VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi-590018

A Project Report

on

"INTELLIGENT AMBULACE USING IOT"

A Project report submitted in partial fulfillment of the requirements for the award of the degree of ***Bachelor of Engineering in Electronics and Communication Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by

BHUMIKA P (1DT19EC010)
DARSHAN M GOWDA (1DT19EC016)
ISHA V (1DT19EC027)
M PHANI TEJA (1DT19EC043)

Under the Guidance of

Mrs. Kalpavi C Y
Assistant Professor, Dept. of ECE

Department Of Electronics And Communication Engineering
DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT
Accredited by NBA, New Delhi, Accredited by NAAC with Grade A+
Udayapura, Kanakapura Road, Bengaluru-560082

2022-2023

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi-590018

A Project Report

on

"INTELLIGENT AMBULANCE USING IOT"

A Project report submitted in partial fulfillment of the requirements for the award of the degree of
***Bachelor of Engineering in Electronics and Communication Engineering** of*
Visvesvaraya Technological University, Belagavi
Submitted by

BHUMIKA P(1DT19EC010)
DARSHAN M GOWDA (1DT19EC016)
ISHA V (1DT19EC027)
M PHANI TEJA (1DT19EC043)

Under the Guidance of

Mrs. Kalpavi C Y
(Assistant Professor, Dept. of ECE)

Department Of Electronics And Communication Engineering
DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT
Accredited by NBA, New Delhi, Accredited by NAAC with Grade A+
Udayapura, Kanakapura Road, Bengaluru-560082
2022-2023

Signature of the Guide	**Signature of the Coordinator**	**Signature of the HoD**
Mrs.Kalpavi C Y	**Dr. Siddaligappagouda Biradar**	**Dr. Mallikarjun P Y**

H.O.D. Department of Electronics & Communicat
Dayananda Sagar Academy of Technology & Manage
Udayapur, Opp: Shri Ravishankar Ashra
Kanakapura Road, Bangalore - 560 08

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Accredited by NBA, New Delhi, Accredited by NAAC with Grade A+

Udayapura, Kanakapura Road, Bengaluru-560082

2022-2023

Department Of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project work entitled **"INTELLIGENT AMBULANCE USING IOT"** is carried out by **BHUMIKA P (1DT19EC010), DARSHAN M GOWDA (1DT19EC016), ISHA V (1DT19EC027) and M PHANI TEJA (1DT19EC043),** bonafide students of **DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT** in partial fulfilment for the award of the degree of **Bachelor of Engineering in Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year **2022-2023**. It is certified that all corrections/ suggestions indicated for internalassessment have been incorporated in the report deposited the departmental library. The project report has been approved as it satisfies the academic requirements with respect of project work prescribed for the award of Bachelor of Engineering Degree.

Signature of the guide	Signature of the HOD	22/5/23 Signature of the Principal
Mrs.Kalpavi C Y	**Dr. Mallikarjun P Y**	**Dr. Ravishankar M.**

H.O.D. Department of Electronics & Communication
Dayananda Sagar Academy of Technology & Management
Udayapur, Opp: Shri Ravishankar Ashram,
Kanakapura Road, Bangalore - 560 082.

External Viva

Sl. No.	Name of the Examiner	Signature with date
1.	Prof. P. Venugopal	
2.	Dr MALLIKARJUN P.Y DSATM, Blre	23/5/23

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to everyone who helped and guided us in completing this project work.

We are grateful to **Dr. M. Ravishankar, Principal of DSATM, Bengaluru** for having encouraged us in our academic endeavors.

We are thankful to **Dr. Mallikarjun P Y, Professor and Head of Department of Electronics and Communication Engineering** for encouraging us to aim higher.

We would like to express our gratitude to our project coordinator **Mrs.Kalpavi C Y**, **Assisstant Professor in Department of Electronics and Communication Engineering** for constant motivation, support and guidance.

We are also thankful to all faculty members of the Department of Electronics and Communication Engineering for their assistance and encouragement.

BHUMIKA P (1DT19EC010)
DARSHAN M GOWDA (1DT19EC016)
ISHA V (1DT19EC027)
M PHANI TEJA(1DT19EC43)

ABSTRACT

Traffic congestion is the primary cause of patient deaths when travelling to hospitals. Any number of factors, including accidents, road construction, a high number of pedestrians crossing the street, etc., can contribute to traffic congestion. In particular, in large cities like Bangalore, Delhi, Hyderabad, etc. People in these cities lead busy lives with little time for themselves and don't even consider making room for emergency vehicles like ambulances. This article offers a "Intelligent Ambulance using IOT" approach to address this issue as well as many others. This paper describes the use of a variety of technologies, including GPS for location, digital image processing for ambulance detection, automatically turning on signal lights when an ambulance is approaching, and monitoring patient health. It also describes the use of nodemcu microcontroller and ChatBot to telegram critical patient data, such as body temperature and heart rate, to the closest hospitals. By transporting patients to hospitals more quickly than is customary, the proposed system is employed to save lives. In addition, the patient monitoring system can use the DHT11 temperature and heart rate sensors to check on the victim's health and report the results to the hospital.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

DIP- digital Image Processing

CNN -Convolutional Neural Networks

YOLO- You Only Look Once

LCD-Liquid Crystal Display

LED-Light Emitting Diode

GPS-Global Positioning System

HMM-Hidden Markov Model

SVM-Support Vector Machine

CHAPTER I

1. INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

India being one amongst the foremost inhabited countries within the world, road traffic congestion and delayed provision of medical attention are the critical issues for loss of life. According to many statistics, each minute a life is lost due to late responses by family members, hospital formalities or unavoidable situations such as traffic congestion. Saving a life is precious to both the patient's family and even the hospital's fame.

Due to the increase in population and busy lifestyle of people they are not able to provide a way for the ambulance. Too many vehicles at the intersection cause traffic jams and heavy traffic. Conventional traffic control (warning signs, stop signs, etc.) is used in many places, but sometimes it is not enough to solve the problem. The main aim of this project is to make sure that the patients reach the hospital as soon as possible. Instead of using the road to accommodate increased traffic, various methods have been designed to control traffic on the road. Therefore, reducing the waiting time at traffic lights can save our society billions of rupees annually.

We leverage emerging technologies such as networking and image processing to make traffic lights smarter by managing traffic demand at each network intersection. The aim is to prevent traffic congestion and reduce the wait time. The system will initially measure the traffic density at various signals and accordingly adjust the traffic light signal delay on the upstream side so that the signal stays green longer. Second, it will also communicate with adjacent junction signals. Both signals will control the traffic together depending on the density.

The traffic lane is called a node. When a node in control with a green traffic signal for an ambulance passes without waiting for an ambulance, it is called ON STATE, because all other nodes are in OFF STATE. This project describes a solution for automatic traffic signal light changing and to send patient's vital information to nearest hospitals so that they can keep the required equipment ready before the patient reaches the hospital.

The proposed system is used to save the life of patients by taking them to hospital in a faster way than the usual traditional way. Earlier the ambulance had to wait for a longer time till the signal gives green light which has no technology attached to it, in this system a set of IR sensors and sound sensors are placed on four lanes of the road connected by traffic signal. These sensors are used to calculate the traffic density on the roads and also the presence of ambulances on the lane, if any ambulances are detected the traffic signal in that lane turns green.

Fig 1.1: Signs of Each Alphabet

Irrespective of which light is should display, and simultaneously the signal on the opposite lane turns red. The presence of heartbeat rate sensor and body temperature sensor will sense the vital details of the patient to the hospital by using nodemcu. The proposed project i.e.," Intelligent ambulance using IOT" helps society to save the life at "golden hour".

1.2 MOTIVATION

The motivation behind the development of a smart ambulance system utilizing Node MCU, GPS, IoT health monitoring, digital image processing for vehicle density, traffic control, and the YOLO v5 algorithm is driven by the urgent need to revolutionize emergency medical services. This integrated approach aims to address critical challenges faced by emergency vehicles, enhance patient care, and improve overall response efficiency.

Firstly, the integration of Node MCU, GPS, and IoT health monitoring technologies offers a holistic solution for real-time monitoring and communication. By equipping ambulances with Node MCU devices, vital health parameters of patients can be continuously monitored during transit. This information can be transmitted securely through IoT connectivity to healthcare professionals in hospitals, enabling them to receive real-time updates and make informed decisions prior to the patient's arrival. This timely communication can significantly improve patient care and increase the chances of positive medical outcomes.

Secondly, incorporating digital image processing for vehicle density and traffic control enables the ambulance system to navigate through congested road networks efficiently. By analysing images captured by strategically placed cameras, the system can detect vehicle density and traffic patterns in real-time. This information can be used to optimize the ambulance's route, identify congestion hotspots, and dynamically adjust traffic signals to create a clear path for emergency vehicles. Such traffic control measures can substantially reduce response times, ensuring that ambulances reach their destinations quickly and safely.

Moreover, the integration of the YOLOv5 algorithm, a state-of-the-art object detection model, enhances the system's ability to identify obstacles and potential hazards on the road. By accurately and efficiently detecting objects in real-time, the algorithm assists the ambulance driver in navigating safely and avoiding collisions. This feature ensures the safety of both the ambulance crew and other road users, contributing to an overall improved emergency response process.

1.3 BLOCK DIAGRAM

Digital image processing (DIP) has been widely applied in various fields such as medical imaging, surveillance, object recognition, and traffic management. In this context, DIP can play a vital role in traffic management by processing images from traffic cameras to detect, track, and analyse vehicles and pedestrians. One of the most advanced object identification algorithms available today is YOLO (You Only Look Once), and its latest version, YOLOv5, is a state-of-the-art convolutional neural network (CNN) that can detect objects in real-time with high accuracy.

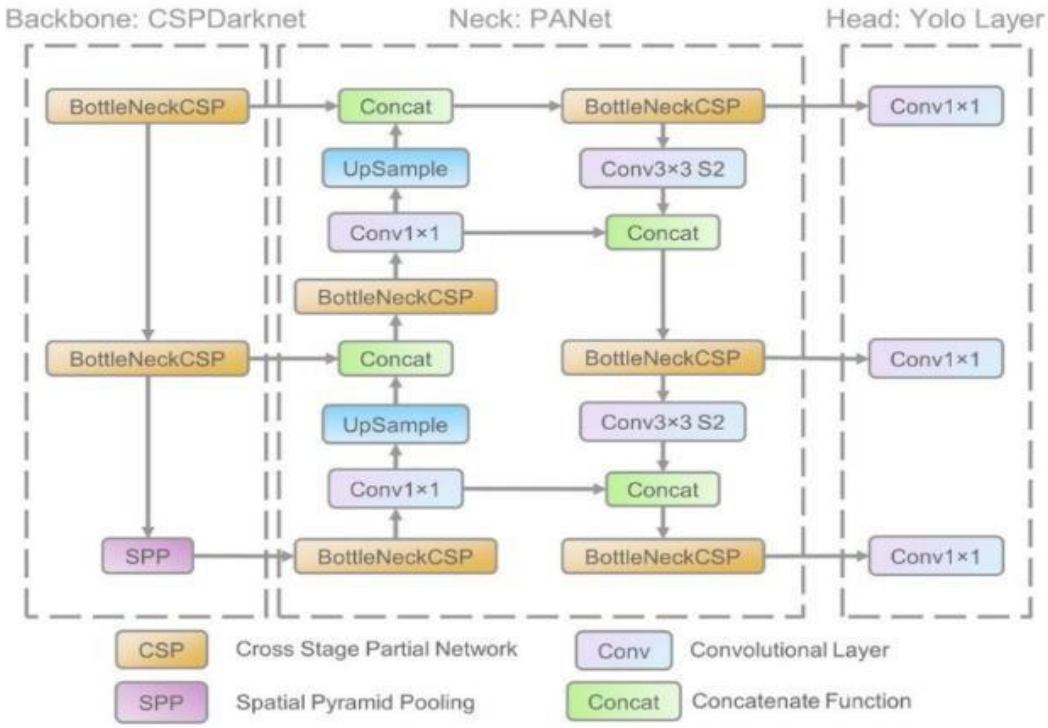


Fig 1.2: Block Diagram of Node-MCU

The use of YOLOv5 in traffic management involves the following steps:

1. Image Acquisition: Traffic cameras are used to capture images of the traffic scene.
2. Image Pre-processing: The acquired images are pre-processed to enhance the quality of the image, remove noise, and correct for distortion.
3. Object Detection: YOLOv5 is applied to detect objects of interest in the image. The algorithm processes the entire image in one go and identifies the location and class of each object present in the image. The object classes include vehicles, pedestrians, cyclists, and traffic signs.
4. Tracking: After detecting the objects, the algorithm tracks them across frames to determine their trajectories and speeds. This information is useful in analysing traffic flow and detecting congestion.
5. Analysis: The tracked objects are analysed to extract meaningful information such as vehicle count, average speed, and occupancy rate. This information can be used to optimize traffic flow and control traffic lights.
6. Communication: The analysed information is communicated to the traffic control center, which can use it to make real-time decisions on traffic management.

1.5 APPLICATIONS

The smart ambulance system incorporating Node MCU, GPS, IoT, health monitoring, digital image processing, vehicle density detection, traffic control, and the YOLOv5 algorithm has a range of applications that can revolutionize emergency medical services and enhance patient care. Some of the key applications of this integrated system include:

1. Rapid and Accurate Emergency Response: The system enables ambulances to respond quickly and efficiently to emergency calls. GPS navigation and real-time communication capabilities ensure that ambulances can reach the scene promptly, even in unfamiliar areas. The YOLOv5 algorithm assists in identifying obstacles and potential hazards on the road, allowing for safer and faster navigation.
2. Remote Patient Monitoring: IoT-enabled sensors and health monitoring capabilities facilitate continuous monitoring of patients' vital signs during transit. Real-time data can be transmitted to healthcare professionals, enabling remote monitoring and early detection of critical changes in the patient's condition. This application improves patient outcomes and allows for timely interventions.
3. Traffic Management and Optimization: Digital image processing and vehicle density detection help identify traffic congestion and optimize the ambulance's route in real-time. By clearing the path and controlling traffic flow, the system reduces response times and ensures that ambulances can reach their destinations without delays caused by traffic congestion.
4. Communication and Coordination: The integration of Node MCU and IoT technology enables seamless communication between ambulance drivers, hospitals, and emergency responders. Real-time updates on the patient's condition can be transmitted, allowing healthcare professionals to prepare necessary resources and interventions in advance. This application enhances coordination and improves the overall effectiveness of emergency medical services.
5. Improved Safety and Collision Avoidance: The YOLOv5 algorithm assists in real-time object detection, helping ambulance drivers navigate safely and avoid collisions. By accurately identifying obstacles and hazards on the road, the system enhances the safety of both the ambulance crew and other road users.

6. Data Analysis and Performance Optimization: The system generates a wealth of data that can be analysed to identify trends, optimize resource allocation, and enhance overall performance. Insights gained from data analysis can contribute to process improvements, cost reductions, and better resource utilization in emergency medical services.

Overall, the applications of the smart ambulance system are far-reaching, encompassing rapid emergency response, remote patient monitoring, traffic management, communication, safety enhancement, and data-driven performance optimization. By leveraging these integrated technologies, emergency medical services can be transformed, leading to improved patient outcomes and more efficient and effective emergency responses.

CHAPTER II

LITERATURE SURVEY

CHAPTER 2

LITERATURE REVIEW

2.1 REVIEW OF INTELLIGENT AMBULANCE USING IOT

These are the literature surveys on different sign language models developed over the years.

In paper [1], we have considered the analysis of finding a shortest path and giving the best route to an ambulance to reach the particular hospital. This paper summarizes the creation of GPS System for which the GPS tracker is built and it is set up with the ambulance to track the path. This technology can be used by everyone even a common man can track all the ambulance in his surroundings. This paper also provided the proper utilization of GPS technology and in case of any emergency this technology will be very useful in finding the nearby ambulances and reach the patient's address of the accident spot. This GPS technology in ambulance is developed to overcome the current difficulty which is faced by many ambulance drivers. Hence this technology uses a shortest path establishing algorithm to reach the destination hospital in minimum amount of time. The overall architecture of this system consists of two sides. One is the ambulance required one's side and the other one is ambulance driver's side. The patient's side is just to track the address of the available ambulances with of a GPS sensor. Then the ambulance driver can check with the exact location of the person who has booked for the ambulance. If the driver is busy or cannot make to reach at that time then he may cancel the users request for ambulance, user will get notified about the non-availability of ambulance but he still gets many lists of ambulance which are nearby. Meanwhile he will also be getting information about the nearest hospitals.

The paper[2] puts up which mainly depends on the very low cost and accurate and also real time traffic controlling system. This system mainly focuses on detecting and monitoring the traffic density and overall traffic volume through the help of infrared sensors (IR Sensors), and changes the signal lights and also timing of the slots. From this paper we have considered the usage of IR sensors for monitoring the traffic and controlling the traffic signals. In this paper [2], we observed the usage of mainly three IR sensors that is IR1, IR2, IR3 one is for Low traffic density, and the other two are for medium and High traffic density respectively. Through this knowledge we have taken in consideration of using a total of four IR sensors for the better results and also to avoid any collisions with the signal. According to this paper, when IR1 sensor is sensed, the LCD Display shows Low traffic density and the signal turns to green light and it will be turned ON for 20 seconds of time. When IR2 sensor is sensed, the LCD Display shows medium traffic density and turns ON green light which will be again ON for 50 seconds of time. When IR3 sensor is detected, the LCD Display shows High traffic density and green light will be kept ON for 60 seconds of time. So, when an ambulance reaches the junction, the IR sensors gets detected and the same sensor will turn that junctions signal to green and other to red, the signal will be turned ON to green for next 60 seconds until the ambulance passes to next junction or the next signal.

In paper [3] Saah H. N et a, proposes the method of implementing a health monitoring system in an ambulance, which helps to have an update about the patient’s health conditions before reaching the particular hospital. The health monitoring system includes the body temperature, patient’s heart beat, pulse sensors etc... In this paper [3], it helped about the usage of various health monitoring sensors for detecting these health-related conditions of the patient in the ambulance while reaching the hospital. We also see the usage of temperature sensor which is used to identify the temperature in blood vessels and also to find out the output of cardiac. In this paper we see that IC lm35 sensor is used to know the body’s temperature. In this paper, we have observed the usage of the new model of microcontroller for developing health monitoring system inside the ambulance. To get the health conditions of the patient, these sensors are connected to the microcontroller. When compared to old models, the speed and performance of this was better. And also compared to other microcontrollers this consumes less power and runs without producing any noise. Through this paper we got an idea about usage of microcontroller for implementing the health monitoring system inside the ambulance.

