Department of Electrical and Computer Engineering
North South University

Senior Design Project

An Intelligent Traffic Control System for Trouble-free Movement of Ambulances and Fire Trucks on Heavy Traffic Roads in Bangladesh Using YOLOv5 Algorithm and GSM Module

TASFIA SAMIN **ID: 1821823042**

FARZANA HAQUE **ID: 1821386042**

NESHAT ANJUMAN MOURI **ID: 1812749042**

Faculty Advisor:

Md. Shahriar Hussain

Senior Lecturer

ECE Department

Fall, 2022

DECLARATION

This is to certify that this project is our original work. No part of this work has been submitted elsewhere, partially or fully, for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students' names & Signatures

1. Tasfia Samin

Tasfia Samin

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

2. Farzana Haque

Farzana Haque

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

3. Neshat Anjuman Mouri

Neshat Anjuman Mouri

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ACKNOWLEDGEMENT

First, we would like to thank almighty Allah for his grace in accomplishing this thesis work timely and complete book. We would like to firmly our innate and genuine appreciation to our respectable research supervisor, Md. Shahriar Hussain, Senior Lecturer, Department of CSE, North South University (NSU), from the core of my heart, for his kind support, cooperation, steerage, formative, superintendence, instructions, and exhortation all through the complete work. His direction and advice always lead us to go on the right track. Not only that, but also his support always helped us to detect any fault in our work. We want to thank our mates and family members for their affection, care, prayer, and support to reach us at this stage.

ABSTRACT

One of the biggest problems developing countries like Bangladesh are facing is traffic problems. Especially metropolitan cities in a country need a proper traffic control system for stress-free movement, so the situation gets less stressful. This work has been designed to create a smooth-movement process for emergency vehicles like ambulances and fire trucks, which needs to be accessible as soon as possible during traffic congestion. The system would notify the traffic controller after detecting the emergency vehicle among all the stuck vehicles. This work uses a dataset constructed and labelled manually. To train the model, we have used the YOLOv5 algorithm. The accuracy of our model is 94%. The model detects emergency vehicles like ambulances and fire trucks as well as other vehicles captured by the CCTV camera of the road. However, only the detection of emergency vehicles would be notified to the traffic controller through Arduino Uno and GSM Module. That way, the traffic controller could clear that road for the easy movement of those emergency vehicles.

TABLE OF CONTENTS

CHAPTER 1 ..... **7**

Introduction **7**

1.1 Motivation 8

1.2 Objective 8

1.3 Report Outline 9

CHAPTER 2 ..... **10**

Related Work **10**

2.1 Literature Review 11

Smart Traffic Control System for Ambulance **14**

CHAPTER 3 ..... **16**

Theoretical Background **16**

3.1 Dependencies, Tools & Equipment 17

3.1.1 Software 17

3.1.2 Hardware 17

3.2 Techniques & Skills (Software) 18

3.2.1 Data Collection and Labelling 18

3.2.2 YOLO Algorithm 19

3.2.3 YOLO v5 Repository 21

3.3 Techniques & Skills (Hardware) 22

3.3.1 Arduino UNO 22

3.3.2 GSM SIM800L 23

CHAPTER 4 ..... **25**

Result and Analysis **25**

4.1 Project Illustration 26

4.2 System Design 27

4.2.1 Process 28

4.3 Result 30

CHAPTER 5 ..... **38**

Conclusion **38**

5.1 Summary of the Work 39

5.2 Project Development Phases 39

5.3 Ethical and Professional Responsibilities 40

5.4 Environmental Impact 40

5.5 Project Sustainability 41

BIBLIOGRAPHY **42**

LIST OF FIGURES 45

LIST OF TABLES 47

CHAPTER 1

Introduction

1.1 Motivation

Traffic congestion is the biggest issue a developing country like Bangladesh currently has. Changing the whole scenario overnight is difficult, but we can resolve it considering priority. This work targets to make changes to the existing system while simultaneously triggering a vital issue in traffic congestion. Our work will provide a safe movement system for vehicles which should be considered with the highest priority, like ambulances and fire trucks. In that matter, our model detects the emergency vehicle out of all the traffic in congestion using a deep learning model and clears that road as soon as possible by notifying the traffic controller regarding this emergency vehicle. The situation worsens for a critical patient stuck on a road or for people stuck in a fire accident if the emergency vehicles cannot contribute on time. So, the main purpose of this work is to ensure human life security by not making huge differences in the current system so that it is simultaneously executable and the problem gets solved.

1.2 Objective

The main objective of this project is to observe the roads where vehicles are stuck in traffic congestion and identify emergency vehicles like ambulances and fire trucks by applying a suitable detection model and notifying the traffic controller. This work focused on incorporating a new system into the existing one for effective execution in traffic conditions. Where the existing system is not required to make massive changes, but the impact of the change is significant. The system will provide a trouble-free movement system for ambulances and fire trucks by detecting them on the heavy traffic road and notifying the traffic controller of that route. The traffic congestion does not make any difference for the emergency vehicles, which may cause the loss of valuable lives. Therefore, an intelligent traffic management system prioritizing emergency vehicles can be proved effective in terms of saving valuable lives.

1.3 Report Outline

In the first chapter, we explained the motivation and objective of our project. In chapter 2, we will be discussing the related works regarding our project. In chapter 3, we will explain the theoretical background of our project. In chapter 4, we are going to describe result analysis which includes project illustration, system diagram, process to meet desired needs within realistic constraints and result as well. Chapter 5 is the conclusion which will include the summary of work, project development phases, ethical and professional responsibilities in different working phases, environmental impact and project sustainability.

CHAPTER 2
Related Work

2.1 Literature Review

Ananya et al. proposed an effective traffic control framework by presenting a detecting system. In this paper [1], their main objective is to use the existing system with a vision to capture the changing traffic pattern and provide a sign to the traffic controller. It also proposes to make an exceptional arrangement for emergency vehicles. In this work, they have used YOLO (You Only Look Once) algorithms for detecting objects from each of the videos and SORT (Simple Online and Real-time Tracking algorithm) for tracking objects of different frames. Moreover, after the detection process, there will be a simple mathematical calculation to detect the intersection of the basis of the vehicles on the previous and current position.

Miss Khushi proposed a breakthrough to the conventional traffic control system in her paper [2]. Here she will be using MATLAB codes for image processing through which the model will bring change to the timing of green amber light and red light depending on the density of traffic and the condition of the road. Here she is using two Arduino. One is for controlling the timing of the green amber light, and the other one is for the red light.

Sung-Dong Kim presents a situation-cognitive traffic light control algorithm that measures the traffic volume using an object detection algorithm called You Only Look Once (YOLO) and controls the traffic signal intervals according to the traffic volume [3]. In this paper, they used YOLO algorithm to determine whether it can detect vehicles in the real road image. The algorithm expects a smooth traffic flow and the reduction of the driver's stress.

Nan Jiang et al. discussed the elaboration of ML procedures in access control streamlining for arbitrary access plans and irregular access plans in their paper [4]. They prove that the ML-based

techniques could be more powerful than the entrance control issue contrasting and Non-ML-based strategies.

