AUTOMATIC REAL-TIME BRIDGE STRUCTURAL HEALTH MONITORING SYSTEM

A PROJECT REPORT

Submitted By

Sayan Paul, Roll No.12617002070, Reg. No. 171260120044

Abhinaba Biswas, Roll No. 12617002063, Reg. No. 171260120037

Mitrajeet Dutta, Roll No. 12617002065, Reg. No. 171260120039

Under the Supervision of

Prof. Hemanta De

Department of Information Technology

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

An Autonomous Institute under

Maulana Abul Kalam Azad University of Technology

Formerly Known as

West Bengal University of Technology

July, 2020

ACKNOWLEDGEMENT

We would like to take this opportunity to thank Prof. Siuli Roy, Head of The Department of Information Technology for giving us the opportunity to work for this project and providing us with all necessary facilities required for this project.

We would also like to thank Prof. Hemanta De, our project mentor for constantly supporting us and guiding us throughout the project. His guidance and his words of encouragement motivated us to achieve our goal and impetus to excel.

We thank all the teachers, faculties and lab assistants of the IT department of Heritage Institute of Technology for playing a pivotal role during the development of this project. Last but not the least, all our friends and peers for their constant encouragement throughout.

Last but not the least I would like to express my deep gratitude and sincerest thanks to all the faculty members of Ogma Techlab Pvt. Ltd for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

Sayan Paul

Abhinaba Biswas

Mitrajeet Dutta

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

An autonomous Institute under

Maulana Abul Kalam Azad University of Technology

Formerly Known as

West Bengal University of Technology

BONAFIDE CERTIFICATE

Certified that this project report "**AUTOMATIC REAL-TIME BRIDGE STRUCTURAL HEALTH MONITORING SYSTEM"** is the bonafide work of "**SAYAN PAUL, ABHINABA BISWAS, MITRAJEET DUTTA"** who carried out the project work under my supervision.

SIGNATURE

Prof. (Dr.) Siuli Roy

HEAD OF THE DEPARTMENT

Dept. of Information Technology

Heritage Institute of Technology

Kolkata-700107

SIGNATURE

Prof. Hemanta De

ASSISTANT PROFESSOR

PROJECT MENTOR

Dept. of Information Technology

Heritage Institute of Technology

Kolkata-700107

SIGNATURE

EXTERNAL EXAMINER

ABSTRACT

Automatic Real-Time Bridge Structural Health Monitoring System is significant to health diagnosis of bridges and flyovers. This report is proposed and developed a novel architecture for large span bridge monitoring. A 3-level distributed structure is adopted in the monitoring system, which includes Central server, Intelligent Acquisition nodes and Local controller. Acquisition Nodes or The Sensors are located across the bridge. All the acquisition nodes are managed by one local controller. Every acquisition node samples different parameters of the bridge. Signals of various sensors are analysed & then processing results are sent to the local controller through wireless networks. IoT is utilized to provide enough bandwidth for real time data transmission between local controller & the central server. The sensors monitor cracks, bend, water level, sustainability, no. of vehicles moving over the bridge etc. At any time if any of these parameters cross their threshold value the system giving an alarm by sending the location for taking precautionary measures. This technology can be called Monitoring based maintenance that enables the highway bridge maintenance authorities monitor the condition of it in real time. We also want to add another feature which is if any vehicle breaks the load capacity & speed rule of the bridge then the system will capture the image of that vehicle & from that the vehicle number as well as the information about the owner of the vehicle can be extracted.

Keywords: Bridge, Structures, Bridge Monitoring, Maintenance, IoT, Sensors,

Image Processing, Vehicle, Number Plate Extraction

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

Table of Contents

Chapter 1: Introduction

1.1 What is Bridge?

A **Bridge** is a structure built to span a physical obstacle, such as a body of water, valley, or road, without closing the way underneath. It is constructed for the purpose of providing passage over the obstacle, usually something that can be detrimental to cross otherwise. There are many different designs that each serve a particular purpose and apply to different situations. Designs of bridges vary depending on the function of the bridge, the nature of the terrain where the bridge is constructed and anchored, the material used to make it, and the funds available to build it.