In Paper [5] The system focusses on providing the ambulance a free way to an emergency unit on the way to a hospital ambulance or to the accident spot when struck in traffic jams or not. So, to overcome this problem researchers of this paper [5], proposed an adaptive way where the ambulance informs the closest traffic signals about their arrival through IR led and the ambulance buzzer. This system tries to develop an algorithm, considering the current and the destination location, in deciding the path that the ambulance must take. Helps also in routing the traffic based on traffic density, by establishing a network between the ambulance and the signal posts. Further in this paper, the health condition of the patient is continuously monitored is discovered by the emergency unit is performed by various sensors on the ambulance. Based on the described parameters the availability of the medical requirement to treat the patient is judged. The various parameters like Heart Rate and Body Temperature are monitored. This data is sent to hospital for further analysis. Proper treatment may not be available if the doctors do not the sufficient medical history of the patient. This system ties in obviating just that. The entire system is enhanced to develop the facility for proving the patient first aid at the earliest in an emergency situation.

In Paper [5], At present, there is no automation system for controlling the traffic signals at emergencies. This paper [5] proposed a basic architectural model of intelligent ambulance by enhancing its features by adding GPS module to the traffic control system which was successfully proposed previously. This model is a complete automation system where the driver has to switch on to the emergency mode through the created mobile application. This activation will help the app track the live location of the ambulance and also help in tracking the nearest traffic control signals with the help of GPS. The destination location is pinned which helps in understanding the best route available to reach at the earliest. Whenever the ambulance approaches in the 500m of the traffic signal, it immediately sends request to control the traffic lights of which the ambulance will have to cross and again send a signal to set it back. When the signal is already green, based on the timing the app automatically requests control to the nearest control signal. If the signal is red, similarly depending on the remaining time, the app requests a control to open that particular the ambulance is travelling by calculating the distance. This helps the ambulance to move freely without getting struck in the traffic. Considering the direction of the vehicle the server toggles the signal. In real time, the system is improved so, the death rate can be reduced during emergency situations. Hence in this paper we come across an idea to control signal from the driver’s smartphone itself.

In Paper [6], A lot of advantages that IoT application offers within the aid sector is most categorized into tracking of patients, staffs, authenticating people and also gathering information and sensing. Paper [6] presents an architectural model consisting of three different modules. The ambulance module is placed inside the ambulance. It consists of microcontroller with various sensor units interfacing to it. The different health parameters of the patient’s health are monitored including heartbeat, temperature, blood glucose levels and various others. After processing these statistics, it can be uploaded into the cloud. The second module is the cloud or WIFI module, which consists of two cloud services. The Think Speak cloud platform supports numerical data to be uploaded. The image uploading is done on Dropboxcloud. The third module is the hospital module which helps the doctor monitor the patients' health condition based on the data which is uploaded into the cloud. This module also consists a simple application to download the data from the cloud. If these real-time data can be made available for the access to the doctor, it can help him to give adequate feedback to the emergency vehicle as well as arrange hospital facilities for the patient.

In paper [7] the author puts forward a paper which mainly emphasizes on. There is conjointly projected phone health observation system for the elderly that works with smart phones and wireless sensors to check their health. The elderly person's smart phone can send an alert to the people who have been pre-assigned in case of an emergency so they can prepare for a machine. This technology provides a platform for distant family members and the elderly person's health state to communicate. The advertisement promises a tool that may display a wide range of physical characteristics of different patient bodies. The base station receives information from the organizer cluster point through the wireless sensors, which emit signals that are concentrated on the patient's skin. In addition to mere measurement the physical attributes the device may also detect irregular activity from them and notify the doctor accordingly. With the help of HMS, the patient will be admitted via an automatic procedure more often. The HMS is designed with a display, bio-medical sensors, and a 1x5 keypad; all of them are not intended to form a group of the patient's critical data. The projected system uses a variety of sensors, including pulse and vital sign sensors as well as fingerprint and vital sign sensors. For displaying the selected mode, an OLED is used. Patient may not be conscious when placed inside the car teaming forces with a relative. Due to the patient's inability to access their vital information, there may be an urgent circumstance when the patient is also not awake. When transporting the patient, the care in the automobile makes a judgement call regarding the patient's current status and selects one of the earlier modes. In order to provide the care confidence in the mode that was selected, the mode is presented on the screen.

In [8] the paper which mainly emphasizes on conjointly a projected phone safety observation system for elder that are experiencing with wireless body sensors, sensible mobiles and observation of the health of elderly. If there is any emergency the sensible mobile with elder human being can provide associate awake to the folks that are preassigned in order that they'll organize for associate machine. this technique creates a platform for family members to communicate on the health of the elderly person regardless of distance. The advertisement suggests a device that is capable of monitoring multiple physical characteristics of various patient bodies. Through an organizer node that communicates with a base station, the signals from the wireless sensors are gathered on the patient's body. In addition to just measuring the physical parameters, the device may also detect problems and alert the physician accordingly. The process continues with the help of the HMS as of right now because the patient has been admitted into the car. This HMS is equipped with bio-medical sensors, a display, and a 1x5 membrane keypad; all of them are intended to gather the patient's vital information. A few of the sensors used in the proposed system include pulse and vital sign sensors as well as fingerprint and vital sign readers. The selected mode is displayed using a degree OLED. Once the patient is in the car, a family member or guardian may or may not be present to assist. There may be emergency situations in which the patient is also unconscious, rendering vital information about the patient unavailable are going to be accessed. The HMS employs a fingerprint sensor component to overcome this problem. After assessing the patient, the helper inside the car determines that the patient is in an emergency situation and selects one of the several options. The selected mode is displayed on the OLED display to give the helper confidence in the selected option.

Paper[9] suggests the chiefly emphasizes on “IoT primarily based auto deployment” technique to manage auto and emergency services. This analysis is economical to hide all the items required to develop a sensible auto management framework however lacks to elucidate however the system will add real time with a mixture of mobile computing, cloud computing and standalone application along. On call-based machine management technique to manage machine and emergency services. This analysis is economical to hide all the items required to develop a sensible machine management framework however lacks to elucidate however the system will add real time with a mix of mobile computing, cloud computing and standalone application along. Remote medical services mistreatment cloud” technique to manage machine and emergency services. This analysis is economical to hide all the items required to develop a sensible machine management framework however lacks to elucidate however the system will add real time with a mix of mobile computing, cloud computing and standalone application along. advance a paper that principally emphasizes on ambulance service supplier mistreatment automaton application technique to manage machine and emergency services. This analysis is economical to hide all the items required to develop a sensible machine management framework however lacks to elucidate however the system will add real time with a mix of mobile computing, cloud computing and standalone application along.

In [10] a discovery mechanism for V2V communication facilitated by traffic jams on the road is suggested. In order to evaluate the severity of the initial traffic jam, a fuzzy regulator is initially constructed with vehicle speed and business viscosity as the input and the original traffic position as the affair. Additionally, the position of indigenous business jam is accomplished supported by the massive sub-sample thesis test as the amount of original business jam of bordering cars is collected supported by V2V communication. To maximize the efficacy of business jam discovery findings, the business jam information is promptly communicated to the affected cars when it has been obtained. Internet of Vehicles (IoVs) communication technology can significantly improve the transmission of traffic information, whereas traditional technologies like electronic information boards, FM radios, and SMS cannot guarantee the effective dispersion of traffic information. have suggested using IOT to automate gestures and discover business viscosity.

CHAPTER III

PROBLEM STATEMENT

CHAPTER 3

PROBLEM STATEMENT

The problem statement addressed by the proposed smart ambulance system using Node MCU, GPS, IoT, health monitoring, digital image processing, vehicle density detection, traffic control, and the YOLOv5 algorithm is the need to improve emergency medical services and enhance patient care in critical situations. The current emergency response process faces several challenges that contribute to longer response times and potential delays in providing timely medical attention to patients. One of the key issues is the lack of efficient navigation systems in ambulances. Without proper guidance, drivers may face difficulties in locating patients' addresses, particularly in unfamiliar areas, leading to delays in reaching the destination. Additionally, traffic congestion poses a significant obstacle to emergency vehicles, further extending response times and potentially jeopardizing patient outcomes. Communication between the ambulance crew and hospital staff is another area that needs improvement. The current lack of a direct and seamless communication channel hinders the transmission of real-time updates on the patient's condition. This lack of information may result in unprepared hospital staff upon the patient's arrival, leading to delays in providing appropriate medical care. Furthermore, the absence of health monitoring capabilities during ambulance transit restricts the availability of vital information about the patient's health status. Without continuous monitoring, critical changes in the patient's condition may go unnoticed, and timely interventions may not be initiated. This limitation can negatively impact patient outcomes and potentially result in avoidable complications.

CHAPTER IV

OBJECTIVES

CHAPTER 4
OBJECTIVES

The three main objectives in order to overcome the challenges faced by the emergency vehicles like ambulance to reduce the response time and to give timely medical attention. The key objectives are listed as below:

1. Implementing a GPS system in emergency vehicles: By installing GPS systems in ambulances, drivers can quickly and efficiently navigate through unfamiliar areas, locate patients' addresses, and find the fastest route to the hospital. This objective can significantly reduce response times and improve patient outcomes.

2. Creating a communication channel between hospital staff and ambulance drivers: By establishing a channel of communication between hospital staff and ambulance drivers, medical staff will be able to get real-time updates on the patient's health and be better prepared for the patient's arrival. This goal can shorten the time it takes for patients to obtain medical attention and increase the emergency medical services' general effectiveness.

3. Establishing traffic free path to reach the hospital: The difficulties faced by emergency vehicles when navigating through traffic congestion can be addressed using digital image processing (DIP) as a tool to develop a traffic management system. Response times can be shortened and emergency medical service effectiveness increased by utilising DIP to identify and track emergency vehicles, monitor and control traffic, forecast traffic congestion, and automatically detect occurrences. To identify and track traffic flow, DIP can process photos from cameras placed at various points on roads.

CHAPTER V

HARDWARE AND SOFTWARE

CHAPTER 5

HARDWARE AND SOFTWARE REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

- Lcd
- Buzzer
- Heartbeat Sensor
- Temperature Sensor
- Node MCU
- Powers supply

5.1.1 Regulated power supply:

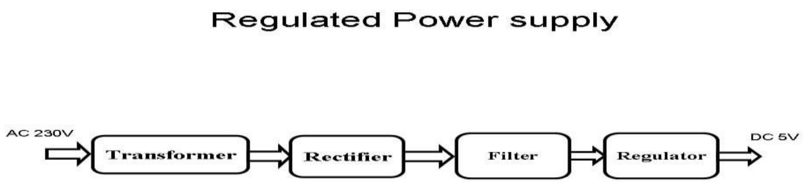


Fig5.1 Regulated power supply

Transformer**:**

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled conductors without changing its frequency. A varying current in the first or primary winding creates a varying magnetic flux in the transformer's core, and thus a varying magnetic field through the secondary winding. This varying magnetic field induces a varying electromotive force (EMF) or "voltage" in the secondary winding. This effect is called mutual induction. If a load is connected to the secondary, an electric current will flow in the secondary winding and electrical energy will be transferred from the primary circuit through the transformer to the load. This field is made up from lines of force and has the same shape as a bar magnet. If the current is increased, the lines of force move outwards from the coil. If the current is reduced, the lines of force move inwards. If another coil is placed adjacent to the first coil then, as the field moves out or in, the moving lines of force will "cut" the turns of the second coil. As it does this, a voltage is induced in the second coil. With the 50 Hz AC mains supply, this will happen 50 times a second. This is called MUTUAL INDUCTION and forms the basis of the transformer.

5.1.2 Temperature Sensor:

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 does not require any external calibration or trimming to provide typical accuracies.

Fig 5.2: Temperature sensor

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄5°Cat room temperature and ±3⁄5°C over a full −55 to +150°Ctemperature range. Low cost is assured by trimming and calibration at the wafer level.

The LM35's low output impedance ,linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −50° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-56 transistor packages, while theLM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and aplastic TO-220 package.

FEATURES

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guarantee able (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Less than 60µA current drain
- Operates from 5 to 30 volts
- Low self-heating, 0.08°C in still air
- Low impedance output, 0.1 W for 1 mA load

Circuit Diagram

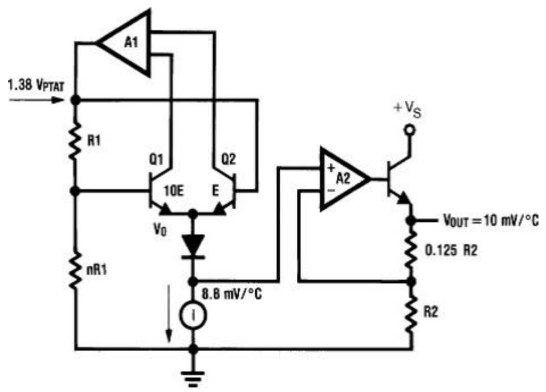


Fig 5.3: Circuit diagram of temperature sensor

Applications:

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01§C of the surface temperature. This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an inter- mediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature. To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-56 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the Vb terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and vanishes such as Humiseal and epoxy paints or dips are often used to in sure that moisture cannot corrode the LM35 or its connections. These devices are sometimes soldered to a small light- weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

5.1.3 Easy Pulse Sensor:

The new version uses the **TCRT1000** reflective optical sensor for photoplethysmography. The use of TCRT100 simplifies the build process of the sensor part of the project as both the infrared light emitter diode and the detector are arranged side by side in a leaded package, thus blocking the surrounding ambient light, which could otherwise affect the sensor performance. I have also designed a printed circuit board for it, which carries both sensor and signal conditioning unit. and its output is a digital pulse which is synchronous with the heart beat.

The output pulse can be fed to either an ADC channel or a digital input pin of a microcontroller for further processing and retrieving the heart rate in beats per minute (BPM).

Theory

This project is based on the principle of photoplethysmography (PPG) which is a non-invasive method of measuring the variation in blood volume in tissues using a light source and a detector. Since the change in blood volume is synchronous to the heart beat, this technique can be used to calculate the heart rate. Transmittance and reflectance are two basic types of photoplethysmography. For the transmittance PPG, a light source is emitted in to the tissue and a light detector is placed in the opposite side of the tissue to measure the resultant light. Because of the limited penetration depth of the light through organ tissue, the transmittance PPG is applicable to a restricted body part, such as the finger or the ear lobe. However, in the reflectance PPG, the light source and the light detector are both placed on the same side of a body part. The light is emitted into the tissue and the reflected light is measured by the detector. As the light doesn't have to penetrate the body, the reflectance PPG can be applied to any parts of human body. In either case, the detected light reflected from or transmitted through the body part will fluctuate according to the pulsatile blood flow caused by the beating of the heart.

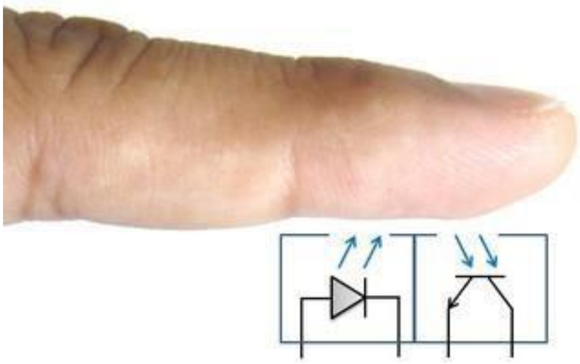


Fig 5.4: Pulse sensor

The PPG signal has two components, frequently referred to as AC and DC. The AC component is mainly caused by pulsatile changes in arterial blood volume, which is synchronous with the heart beat. So, the AC component can be used as a source of heart rate information. This AC component is superimposed onto a large DC component that relates to the tissues and to the average blood volume. The sensor used in this project is TCRT1000, which is a reflective optical sensor with both the infrared light emitter and phototransistor placed side by side and are enclosed inside a leaded package so that there is minimum effect of surrounding visible light. The circuit diagram below shows the external biasing circuit for the TCRT1000 sensor. Pulling the Enable pin high will turn the IR emitter LED on and activate the sensor. A fingertip placed over the sensor will act as a reflector of the incident light.

The output (VSENSOR) from the sensor is a periodic physiological waveform attributed to small variations in the reflected IR light which is caused by the pulsatile tissue blood volume inside the finger. The waveform is, therefore, synchronous with the heart beat. The following circuit diagram describes the first stage of the signal conditioning which will suppress the large DC component and boost the weak pulsatile AC component, which carries the required information. In the circuit shown above, the sensor output is first passed through a RC high-pass filter (HPF) to get rid of the DC component. The cut-off frequency of the HPF is set to 0.7 Hz. Next stage is an active low-pass filter (LPF) that is made of an Op-Amp circuit. The gain and the cut-off frequency of the LPF are set to 101 and 2.35 Hz, respectively. Thus the combination of the HPF and LPF helps to remove unwanted DC signal and high frequency noise including 60 Hz (50 Hz in some countries) mains interference, while amplifying the low amplitude pulse signal .The output from the first signal conditioning stage goes to a similar HPF/LPF combination for further filtering and amplification (shown below). So, the total voltage gain achieved from the two cascaded stages is 101*101 = 10201. The two stages of filtering and amplification converts the input PPG signals to near TTL pulses and they are synchronous with the heart beat.

5.1.4 Jumper Wires

Fig5.5 . Jumper wires

A **jump wire** (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

5.1.5 Node MCU

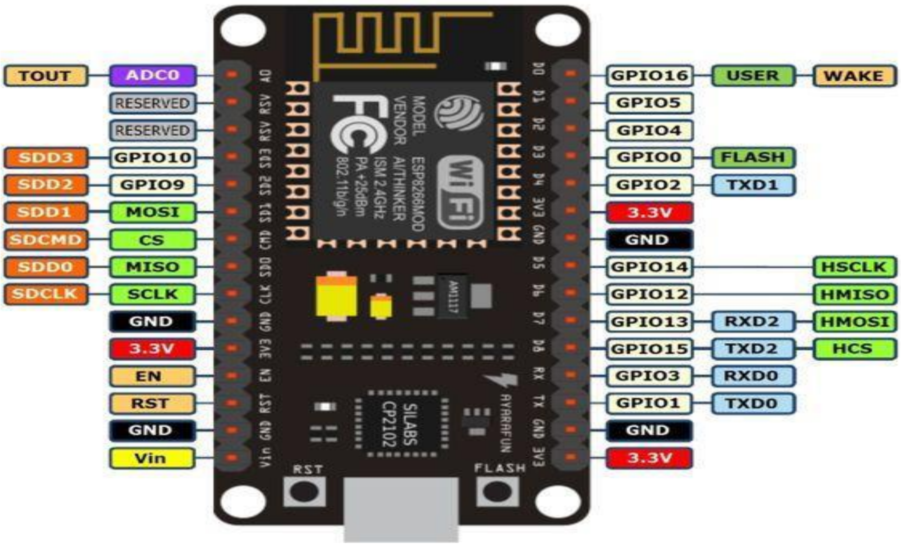


Fig 5.6: Node MCU

The ESP8266 Node MCU has 16 GPIO pins and one analog input pin shown in the image bellow. However only 10 of these GPIO pins can be used for digital input and output operations. These are listed on the table below.