Miss Varsha et al. have proposed a system for reducing traffic congestion using image processing [5]. The system will detect vehicles through images instead of using electronic sensors. They also plan to provide a suitable solution for emergency vehicles stuck in traffic to clear the route by using Bluetooth. When an ambulance is detected, the signal is green, and the timing is extended by 20 seconds. This can be extended to any number of signals along the ambulance's path so that emergency cases can be served immediately.

Rajeshwari S. et al. proposed an Intelligent Traffic Control System for Ambulance Clearance in their paper [6]. There are two parts in this work. The first part is the ZigBee transmitter, placed in the emergency vehicle. When the switch is pressed, it will transmit the signal. The signal contains a unique id and security code. The transmitter contains a microcontroller and ZigBee module. The microcontroller sends the commands and data to the ZigBee. The second part is the receiver, placed at the traffic pole. It also contains a microcontroller and ZigBee module. The receiver compares the security code received to the security code present in its database. If it matches, then it will turn the green light on.

Stevlin Antonov et al. presented a method for signaling the traffic light by applying strobe lights as an indicator [7]. In this technique, flashing strobe lights of emergency vehicles will be sensed by cameras and strobe sensors which are directly connected to the traffic light controller, hence signaling the presence of emergency vehicles.

Al Hussain Akoum et al. suggest implementing a smart traffic controller using real-time image processing, which uses image processing techniques to count the number of vehicles on the road

and estimate the density [8]. The number of vehicles found can be used for surveying or controlling the traffic signal. This research uses software that takes a picture or video.

Table 2.1.1 shows the works, algorithm, components and accuracy of the works related to our study.

	Paper Tittle	**Author Name**	**Work**	**Algorithm/ Components**	**Accuracy**
1	Intelligent Traffic Light Control System Based On Traffic Environment Using Deep Learning	Moolchand Sharma et al	Detects and tracks vehicles on a video stream and counts those vehicles going across a predefined line.	YOLO, SORT	63.4%
2	Smart Control of Traffic Light System using Image Processing	Khushi	Proposes a traffic control system based on image processing using MATLAB code which changes the time of green, amber and red light with respect to the traffic density and traffic count.	MATLAB, Arduino UNO	70-80%
3	Situation-cognitive traffic light control based on object detection using YOLO algorithm	Sung-Dong Kim	Measures the traffic volume and controls the traffic signal intervals according to the traffic volume.	YOLOv2, Raspberry Pi	90.38%
4	Traffic Prediction and Random-Access Control Optimization: Learning and Non-Learning-Based Approaches	Nan Jiang et al	This article first summarizes the general structure of optimization for different Random Access Channel (RACH) schemes and then reviews existing RACH optimization methods	DA estimator, MoM estimator, MLE estimator, RL-based optimizer SL-based optimizer, DLS-based optimizer	…

			based on ML and non-ML techniques.		
5	Smart traffic control with ambulance detection	Varsha Srinivasan et al	Detects ambulances through images, and once the ambulance is near the signal, the driver can send a command to the Bluetooth module, thereby guiding the traffic signal to change accordingly.	Blob, Bluetooth Module	…
6	Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance and Stolen Vehicle Detection	Rajeshwari S. et al	A switch pressed from emergency vehicles will transmit a signal which turns the traffic signal green as long as the emergency vehicle is waiting in the traffic junction.	PIC16F877A Microcontroller, ZigBee module	…
7	Smart Traffic Control System for Ambulance	Stevlin Antonov et al	Cameras & strobe sensors connected to the traffic light controller will sense the flashing strobe lights of emergency vehicles & the green light will be turned on.	Strobe Light, Strobe Sensor	…
8	Automatic Traffic Using Image Processing	Al Hussain Akoum	This method uses image processing techniques to count the number of vehicles on the road and estimate the density, which can be used for controlling the traffic signal.	Filtering Method	=< 90%

Table 2.1.1: Works, Algorithms, Components & Accuracy of the Related Works

Studies [1] and [8] detect and count the number of vehicles on the road and estimate the density using the YOLO algorithm and filtering method, respectively. In [2] and [3], these works measure traffic volume and control the traffic signals according to traffic congestion. In [5], [6], and [7], a command, switch, and strobe light is used respectively from emergency vehicles to control the traffic lights. Our work detects ambulances, fire trucks and other vehicles using the YOLOv5 algorithm. It sends SMS to the traffic controller police after detecting any emergency vehicle automatically through Arduino UNO and GSM Module so that the traffic police can clear the road where an emergency vehicle is present. We have created and labelled the dataset manually to train our model, and our model's accuracy is 94%.

CHAPTER 3
Theoretical Background

3.1 Dependencies, Tools & Equipment

3.1.1 Software

1. Object Detection:
    - Make Sense AI
    - Google Collab
    - YOLO_v5 Repository
2. Real-time Detection:
    - CUDA 11.7
    - PyTorch
    - Pycocotools
    - Microsoft C++ build tools

3.1.2 Hardware

1. Simulation
    - Proteus 8 Professional
    - Arduino IDE 2.0.2
2. Hardware Equipment
    - Webcam
    - Arduino UNO
    - GSM SIM800L
    - Mobile Phone
    - SIM

- 9V 2A Adapter
- Buck Converter
- Wires

3.2 Techniques & Skills (Software)

3.2.1 Data Collection and Labelling

We collected the images of ambulances, fire trucks and other vehicles from the Open source Images and google. We used Make Sense AI for labelling the images [9]. Make Sense AI is a free-to-use online tool for labelling images which makes the process of preparing a dataset much easier and faster [10]. YOLOv5 uses its specific representation of bounding box coordinates. After labelling, we get two files with the same name but different extensions.

One is the JPG image file, and the other is the corresponding .txt text file. Each .txt file has five parameters:

- The index of objects class
- (X, Y) coordinates that represent the center of the bounding box (x and y are each a parameter)
- Width of the bounding box
- Height of bounding box.

The coordinates and bounding box dimensions are normalized between zero and one as a percentage of image dimensions [11]. We labelled a dataset of 2009 images containing ambulances, fire trucks and other vehicles. The class ambulance is labelled 0, the class fire truck is labelled 1, and other vehicles are labelled 2.

Figure 3.2.1.1: Image Labelling

Figure 3.2.1.1 shows the .txt file of the corresponding JPG file, which contains object-class-ID, X center, Y center, Box width and Box height.

3.2.2 YOLO Algorithm

YOLO (You Only Look Once) is a deep learning algorithm that uses neural networks to provide real-time object detection; popular because of its speed and accuracy. It works using three techniques:

1. Residual blocks: The image is divided into several grids. Each of the grids has a dimension of S x S.
2. Bounding box regression: A bounding box is an outline that highlights an object in the image. Each bounding box in the image consists of attributes; Class, Center of bounding box (X, Y), Width and Height.

3. Intersection Over Union (IOU): IOU describes how the bounding boxes overlap. YOLO algorithm uses IOU to provide an output box that surrounds the objects perfectly.