Figure 1: Bridge

Most likely the earliest bridges were fallen trees and stepping stones, while Neolithic people built boardwalk bridges across marshland. The Arkadiko Bridge dating from the 13th century BC, in the Peloponnese, in southern Greece is one of the oldest arch bridges still in existence and use [14].

1.2 Types of Bridges

Bridges can be categorized in several different ways. Common categories include the type of structural elements used, by what they carry, whether they are fixed or movable, and by the materials used [14].

1.2.1 Structure Types

Bridges may be classified by how the actions of tension, compression, bending, torsion & shear are distributed through their structure.

1.2.1.1 Beam Bridge

Beam Bridges are horizontal beams supported at each end by substructure units & can be either simply supported when the beams only connect across a single span or continuous when the beams are connected across two or more spans.

Figure 2: Beam Bridge

World's longest beam bridge is **Lake Ponchartrain Causeway** in Louisiana, United States. Beam bridges are simplest, oldest & popular type of bridge in use today.

1.2.1.2 Truss Bridge

A Truss Bridge is a bridge whose load-bearing superstructure is composed of a truss. This truss is a structure of connected elements forming triangular units. Truss Bridges are one of the oldest types of modern bridges.

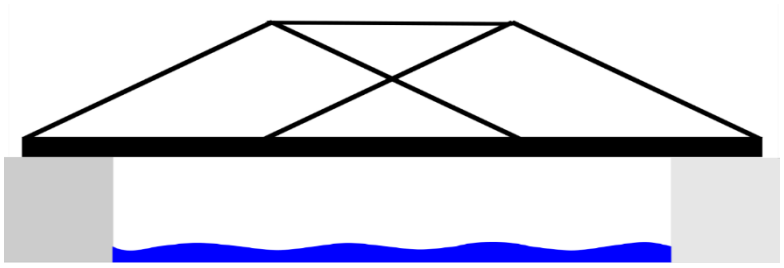

Figure 3: Truss Bridge

1.2.1.3 Cantilever Bridge

Cantilever Bridges are built with cantilevers-horizontal beams supported on only one end. Most

Cantilever bridges use a pair of continuous spans that extend from opposite sides of the supporting piers to meet at the center of the obstacle the bridge crosses.

Figure 4: Cantilever Bridge

The largest Cantilever bridge is the **Quebec Bridge** in Quebec, Canada.

1.2.1.4 Arch Bridge

Arch Bridges have abutments at each end. The weight of the bridge is thrust into the abutments at either side.

Figure 5: Arch Bridge

The world's largest Arch Bridge is the **Chaotianmen Bridge** in Chongqing, China.

1.2.1.5 Tied Arch Bridge

Tied Arch Bridges have an arch-shaped superstructure but differ from conventional arch bridges. Instead of transferring the weight of the bridge & traffic loads into thrust forces into the abutments, the ends of the arches are restrained by tension of the bottom chords of the structure.

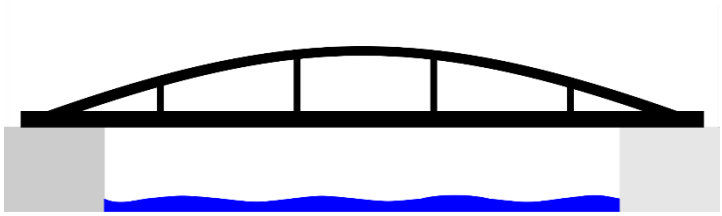

Figure 6: Tied Arch Bridge

1.2.1.6 Suspension Bridge

Suspension Bridges are suspended from cables. The cables hang from towers that are attached to caissons or cofferdams. The caissons or cofferdams are implanted deep into the bed of the lake, river or sea.