In the ESP8266 firmware for the Arduino IDE pin numbers are defined as follows:

Pin Name on the Board	Function	Pin Number in Arduino IDE	Alias Name in Arduino IDE
D3	GPIO 0	0	D3
TX	GPIO 1	1	D10
D5	GPIO 2	2	D5
RX	GPIO 3	3	D9
D2	GPIO 5	5	D2
D1	GPIO 5	5	D1

D6	GPIO 12	12	D6
D7	GPIO 13	13	D7
D5	GPIO 15	15	D5
D8	GPIO 15	15	D8
D0	GPIO 16	16	D0, LED_BUILTIN
A0	ADC0	A0	analog_ip

Table 5.1 pin configuration

Pin Functions

Pin numbers in the Arduino IDE correspond directly to the ESP8266 GPIO pin numbers. **Pin Mode**, **digital Read**, and **digital Write** functions work as usual, so to read GPIO2, call **digital Read(2)** or its alias name **digital Read(D10)**.

At startup, pins are configured as **INPUT**. Digital pins 0—15 can be **INPUT**, **OUTPUT**, or **INPUT_PULLUP.**

Pins may also serve other functions, like Serial, I2C, SPI. These functions are normally activated by the corresponding library. The diagram above shows the pin mapping for the popular ESP8266 Node MCU module.

Pin interrupts are supported through **attach Interrupt**, functions. Interrupts may be attached to any GPIO pin, except GPIO16. Standard Arduino interrupt types are supported: **CHANGE**, **RISING**, **FALLING**.

Reserved Pins

GPIO pins 6—11 are not shown on this diagram because they are used to connect flash memory chip on most modules. Trying to use these pins as IOs will likely cause the program to crash.

Note that some boards and modules (ESP-12ED, NodeMCU 1.0) also break out pins 9 and 11. These may be used as IO if flash chip works in DIO mode (as opposed to QIO, which is the default one).

Vin,3V3,GND

Vin is the NodeMcu's voltage input that is connected to its internal voltage regulator allowing an input voltage range of 5.75V to10V. It will be regulated to 3.3V. Alternatively an external voltage source of 3.3V can be directly connected to the NodeMcu's 3V3 pins. The **3V3** pin can be also a voltage source to other components such as LEDs. **GND** is the common ground of the board.

Analog Input

ESP8266 has a single ADC channel available to users. It may be used either to read voltage at ADC pin, or to read module supply voltage (VCC).

To read external voltage applied to ADC pin, use **analog Read(A0)**. Input voltage range is 0 — 1.0V.

To read VCC voltage, use **Eupeptic()** while the ADC pin must be kept unconnected. Additionally, the following line has to be added to the sketch:

ADC_MODE(ADC_VCC);

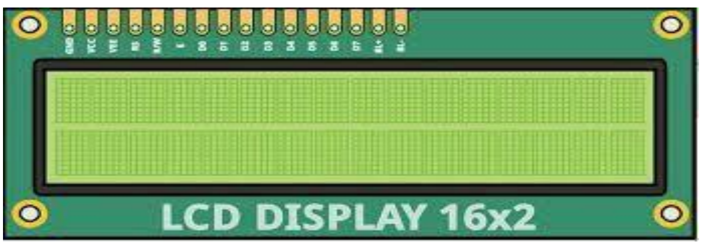
This line has to appear outside of any functions, for instance right after the **#include** lines of your sketch.

Analog Output

Analog Write(pin,value) enables software PWM on the given pin. PWM may be used on pins 0 to 16. Call **analog Write(pin,0)** to disable PWM on the pin. **value** may be in range from 0 to **PWMRANGE**, which is equal to 1023 by default. A value of 0, 512 and 1023 sets the PWM duty cycle to 0%, 50% and 100%, respectively. Optionally, the PWM range may be changed by calling **analog Write Range(new_range)**.

PWM frequency is 1kHz by default. Call **analog Write Freq(new_frequency)** to change the frequency. The unit representation is in [Hz]. Example:

5.1.6 LCD display



5.7:16*2 LCD display

A **liquid-crystal display** (**LCD**) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in color or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

LCD is used in wide range application including computer monitors, televisions, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common importable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big- screen television sets.

Since LCD screens do not use phosphors, they do not suffer image burn-in when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft schedule on an indoor sign). LCDs are, however, susceptible to image persistence.

Interfacing an LCD .

The 16x2 LCD has a total of 16 pins. As shown in the table below, eight of the pins are data lines (pins 7-15), two are for power and ground (pins 1 and 16), three are used to control the operation of LCD (pins 5-6), and one is used to adjust the LCD screen brightness (pin 3). The remaining two pins (15 and 16) power the backlight. The details of the LCD terminals are as follows:

Terminal 1	GND
Terminal 2	+5V
Terminal 3	Mid terminal of potentiometer (for brightness control)

Terminal 5	Register Select (RS)
Terminal 5	Read/Write (RW)
Terminal 6	Enable (EN)
Terminal 7	DB0
Terminal 8	DB1
Terminal 9	DB2
Terminal 10	DB3
Terminal 11	DB5
Terminal 12	DB5
Terminal 13	DB6
Terminal 15	DB7
Terminal 15	+5.2-5V
Terminal 16	GND

Table 5.2. LCD terminals

5.1.7 LCDMODULE

Fig5.8. Output of LCD

The name and functions of each pin of the 16×2 LCD module is given below. Pin1 (Vss): Ground pin of the LCD module.

Pin2 (Vcc): Power to LCD module (+5V supply is given to this pin).

Pin3 (VEE): Contrast adjustment pin. This is done by connecting the ends of a 10K potentiometer to +5V and ground and then connecting the slider pin to the VEE pin. The voltage at the VEE pin defines the contrast. The normal setting is between 0.5 and 0.9V.

Pin5(RS): Register select pin. Logic HIGH at RS pin selects data register and logic LOW at RS pin selects command register. If we make the RS pin HIGH and feed an input to the data lines

(DB0 to DB7), this input will be treated as data to display on LCD screen. If we make the RS pin LOW and feed an input to the data lines, then this will be treated as a command (a command to be written to LCD controller – like positioning cursor or clear screen or scroll).
Pin5 (R/W): Read/Write modes. This pin is used for selecting between read and write modes. Logic HIGH at this pin activates read mode and logic LOW at this pin activates write mode.
Pin6 (E): This pin is meant for enabling the LCD module. A HIGH to LOW signal at this pin will enable the module.
Pin7 (DB0) to Pin15(DB7): These are data pins. The commands and data are fed to the LCD module though these pins.
Pin15 (LED+): Anode of the back light LED. When operated on 5V, a 560 ohm resistor should be connected in series to this pin. In NodeMCU based projects the back light LED can be powered from the 3.3V source on the NodeMCU board.
Pin16 (LED-): Cathode of the back light LED.

RS pin of the LCD module is connected to digital pin 12 of the Arduino. R/W pin of the LCD is grounded. Enable pin of the LCD module is connected to digital pin 11 of the Arduino. This method is very simple, requires less connections and we can almost utilize the full potential of the LCD module. Digital lines DB5, DB5, DB6 and DB7 are interfaced to digital pins 5, 5, 3 and 2 of the NodeMCU. The 10K potentiometer is used for adjusting the contrast of the display. The NodeMCU can be powered through the external power jack provided on the board. +5V required in some other parts of the circuit can be tapped from the 5V source on the Arduino board. The Arduino can be also powered from the PC through the USB port.

5.1.8 TELEGRAM BOTS

Bots: An introduction for developers

Bots are third-party applications that run inside Telegram. Users can interact with bots by sending them messages, commands and inline requests. You control your bots using HTTPS requests to our Bot API.

What can I do with bots?

A chat with @TechCrunchBot also showing search results from the @gif inline-bot

- To name just a few things, you could use bots to:
- Get customized notifications and news. A bot can act as a smart newspaper, sending you relevant content as soon as it's published.

- Integrate with other services. A bot can enrich Telegram chats with content from external services.
- Gmail Bot, GIF bot, IMDB bot, Wiki bot, Music bot, you tube bot, GitHub bot
- Accept payments from Telegram users. A bot can offer paid services or work as a virtual storefront.
- Demo Shop Bot
- Create custom tools. A bot may provide you with alerts, weather forecasts, translations, formatting or other services.
- Markdown bot, Sticker bot, Vote bot, Like bot
- Build single- and multiplayer games. A bot can offer rich HTML5 experiences, from simple arcades and puzzles to 3D-shooters and real-time strategy games.
- GameBot, Gamee
- Build social services. A bot could connect people looking for conversation partners based on common interests or proximity.
- Do virtually anything else. Except for dishes — bots are terrible at doing the dishes

How do bots work?

At the core, Telegram Bots are special accounts that do not require an additional phone number to set up. Users can interact with bots in two ways:

Send messages and commands to bots by opening a chat with them or by adding them to groups. This is useful for chat bots or news bots like the official TechCrunch bot.

Send requests directly from the input field by typing the bot's @username and a query. This allows sending content from inline bots directly into any chat, group or channel.

Messages, commands and requests sent by users are passed to the software running on your servers. Our intermediary server handles all encryption and communication with the Telegram API for you. You communicate with this server via a simple HTTPS-interface that offers a simplified version of the Telegram API. We call that interface our Bot API.

How do I create a bot?

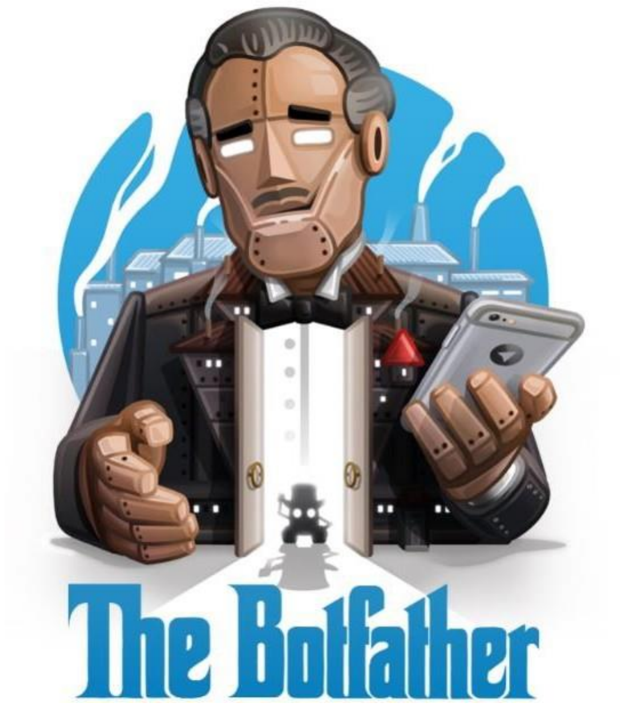


Fig 5.9 The Botfather.

There's a… bot for that. Just talk to BotFather (described below) and follow a few simple steps. Once you've created a bot and received your authorization token, head down to the Bot API manual to see what you can teach your bot to do.

5.2 SOFTWAREREQUIRMENTS

5.2.1 Arduino IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

- **setup():** This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
- **loop():** After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

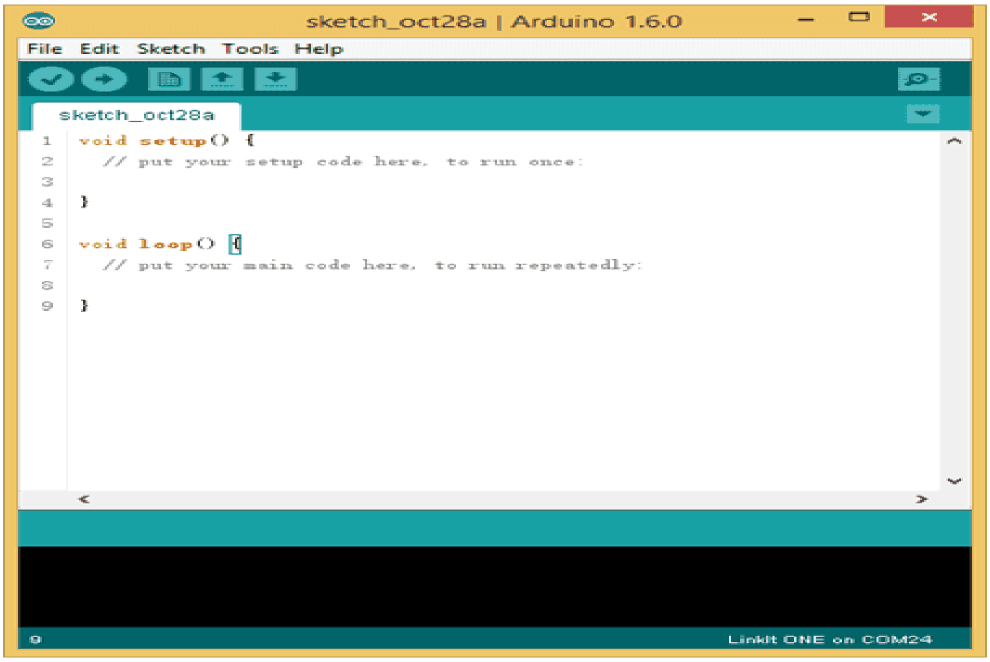


Fig 5.10 Arduino IDE

5.2.2 Embedded C:

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.

Programming Embedded Systems

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart of the Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e. we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like Assembly (low-level Programming Language), C, C++, JAVA (high-level programming languages), Visual Basic, JAVA Script (Application level Programming Languages), etc.

In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

Introduction to Embedded C Programming Language

Before going in to the details of Embedded C Programming Language and basics of Embedded C Program, we will first talk about the C Programming Language.

The C Programming Language, developed by Dennis Ritchie in the late 60's and early 70's, is the most popular and widely used programming language. The C Programming Language provided low level memory access using an uncomplicated compiler (a software that converts programs to machine code) and achieved efficient mapping to machine instructions.

The C Programming Language became so popular that it is used in a wide range of applications ranging from Embedded Systems to Super Computers.

Embedded C Programming Language, which is widely used in the development of Embedded Systems, is an extension of C Program Language. The Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc.

The extension in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.Basics of Embedded C Program

Now that we have seen a little bit about Embedded Systems and Programming Languages, we will dive in to the basics of Embedded C Program. We will start with two of the basic features of the Embedded C Program: Keywords and Datatypes.

CHAPTER VI

METHODOLOGY

CHAPTER 6

METHODOLOGY

6.1 EXISTING METHODOLOGY

Several methods have been developed for recognizing sign language, including Hidden Markov Model (HMM) and Support Vector Machine (SVM). Previous approaches primarily relied on video sequences to identify fingerspelling, as it often involves movement. However, certain fingerspelling can be recognized with a single static image, as it does not require any motion. In this research, a recognition method is proposed specifically for fingerspelling without movement in Japanese sign language. The objective is to recognize 51 characters without movement, utilizing only one image. To address the varying levels of difficulty in recognizing different fingerspelling gestures, the method combines pattern recognition with classification trees for easy gestures and machine learning with SVM for challenging ones. Experimental results indicate an average recognition rate of 86%. Sign language plays a vital role in facilitating communication for individuals with hearing impairments, and various recognition methods like HMM and SVM have been explored. While fingerspelling is typically recognized using video sequences captured by devices such as Microsoft Kinect or Leap Motion controllers to capture finger movements, some fingerspelling characters can be identified with a single snapshot. This study introduces a method specifically designed for recognizing fingerspelling without movement in Japanese sign language. The method targets 51 characters and relies on a single image for recognition. It employs pattern recognition with classification trees for easily recognizable gestures and machine learning with SVM for more challenging ones. Experimental evaluation demonstrates an average recognition rate of 86%.

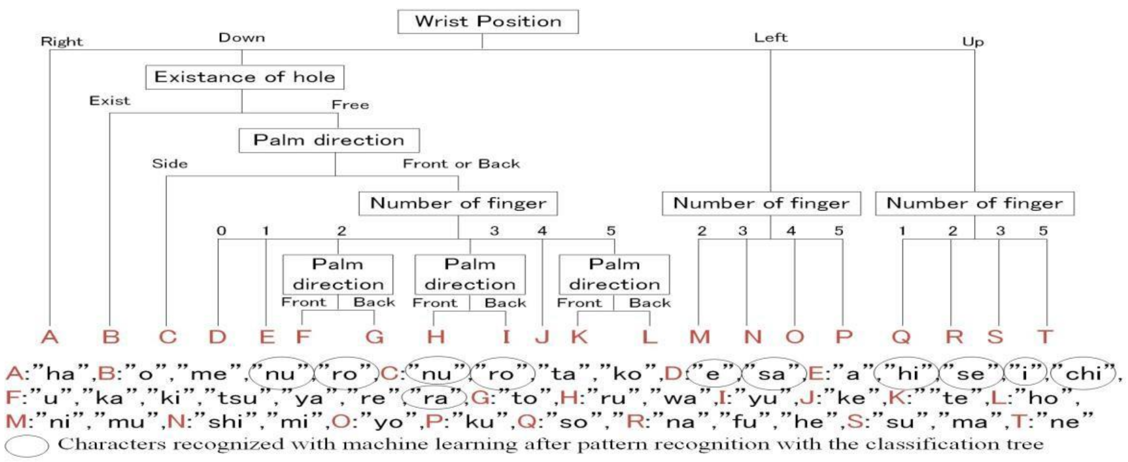


Fig 6.1: Classification Tree

6.2 PROPOSED METHODOLOGY

Convolutional Neural Networks (CNN) are a type of artificial neural network that are particularly suited for processing images

6.2.1 DATA SET GENERATION

A CNN consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers are responsible for extracting features from the input image, while the pooling layers reduce the spatial dimensions of the features. The fully connected layers then use these features to make a prediction about the input image.

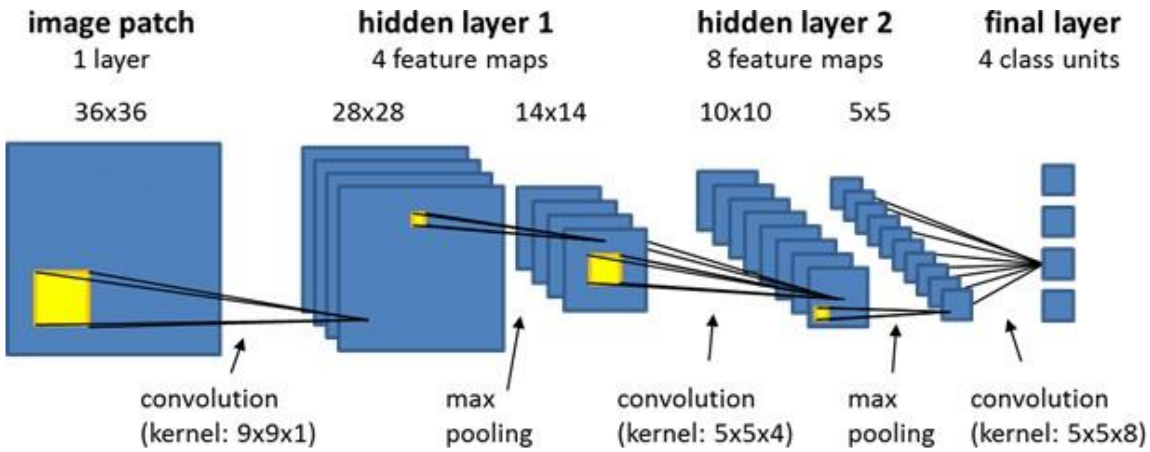


Fig 6.2: Layers of CNN model

In traffic management systems, CNNs can be used to detect and classify vehicles, pedestrians, and other objects on the road. For example, a CNN can be trained to detect cars, trucks, and motorcycles and then use this information to estimate the traffic density on a particular road. This can be useful for traffic flow control, as traffic lights can be adjusted based on the volume of traffic in each lane.