Combination of the three techniques:

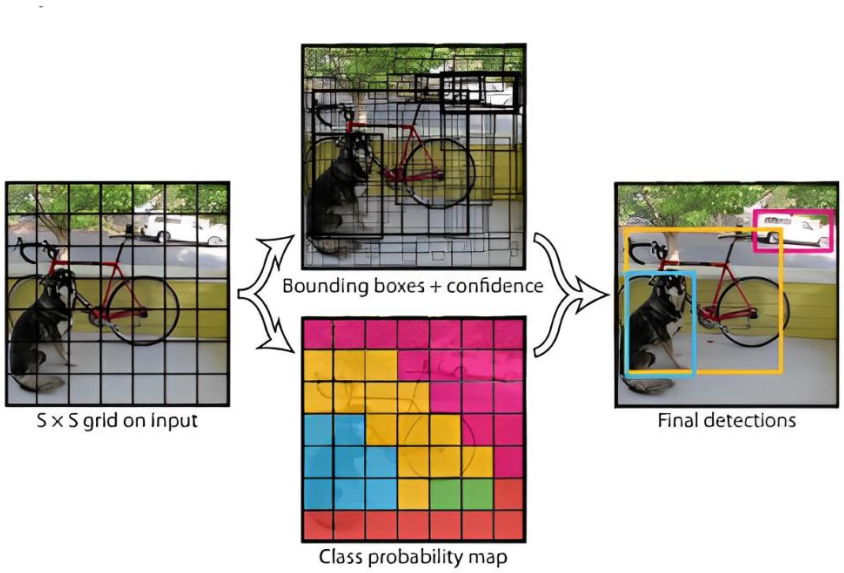

Figure 3.2.2.1: Combination of Residual blocks, Bounding box regression & IOU

Figure 3.2.2.1 illustrates how the three techniques are applied to produce the final detection results [12].

We have used YOLOv5 algorithm in our project. YOLOv5 labelling format is < object-class-ID> <X center> <Y center> <Box width> <Box height>. The YOLOv5 GitHub repository is created and maintained by Ultralytics. To comply with Ultralytics directories structure, we created the dataset following the structure in figure 3.2.2.1.

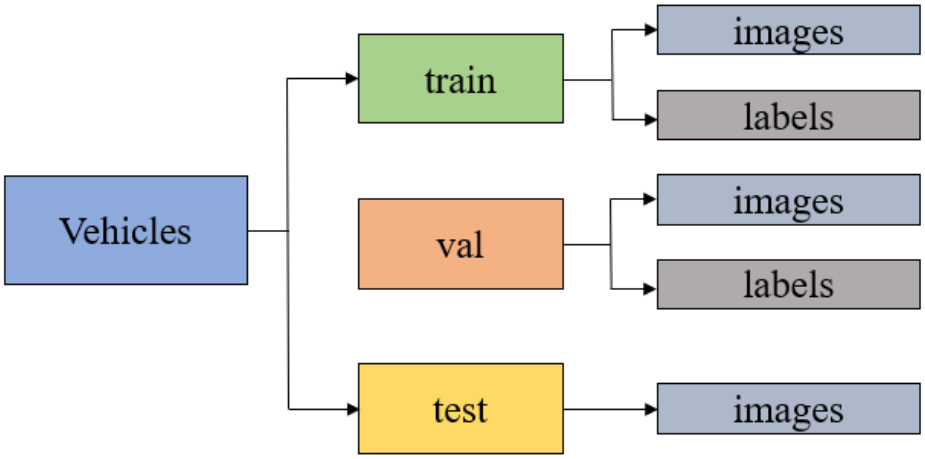

Figure 3.2.2.2: Dataset Structure for YOLOv5

Figure 3.2.2.2 shows the structure of dataset for YOLOv5 algorithm of our project. The Vehicle folder contains three folders named train, val and test. The train and val folders have two folders each; images and labels. The images folder contains the pictures of ambulances, fire trucks and other vehicles in JPG format, and the labels folder contains the corresponding .txt files of the images. The train folder is for training the dataset, the val folder is for validation, and the test folder is for testing the model [13] [14].

3.2.3 YOLO v5 Repository

To set up YOLOv5, a YOLOv5 GitHub repository created and maintained by Ultralytics must be cloned [14]. We have cloned the YOLOv5 GitHub repository in Google Collab for object detection. For Real-time detection, we have downloaded the GitHub repository [20].

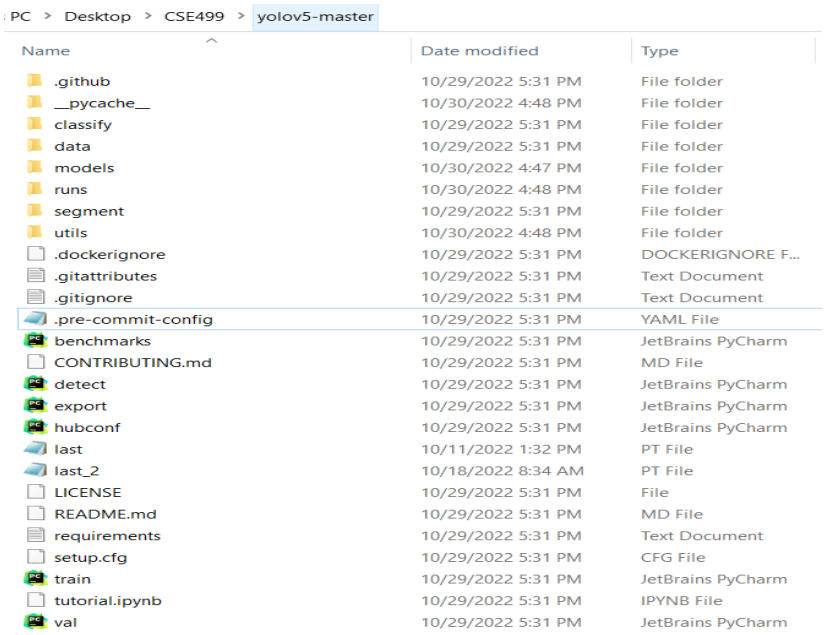

Figure 3.2.3.1: YOLOv5 GitHub Repository

Figure 3.2.3.1 shows the GitHub Repository of YOLOv5 algorithm. Here, the detect file is required for object detection.

3.3 Techniques & Skills (Hardware)

3.3.1 Arduino UNO

For making an object interactive and accessible, the single-board microcontroller required is Arduino UNO. It is designed around an 8-bit Atmel AVR microcontroller open-source hardware board. This interface consists of a USB interface, six analogue input pins and 14 digital I/O pins, allowing the user to attach an extension board [15]. Among 14 i/o pins, six pins can be used as PWM outputs. To get started, it required an adapter battery of AC to DC.

In our project, we have used pin 7 and pin 8 as external pins where pin 7 is the RX pin or the receiving pin, and 8 is the TX pin or the sending pin.

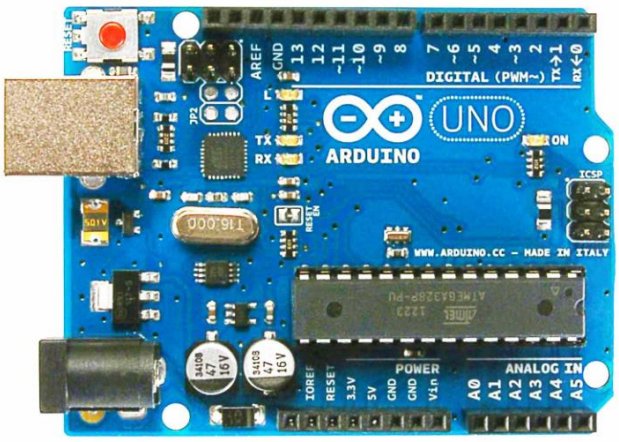

Figure 3.3.1.1: Arduino Uno [18]

Figure 3.3.1.1 shows the picture of Arduino Uno.