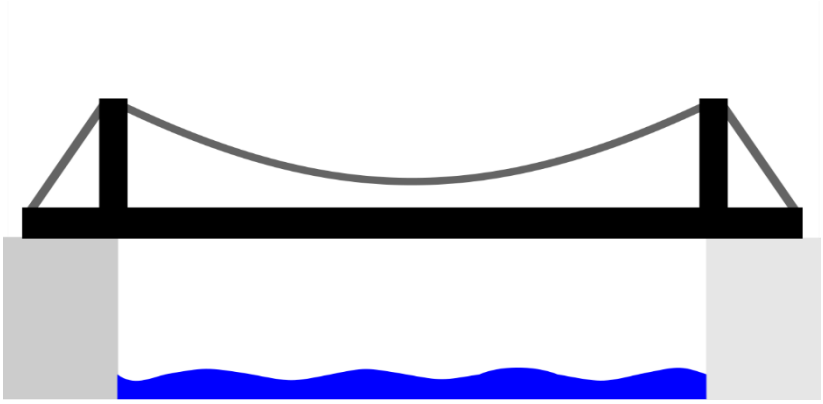

Figure 7: Suspension Bridge

The longest suspension bridge in the world is **Akashi Kaikyo Bridge** in Japan.

1.2.1.7 Cable-Stayed Bridge

Cable-Stayed Bridges like Suspension Bridges are held up by Cables. Here less cables are required & the towers holding the cables are proportionately higher.

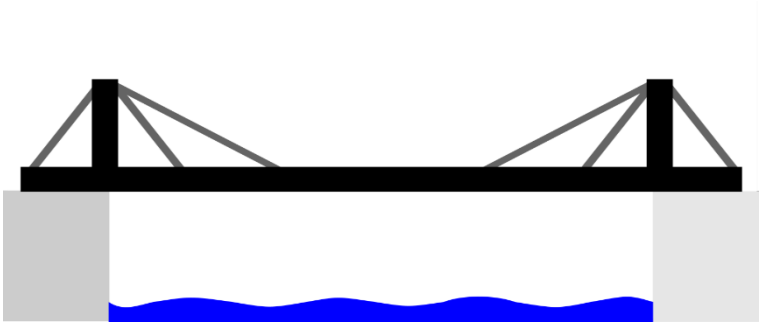

Figure 8: Cable-Stayed Bridge

The longest cable-stayed bridge is **Russky Bridge** in Vladivostok, Russia.

1.2.2 Fixed or Movable Bridges

Most bridges are fixed bridges, meaning they have no moving parts and stay in one place until they fail or are demolished. Temporary bridges, such as Bailey bridges, are designed to be assembled, and taken apart, transported to a different site, and re-used. Movable bridges are designed to move out of the way of boats or other kinds of traffic, which would otherwise be too tall to fit. These are generally electrically powered.

Figure 9: Movable Bridge

1.2.3 Double-Decked Bridges

Double-decked (or double-decker) bridges have two levels, such as the **George Washington Bridge**, connecting New York City to Bergen County, New Jersey, US, as the world's busiest bridge, carrying 102 million vehicles annually; truss work between the roadway levels provided stiffness to the roadways and reduced movement of the upper level when the lower level was installed three decades after the upper level.

Figure 10: Double-Decked Bridge

1.2.4 Viaducts

A Viaduct is made up of multiple bridges connected into one longer structure. The longest and some of the highest bridges are viaducts, such as the **Lake Pontchartrain Causeway** and **Millau Viaduct**.

Figure 11: Viaduct

1.2.5 Multi-Way Bridge

A Multi-way bridge has three or more separate spans which meet near the center of the bridge. Multi-way bridges with only three spans appear as a "T" or "Y" when viewed from above. Multi-way bridges are extremely rare. **The Tridge, Margaret Bridge**, and **Zanesville Y-Bridge** are examples.