One popular application of CNNs in traffic management is object detection. Object detection involves identifying the presence and location of objects within an image or video feed. This can be done using a combination of bounding boxes and classification scores. For example, a CNN can be trained to detect vehicles and pedestrians in a video feed from a traffic camera. The CNN would then output a list of bounding boxes and classification scores for each detected object.

To train a CNN for object detection, a large dataset of labelled images is required. The labelled images should contain examples of the objects that the CNN is being trained to detect. For example, if the CNN is being trained to detect cars, the dataset should contain images of cars in different orientations, lighting conditions, and backgrounds. The labelled images are used to train the CNN to recognize the features of the objects of interest.

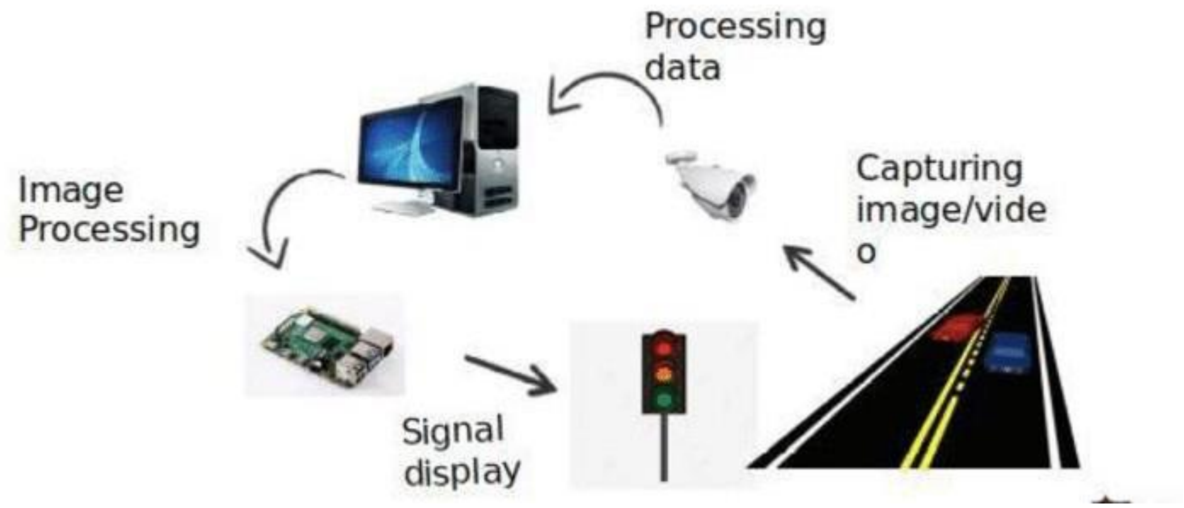


Fig 6.3: Steps included in training process

Once the CNN has been trained, it can be used to process video feeds from traffic cameras in real-time. The video feed is divided into individual frames, which are processed by the CNN one at a time. The CNN identifies the objects in each frame and outputs a list of bounding boxes and classification scores. The bounding boxes can be used to track the movement of the objects over time, which can be useful for predicting traffic flow and identifying potential accidents.

Overall, CNNs are a powerful tool for traffic management systems. They can be used for a wide range of tasks, including object detection, traffic flow control, and accident detection. By processing video feeds from traffic cameras in real-time, CNNs can help to improve the efficiency and safety of our roads.

6.2.2 YOLO V5 ALGORITHM

Digital image processing (DIP) has been widely applied in various fields such as medical imaging, surveillance, object recognition, and traffic management. In this context, DIP can play a vital role in traffic management by processing images from traffic cameras to detect, track, and analyze vehicles and pedestrians. One of the most advanced object identification algorithms available today is YOLO (You Only Look Once), and its latest version, YOLOv5, is a state-of-the-art convolutional neural network (CNN) that can detect objects in real-time with high accuracy.

The use of YOLOv5 in traffic management involves the following steps:

1. Image Acquisition: Traffic cameras are used to capture images of the traffic scene.
2. Image Pre-processing: The acquired images are pre-processed to enhance the quality of the image, remove noise, and correct for distortion.
3. Object Detection: YOLOv5 is applied to detect objects of interest in the image. The algorithm processes the entire image in one go and identifies the location and class of each object present in the image. The object classes include vehicles, pedestrians, cyclists, and traffic signs.
4. Tracking: After detecting the objects, the algorithm tracks them across frames to determine their trajectories and speeds. This information is useful in analysing traffic flow and detecting congestion.
5. Analysis: The tracked objects are analysed to extract meaningful information such as vehicle count, average speed, and occupancy rate. This information can be used to optimize traffic flow and control traffic lights.
6. Communication: The analysed information is communicated to the traffic control centre, which can use it to make real-time decisions on traffic management.

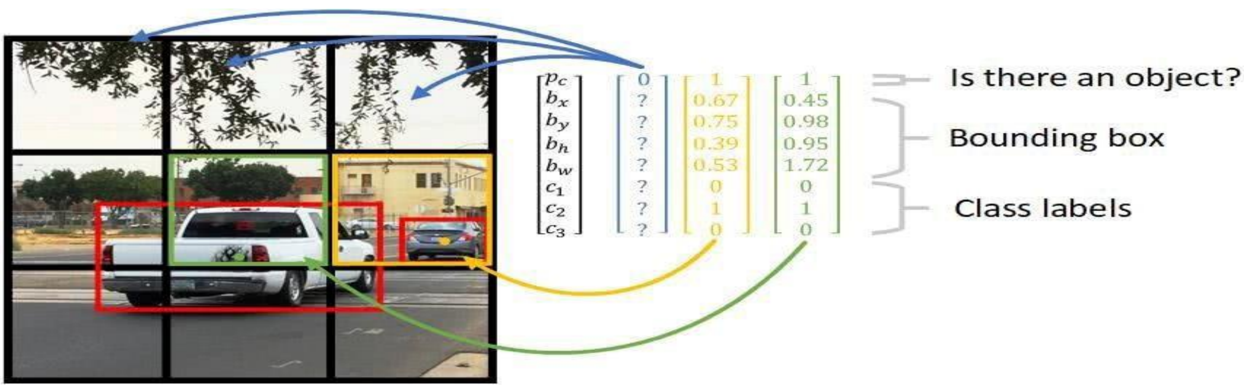


Fig 6.4: Detecting of images using classes

The block diagram of the YOLOv5-based traffic management system is shown in Figure 1. The system comprises a camera module, a processing module, and a communication module.

Block Diagram:

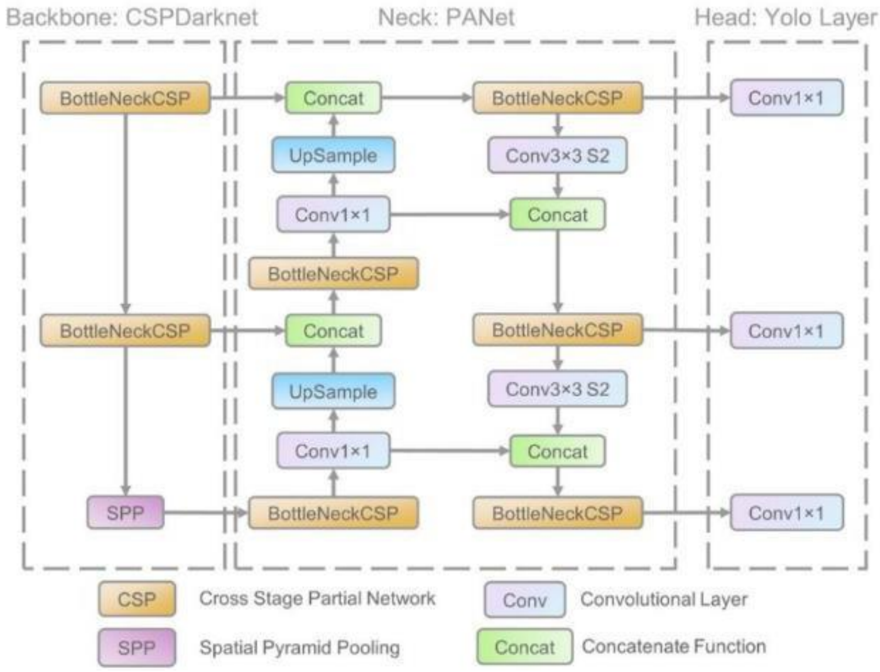


Fig 6.5: Block diagram of CNN

The camera module captures the images of the traffic scene and sends them to the processing module. The processing module consists of an image pre-processing block, an object detection block, a tracking block, and an analysis block. The image pre-processing block enhances the quality of the images, removes noise, and corrects for distortion. The object detection block uses YOLOv5 to detect objects of interest in the image. The tracking block tracks the objects across frames to determine their trajectories and speeds. The analysis block extracts meaningful information from the tracked objects, such as vehicle count, average speed, and occupancy rate.

The communication module receives the analysed information from the processing module and sends it to the traffic control centre. The traffic control centre can use this information to optimize traffic flow and control traffic lights in real-time.

6.2.3 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) have significantly transformed the field of computer vision and image processing, propelling advancements in tasks such as image classification, object recognition, and scene understanding. In this comprehensive explanation, we will explore the inner workings of CNNs, including their architecture, key components, and their contributions to the field of computer vision.

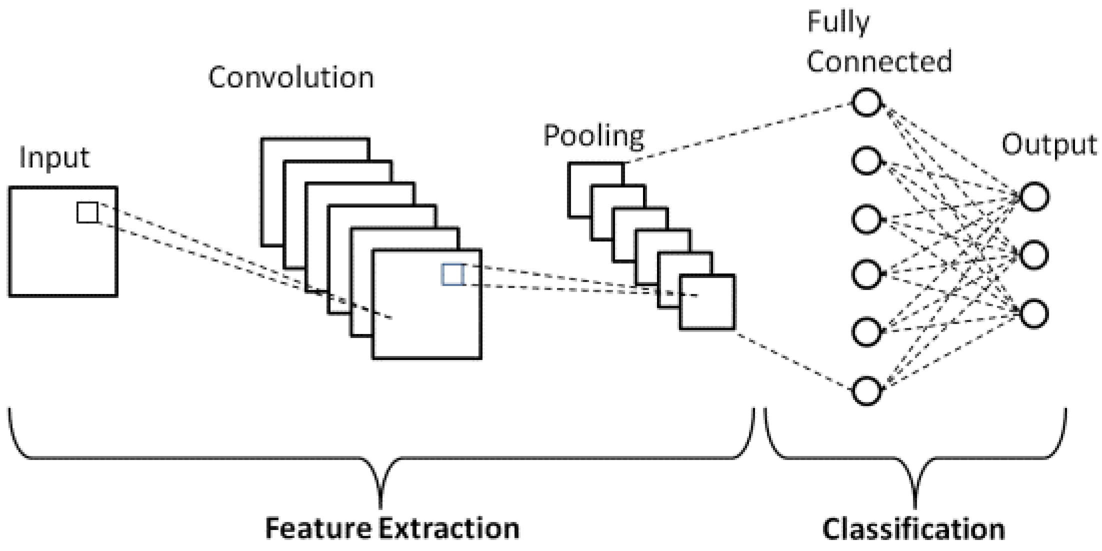


Fig 6.6: Outline of CNN

Convolutional Neural Networks, also known as ConvNets or CNNs, draw inspiration from the visual processing mechanisms of the human brain. These deep learning models excel at analysing grid-like structured data, particularly images, due to their ability to capture local spatial patterns and learn features at different scales.

. Understanding Convolutional Layers:

The core element of CNNs is the convolutional layer. This layer applies learnable filters or kernels to input images, performing convolutions to extract local features. By sliding these filters across the image, CNNs can identify spatial patterns and create feature maps that highlight relevant characteristics.

3. Pooling Layers for Down sampling:

To reduce the dimensionality of feature maps generated by convolutional layers and capture the most important information, CNN architectures often incorporate pooling layers.

Techniques like max pooling and average pooling are commonly employed to down sample the feature maps while retaining key features. Pooling aids in achieving translational invariance and optimizing computational efficiency.

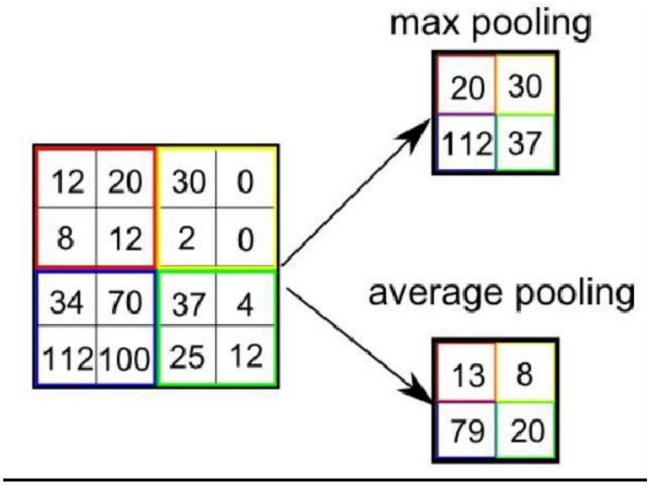


Fig 6.7: Types of pooling

5. Activation Functions:

Activation functions introduce non-linearities into CNNs, allowing them to model complex relationships between inputs and outputs. Popular activation functions include Rectified Linear Unit (ReLU) and its variants, such as Leaky ReLU and Parametric ReLU, which address the vanishing gradient problem and promote sparsity.

5. Fully Connected Layers:

At the end of CNN architectures, fully connected (FC) layers are often utilized. These layers establish connections between all neurons in the preceding layer and the subsequent layer, enabling the learning of global relationships. FC layers are commonly employed for classification tasks, mapping high-level features to specific output classes.

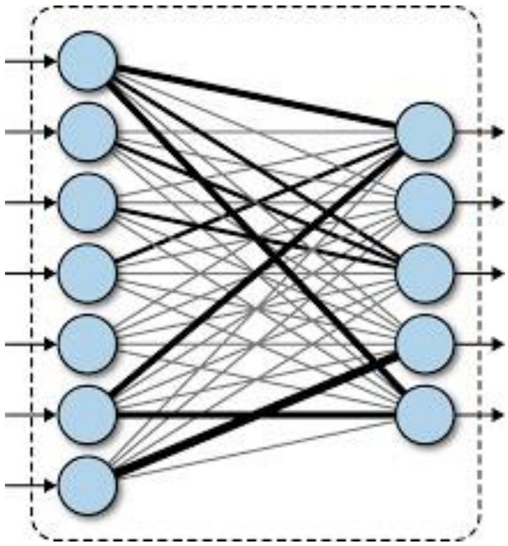


Fig 6.8: Fully Connected Layer

6. Backpropagation and Training:

Training CNNs involves iteratively adjusting the network's weights using backpropagation, a variant of gradient descent. By minimizing a defined loss function, CNNs learn to make accurate predictions. Optimization algorithms like Stochastic Gradient Descent (SGD), Adam, or RMSprop are commonly employed to update the network's parameters.

7. Overfitting and Regularization Techniques:

Addressing overfitting, which occurs when a CNN performs well on training data but poorly on new data, is a significant concern in deep learning. Regularization techniques such as dropout, batch normalization, and weight decay are employed to combat overfitting, enhance generalization, and improve the model's performance on unseen data.

8. Pretrained Models and Transfer Learning:

The concept of transfer learning has greatly benefited CNNs. Pretrained models, trained on large-scale datasets like ImageNet, serve as a starting point for fine-tuning models on specific tasks. Transfer learning allows CNNs to leverage learned features and weights, even when limited labeled data is available, resulting in improved performance and faster convergence.

9. CNN Architectures:

Numerous influential CNN architectures have emerged, each contributing unique architectural features. Examples include LeNet-5, AlexNet, VGGNet, GoogLeNet (Inception), and ResNet.

These architectures vary in terms of depth, filter sizes, skip connections, and computational complexity, addressing specific challenges in computer vision tasks.

10. Recent Advances and Future Directions:
Recent advancements in CNNs include attention mechanisms, which enable the network to focus on important regions, and capsule networks, which capture hierarchical relationships between object parts. CNN research also focuses on interpretability, visualizing learned features and decision-making processes to enhance transparency and trust.

6.2.5 TRAINING AND TESTING:

Our image processing pipeline initiates by converting the RGB input images into grayscale, followed by the application of Gaussian blur to reduce noise. To distinguish the hand from the background, adaptive thresholding is employed, and the images are downscaled to a resolution of 128x128 pixels. Once the input images are prepared, we proceed with training and testing our model.

In order to enhance the accuracy of our model, we trained our neural networks using labelled data. Cross-entropy was employed as a performance metric for classification tasks. The cross-entropy function assigns a positive value when the predicted output deviates from the actual output, and a value of zero when they match perfectly. By minimizing the cross-entropy, we aimed to improve the overall performance of our model.

To achieve this, we leveraged the built-in cross-entropy calculation tool provided by TensorFlow. This tool allowed us to compute the cross-entropy loss and use it to adjust the weights of our neural networks at the network layer. By iteratively updating the weights based on the gradients computed during backpropagation, we aimed to optimize the model's performance.

In our optimization process, we employed the Adam optimizer, a widely used gradient descent optimization algorithm. Adam combines the benefits of two other optimization techniques, namely Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation

(RMSProp). This powerful combination helps in improving the performance of our model by adapting the learning rate for each parameter, considering the historical gradient information.

The Adam optimizer adjusts the weights of the neural network by computing the gradients of the cross-entropy loss function and updating the parameters accordingly. It takes into account the magnitudes of the gradients and the historical gradients to determine the appropriate step size for weight updates. This adaptive nature of the Adam optimizer allows it to efficiently navigate the optimization landscape and converge towards an optimal solution.

By utilizing the Adam optimizer, we aimed to enhance the convergence speed and accuracy of our model during the training process. The adaptive learning rate and momentum provided by the optimizer enabled our neural networks to effectively optimize the cross-entropy loss function and improve the model's ability to classify the input images correctly.