3.3.2 GSM SIM800L

For wireless radiation monitoring through Short Messaging Service, a customized Global System for Mobile Communication (GSM) module is created (SMS). This module can provide text messages as serial data to host servers from radiation monitoring devices like survey meters and area monitors [16]. In our project, we used a sim800L GSM module. Any microcontroller with the SIM800L GSM module from Simcom may connect to the mobile network to make and receive phone calls, send and receive text messages, and connect to the internet via GPRS, TCP, or IP. Another benefit is that the board can be used anywhere in the world because it uses of current mobile frequencies. In the GSM module, we used pins 4, 5, and 6. Here, pin 4 is for RXD, which

is for serial data input, pin 5 (TXD) is for serial data output, and pin 6 is for (GND) module ground reference [17].

Figure 3.3.2.1: GSM SIM800L

Figure 3.3.2.1 shows the picture of GSM SIM800L.

.

CHAPTER 4

Result and Analysis

4.1 Project Illustration

We have developed an intelligent traffic control management system. The problem we are dealing with is the traffic jam that is causing the problem for the ambulances and the fire trucks. The roads of Bangladesh are congested, which is causing big trouble in our economy and social life, especially for emergency vehicles. We developed a system using deep learning that can detect ambulances and fire trucks and give them a prioritized movement by sending an SMS to traffic police regarding the presence of an emergency vehicle. Our system detects ambulances and fire trucks using the image processing technique. We have used YOLOv5 algorithm to detect emergency vehicles. First, we labelled a dataset of 2009 images. Our dataset has three classes: ambulance, fire truck and other vehicles. Then we trained 80 percent of our dataset using 300 epochs. We executed the detect.py file in the YOLOv5 repository for real-time emergency vehicle detection using our trained model and a real-time video. In this step, we used the in-built webcam of our laptop to capture the video. Next, we connected the detect.py file with Arduino Uno through serial communication. Using serial communication, ambulance detection or fire truck detection signal will be sent to the Arduino first. Then, using GSM SIM800L, an 'ambulance detected' or 'fire truck detected' SMS will be sent to the traffic controller police.

4.2 System Design

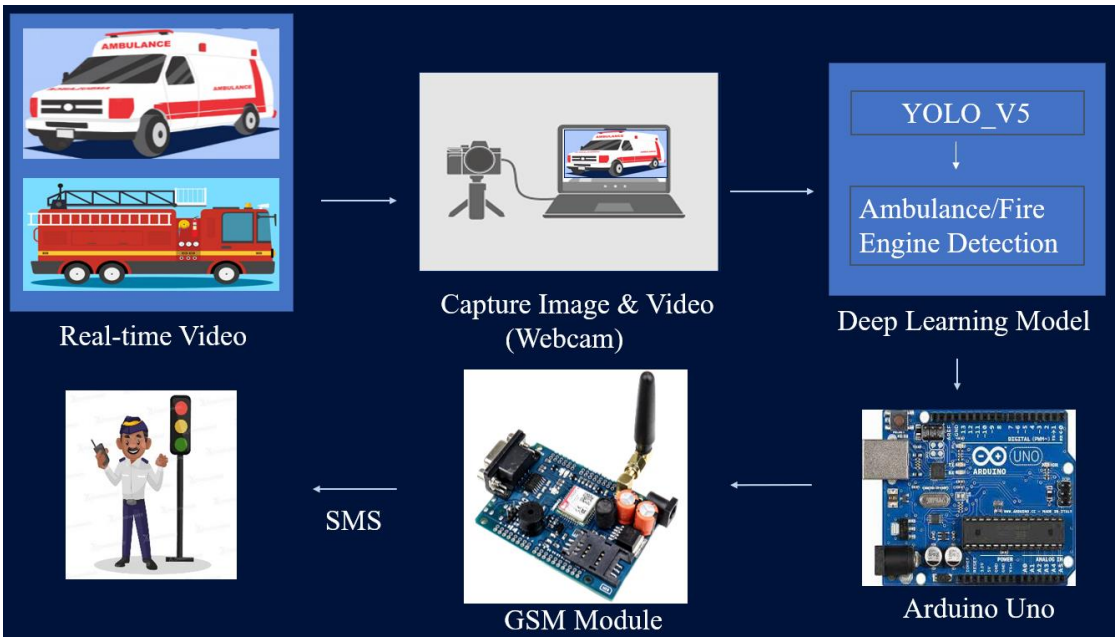

Figure 4.2.1: System Diagram

Figure 4.2.1 shows the system diagram of our project. First, our webcam will capture ambulances or fire trucks in real-time video. Here, we have used the in-built webcam of our laptop. We have labelled a dataset of 2009 images. Our dataset contains images of ambulances, fire trucks and other vehicles. We trained them using the YOLOv5 algorithm. After capturing the video, our trained model is able to detect emergency vehicles. Once the model detects any emergency vehicle, this detection data will be passed to the Arduino through serial communication. Arduino is connected to GSM Module with RX and TX pins. Then an emergency vehicle detection SMS will be sent to the traffic police through the GSM Module so that the traffic police can clear the road where an emergency vehicle is present. We have other vehicles in our dataset, along with ambulances and fire trucks, and our model can detect three of the classes. However, in the case of sending an SMS to the traffic controller, an SMS will be sent only when an ambulance or fire truck is detected.

4.2.1 Process

At first, we collected images of ambulances, fire trucks and other vehicles from google and Open Source Images. Then we labelled the images using an online tool named Make Sense AI. We used Google Collab for coding. After completing the dataset, we implemented our code. We installed some required libraries, connected google drive and then connected the google drive folder, which contains the dataset. Then we cloned the YOLOv5 from Git Hub. Next, we trained, tested and validated 80, 10 and 10 percent of the dataset, respectively, using 300 epochs. We trained and tested our dataset using the YOLOv5 algorithm.

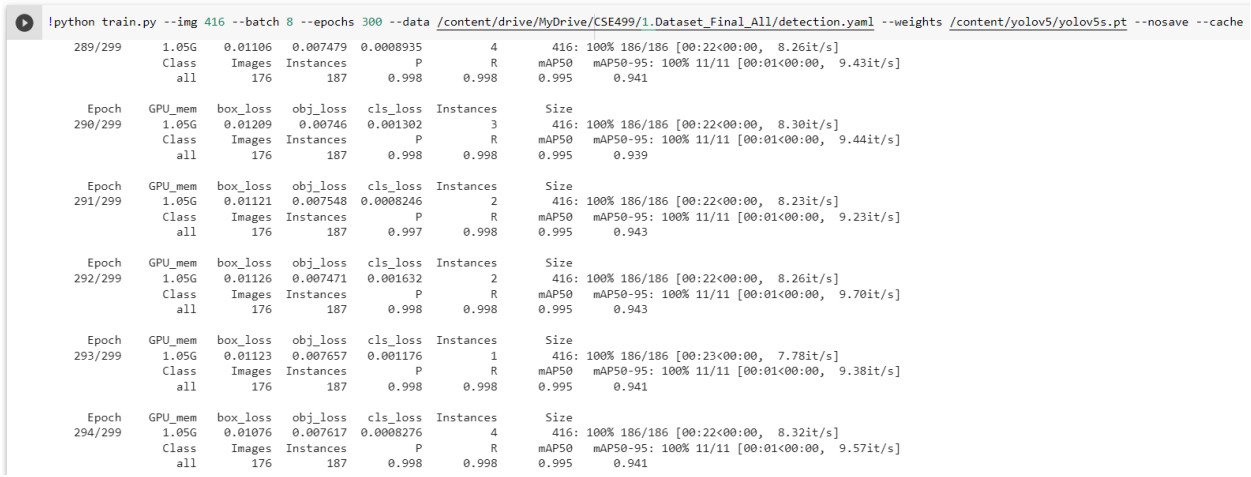

Figure 4.2.1.1: Dataset Training

Figure 4.2.1.1 shows training of our dataset using 300 epochs.