Figure 12: Multi-Way Bridge

1.3 Bridge Maintenance

Bridge maintenance consisting of a combination of structural health monitoring and testing. This is regulated in country-specific engineer standards and includes an ongoing monitoring every three to six months, a simple test or inspection every two to three years and a major inspection every six to ten years. In Europe, the cost of maintenance is considerable and is higher in some countries than spending on new bridges. The lifetime of welded steel bridges

can be significantly extended by aftertreatment of the weld transitions. This results in a potential high benefit, using existing bridges far beyond the planned lifetime [14].

1.4 List of Notable Bridge Failures in 21st Century

Bridge	Location	Date	Reason	Casualties
Hintze Ribeiro Disaster	Portugal	4th March,2001	Pillar foundation became compromised due to illegal but permitted sand extraction & central span collapsed.	59 Killed
Pedestrian Bridge	Bhagalpur, India	Dec,2006	150 years old pedestrian bridge collapsed on to a train as it was passing underneath.	More than 30 Killed
Unknown	Guinea	March,2007	Bridge collapsed under the weight of a truck packed with passengers & merchandise.	65 Killed

Bridge	Location	Date	Reason	Casualties
Minneapolis I-35W Bridge	United States	1 Aug, 2007	Undersized Gusset plates & increased concrete surfacing load	13 Killed & 145 Injured
Chhinchu Suspension Bridge	Nepal	25th Dec, 2007	Overcrowded Suspension Bridge Collapsed	19 Killed & 15 Missing
Tarcoles Bridge	Costa Rica	22nd Oct, 2009	Overload by Heavy Trucks & Dead Loads	5 Killed & 30 Injured
Gongguan Bridge	China	14th July, 2011	Overloading	1 Killed & 22 Injured
Vivekananda Flyover Bridge	Kolkata, India	31st March, 2016	Bolts holding together a section of the bridge snapped.	27 Killed & 80+ Injured
Savitri River Bridge	Maharashtra, India	2nd Aug, 2016	Dilapidated Condition	28 Killed
Yellow 'Love' Bridge	Indonesia	16th Oct, 2016	Snapped Sling due to Overloading	9 Killed & 30 Injured
Sanvordem River Bridge	Goa, India	18th May, 2017	Dilapidated Condition	2 Killed & 30 Injured

Ponte Morandi Motorway Bridge	Italy	14th Aug, 2018	Improper Maintenance	43 Dead
Majherhat Bridge	Kolkata, India	4th Sep, 2018	Possibly had become too heavy & needed to shed load	3 Dead & 25 Injured
CST Foot over Bridge	Mumbai, India	14th March, 2019	The structural audit has been conducted in an irresponsible & Negligent manner.	6 Dead & 30 Injured

Table 1: List of Notable Bridge Failures in 21st Century [15]

1.5 Current Bridge Health Monitoring Techniques

Here are several methods used to monitor the condition of large structures like bridges. Many long-span bridges are now routinely monitored with a range of sensors. Many types of sensors are used, including strain transducers, accelerometers, tiltmeters, and GPS. **Accelerometers** have the advantage that they are inertial, i.e., they do not require a reference point to measure from. This is often a problem for distance or deflection measurement, especially if the bridge is over water.

An option for structural-integrity monitoring is "non-contact monitoring", which uses the **Doppler effect (Doppler shift)**. A laser beam from a Laser Doppler Vibrometer is directed at the point of interest, and the vibration amplitude and frequency are extracted from the Doppler shift of the laser beam frequency due to the motion of the surface.[74] The advantage of this method is that the setup time for the equipment is faster and, unlike an accelerometer, this makes measurements possible on multiple structures in as short a time as possible. Additionally, this method can measure specific points on a bridge that might be difficult to access. However, vibrometers are relatively expensive and have the disadvantage that a reference point is needed to measure from.

Snapshots in time of the external condition of a bridge can be recorded using **Lidar** to aid bridge inspection. This can provide measurement of the bridge geometry (to facilitate the building of a computer model) but the accuracy is generally insufficient to measure bridge deflections under load.