In summary, our image processing pipeline involves various steps such as grayscale conversion, Gaussian blur, adaptive thresholding, and downsampling. The training process utilizes labeled data and employs cross-entropy as a performance metric. By leveraging the Adam optimizer, we optimize the cross-entropy loss function and update the weights of our neural networks to improve the model's accuracy. The combination of these techniques allows us to effectively process images and classify them with high precis

CHAPTER VII

EXPERIMENTAL RESULTS AND DISCUSSION

CHAPTER 7

EXPERIMENTAL RESULTS AND DISCUSSION

7.1 PERFORMANCE MEASURES

The signal light is initially configured such that every lane has a default time duration of 5 seconds. Digital image processing is used to identify the emergency vehicle, Whenever Digital image processing recognizes emergency vehicles on ,the lane traffic signal in that lane turns green in contrast the opposite lane turns red. The project's overall view is depicted in the following image, which also includes a traffic management system, a webcam to implement digital image processing, a temperature sensor, a heart rate sensor, a GPS module, and a nodeMCU microcontroller board.

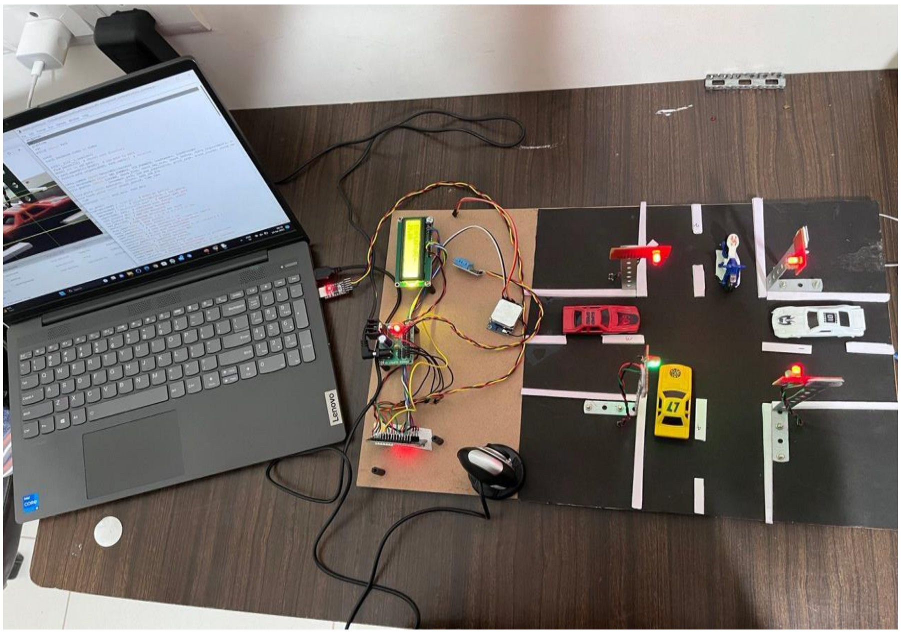


Fig 7.1: Overview of the model

The study and manipulation of digital images through the application of computer algorithms is known as digital image processing. Digital image processing is used to recognize vehicles (bike, car, Ambulance, etc.) as seen in the figures above. To accomplish this task, machine learning techniques are employed to train computers to detect various vehicles based on their look in digital images. Typically, there are numerous steps to this procedure, including picture segmentation, feature extraction, and classification.

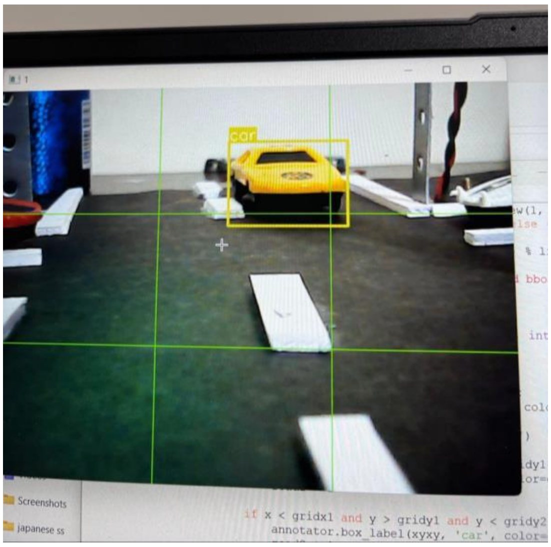


Fig 7.2: DIP recognizing image as car

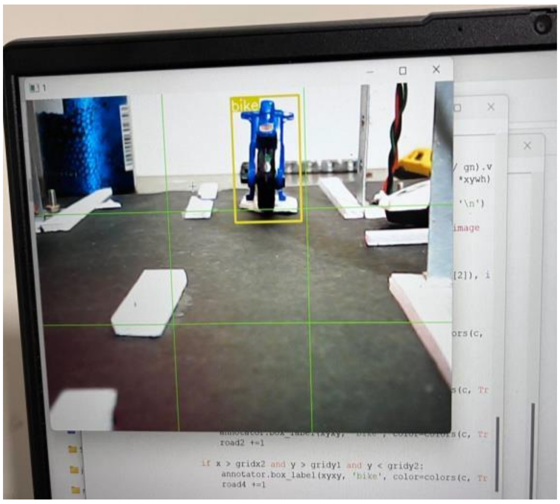


Fig 7.3: DIP recognizing image as bike

Fig 7.4: DIP recognizing image as Ambulance

Two popular medical gadgets used to track a patient's health are temperature and heart rate sensors. These sensors are affixed to the patient's body and transmit data to a monitoring system in real-time. This data is then transmitted to hospitals where medical staff monitors the patient's vital signs, diagnoses any issues, and ensures the necessary equipment is on hand. While the temperature sensor measures the patient's body temperature and displays it in Celsius, the heart rate sensor measures the electrical activity of the heart and displays the patient's heart rate in beats per minute. Healthcare professionals can promptly spot any symptoms and take the necessary action to stop any issues by continually monitoring these vital signs.

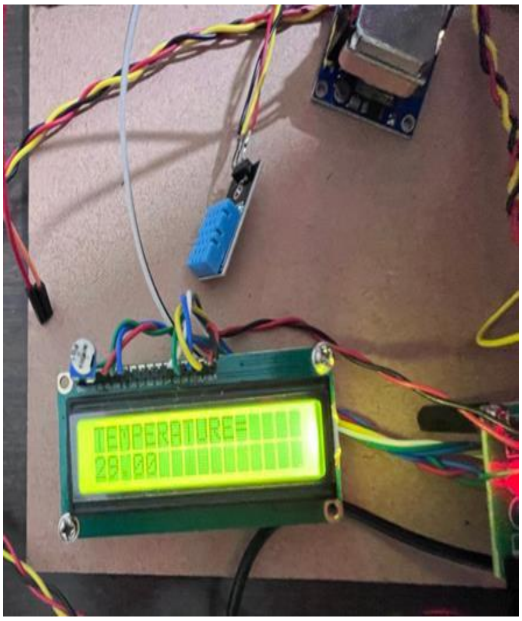


Fig 7.5: LED display showing temperature

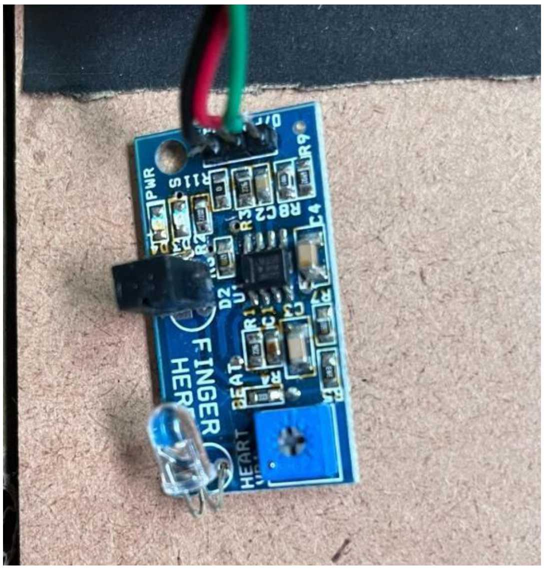


Fig 7.6: Heart-beat sensor

The NodeMCU is a versatile and portable device that can be used to transmit real-time data from ambulances to hospitals. By connecting the NodeMCU with heart rate and temperature sensors, the device can collect vital sign data from the patient and transmit it wirelessly to the hospital. The device can be programmed to send data continuously or at set intervals, allowing healthcare professionals to monitor the patient's condition in real time. Real-time data transmission can ensure that hospitals have the resources they need to prepare for critical patients and provide quick and efficient care. The image above shows the patient's actual data in terms of temperature and heart rate..

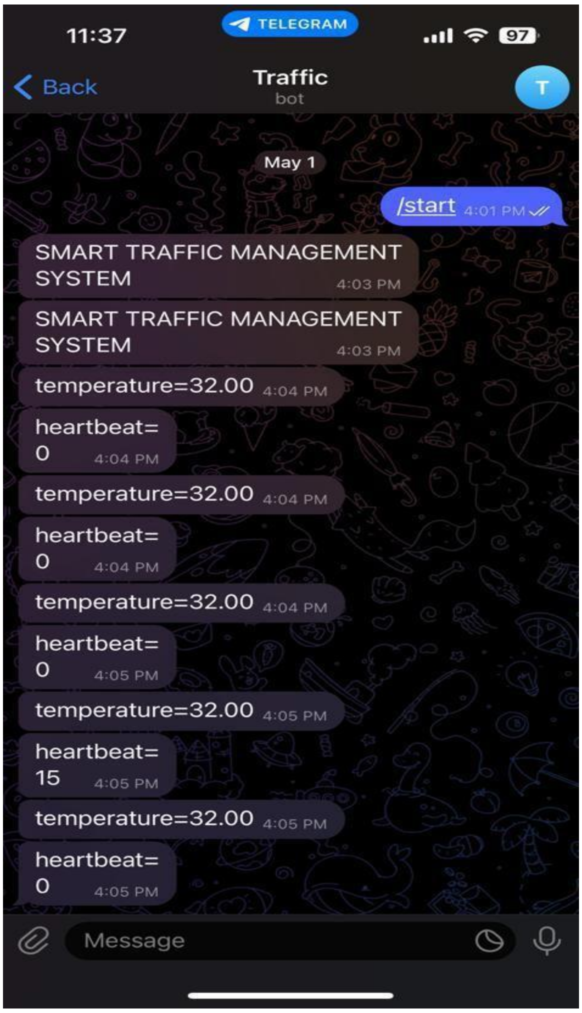


Fig 7.7: Telegram chatbot

CHAPTER VIII

CONCLUSION AND FUTURE SCOPE

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The proposed methodology combining YOLO Version 5 with GPS Ambulance Tracking System and traffic flow control has the potential to greatly improve emergency response times and patient outcomes. By using a novel convolutional neural network that detects objects in real-time with high accuracy, the system can quickly identify and clear traffic for ambulances. Additionally, the intelligent ambulance's extra features such as health status detection and communication with hospitals through nodemcu can further improve patient care. The use of digital image processing and automatic mode design also makes the system efficient and user-friendly. With the integration of these advanced technologies, the proposed methodology has the potential to revolutionize emergency medical services.

8.2 FUTURE SCOPE

The future scope of the smart ambulance system utilizing Node MCU, GPS, IoT, health monitoring, digital image processing, vehicle density detection, traffic control, and the YOLOv5 algorithm holds immense potential for further advancements in emergency medical services. One potential future development is the integration of artificial intelligence (AI) and machine learning (ML) algorithms into the system. By leveraging AI and ML techniques, the smart ambulance can continuously learn and adapt to changing traffic patterns, optimize routing algorithms, and improve object detection accuracy. This can further enhance response times and traffic management, ensuring faster and safer transportation of patients. Another area of future development lies in the expansion of health monitoring capabilities. By incorporating wearable devices and sensors, the system can monitor a broader range of vital signs, gather more comprehensive patient data, and provide real-time updates to medical professionals. Additionally, advancements in telemedicine can enable remote consultations between ambulance crews and specialized healthcare providers, facilitating prompt medical interventions and improving patient care during transit.

REFERENCES

[1] Shantanu Sarkar, Ambulance Assistance for Emergency Services Using GPS Navigation in International Research Journal of Engineering and Technology (IRJET), September,2016.

[2] G.araprasad, “High stable power aware multicast algorithm for mobile ad hoc networks,” “IEEE Sensors J.”, Volume: 05 Issue: 09 | Sep 2018, page no:523-526.

[3] Saha, H. N., Raun, N. F., & Saha, M. (2017, August). Monitoring patient's health with smart ambulance system using Internet of Things (IOTs). In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) (pp. 91-95). IEEE.

[4] Saurabh Kapoor, Parul Gupta, Pooja Sharma, Prabhu Nath Singh Intelligent Ambulance with Automatic Traffic Control ,International Research Journal of Engineering and Technology (IRJET) Volume:05 Issue:05 ,2017,page no-1265-1266

[5] M. Abirami, G. Archana, R. S. Dheepika ,M. Keerthana “GPS BASED TRAFFIC SIGNAL CONTROL SYSTEM FOR AMBULANCE USING MACHINE LEARNING” , International Research Journal of Engineering and Technology(IRJET) Volume: 08 Issue: 05, 2021,page no-228-231

[6] Himadri Nath Saha , Arpita Konar, Shreya Guha , Avik Mukherjee , Shayeli Sarker , Saunak Chowdhury , Saanya Khan , Gourav Saha, Rituparna Guchhait , Aniket Mittra , Sanhita Bishal , Srutayu Biswas Automatic Ambulance System using IOT International Journal of Engineering Science Invention (IJESI), Volume 9 Issue 5 Series I ,May 2020 , PP 01-05

[7] Prof. Dr. Deepak .B. Kadam, Prof. “Patient Health Care & Ambulance Monitoring System using Controller & Wireless Technology”, IJIREEICE, Vol. 5, Issue 9, September

[8] Venkata Krishna Kota, Nagendra Kumar Mangali, Thirumal Kumar Kanakurthi, "Automated Accident Detection and Rescue System", WiSPNET Conference, 2017.

[9] Tugay Akca, Emre Kocyigit, Ozgur Koray Sahingoz and Mucahid Tozal., "IntelligentAmbulance Management System in Smart Cities.," in IEEE-2020.

[10] Rajeshwari S., Santhoshs Hebbar, Varaprasad Golla: "Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance and Stolen Vehicle Detection", IEEE Sensors Journal, 2013, Vol 10, pp. 1530-537X(c).

Annexure-I

MAT JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled

Intelligent Ambulance Using IOT

Submitted by Kalpavi C Y has been published in

Journal of Electronics and Communication Systems

Volume 8 Issue 1 Year 2023

MAT JOURNALS * INDIRAPURAM * GHAZIABAD U.P.

Arvind Sharma

Arvind Sharma

Managing Director, MAT Journals Pvt. Ltd.

MAT JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled

Intelligent Ambulance Using IOT

Submitted by Bhumika P has been published in

Journal of Electronics and Communication Systems

Volume 8 Issue 1 Year 2023

MAT JOURNALS GHAZIABAD U.P. ★ INDIRAPURAM ★

Arvind Sharma

Arvind Sharma

Managing Director, MAT Journals Pvt. Ltd.

MAT JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled

Intelligent Ambulance Using IOT

Submitted by Isha V has been published in

Journal of Electronics and Communication Systems

Volume 8 Issue 1 Year 2023

MAT JOURNALS * GHAZIABAD U.P. * INDIRAPURAM

Arvind Sharma

MAT
JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled

Intelligent Ambulance Using IOT

Submitted by Darshan M has been published in

Journal of Electronics and Communication Systems

Volume 8 Issue 1 Year 2023

MAT JOURNALS ★ GHAZIABAD U.P. ★ INDIRAPURAM

Arvind Sharma
Managing Director, MAT Journals Pvt. Ltd.

MAT JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled

Intelligent Ambulance Using IOT

Submitted by M Phani Teja has been published in

Journal of Electronics and Communication Systems

Volume 8 Issue 1 Year 2023

MAT JOURNALS * GHAZIABAD U.P. * INDIRAPURAM

Arvind Sharma

Arvind Sharma

Managing Director, MAT Journals Pvt. Ltd.

Annexure-II

PLAGIARISM REPORT

Survey Paper: 1

survey ambulance

ORIGINALITY REPORT

11%	10%	6%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijert.org Internet Source	2%
2	www.ijraset.com Internet Source	1%
3	ijesi.org Internet Source	1%
4	imanagerpublications.com Internet Source	1%
5	M. Anantha Sunil, J. Nagaraju, G. Mohan Rao. "Tuning the opto-electrical properties of AgInS 2 films prepared by two- step process", Surfaces and Interfaces, 2018 Publication	1%
6	ijariie.com Internet Source	1%
7	ijritcc.org Internet Source	1%
8	www.turcomat.org Internet Source	<1%

Survey paper :2

ambulance

ORIGINALITY REPORT

12%	9%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Visvesvaraya Technological University Student Paper	1%
2	www.ijraset.com Internet Source	1%
3	Nobuhiko Mukai, Naoto Harada, Youngha Chang. "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning", 2017 Nicograph International (NicoInt), 2017 Publication	1%
4	ijesi.org Internet Source	1%
5	www.ijert.org Internet Source	1%
6	www.irjet.net Internet Source	1%
7	Submitted to Queen Mary and Westfield College Student Paper	1%

Annexure-III

1. TENSORFLOW:

TensorFlow is a complete open-source Machine Learning platform. It features a rich, flexible ecosystem of tools, libraries, and community resources that enable researchers to push the boundaries of Machine Learning and developers to simply design and deploy Machine Learning-powered applications.

TensorFlow has various layers of abstraction, allowing you to select the best one for your purposes. Create and train models with the Keras API, which simplifies getting started with TensorFlow and machine learning. If you require additional freedom, eager execution enables instant iteration and intuitive debugging. Use the Distribution Strategy API for distributed training on diverse hardware configurations without modifying the model definition for large ML training projects.

2. KERAS:

Keras is a Python-based high-level neural network library that acts as a wrapper for TensorFlow. It is utilised when we need to quickly develop and test a neural network with few lines of code. It includes implementations of commonly used neural network parts such as layers, objectives, activation functions, optimizers, and tools for working with images and text data.

3. OPENCV:

OpenCV (Open-Source Computer Vision) is a free and open-source programming library for real-time computer vision.

It is mostly used for image processing, video capture, and feature analysis such as face and object identification. It is written in C++ and has bindings for Python, Java, and MATLAB/OCTAVE as its primary interface.