Once our model was ready, we tested our model through some images and videos which contained emergency vehicles. For real-time emergency vehicle detection, we downloaded the YOLOv5 repository and installed the required libraries and dependencies. Then we incorporated our model

into the detect.py file. Finally, we executed the detect.py file, which is the real-time detection, with a real-time video using the webcam of our laptop.

For hardware implementation, we selected pin 7 & pin 8 of Arduino for serial communication. Pin 7 is the RX pin of Arduino & pin 8 of the TX pin. We connected the RX pin of Arduino with the TX pin of the GSM module and the TX pin of Arduino with the RX pin of the GSM module. We have used a buck converter to convert 9V of the adapter to 4.2 volts to provide power to the GSM Module. At last, we inserted a 4.5g Grameen phone Sim into the GSM Module.

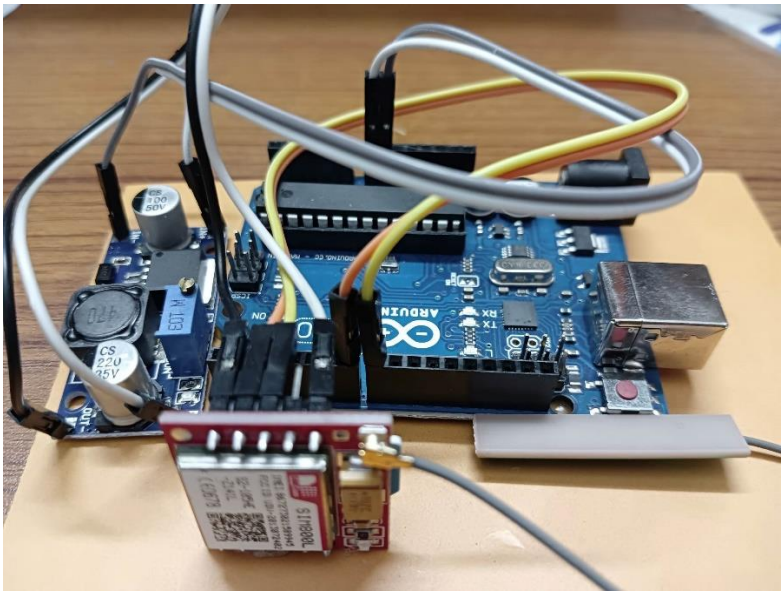

Figure 4.2.1.2: Hardware Configuration

Figure 4.2.1.2 shows the hardware configuration of our project.

Though we have other vehicles in our dataset, along with ambulances and fire trucks, and our model can detect three of the classes, in the case of sending an SMS to the traffic controller, an SMS will be sent only when an ambulance or fire truck is detected.

4.3 Result

Our model is successfully able to detect ambulances, fire trucks and other vehicles from real-time video. The accuracy of our model is 94%. Our system is also successfully able to send SMS to the traffic controller when an ambulance or fire truck is detected.

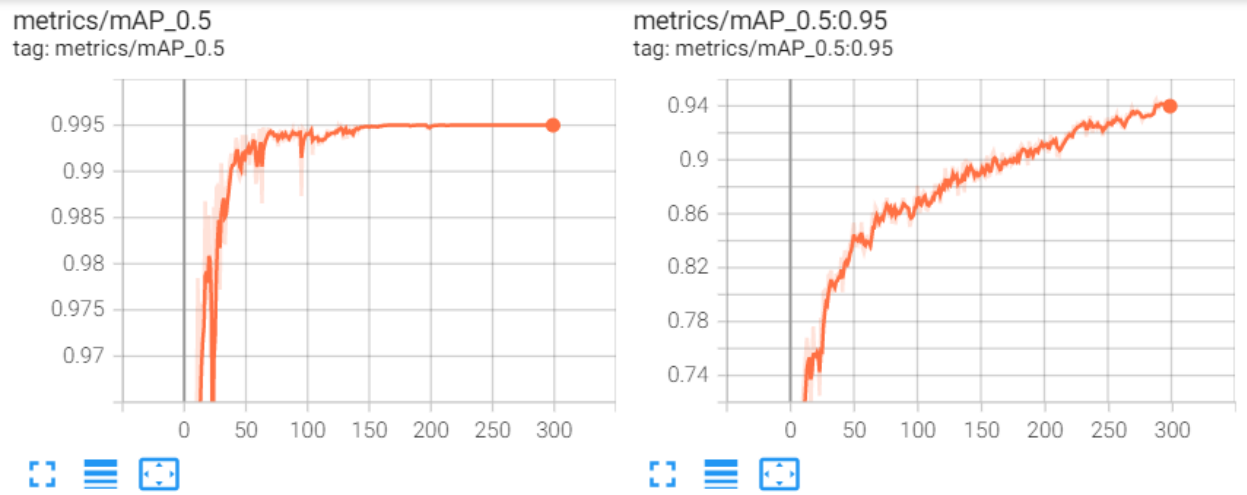

Figure 4.3.1: Accuracy Graph

Figure 4.3.1 illustrates the Mean Average Precision (mAP) of our model. For 300 epochs, we got 94% accuracy. Here, 'mAP_0.5' is the mean Average Precision (mAP) at the IOU (Intersection over Union) threshold of 0.5. 'mAP_0.5:0.95' is the average mAP over different IOU thresholds, ranging from 0.5 to 0.95 [13].

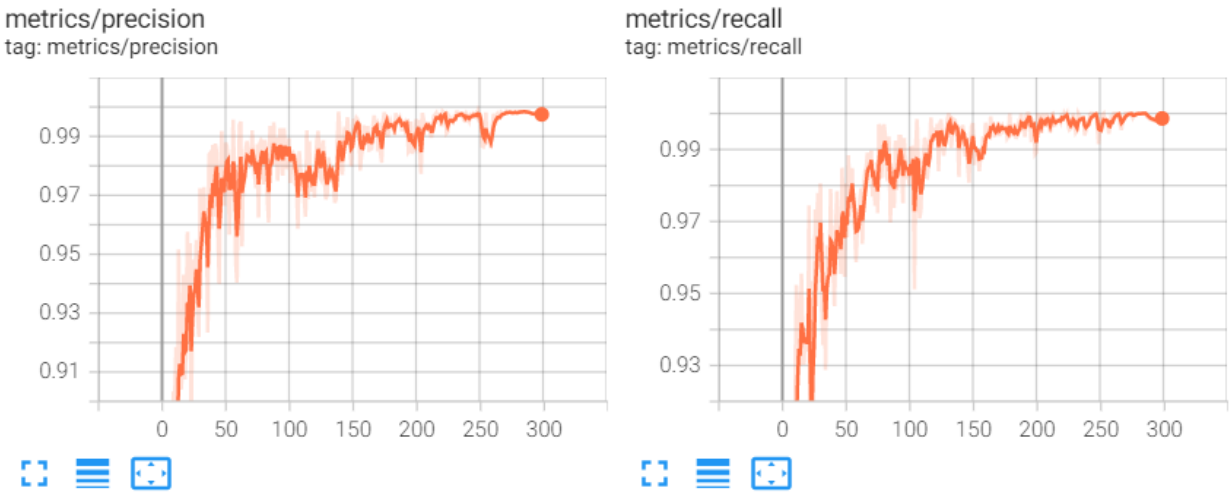

Figure 4.3.2: Precision & Recall graph

In figure 4.3.2, we can see the graph of precision and recall of our model. Here, Precision measures how much of the bounding box predictions are correct, and recall measures how much of the true bounding box was correctly predicted.