While larger modern bridges are routinely monitored electronically, smaller bridges are generally inspected visually by trained inspectors. There is considerable research interest in the challenge of smaller bridges as they are often remote and do not have electrical power on site. Possible solutions are the installation of sensors on a specialist inspection vehicle and the use of its measurements as it drives over the bridge to infer information about the bridge

condition. These vehicles can be equipped with accelerometers, gyro meters, Laser Doppler Vibrometers and some even have the capability to apply a resonant force to the road surface in order to dynamically excite the bridge at its resonant frequency.

1.6 Objective of The Project

Flyovers and highway bridge systems are critical in many regions, being used over several decades. It is critical to have a system to monitor the health of these bridges and report when and where maintenance operations are needed. Advancements in sensor technology have brought the automated real-time bridge health monitoring system. However, current system uses complicated and high cost wired network amongst sensors in the bridge and high cost optical cable between the bridge and the management centre. The complicated wiring also makes the installation and repair/replacement process difficult and expensive.

In this project an idea of bridge monitoring system using IoT is proposed. This technology can be called MBM (Monitoring Based Maintenance) that enables the bridge maintenance engineers monitor the condition of the bridge in real time. The sensors installed on various parts of the bridge monitors the bend, traffic, weight of the vehicles etc. At any point of time if any of these parameters cross their threshold value the communication system informs the management centre giving an alarm for taking precautionary measures.

The main objectives of our project are as follows:

- Design of Automatic Real-time Bridge Structural Health Monitoring System
- Provide 24*7 Camera Surveillance
- Monitor the Water level.
- Monitor Temperature & Humidity of the bridge.
- Monitor the Bending & the Vibrations of the bridge.
- Monitor the loads on the bridge.
- Count the no. of vehicles moving over the bridge.
- Transmit these above-mentioned parameters to the Control Station using Wireless Network.
- Display the parameters in the Control Station using any visual interface & analyse the parameters for real-time monitoring.
- At any point of time if any of these parameters cross their threshold value the system will alarm the Control Station for taking any precautionary measures.
- Capture the image of the vehicle which breaks the weight & speed laws.
- Extract the vehicle number plate from the image & identify the owner of that vehicle from that vehicle number to take actions against him/her.
- Construct simple & cost-effective design of the system.
- Provide high reliability of Bridge monitoring.

Chapter 2: Literature Survey

Author Name	Paper Name	Year	Publication
Ms.Arohi. D. Sonawane, Ms.Pooja. P. Vichare, Mr.Shubham. S. Patil and Mr.Nitin. P. Chavande	Bridge Monitoring System Using IoT	2018	In countries like India there is powerful focus on national infrastructure. New bridges are built each year and the maintenance of those bridges is frequently ignored. The present structures use very complex and excessive fee wired network and it additionally required high upkeep for optical fibre machine. So the primary objective of this task is to build a cheap bridge tracking machine for developing international locations like India. This project aim to simplify the system for selecting bridge tracking devices. Many bridges within the India are obsolete or structurally deficient to safely increase the life of those bridges, inspection would be vital. Bridge engineers have many duties and it's far not possible to expect one to know. Our device will sense the crack inside the bridge and signal might be given to govern room immediately to stop cars [1].
Varsha Kusal, Amrita Argade, Sanika Chiplunkar, Rohini Kumbhar, Swati A. Khodke	Bridge Monitoring and Alert Generation System Using IOT	2017	Many of the bridges in cities built on the river are subject to deterioration as their lifetime is expired but they are still in use. They are dangerous to bridge users. Due to heavy load of vehicles, high water level or pressure, heavy rains these bridges may get collapse which in turn leads to disaster. So, these bridges require continuous monitoring. So we are proposing a system which consists of a weight sensor, water level point contact sensor, Wi-Fi module, and Arduino microcontroller. This system detects the load of vehicles, water level, and pressure. If the water level, water pressure and vehicle load on the bridge cross its threshold value then it generates the alert through buzzer and auto barrier. If it is necessary, then the admin assign the task to the employees for maintenance [2].