Annexure-IV

CODE OF THE PROJECT

```
#include<DHT.h>
int red1=2;
int green1=15;
int red2=5;
int green2=4;
int red3=23;
int green3=22;
int red4=19;
int green4=18;

#define Type DHT11
int DHT11PIN = 21;
DHT HT(DHT11PIN, Type);
float tempC;


#include <LiquidCrystal.h>
const int rs =13, en = 12, d4 = 14, d5 = 27, d6 = 26, d7 = 25;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  Serial.begin(9600);
  pinMode(DHT11PIN, INPUT);
  HT.begin();
  pinMode(red1,OUTPUT);
  pinMode(green1,OUTPUT);
  pinMode(red2,OUTPUT);
  pinMode(green2,OUTPUT);
  pinMode(red3,OUTPUT);
  pinMode(green3,OUTPUT);
  pinMode(red4,OUTPUT);
  pinMode(green4,OUTPUT);

  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("TRAFIC CLEARANCE");
  lcd.setCursor(0, 1);
  lcd.print("FOR AMBULANCE");
//  Serial.println("welcome");

}

void loop() {
  initial_position();
```

```
}

void serial_event(){
  if(Serial.available() > 0){
    String data = Serial.readString();
    char e = data[0];
    if(e == 'A'){

        lcd.clear();
          lcd.print("AMBULENCE  DETECT");
          lcd.setCursor(0, 1);
          lcd.print("IN ROAD ONE ");
          road1(5);
    }
    else if(e == 'B'){

        lcd.clear();
          lcd.print("AMBULENCE  DETECT");
          lcd.setCursor(0, 1);
          lcd.print("IN ROAD TWO ");
          road2(5);
    }
    else if(e == 'C'){

        lcd.clear();
          lcd.print("AMBULENCE  DETECT");
          lcd.setCursor(0, 1);
          lcd.print("IN ROAD THREE ");
          road3(5);
    }
    else if(e == 'D'){

        lcd.clear();
          lcd.print("AMBULENCE  DETECT");
          lcd.setCursor(0, 1);
          lcd.print("IN ROAD FOUR ");
          road4(5);
    }
    else{
      int r1 = data[0]-48;
      int r2 = data[1]-48;
      int r3 = data[2]-48;
      int r4 = data[3]-48;

      int a, i, j, k, m;
      char b;
      int num[] = {r1, r2, r3, r4};
      char rd[] = "ABCD";

      for (i = 0; i < 4; ++i){
```

```
        for (j = i + 1; j < 4; ++j){
          if (num[i] < num[j]){
              a = num[i];
              b = rd[i];
              num[i] = num[j];
              rd[i] = rd[j];
              num[j] = a;
              rd[j] = b;
          }
        }
      }
      for(m=0; m<4;m++){
        if(rd[m] == 'A'){

          lcd.clear();
          lcd.print("R1 GREEN SIGNAL");
          lcd.setCursor(0, 1);
          lcd.print("TOTAL VEHICLE "+String(num[m]));
          road1(num[m]);
        }
        if(rd[m] == 'B'){

          lcd.clear();
          lcd.print("R2 GREEN SIGNAL");
          lcd.setCursor(0, 1);
          lcd.print("TOTAL VEHICLE "+String(num[m]));
          road2(num[m]);
        }
        if(rd[m] == 'C'){

          lcd.clear();
          lcd.print("R3 GREEN SIGNAL");
          lcd.setCursor(0, 1);
          lcd.print("TOTAL VEHICLE "+String(num[m]));
          road3(num[m]);
        }
        if(rd[m] == 'D'){

          lcd.clear();
          lcd.print("R4 GREEN SIGNAL");
          lcd.setCursor(0, 1);
          lcd.print("TOTAL VEHICLE "+String(num[m]));
          road4(num[m]);
        }
      }
    }
  }
}

void initial_position(){
```

```
  Serial.print("S");
  lcd.begin(16, 2);
  lcd.clear();
  lcd.print("TRAFIC CLEARANCE");
  lcd.setCursor(0, 1);
  lcd.print("FOR AMBULANCE");
  road1(1);
  road2(1);
  road3(1);
  road4(1);
}

void road1(int C)
{
  digitalWrite(red1,LOW);
  digitalWrite(green1,HIGH);
  digitalWrite(red2,HIGH);
  digitalWrite(green2,LOW);
  digitalWrite(red3,HIGH);
  digitalWrite(green3,LOW);
  digitalWrite(red4,HIGH);
  digitalWrite(green4,LOW);
  delay(C*2000);
  serial_event();
  temp_check();
}

void road2(int C)
{
  digitalWrite(red1,HIGH);
  digitalWrite(green1,LOW);
  digitalWrite(red2,LOW);
  digitalWrite(green2,HIGH);
  digitalWrite(red3,HIGH);
  digitalWrite(green3,LOW);
  digitalWrite(red4,HIGH);
  digitalWrite(green4,LOW);
  delay(C*2000);
  serial_event();
  temp_check();
}

void road3(int C)
{
  digitalWrite(red1,HIGH);
  digitalWrite(green1,LOW);
  digitalWrite(red2,HIGH);
  digitalWrite(green2,LOW);
  digitalWrite(red3,LOW);
  digitalWrite(green3,HIGH);
```

```
  digitalWrite(red4,HIGH);
  digitalWrite(green4,LOW);
  delay(C*2000);
  serial_event();
  temp_check();
}

void road4(int C)
{
  digitalWrite(red1,HIGH);
  digitalWrite(green1,LOW);
  digitalWrite(red2,HIGH);
  digitalWrite(green2,LOW);
  digitalWrite(red3,HIGH);
  digitalWrite(green3,LOW);
  digitalWrite(red4,LOW);
  digitalWrite(green4,HIGH);
  delay(C*2000);
  serial_event();
  temp_check();
}
void temp_check()
{
//Humidity = HT.readHumidity();
  tempC = HT.readTemperature();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TEMPERATURE=");
  lcd.setCursor(0,1);
  lcd.print(tempC);
    delay(1000);
//   delay(1000);
// Serial.println("soundval="+String(soundval));
  if(tempC>35)
  {
        lcd.clear();
        lcd.print("MORE TEMPERATURE");
        Serial.print("T");
//          lcd.setCursor(0, 1);
//          lcd.print("IN ROAD ONE ");
//          road1(5);
        delay(1000);
  }
//  delay(1000);
}
```

Python code for DIP:

```
import argparse
import os
import platform
import sys
from pathlib import Path

import torch
import torch.backends.cudnn as cudnn

FILE = Path(_file_).resolve()
ROOT = FILE.parents[0]  # YOLOv5 root directory
if str(ROOT) not in sys.path:
    sys.path.append(str(ROOT))  # add ROOT to PATH
ROOT = Path(os.path.relpath(ROOT, Path.cwd()))  # relative

from models.common import DetectMultiBackend
from utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages,
LoadStreams
from utils.general import (LOGGER, check_file, check_img_size, check_imshow,
check_requirements, colorstr, cv2,
                           increment_path, non_max_suppression, print_args, scale_coords,
strip_optimizer, xyxy2xywh)
from utils.plots import Annotator, colors, save_one_box
from utils.torch_utils import select_device, time_sync
import time
from serialtest import send_data, read_data

@torch.no_grad()
def run(weights=ROOT / 'toys.pt',  # model.pt path(s)
        source=ROOT / '0',  # file/dir/URL/glob, 0 for webcam
        data=ROOT / 'data/coco128.yaml',  # dataset.yaml path
        imgsz=(640, 640),  # inference size (height, width)
        conf_thres=0.30,  # confidence threshold
        iou_thres=0.45,  # NMS IOU threshold
        max_det=1000,  # maximum detections per image
        device='',  # cuda device, i.e. 0 or 0,1,2,3 or cpu
        view_img=False,  # show results
        save_txt=False,  # save results to *.txt
        save_conf=False,  # save confidences in --save-txt labels
        save_crop=False,  # save cropped prediction boxes
        nosave=False,  # do not save images/videos
        classes=None,  # filter by class: --class 0, or --class 0 2 3
        agnostic_nms=False,  # class-agnostic NMS
        augment=False,  # augmented inference
        visualize=False,  # visualize features
        update=False,  # update all models
```

```
        project=ROOT / 'runs/detect',  # save results to project/name
        name='exp',  # save results to project/name
        exist_ok=False,  # existing project/name ok, do not increment
        line_thickness=3,  # bounding box thickness (pixels)
        hide_labels=False,  # hide labels
        hide_conf=False,  # hide confidences
        half=False,  # use FP16 half-precision inference
        dnn=False,  # use OpenCV DNN for ONNX inference
        ):
    source = str(source)
    save_img = not nosave and not source.endswith('.txt')  # save inference images
    is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
    is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
    webcam = source.isnumeric() or source.endswith('.txt') or (is_url and not is_file)
    if is_url and is_file:
        source = check_file(source)  # download

    # Directories
    save_dir = increment_path(Path(project) / name, exist_ok=exist_ok)  # increment run
    (save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True, exist_ok=True)  # make
dir

    # Load model
    device = select_device(device)
    model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data)
    stride, names, pt, jit, onnx, engine = model.stride, model.names, model.pt, model.jit,
model.onnx, model.engine
    imgsz = check_img_size(imgsz, s=stride)  # check image size
    print(names)

    # Dataloader
    if webcam:
        view_img = check_imshow()
        cudnn.benchmark = True  # set True to speed up constant image size inference
        dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt)
        bs = len(dataset)  # batch_size
    else:
        dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt)
        bs = 1  # batch_size
    vid_path, vid_writer = [None] * bs, [None] * bs

    # Run inference
    model.warmup(imgsz=(1 if pt else bs, 3, *imgsz))  # warmup
    dt, seen = [0.0, 0.0, 0.0], 0
    global m
    m = 1
    global np
    np = 1
    count = 0
    for path, im, im0s, vid_cap, s in dataset:
```

```
        t1 = time_sync()
        im = torch.from_numpy(im).to(device)
        im = im.half() if half else im.float()  # uint8 to fp16/32
        im /= 255  # 0 - 255 to 0.0 - 1.0
        if len(im.shape) == 3:
            im = im[None]  # expand for batch dim
        t2 = time_sync()
        dt[0] += t2 - t1

        # Inference
        visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize else
False
        pred = model(im, augment=augment, visualize=visualize)
        t3 = time_sync()
        dt[1] += t3 - t2

        # NMS
        pred = non_max_suppression(pred, conf_thres, iou_thres, classes, agnostic_nms,
max_det=max_det)
        dt[2] += time_sync() - t3

        # Second-stage classifier (optional)
        # pred = utils.general.apply_classifier(pred, classifier_model, im, im0s)
        road1=0
        road2=0
        road3=0
        road4=0
        counting=0
        # Process predictions
        for i, det in enumerate(pred):  # per image
            seen += 1
            if webcam:  # batch_size >= 1
                p, im0, frame = path[i], im0s[i].copy(), dataset.count
                s += f'{i}: '
            else:
                p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)

##              im0 = cv2.resize(im0, (1200, 900))
            p = Path(p)  # to Path
            save_path = str(save_dir / p.name)  # im.jpg
            txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode == 'image' else
f'_{frame}')  # im.txt
            s += '%gx%g ' % im.shape[2:]  # print string
            gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]  # normalization gain whwh
            imc = im0.copy() if save_crop else im0  # for save_crop
            annotator = Annotator(im0, line_width=line_thickness, example=str(names))

            height, width, _ = im0.shape

            gridx1 = int(width/3)
```

```
gridx2 = gridx1 + gridx1

gridy1 = int(height/3)
gridy2 = gridy1 + gridy1

cv2.line(im0, (gridx1, 0), (gridx1, height), (0, 255, 0), 1)
cv2.line(im0, (gridx2, 0), (gridx2, height), (0, 255, 0), 1)
cv2.line(im0, (0, gridy1), (width, gridy1), (0, 255, 0), 1)
cv2.line(im0, (0, gridy2), (width, gridy2), (0, 255, 0), 1)

if len(det):
    # Rescale boxes from img_size to im0 size
    det[:, :4] = scale_coords(im.shape[2:], det[:, :4], im0.shape).round()

    # Print results
    for c in det[:, -1].unique():
        n = (det[:, -1] == c).sum()  # detections per class
        s += f"{n} {names[int(c)]}{'s' * (n > 1)}, "  # add to string

    # Write results
    for *xyxy, conf, cls in reversed(det):
        if save_txt:  # Write to file
            xywh = (xyxy2xywh(torch.tensor(xyxy).view(1, 4)) / gn).view(-1).tolist()  #
normalized xywh
            line = (cls, *xywh, conf) if save_conf else (cls, *xywh)  # label format
            with open(txt_path + '.txt', 'a') as f:
                f.write(('%g ' * len(line)).rstrip() % line + '\n')

        if save_img or save_crop or view_img:  # Add bbox to image
            c = int(cls)  # integer class
            counting += 1

            x, y, w, h = int(xyxy[0]), int(xyxy[1]), int(xyxy[2]), int(xyxy[3])

            D = read_data()

            if names[c] == 'Ambulance' and D == 'S':
                annotator.box_label(xyxy, names[c], color=colors(c, True))
                send_data('A')
                print('Ambulence detected in road 1')

            if x > gridx1 and x < gridx2 and y < gridy1:
                annotator.box_label(xyxy, 'bike', color=colors(c, True))
                road1 +=1

            if x < gridx1 and y > gridy1 and y < gridy2:
                annotator.box_label(xyxy, 'bike', color=colors(c, True))
                road2 +=1

            if x > gridx2 and y > gridy1 and y < gridy2:
```

```
                annotator.box_label(xyxy, 'bike', color=colors(c, True))
                road4 +=1

            if x > gridx1 and x < gridx2 and y > gridy2:
                annotator.box_label(xyxy, 'bike', color=colors(c, True))
                road3 +=1


        if D == 'S' and road1 > 0 or road2 > 0 or road3 > 0 or road4 > 0:
            print('Total Vehicles in road1 {}'.format(road1))
            print('Total Vehicles in road2 {}'.format(road2))
            print('Total Vehicles in road3 {}'.format(road3))
            print('Total Vehicles in road4 {}'.format(road4))
            count1='{}{}{}{}'.format(road1, road2, road3, road4)
            send_data(count1)
            print('==========================================')

        # Stream results
        im0 = annotator.result()
        cv2.imshow(str(p), im0)
        cv2.waitKey(1)  # 1 millisecond
        fps, w, h = 30, im0.shape[1], im0.shape[0]
        save_path = str(Path(save_path).with_suffix('.mp4'))  # force *.mp4 suffix on results
videos
        vid_writer[i] = cv2.VideoWriter(save_path, cv2.VideoWriter_fourcc(*'mp4v'), fps,
(w, h))
        vid_writer[i].write(im0)

def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'toys.pt',
help='model path(s)')
    parser.add_argument('--source', type=str, default='1', help='file/dir/URL/glob, 0 for
webcam')
    parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml',
help='(optional) dataset.yaml path')
    parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640],
help='inference size h,w')
    parser.add_argument('--conf-thres', type=float, default=0.80, help='confidence threshold')
    parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')
    parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per
image')
    parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')
    parser.add_argument('--view-img', action='store_true', help='show results')
    parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')
    parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt
labels')
    parser.add_argument('--save-crop', action='store_true', help='save cropped prediction
boxes')
    parser.add_argument('--nosave', action='store_true', help='do not save images/videos')
```

```
    parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --
classes 0 2 3')
    parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')
    parser.add_argument('--augment', action='store_true', help='augmented inference')
    parser.add_argument('--visualize', action='store_true', help='visualize features')
    parser.add_argument('--update', action='store_true', help='update all models')
    parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to
project/name')
    parser.add_argument('--name', default='exp', help='save results to project/name')
    parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do
not increment')
    parser.add_argument('--line-thickness', default=2, type=int, help='bounding box thickness
(pixels)')
    parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')
    parser.add_argument('--hide-conf', default=False, action='store_true', help='hide
confidences')
    parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
    parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX
inference')
    opt = parser.parse_args()
    opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1  # expand
    print_args(vars(opt))
    return opt


def main(opt):
    check_requirements(exclude=('tensorboard', 'thop'))
    run(**vars(opt))

if _name_ == "_main_":
    opt = parse_opt()
    main(opt)
```

Journal of
Electronics and Communication Systems
Volume-8, Issue-1 (January-April, 2023)

www.matjournals.com

Intelligent Ambulance Using IOT

Kalpavi C Y[1*], Bhumika P[2], Darshan M[3], Isha V[4], M Phani Teja[5]
[1]Associate Professor, [2-5]UG Student, Department of Electronics and Communication Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India

*Corresponding Author: Kalpavi-ece@dsatm.edu.in

Received Date: February 7, 2023 *Published Date: February 16, 2023*

ABSTRACT

The main reason for the death of patients while going to hospitals is traffic congestion Traffic congestion may be caused due to many reasons such as accidents, road construction works, large numbers of pedestrians crossing roads, etc. Especially in metropolitan cities like Bangalore, Delhi, Hyderabad, etc. In such cities, people live a busy lifestyle where they have no time for themselves, and they don't even think of giving way to emergency services like ambulances. To overcome this problem as well as many such problems this paper gives a solution that is "Intelligent Ambulance using IOT". This paper describes the use of many technologies such as GPS for providing location, IOT for calculating the density of vehicles and automatically changing signal light to green if an ambulance is on the way and to monitor the health of the patient, Arduino microcontroller and Nodemcu technology to send vital information such as heartbeat rate and body temperature of the patient to nearest hospitals via telegram. The proposed system is used to save the life of patients by taking them to the hospital in a faster way than the usual traditional way. Additionally to this, the patient monitoring system can check the victim's health condition using a heart rate sensor and temperature sensor, i.e. LM35, and send the information to the hospital.

Keywords- GPS, Health monitoring, IoT, Microcontroller, Nodemcu

INTRODUCTION

India is one of the foremost inhabited countries in the world, and road traffic congestion and delayed provision of medical attention are critical issues for loss of life. According to many statistics, each minute of life is lost due to late responses by family members, hospital formalities, or unavoidable situations such as traffic congestion. Saving a life is precious to both the patient's family and even the hospital's name. Due to the increase in population and busy lifestyle of people, they are not able to provide a way for the ambulance. This paper describes a solution for automatic traffic signal light changing and sending patients' vital information to the nearest hospitals so that they can keep the required equipment ready before the patient reaches the hospital. The proposed system is used to save the life of patients by taking them to the hospital in a faster way than the usual traditional way. Earlier the ambulance had to wait for a longer time till the signal gives the green light which has no technology attached to it, in this system a set of IR sensors and sound sensors are placed on four lanes of the road connected by a traffic signal. These sensors are used to calculate the traffic density on the roads and also the presence of ambulances in the lane, if any ambulances are detected the traffic signal in that lane turns green irrespective of which light it should display, and simultaneously the signal on the opposite lane turns red. The presence of a heartbeat rate sensor and body temperature sensor will sense the vital details of the patient and this system uses
nodemcu.

LITERATURE SURVEY

In paper [1], we have considered the analysis of finding the shortest path and giving the best route to an ambulance to reach the particular hospital. This paper summarizes the creation of a GPS System for which the GPS tracker is built and it is set up with an ambulance to track the path. This technology can be used by everyone, even a common man can track all the ambulances in his surroundings. This paper also provided the proper utilization of GPS technology and in case of any emergency; this technology will be very useful in finding nearby ambulances and reaching the patient's address at the accident spot. This GPS technology in an ambulance is developed to overcome the current difficulty which is faced by many ambulance drivers. Hence this technology uses the shortest path-establishing algorithm to reach the destination hospital in a minimum amount of time. The overall architecture of this system consists of two sides. One is the ambulance required on one side and the other one is the ambulance driver's side. The patient's side is just to track the address of the available ambulances with a GPS sensor. Then the ambulance driver can check the exact location of the person who has booked the ambulance. If the driver is busy or cannot make it to reach at that time then he may cancel the user's request for an ambulance, the user will get notified about the non-availability of the ambulance but he still gets many lists of the ambulance which are nearby. Meanwhile, he will also be getting information about the nearest hospitals.