Precision = True positives / (True positives + False positives)

Recall = True positives / (True positives + False negatives) [13]

Figure 4.3.3: Ambulance Detected

Figure 4.3.3 shows an ambulance detected with 96 percent accuracy.

Figure 4.3.4: Fire Truck Detected

Figure 4.3.4 shows a fire truck detected with 95 percent accuracy.

For real-time emergency vehicle detection, we downloaded the YOLOv5 repository first and then installed the required libraries and dependencies. After that, we incorporated our model into the

detect.py file. Next, we executed the detect.py file, which is the real-time detection file. We performed it with a real-time video using the webcam of our laptop.

Figure 4.3.5: Fire Truck Detected from Real-time Video

Figure 4.3.5 shows a fire truck detected with 89 percent accuracy from a real-time video.

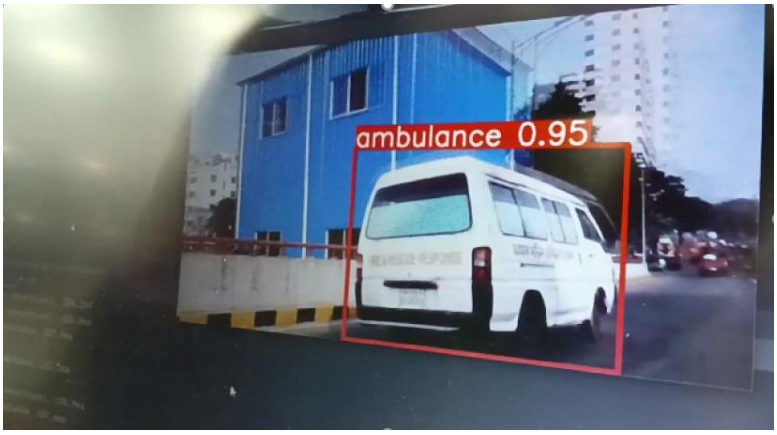

Figure 4.3.6: Ambulance Detected from Real-time Video

Figure 4.3.6 shows an ambulance detected with 95 percent accuracy from a real-time video.

After detecting emergency vehicles, we modified the detect.py file in the yolov5 repository to connect the real-time detection with Arduino and GSM Module for sending SMS. For sending signals to Arduino, we used the Serial Communication technique.

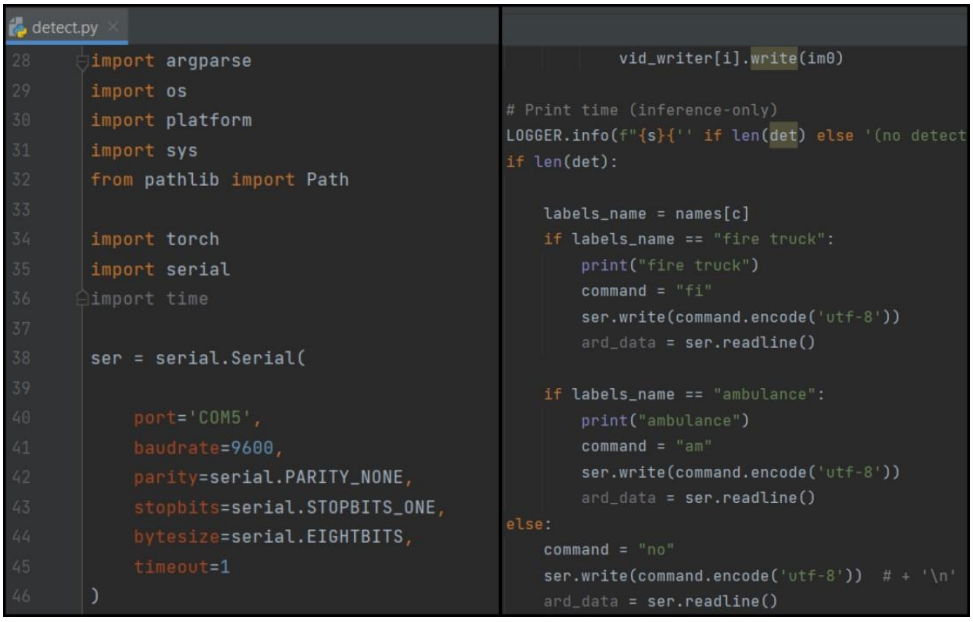

Figure 4.3.7: detect.py file

Figure 4.3.7 shows the detect.py file. In the first modification step, we imported the serial library and then wrote the code for serial configuration. Here, COM5 is the USB serial port. This port is used to ensure serial port communication. Then we wrote the conditions for the fire truck, ambulance and other vehicles inside the 'LOGGER.info()' function. When the fire truck is detected, the serial communication port sends a command 'fi' for serial communication to pass the signal to the Arduino using 'ser.write()' function. When the ambulance is detected, the serial communication port sends a command 'am' to pass the signal to the Arduino. When other vehicles are detected, the serial communication port sends a command 'no' to pass the signal to the Arduino. Here, 'utf-8'

means 8-bit processor, as our Arduino has an 8-bit processor. The 'readline()' function checks the connection to the Arduino.

```
void loop() {


if (Serial.available() > 0) {
  String scommand = Serial.read();
  scommand.trim();
  // Serial.println(scommand);

  if (scommand == "fi" && flag == 0)
  {
    digitalWrite(13, HIGH);
    firealeart();
    firealeart1();
    firealeart2();
    flag = 1;
  }
```

```
// first sms
void firealeart() {
  myserial.println("AT\r");
  delay(1000);
  myserial.print("AT+CMGF=1\r");  // AT command
  delay(1000);
  myserial.println("AT+CMGS=\"+8801316847091\"")
  delay(1000);
  myserial.println("fire truck Detected.");
  delay(1000);
  myserial.println((char)26);  // End AT command
  delay(3000);
  myserial.println();
  delay(1000);
}
```

Figure 4.3.8: Arduino Code

Figure 4.3.8 shows the code which has been uploaded into Arduino. First, it checks if any of the serial data is available. The serial data are 'fi', 'am' and 'no'. Then Arduino reads the serial data, and if it receives 'fi' or 'am', it calls the function 'firealert()' or 'ambualert()' accordingly. These two functions are for sending SMS to the traffic controller through GSM Module using the SIM inserted in it. The two functions will not be called if it receives 'no', which is the command for other vehicles' detection. Therefore, the traffic controller will only receive SMS when ambulances and fire trucks are detected.

