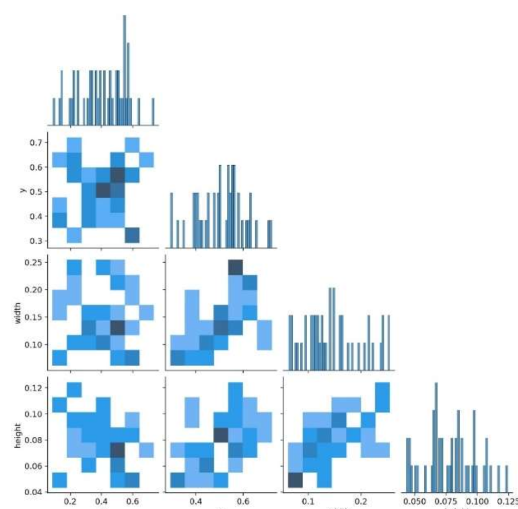
Appendix A: 5 Searching information by filtering date

Labels in the training process



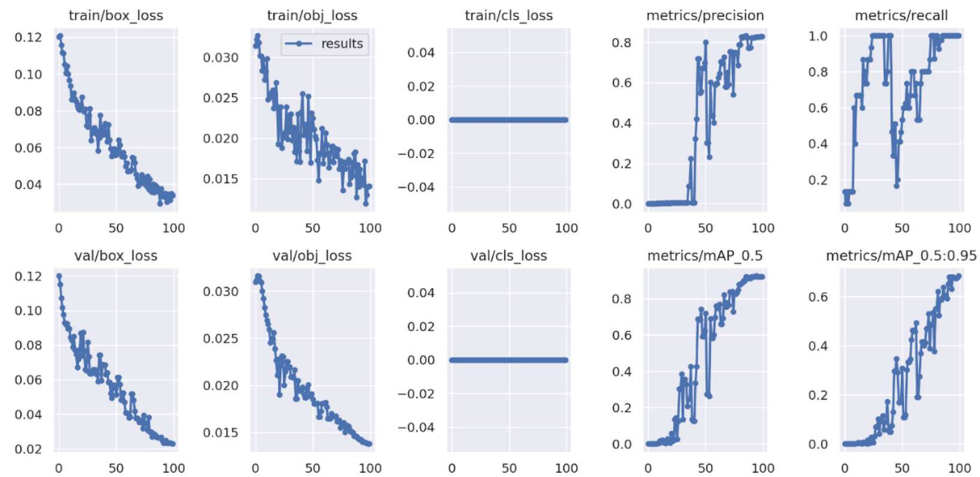
Appendix A:6 Labels in the training process

Labels Correlogram



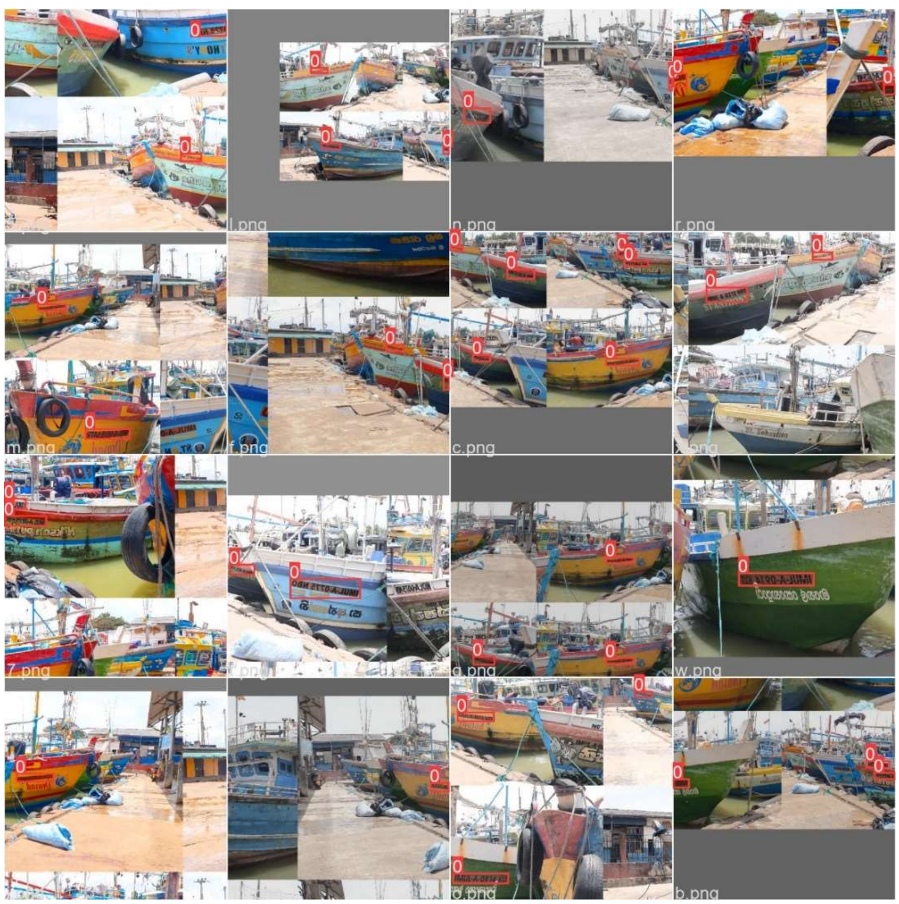
Appendix A: 7 Labels Correlogram

Results of the Training Process



Appendix A: 8 Results of the Training Process

Training Batch



Appendix A: 9 Training Batch

Validation Batch

Appendix A: 10 Validation Batch