G. Ramprasad et al. [2] put a paper that mainly depends on the very low cost and accurate and also real-time traffic controlling system. This system mainly focuses on detecting and monitoring the traffic density and overall traffic volume through the help of infrared sensors (IR Sensors), and changes the signal lights and also the timing of the slots. In this paper, we have considered the usage of IR sensors for monitoring traffic and controlling traffic signals. In this paper [2], we observed the usage of mainly three IR1, IR2, and IR3 for low traffic density, and the other two for medium and high traffic density respectively. Through this knowledge, we have taken into consideration the use of a total of four IR sensors for better results and also to avoid any collisions with the signal. According to this paper, when the IR1 sensor is sensed, the LCD Display shows Low traffic density and the signal turns to green light and it will be turned ON for 20 seconds. When the IR2 sensor is sensed, the LCD Display shows medium traffic density and turns ON the green light which will be again ON for 40 seconds. When the IR3 sensor is detected, the LCD Display shows High traffic density and the green light will be kept ON for 60 seconds. So, when an ambulance reaches the junction, the IR sensors get detected and the same sensor will turn those junction signals to green and others to red, the signal will be turned ON to green for the next 60 seconds until the ambulance passes to the next junction or the next signal.

In paper [3] Saah H. N et al proposes the method of implementing a health monitoring system in an ambulance, which helps to have an update about the patient's health conditions before reaching the particular hospital. The health monitoring system includes the body temperature, patient's heartbeat, pulse sensors, etc. This paper [3], helped with the usage of various health monitoring sensors for detecting these health-related conditions of the patient in the ambulance while reaching the hospital. We also see the usage of temperature sensor which is used to identify the temperature in blood vessels and also to find out the output of cardiac. In this paper, we see that the IC lm35 sensor is used to know the body's temperature. In this paper, we have observed the usage of the new model of Raspberry Pi which is the Raspberry Pi 4 microcontroller for developing a health monitoring system inside the ambulance. To get the health conditions of the patient, these sensors are connected to the Raspberry Pi. When compared to old raspberry models, the speed and performance of this were better. And also compared to other microcontrollers, this consumes less power and runs without producing any noise. The Raspberry Pi 4 has upgraded USB capacity; it also has two USB 2 ports and two USB 3 ports, which help in transferring data ten times faster. Different varieties of raspberry models are available depending on how much RAM is needed for the particular application. The raspberry model is more powerful and can multitask at a single time. Through this paper, we got an idea about the usage of the microcontroller for implementing the health monitoring system inside the ambulance.

In Paper [4] the system focuses on providing the ambulance a free way to an emergency unit on the way to a hospital ambulance or to the accident spot when struck in traffic jams or not. So, to overcome this problem, researchers of this paper [4], proposed an adaptive way where the ambulance informs the closest traffic signals about their arrival through IR led and the ambulance buzzer. This system tries to develop an algorithm, considering the current and the destination location, in deciding the path that the ambulance must take. Helps also route traffic based on traffic density by establishing a network between ambulance and signal posts. Further in this paper, the health condition of the patient is continuously monitored by the emergency unit and is performed by various sensors on the ambulance. Based on the described parameters, the availability of the medical requirement to treat the patient is judged. Various parameters like Heart Rate and Body Temperature are monitored. This data was sent to the hospital for further analysis. Proper treatment may not be available if the doctors do not have a sufficient medical history of the patient. The system ties in, obviating just that, the entire system is enhanced to develop the facility for proving the patient first aid at the earliest in an emergency.

In Paper [5], at present, there is no automation system for controlling traffic signals in emergencies. This paper [5] proposed a basic architectural model of an intelligent ambulance by enhancing its features by adding a GPS module to the traffic control system which was successfully proposed previously. This model is a complete automation system where the driver has to switch to emergency mode through the created mobile application. This activation will help the app track the live location of the ambulance and also help in tracking the nearest traffic control signals with the help of GPS. The destination location is pinned which helps in understanding the best route available to reach the earliest. Whenever the ambulance approaches the 500m traffic signal, it immediately sends a request to control the traffic lights which the ambulance will have to cross, and again sends a signal to set it back. When the signal is already green, based on the timing, the app automatically requests control over the nearest control signal. If the signal is red, similarly depending on the remaining time, the app requests a control to open that particular ambulance is traveling by calculating the distance. This helps the ambulance to move freely without getting stuck in traffic. Considering the direction of the vehicle, the server toggles the signal. In real-time, the system is improved so the death rate can be reduced during emergencies. Hence in this paper, we come across an idea to control signal from the driver's smartphone itself.

In Paper [6], a lot of advantages that IoT application offers within the aid sector are mostly categorized into the tracking of patients and staff, authenticating people, and also gathering information and sensing. Paper [6] presents an architectural model consisting of three different modules. The ambulance module is placed inside the ambulance. It consists of Raspberry Pi with various sensor units interfacing with it. The different health parameters of the patient's health are monitored including heartbeat, temperature, blood glucose levels, and various others. After processing these statistics, they can be uploaded into the cloud. The second module is the cloud or WIFI module, which consists of two cloud services. The Think Speak cloud platform supports numerical data being uploaded. The image upload is done on Dropbox cloud. The third module is the hospital module which helps the doctor monitor the patient's health condition based on the data which is uploaded into the cloud. This module also consists of a simple application to download the data from the cloud. If this real-time data can be made available for access to the doctor, it can help him to give adequate feedback to the emergency vehicle as well as arrange hospital facilities for the patient.

In 2019 Dr. Deepak B et al. [7] put forward a paper that mainly emphasizes. There is a conjointly projected phone health observation system for the elderly that works with smartphones and wireless sensors to check their health. The elderly person's smartphone can send an alert to the people who have been pre- assigned in case of an emergency so they can prepare for a machine. This technology provides a platform for distant family members and the elderly person's health state to communicate. The advertisement promises a tool that may display a wide range of physical characteristics of different patient bodies. The base station receives information from the organizer cluster point through the wireless sensors, which emit signals that are concentrated on the patient's skin. In addition to mere measurement of the physical attributes, the device may also detect irregular activity from them and notify the doctor accordingly. With the help of HMS, the patient will be admitted via an automatic procedure more often. HMS is designed with a display, bio-medical sensors, and a 1x4 keypad; all of them are intended to form a group of the patient's critical data. The projected system uses a variety of sensors, including pulses and vital sign sensors as well as fingerprint and vital sign sensors. To display the selected mode, an OLED is used. The patient may not be conscious when placed inside the car teaming forces with a relative. Due to the patient's inability to access their vital information, there may be urgent circumstances when the patient is also not awake. When transporting the patient, the care in the automobile makes a judgment call regarding the patient's current status and selects one of the earlier modes. To provide careful confidence in the mode that was selected, the mode is presented on the screen.

In 2019 Venkata Krishna Kota et al. [8] put forward a paper that mainly emphasizes. There is conjointly a projected phone safety observation system for the elders that are experiencing wireless body sensors, sensible mobiles, and observation of the health of the elderly. If there is any emergency, the sensible mobile with an older human being can provide associate awakening to the folks that are pre- assigned so that they'll organize for the associate machine. This technique creates a platform for family members to communicate about the health of the elderly person regardless of distance. The advertisement suggests a device that is capable of monitoring multiple physical characteristics of various patient bodies. Through an organizer node that communicates with a base station, the signals from the wireless sensors are gathered on the patient's body. In addition to just measuring the physical parameters, the device may also detect problems and alert the physician accordingly. The process continues with the help of the HMS as of right now because the patient has been admitted into the car. HMS is equipped with bio-medical sensors, a display, and a 1x4 membrane keypad; all of them are intended to gather the patient's vital information. A few of the sensors used in the proposed system include pulses and vital sign sensors as well as fingerprints and vital sign readers. The selected mode is displayed using a degree OLED. Once the patient is in the car, a family member or guardian may or may not be present to assist. There may be emergencies in which the patient is also unconscious, rendering vital information about the patient unavailable to be accessed. HMS employs a fingerprint sensor component to overcome this problem. After assessing the patient, the helper inside the car determines that the patient is in an emergency and selects one of the several options. The selected mode is displayed on the OLED display to give the helper confidence in the selected option.

In 2020 Tugay Akca et al. [9] suggests a paper that chiefly emphasizes "IoT primarily based auto-deployment" techniques to manage auto and emergency services. This analysis is economical to hide all the items required to develop a sensible auto management framework, however, it lacks to elucidate how the system will add real-time to a mixture of mobile computing, cloud computing, and standalone applications. On-call-based machine management techniques to manage machine and emergency services. This analysis is economical to hide all the items required to develop a sensible machine management framework, however, cannot elucidate how the system will add real-time to a mix of mobile computing, cloud computing, and standalone applications. Remote medical services mistreat cloud" techniques to manage machines and emergency services. This analysis is economical to hide all the items required to develop a sensible machine management framework however lacks to elucidate how the system will add real-time to a mix of mobile computing, cloud computing, and standalone applications. Advance a paper that principally emphasizes ambulance service supplier mistreatment automation application techniques to manage machine and emergency services. This analysis is economical to hide all the items required to develop a sensible machine management framework, however, cannot elucidate how the system will add real-time to a mix of mobile computing, cloud computing, and standalone applications.

In 2019 Rajeshwari S et al. [10] a discovery mechanism for V2V communication facilitated by traffic jams on the road is suggested. To evaluate the severity of the initial traffic jam, a fuzzy regulator was initially constructed with vehicle speed and business viscosity as the input and the original traffic position as the affair. Additionally, the position of indigenous business jam is accomplished supported by the massive sub-sample thesis test as the amount of original business jam of bordering cars is collected supported by V2V communication. To maximize the efficacy of business jam discovery findings, the business jam information is promptly communicated to the affected cars when it has been obtained. Internet of Vehicles (IoVs) communication technology can significantly improve the transmission of traffic information, whereas traditional technologies like electronic information boards, FM radios, and SMS cannot guarantee the effective dispersion of traffic information have suggested using IOT to automate gestures and discover business viscosity. This design includes IR detectors that monitor the viscosity of the business position on a certain side. Both IR detectors are placed at specific distances so that it is possible to simultaneously measure the viscosity of each side.

CONCLUSION

In India, traffic signals are fixed with a specified time for every signal regardless of the density of traffic in that lane. Ambulance service is one of the most important systems, and if it gets delayed the result is the death of the patient. To overcome this problem, we have designed this system which has three sub-systems i.e., to automatically change the traffic lights in the path of the ambulance, to help the driver with the nearest way to the hospital, and at last monitor the health of the patient and send real-time vital information to the nearest hospital. This project can be a real-life saver project which saves lives at the "golden hour".

REFERENCES

1. S Sarkar (2016). Ambulance assistance for emergency services using GPS navigation, *International Research Journal of Engineering and Technology*, 3(9), 1328-1331, Available at: https://www.irjet.net/archives/V3/i9/IRJET-V3I9246.pdf.
2. G. Varaprasad (2013). High stable power aware multicast algorithm for mobile Ad Hoc networks, *IEEE Sensors Journal*, 13(5), 1442-1446, Available at: https://doi.org/10.1109/JSEN.2012.2236552 .
3. H Nath Saha, N Firdaush Raun and M Saha (2017). Monitoring patient's health with smart ambulance system using internet of things (IoTs). *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*. IEEE, Available at: https://doi.org/10.1109/IEMECON.2017.8079568.
4. K. Priyadharshini and S.K. Manikandan (2019). Automatic traffic control system based on the vehicular density, *International Research Journal of Engineering and Technology*, 6(4), 66-72, Available at: https://www.irjet.net/archives/V6/i4/IRJET-V6I414.pdf.
5. M. Abirami, G. Archana, R. S. Dheepika and M. Keerthana (2021). GPS based traffic signal control system for ambulance using machine learning, *International Research Journal of Engineering and Technology*, 8(4), 228-231, Available at: https://irjet.com/archives/V8/i4/PIT/ICIETET-46.pdf.
6. H Nath Saha, A Konar, S Guha, et al (2020). Automatic ambulance system using IOT, *International Journal of Engineering Science Invention*, 9(5), 1-4, Available at: http://www.ijesi.org/papers/Vol(9)i5/Series-1/A0905010103.pdf.
7. D. B. Kadam and Vinay .S. Suryawanshi (2016). Patient health care and ambulance monitoring system using controller & wireless technology, *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 4(9), 32-34, Available at: https://www.ijireeice.com/upload/2016/september-16/IJIREEICE%207.pdf.
8. V Krishna Kota, N Kumar Mangali, T Kumar Kanakurthi, et al (2017). Automated accident detection and rescue system. *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, Available at: https://doi.org/10.1109/WiSPNET.2017.8300000.
9. T Akca, O Koray Sahingoz, E Kocyigit and M Tozal (2020). Intelligent ambulance management system in smart cities. *2020 International Conference on Electrical Engineering (ICEE)*. IEEE, Available at: https://doi.org/10.1109/ICEE49691.2020.9249959.

10. R Sundar, S Hebbar and V Golla (2015). Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection, *IEEE Sensors Journal*, 15(2), 1109-1113, Available at: https://doi.org/10.1109/JSEN.2014.2360288.

CITE THIS ARTICLE

Kalpavi C Y et al. (2023). Intelligent Ambulance Using IOT, *Journal of Electronics and Communication Systems*, 8(1), 7-12.

INTELLIGENT AMBULANCE USING IOT

Bhumika P, Darshan M Gowda, Isha V, M Phani Teja
Dept. of Electronics and Communication Engineering
Dayananda Sagar Academy of Technology and Management
Bangalore-Karnataka, India
[bhumika01][darshgowda67@][lishavkumar120][mphaniteja01] @gmail.com

Prof. Kalpavi CY
Asst Professor
Dept. of Electronics and Communication Engineering
Dayananda Sagar Academy of Technology and Management
Bangalore-Karnataka, India

Abstract— Traffic congestion is the primary cause of patient deaths when travelling to hospitals. Any number of factors, including accidents, road construction, a high number of pedestrians crossing the street, etc., can contribute to traffic congestion. In particular, in large cities like Bangalore, Delhi, Hyderabad, etc. People in these cities lead busy lives with little time for themselves and don't even consider making room for emergency vehicles like ambulances. This article offers a "Intelligent Ambulance using IOT" approach to address this issue as well as many others. This paper describes the use of a variety of technologies, including GPS for location, digital image processing for ambulance detection, automatically turning on signal lights when an ambulance is approaching, and monitoring patient health. It also describes the use of nodemcu microcontroller and ChatBot to telegram critical patient data, such as body temperature and heart rate, to the closest hospitals. By transporting patients to hospitals more quickly than is customary, the proposed system is employed to save lives. In addition, the patient monitoring system can use the DHT11 temperature and heart rate sensors to check on the victim's health and report the results to the hospital.

I. INTRODUCTION

The issue of road traffic congestion and delayed medical attention is a critical problem in India, which results in loss of life. According to statistics, every minute, a life is lost due to late responses by family members, hospital formalities, or traffic congestion. Due to the increasing population and busy lifestyle, people are unable to make way for ambulances, causing further delays. Conventional traffic control methods, such as warning signs and stop signs, are not always sufficient to solve the problem. The main aim of this project is to ensure that patients reach the hospital as soon as possible. Instead of accommodating increased traffic, the project seeks to control traffic on the road using emerging technologies such as networking and image processing to make traffic lights smarter. By reducing the waiting time at traffic lights, the system can save society billions of moneys annually. The system measures the traffic density at various signals and adjusts the traffic light signal delay accordingly. It also communicates with adjacent junction signals to control the traffic together depending on the density. The proposed system uses a cameras and sound sensors placed on four lanes of the road connected by traffic signals to identify the ambulance on the roads .If an ambulance is detected, the traffic signal in that lane turns green, regardless of which light it should display, and the signal on the opposite lane turns red. Additionally, the system uses a heartbeat rate sensor and body temperature sensor to sense the vital details of the patient and send the information to the nearest hospital using nodemcu. Overall, the "Intelligent ambulance using IoT" project aims to save lives by taking patients to the hospital faster than the traditional way. The system ensures that ambulances can bypass traffic congestion and arrive at the hospital quickly. The proposed solution can significantly reduce the response time and increase the chances of survival for patients, especially during the "golden hour."

II. LITERATURE SURVEY

In paper [1], we looked at the study of determining the quickest path and optimum route for an ambulance to take in order to get to the specific hospital. The development of the GPS system, for which the GPS tracker was constructed and installed with the ambulance to track the path, is summarized in this paper. Anyone may use this technology; the average person can even keep track of all the ambulances in his neighborhood. The proper use of GPS technology was also covered in this article, and in an emergency, it will be very helpful in locating nearby ambulances and locating the patient's address at the scene of the accident. The current challenge that many ambulance drivers are currently experiencing has led to the development of GPS technology in ambulances. Therefore, this technology establishes the quickest path using an algorithm to get to the target hospital in the shortest possible time. Two sides make up the system's overall architecture. One is the side of the ambulance that is needed, and the other is the side of the ambulance driver. Simply using a GPS sensor, the patient's side must keep track of the addresses of the available ambulances. The ambulance driver can then confirm the individual who requested the ambulance's precise location. The user will be informed of the cancellation of the request for an ambulance but will still receive many lists of nearby ambulances if the driver is too busy or unable to arrive at that moment. He will also be learning about the closest hospitals in the meanwhile.

Saah H. N et al. suggest a technique for integrating a health monitoring system in an ambulance, which aids in getting a current assessment of the patient's health problems prior to reaching the specific hospital. The patient's heart rate, pulse sensors, and body temperature are all included in the health monitoring system. The use of various health monitoring sensors for identifying these health-related conditions of the patient in the ambulance as they were being transported to the hospital was helpful in this paper [2]. We also observe the employment of temperature sensors, which are employed to determine the blood vessel temperature and the cardiac output. This paper demonstrates the usage of an IC LM35 sensor to determine the body's temperature. In this paper, we have seen how a new form of microcontroller is being used to create an internal health monitoring system for an ambulance. These sensors are connected to the microcontroller in order to obtain the patient's health conditions. The speed and performance of this were superior to older ones. Additionally, this microprocessor uses less power and operates quietly when compared to other microcontrollers. We learned how to use a microcontroller to construct a health monitoring system inside an ambulance through this paper.

Paper [3] cites The tracking of patients, personnel, authenticating persons, as well as information collection and sensing, are among the many benefits that IoT applications offer within the humanitarian sector. Three distinct components make up the architectural model shown in Paper [3]. Placed within the ambulance is the ambulance module. It consists of a microprocessor that is interfaced with different sensor units. The patient's heartbeat, temperature, blood glucose levels, and a number of other health indicators are also monitored. These statistics may be uploaded into the cloud after processing. The second module, which comprises of two cloud services, is the cloud or WIFI module. The Think Speak cloud platform allows users to upload numerical data. On Dropboxcloud, the image uploading is completed. The third module is the hospital module, which aids the doctor in keeping track of the health of the patient’s using information uploaded to the cloud. A straightforward application to get data from the cloud is also included in this module. The doctor will be able to organise hospital accommodations for the patient and provide the emergency vehicle with appropriate feedback if these real-time data are made accessible to him.