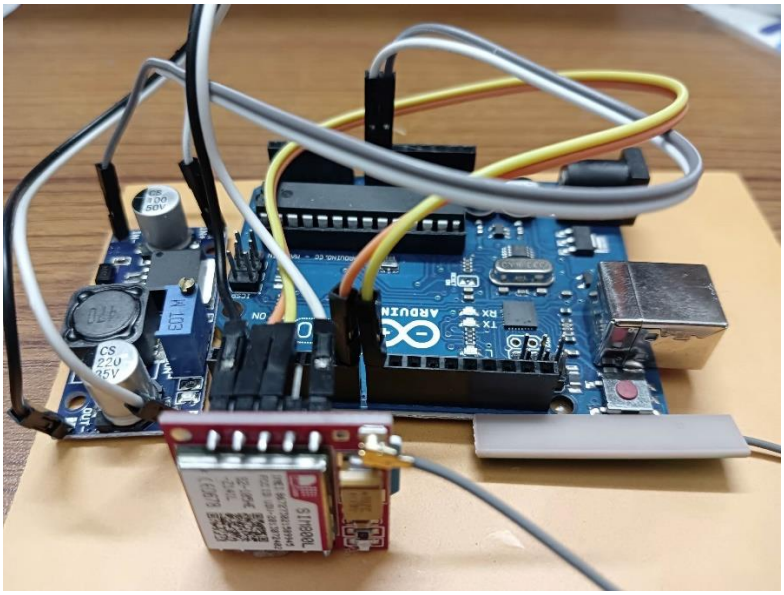

Figure 4.3.9: Hardware Configuration

Figure 4.3.9 shows the hardware configuration of our project. We selected external serial pin 7 & pin 8 of Arduino for serial communication. Pin 7 is the RX pin of Arduino & pin 8 of the TX pin. We connected the RX pin of Arduino with the TX pin of the GSM module and the TX pin of Arduino with the RX pin of the GSM module. We have used a buck converter to convert 9V of the adapter to 4.2 volts to provide power to the GSM Module. At last, we inserted a 4.5g Grameen phone Sim into the GSM Module.

The Arduino Uno gets the command of emergency detection through the USB port. As the receiving data (RX) pin of GSM is connected with the sending data (TX) pin of Arduino, GSM receives the serial data from Arduino and then sends SMS to the traffic controller using the SIM inserted in it.

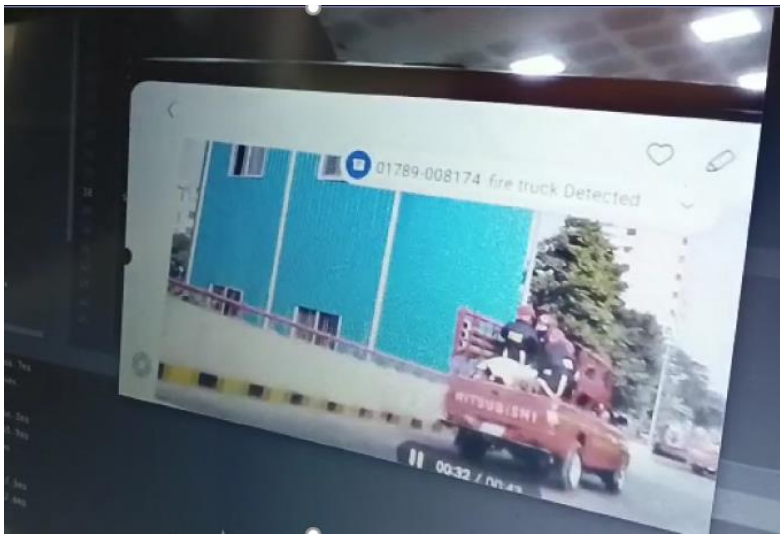

Figure 4.3.10: ‘fire truck Detected’ SMS Delivered

In figure 4.3.10, we can see the ‘fire truck Detected’ SMS delivered to the receiver.

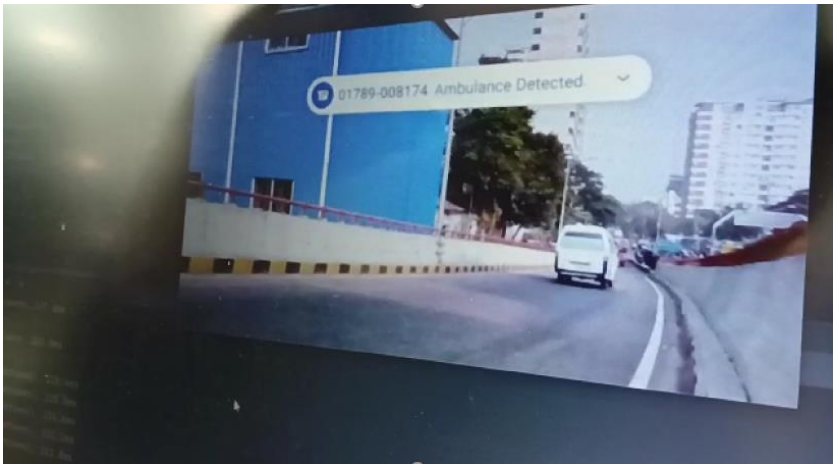

Figure 4.3.11: ‘Ambulance Detected’ SMS Delivered

In figure 4.3.11, we can see the ‘Ambulance Detected’ SMS delivered to the receiver.

CHAPTER 5
Conclusion

5.1 Summary of the Work

At first, we collected images of ambulances, fire trucks and other vehicles from google and Open Source Images. We created a dataset of 2009 images. After that, we labelled the images using an online image labelling tool, Make Sense AI. Then we did train, testing and validation with 300 epochs using the YOLOv5 algorithm. We got 94% accuracy from our model. After that, we tested the model using videos and then performed real-time detection. Finally, with the help of Arduino and the GSM module, we are successfully able to send SMS to the traffic controller after emergency vehicle detection so that the traffic controller can clear the road where an emergency vehicle is present, which would have a massive impact on our day-to-day lives.

5.2 Project Development Phases

In 499A, in the first five weeks, we did the planning for topic selection and literature review of the selected topic. In the next two weeks, we collected the dataset and labelled it. In eight weeks, we learned the process the image detection. In the ninth week, we did the training part. In the tenth week, we did the detection part of the ambulance. Last week, we did the final presentation.

In 499B, in the first two weeks, we created the entire dataset of 2009 images of ambulances, fire trucks and other vehicles. In the next two weeks, we did the image detection and ambulances, fire trucks and other vehicles from videos. In weeks 5 and 6, we completed the emergency vehicle detection from the real-time videos using a webcam. In week 7, we did the simulation of hardware configuration. In weeks 8,9, and 10, we worked on implementing hardware. In week 11, we successfully completed our entire project.

5.3 Ethical and Professional Responsibilities

In our data collection phase, we collected the data from scratch with the help of Google. Also, we have used some real-time pictures to make the dataset more relatable. In the dataset processing, model, real-time detection, and hardware part, we have tried to ensure 100% transparency, and the evidence is present in our work. Also, this report contains multiple citations in terms of the additions that were not originated by us.