According to Paper [4], there is currently no automation mechanism in place to handle traffic lights during crises. This research [4] advanced the characteristics of the previously successfully proposed traffic control system by adding a GPS module to the basic architectural model of an intelligent ambulance. This model is a fully automated system, and the driver must activate the emergency mode using a specially developed mobile application. With the aid of GPS, this activation will enable the app to track the ambulance's real-time location as well as the closest traffic signals. Every time the ambulance comes within 500 metres of a traffic signal, it immediately sends a request to control the lights that it must cross and then sends another signal to set it back. Based on time, the app immediately requests control to the closest control signal when the signal is already green. If the light is red, the app requests a control to open that specific door while the ambulance is moving by calculating the distance, again dependent on the amount of time left. This makes it possible for the ambulance to move about without being hit by a car. The server changes the signal based on the direction of the car. The system is enhanced in real-time so that the emergency death rate can be decreased.

The paper [5] focuses primarily on a projected phone safety observation system for old people who are using wireless body sensors, intelligent mobile devices, and elderly person health monitoring. If there is an emergency, a smart phone with an elderly person inside can wake up the people who have been allocated so they can prepare for a machine. With the use of this method, family members can communicate about the old person's health from a distance. A device that can track several physical features of different patient bodies is described in the advertising. The signals from the wireless sensors are collected on the patient's body by a base station-communicating organiser node. The tool may do more than just measure the physical parameters; it may also spot issues and notify the doctor accordingly. The patient has been admitted into the car, thus the process is currently being carried out with the assistance of the HMS. This HMS has bio-medical sensors, a display, and a 1x4 membrane keypad that are all designed to collect the patient's essential data. Pulse and vital sign sensors, fingerprint and vital sign readers, and other sensors are used in the proposed system. A degree OLED is used to show the selected mode. A family member or guardian may or may not be present to help after the patient is in the automobile.

III. METHODOLOGY

The methodology or the method used for designing the project mainly consists of steps. These steps come in order as these define the respective order of the methods used to construct this project.

A. HARDWARE SETUP

The NodeMCU ESP32, DHT11 sensor, TCRT100 Heartbeat Sensor, GPS Module, Jumper wires, L239 sensor, battery, and LED Display will all be part of the hardware setup. The microcontroller will serve as the main function and direct its motions in accordance with sensor data.

B. BLOCK DIARGAM

The three objectives outlined in the project aim to address the challenges faced by emergency vehicles, particularly ambulances, in reducing response times and providing timely medical attention. Let's examine each objective in more detail:

Implementing a GPS system in emergency vehicles: By equipping ambulances with GPS systems, drivers can efficiently navigate through unfamiliar areas and quickly locate patients' addresses. The GPS system also assists in finding the fastest and most optimal route to the hospital. This objective significantly reduces response times by ensuring that ambulances reach the destination promptly, thereby improving patient outcomes.

Creating a communication channel between hospital staff and ambulance drivers: Establishing effective communication between hospital staff and ambulance drivers is crucial for real-time updates on the patient's health and to better prepare medical staff for the patient's arrival. By having a direct channel of communication, medical professionals can receive vital information about the patient's condition, relay guidance or instructions to the ambulance crew, and make necessary preparations in advance. This objective reduces the time it takes for patients to receive medical attention and enhances the overall effectiveness of emergency medical services.

Establishing a traffic-free path to reach the hospital: Traffic congestion is a significant challenge that emergency vehicles face when trying to reach their destination swiftly. Utilizing digital image processing (DIP) as a tool to develop a traffic management system can address this issue. DIP can be employed to identify and track emergency vehicles, monitor and control traffic, forecast traffic congestion, and automatically detect incidents on the road. By leveraging DIP and processing images from cameras placed at strategic points, this objective aims to identify and track traffic flow, thereby facilitating the creation of a traffic-free path for emergency vehicles. This objective reduces response times, enhances emergency medical service effectiveness, and ensures that ambulances can reach hospitals without delays caused by traffic congestion.

By accomplishing these objectives, the project aims to optimize emergency medical services by improving navigation, communication, and traffic management. This comprehensive approach can significantly reduce response times, enhance patient outcomes, and revolutionize the way emergency vehicles operate in critical situations.

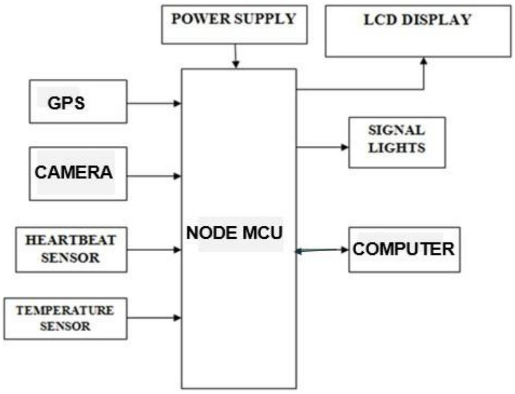


Figure 1. Block Diagram.

C. ALGORITHM

1. Initialization: All of the sensors, and other hardware components will be initialized by the ESP32 microcontroller.

2. Tracking: Using GPS technology, the ambulance finds the shortest path to the nearest hospital. This also helps in tracking the ambulance location with respect the patient.

3. Ambulance Detection: Based on digital image processing, the ambulance can be identified amongst the other vehicles on road, using YOLO algorithm.

4. Movement Control: Once the ambulance is detected ,the movement of the vehicles is controlled in the signal junction and henceforth the signal toggles in order to allow free movement of the ambulance.

5. Health monitoring: The patient's health condition, that is the temperature and the heart beat, is monitored using temperature sensor and the heart beat sensor. These parameters are monitored until the patient reaches the hospital.

6. Communication Control: The monitored health parameters are communicated to the hospital through telegram application over the ChatBot from the ambulance.

7. Refinement: The smart ambulance model can be further improved in performance based on the current real-time scenario.

8. Deployment: After the model has been successfully tested and improved, it can be used in actual crisis situations.

All these steps can be efficiently done using a certain algorithm which must be followed throughout the work by the robot. The basic algorithm is given below.

- START
- red=0,green=1;
- ambdetect=0;
- while(abmdect=0)
- SigState = 0;
- delay(5);
- SigState=1;
- Endwhile
- If(ambdect=1)
- LCD display " Ambulance detected on x road";
- lcd.clear()
- Delay(2)
- SigState=1
- LCD display " x road green signal";
- lcd.clear()
- LCD display "traffic clearance";
- For n=0;n<5;n++;
- lcd.clear()
- tempC = HT.readTemperature();
- LCD display "Temperature=tempC";
- lcd.clear();
- Heatb=HB.readHeartbeat();
- endfor

The basic flow diagram for the above algorithm is given below:

Figure 3. Flow diagram of the algorithm.

IV. EXPERIMENTAL RESULTS

The signal light is initially configured such that every lane has a default time duration of 5 seconds. Digital image processing is used to identify the emergency vehicle, Whenever Digital image processing recognizes emergency vehicles on ,the lane traffic signal in that lane turns green in contrast the opposite lane turns red. The project's overall view is depicted in the following image, which also includes a traffic management system, a webcam to implement digital image processing, a temperature sensor, a heart rate sensor, a GPS module, and a nodeMCU microcontroller board.

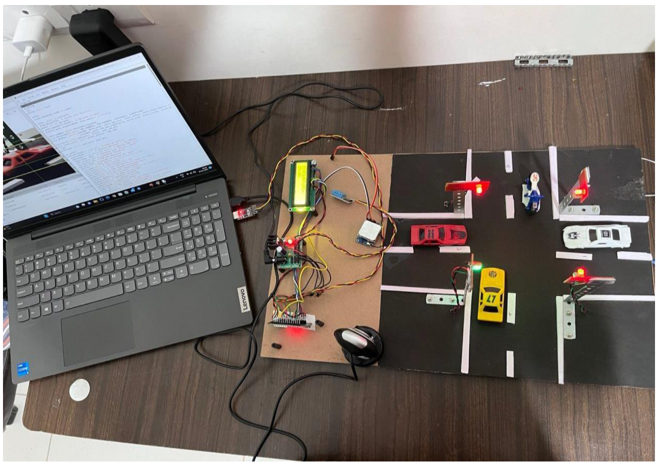


Figure 4. Overview of the project

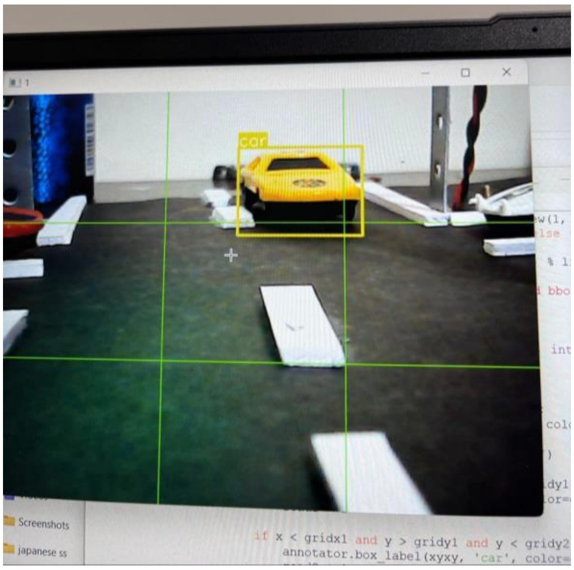


Figure 5. Detection of the Car by DIP

A. The study and manipulation of digital images through the application of computer algorithms is known as digital image processing. Digital image processing is used to recognize vehicles (bike, car, Ambulance, etc.) as seen in the figures above. To accomplish this task, machine learning techniques are employed to train computers to detect various vehicles based on their look in digital images. Typically, there are numerous steps to this procedure, including picture segmentation, feature extraction, and classification.

Two popular medical gadgets used to track a patient's health are temperature and heart rate sensors. These sensors are affixed to the patient's body and transmit data to a monitoring system in real-time. This data is then transmitted to hospitals where medical staff monitors the patient's vital signs, diagnoses any issues, and ensures the necessary .

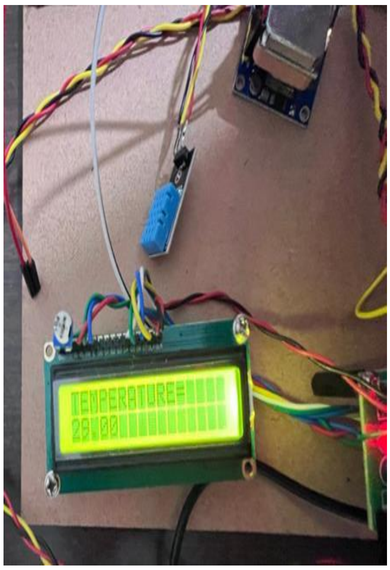


Figure 8. Temperature Sensor updating in realtime

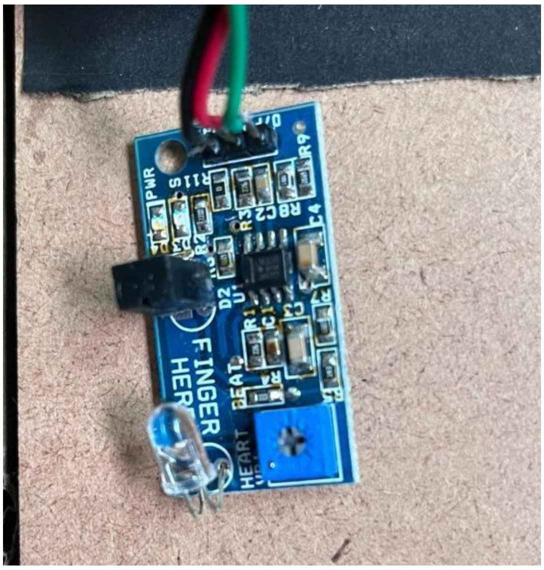


Figure 9. Heartbeat Sensor

While the temperature sensor measures the patient's body temperature and displays it in Celsius.

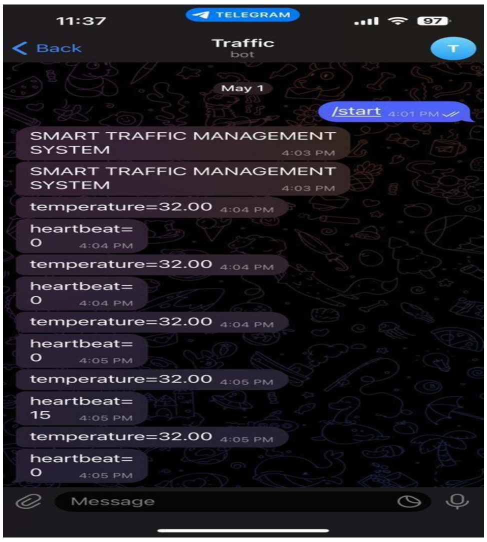


Figure 10. Telegram message

V. CONCLUSION

The proposed methodology combining YOLO Version 5 with GPS Ambulance Tracking System and traffic flow control indeed holds significant potential for enhancing emergency response times and patient outcomes. By integrating a state-of-the-art convolutional neural network like YOLO Version 5, the system can accurately and efficiently detect objects in real-time, allowing for quick identification and clearance of traffic obstacles for ambulances.

The integration of a GPS Ambulance Tracking System provides valuable information about the location and status of ambulances, enabling centralized control and coordination of emergency vehicles. This feature allows for optimized routing and traffic flow control, ensuring ambulances reach their destinations faster and more safely.

Moreover, the additional functionalities of the intelligent ambulance, such as health status detection and communication with hospitals via nodemcu, contribute to improved patient care. The system can monitor vital signs and relay crucial information to healthcare professionals, enabling them to make informed decisions and prepare for the arrival of the patient. Seamless communication with hospitals facilitates timely updates and coordination, leading to more efficient and effective medical interventions.

The incorporation of digital image processing and automatic mode design enhances the system's efficiency and usability. Leveraging advanced technologies in these areas streamlines the workflow and reduces manual intervention, enabling emergency responders to focus more on critical tasks.

Overall, the proposed methodology has the potential to revolutionize emergency medical services by leveraging cutting-edge technologies. It addresses key challenges in emergency response, including traffic congestion, real-time object detection, and communication with healthcare facilities. By enhancing response times and optimizing patient care, this system holds promise for significantly improving emergency outcomes.

VI. REFERENCES

[11] Shantanu Sarkar, Ambulance Assistance for Emergency **Services Using GPS Navigation** in International Research Journal of Engineering and Technology (IRJET), September,2016.

[12] G.araprasad, "High stable power aware multicast algorithm for mobile ad hoc networks," "IEEE Sensors J.", Volume: 05 Issue: 09 | Sep 2018, page no:423-426.

[13] Saha, H. N., Raun, N. F., & Saha, M. (2017, August). Monitoring patient's health with **smart ambulance system using Internet of Things (IOTs)**. In 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON) (pp. 91-95). IEEE.

[14] Saurabh Kapoor, Parul Gupta, Pooja Sharma, Prabhu Nath Singh **Intelligent Ambulance with Automatic Traffic Control** ,International Research Journal of Engineering and Technology (IRJET) Volume:04 Issue:04 ,2017,page no-1264-1266

[15] M. Abirami, G. Archana, R. S. Dheepika ,M. Keerthana "**GPS BASED TRAFFIC SIGNAL CONTROL SYSTEM FOR AMBULANCE USING MACHINE LEARNING"** , International Research Journal of Engineering and Technology(IRJET) Volume: 08 Issue: 04, 2021,page no-228-231

[16] Himadri Nath Saha , Arpita Konar, Shreya Guha , Avik Mukherjee , Shayeli Sarker , Saunak Chowdhury , Saanya Khan , Gourav Saha, Rituparna Guchhait , Aniket Mittra , Sanhita Bishal , Srutayu Biswas **Automatic Ambulance System using IOT** International Journal of Engineering Science Invention (IJESI), Volume 9 Issue 5 Series I ,May 2020 , PP 01-04

[17] Prof. Dr. Deepak .B. Kadam, Prof. "**Patient Health Care & Ambulance Monitoring System using Controller & Wireless Technology**", IJIREEICE, Vol. 4, Issue 9, September

[18] Venkata Krishna Kota, Nagendra Kumar Mangali, Thirumal Kumar Kanakurthi, "**Automated Accident Detection and Rescue System**", WiSPNET Conference, 2017.

[19] Tugay Akca, Emre Kocyigit, Ozgur Koray Sahingoz and Mucahid Tozal., "**IntelligentAmbulance Management System in Smart Cities**.," in IEEE-2020.

[20] Rajeshwari S., Santhoshs Hebbar, Varaprasad Golla: "**Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance and Stolen Vehicle Detection**", IEEE Sensors Journal, 2013, Vol 10, pp. 1530-437X(c).

BIOGRAPHY

Kalpavi C Y, Presently working as an Assistant Professor at Dayananda Sagar Academy of Technology and Management, Bangalore with an experience of 15+ years in teaching. Completed BE in Electronics and Communication Engineering, in 2006 from GMIT, Davangere. Obtained her M.Tech in DECS stream from MCE, Hassan, under VTU University. She has published 7 papers in International Journals, participated in 8 international conferences. She attended 50+ workshops, Seminars, Webinars and FDPs. Her area of interest are Sensor Networks,Computer,Networking,IoT, Wireless Sensor Networks etc.

Bumika P is attending Dayananda Sagar Academy of Technology and Management in Karnataka, India to complete his undergraduate degree in electronics and communication engineering. Her current focus involves working with embedded systems, VLSI, and machine learning. She is currently in his last year. She worked effectively in a team of four to undertake research and development on an audible glove utilizing Arduino.

Darshan M Gowda is a 21-year-old Electronics and Communication Engineering student from Banglore who is now enrolled in his fourth year of study at the Dayananda Sagar Academy of Technology and Management. He is an energetic and perceptive person who constantly adopts a humble attitude when working to accomplish his goals. Pavan is renowned for his upbeat outlook and commitment to achieving both his academic and personal objectives

Isha V is an engineering student at Dayananda Sagar Academy of Technology and Management who is currently 22 years old and in his fourth year of engineering. She is working on a second scholarly project after finishing the first one. Isha V is renowned for being a modest and outgoing individual who enjoys interacting with others. He consistently works hard and is goal-oriented in his pursuit of his goals

M Phani Teja is a student at Dayanand Sagar Academy of Technology and Management who is pursuing electronics and communication engineering in his fourth year. He has high aspirations for his future. A skilled individual and productive teammate, this 21-year-old boy has developed over the course of four years of projects, seminars, and research. Rajeev firmly believes in the importance of social work and feels that helping people improve their life is more valuable than any financial reward.