5.4 Environmental Impact

The main idea behind choosing such a topic is to solve a real-time social problem. That way, our problem has a great social impact as well as it promotes human life security.

Social Impact

In our work, we have targeted solving regular as well as national problems to solve, and that way, we came up with the traffic system idea. We are proposing to create an emergency pathway on heavy-traffic roads for ambulances and fire trucks. This will decrease the death rate of people as well as it will reduce the hamper that may cause due to fire outbreaks.

Health & Safety

Since we are trying to create an immediate safety lane for the emergency patients stuck in an ambulance, along with that, we are also proposing the same for a fire truck to save human life along with resources from fire damage. That way, our work promotes human life security, which was the main motivation behind choosing this topic.

5.5 Project Sustainability

Our project offers a mechanism which would help not only the entire nation but also save valuable human lives. Ambulance services and fire brigade departments can buy our model to improve their services and ensure the safety of human lives on the road for any emergency patient and people stuck in a fire accident. Apart from that, this project would always be helpful as this traffic congestion is a constant issue. Although we have done the project using an in-built webcam of our laptop, in a broader manner, it would be executed through Closed Circuit Television.

BIBLIOGRAPHY

1. Sharma, M., Bansal, A., Kashyap, V., Goyal, P., & Sheikh, T. H. (2021). Intelligent Traffic Light Control System Based On Traffic Environment Using Deep Learning. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1022, No. 1, p. 012122). IOP Publishing.
2. Khushi, "Smart Control of Traffic Light System using Image Processing," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India, 2017, pp. 99-103, doi: 10.1109/CTCEEC.2017.8454966.
3. Kim, Sung. (2020). Situation-cognitive traffic light control based on object detection using YOLO algorithm. International Journal of Computational Vision and Robotics. 10. 133. 10.1504/IJCVR.2020.105682.
4. N. Jiang, Y. Deng and A. Nallanathan, "Traffic Prediction and Random Access Control Optimization: Learning and Non-Learning-Based Approaches," in IEEE Communications Magazine, vol. 59, no. 3, pp. 16-22, March 2021, doi: 10.1109/MCOM.001.2000099.
5. Srinivasan, Varsha & Rajesh, Yazhini & Yuvaraj, S & Muniraj, Manigandan. (2018). Smart traffic control with ambulance detection. IOP Conference Series: Materials Science and Engineering. 402. 012015. 10.1088/1757-899X/402/1/012015.
6. R. Sundar, S. Hebbar and V. Golla, "Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection," in IEEE Sensors Journal, vol. 15, no. 2, pp. 1109-1113, Feb. 2015, doi: 10.1109/JSEN.2014.2360288.

7. Antonov, Svetlin. (2016). SMART TRAFFIC CONTROL SYSTEM FOR AMBULANCE. Advanced aspects of theoretical electrical engineering. 95-98.
8. Akoum, Al. (2017). Automatic Traffic Using Image Processing. Journal of Software Engineering and Applications. 10. 765-776. 10.4236/jsea.2017.109042.
9. *Make sense*. Make Sense. (n.d.). Retrieved January 5, 2023, from https://www.makesense.ai/
10. SkalskiP. (n.d.). *Skalskip/make-sense: Free to use online tool for labelling photos. https://makesense.ai*. GitHub. Retrieved January 5, 2023, from https://github.com/SkalskiP/make-sense
11. Davies, D. (2021, December 14). *Collect and label images to train a YOLOV5 object detection model in Pytorch*. W&B. Retrieved January 5, 2023, from https://wandb.ai/onlineinference/YOLO/reports/Collect-and-Label-Images-to-Train-a-YOLOv5-Object-Detection-Model-in-PyTorch--VmlldzoxMzQxODc3
12. *Introduction to yolo algorithm for object detection*. Section. (n.d.). Retrieved January 15, 2023, from https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/
13. Lihi Gur Arie, P. D. (2022, December 15). *The Practical Guide for Object Detection with Yolov5 algorithm*. Medium. Retrieved January 5, 2023, from https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843
14. Garg, A. (2021, December 14). *Use Yolo V5 object detection algorithm for Custom Object Detection*. Analytics Vidhya. Retrieved January 5, 2023, from

https://www.analyticsvidhya.com/blog/2021/12/how-to-use-yolo-v5-object-detection-algorithm-for-custom-object-detection-an-example-use-case/

15. Admin. (2021, October 14). *Arduino Uno Board : Features and real time applications*. WatElectronics.com. Retrieved January 6, 2023, from https://www.watelectronics.com/arduino-uno-board-tutorial-and-its-applications/#:~:text=The%20Arduino%20Uno%20board%20is,required%20support%20needed%20for%20microcontroller.
16. Abdul Rahman, Nur & Ibrahim, Noor & Lombigit, Lojius & Azman, Azraf & Jaafar, Zainudin & Abdullah, Nor & Mohamad, Glam. (2018). GSM module for wireless radiation monitoring system via SMS. IOP Conference Series: Materials Science and Engineering. 298. 012040. 10.1088/1757-899X/298/1/012040.
17. *SIM800L GSM module*. Components101. (n.d.). Retrieved January 6, 2023, from https://components101.com/wireless/sim800l-gsm-module-pinout-datasheet-equivalent-circuit-specs?fbclid=IwAR1fx5S6pZqqQR6uJ0OXmNGTHmXdhgEX_nPSpzSd5aubDxdw7zEfxSjOJRs
18. *Arduino Uno R3 (China)*. Tech Bazar || টেক বাজার. (2022, November 30). Retrieved January 6, 2023, from https://www.techbazar.com.bd/product/arduino-uno-r3-china/
19. *SIM800L GSM module*. Components101. (n.d.). Retrieved January 6, 2023, from https://components101.com/wireless/sim800l-gsm-module-pinout-datasheet-equivalent-circuit-specs
20. Ultralytics. (n.d.). *Ultralytics/yolov5: Yolov5 🚀 in PyTorch > ONNX > CoreML > TFLite*. GitHub. Retrieved January 6, 2023, from https://github.com/ultralytics/yolov5

LIST OF FIGURES

Figure 3.2.1.1: Image Labelling

Figure 3.2.2.1: Combination of Residual blocks, Bounding box regression & IOU

Figure 3.2.2.2: Dataset Structure for YOLOv5

Figure 3.2.3.1: YOLOv5 GitHub Repository

Figure 3.3.1.1: Arduino Uno

Figure 3.3.2.1: GSM SIM800L

Figure 4.2.1: System Diagram

Figure 4.2.1.1: Dataset Training

Figure 4.2.1.2: Hardware Configuration

Figure 4.3.1: Accuracy Graph

Figure 4.3.2: Precision & Recall graph

Figure 4.3.3: Ambulance Detected

Figure 4.3.4: Fire Truck Detected

Figure 4.3.5: Fire Truck Detected from Real-time Video

Figure 4.3.6: Ambulance Detected from Real-time Video

Figure 4.3.7: detect.py file

Figure 4.3.8: Arduino Code

Figure 4.3.9: Hardware Configuration

Figure 4.3.10: ‘fire truck Detected’ SMS Delivered

Figure 4.3.11: ‘Ambulance Detected’ SMS Delivered

LIST OF TABLES

Table 2.1.1: Works, Algorithms, Components & Accuracy of the Related Works