Author Name	Paper Name	Year	Publication
Amro Al-Radaideh , A. R. Al-Ali, Salwa Bheiry , Sameer Alawnah	A Wireless Sensor Network Monitoring System for Highway Bridges	2015	As wireless smart sensor networks and geographical information systems (GIS) are evolving nowadays, applications of remote monitoring in wide spread geographical areas are becoming cost-effective and possible. An example of such applications is the structural health status monitoring of highway bridges that connect roads in both rural and urban areas. Many of these bridges are subject to deterioration due to external and internal factors. Online, real-time structural health monitoring is a resourceful complimentary tool to facilitate rapid field inspection. Bridge maintenance and infrastructure managers can easily use this application to safeguard the performance and safety of these vital structures. This paper presents an autonomous wireless sensor network system to monitor structural health in highways bridges. The proposed system consists of a wireless Data Acquisition Unit (DAQ), a mobile public network, a structural health data evaluation, a management middleware, a GIS and graphical user interface module. The sensors in the DAQ gather the bridge health signs and transmit them promptly via the public mobile networks to the management and evaluation middleware for further processing. Based on the national bridge inventory rating scale, an early warning fuzzy logic based engine is developed to process the status of a given bridge and alert the concerned operator/s regarding any abnormality. Furthermore, an interactive Google map is used to show the status of each bridge along with its exact location. A prototype was built in the laboratory to validate the proposed system. Analysis of testing results and comparisons with

			existing monitoring systems are also discussed. Operators can access the bridge real-time data through mobile phone. The system is cost effective and user friendly [3].
Snehal Sonawane, Nikita Bhadane, Sayali Zope, Ashitosh Pangavhane and V. S. Tidake	Design of Bridge Monitoring System based on IoT	2018	Bridges may get collapsed or tilted due to flooding or some concrete problem, natural calamities. So there is a need to design a system which will continuously monitor condition of bridges. It is useful for public safety and reduction in human losses. Such system will help in disaster management and recovery. IoT based bridge safety monitoring system is developed using the WSN Technology. This system is composed of: Monitoring devices installed in the bridge environment, communication devices connecting the bridge monitoring devices and the cloud based server, a dynamic database that stores bridge condition data, cloud based server calculates and analyzes data transmitted from the monitoring devices. This system can monitor and analyze in real time the condition of a bridge and its environment, including the water levels and other safety conditions. This paper presents a comprehensive survey of SHM using WSNs outlining and algorithm like damage detection and localization, network design challenges and future research direction [4].
Jin-Lian Lee, Yaw-Yauan Tyan, Ming-Hui Wen, Yun-Wu Wu	Development of an IoT-based Bridge Safety Monitoring System	2017	IoT-based bridge safety monitoring system is developed using the ZigBee technology. This system is composed of: monitoring devices installed in the bridge environment; communication devices connecting the bridge monitoring devices and the cloud-based server; a dynamic database that stores bridge condition data; and a cloud-based server that calculates and analyzes data transmitted from the monitoring devices [5].

Chapter 3: System Description

3.1 System Components

3.1.1 Arduino Uno

The **Arduino UNO** is an open-source microcontroller board based on the Micro chip ATmega 328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of the Arduino Software. The Uno board is the first in a series of USB-based Arduino boards, and it and version 1.0 of the Arduino IDE were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes pre-programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter [37].

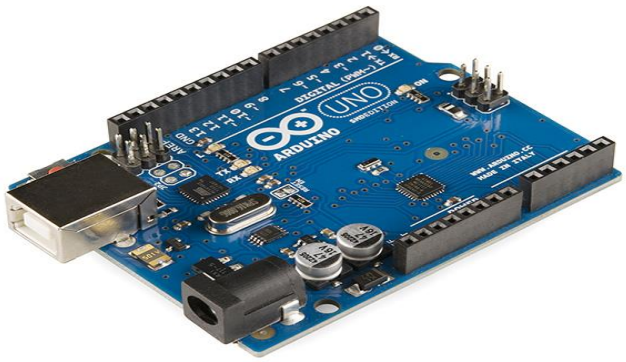

Figure 13: Arduino Uno

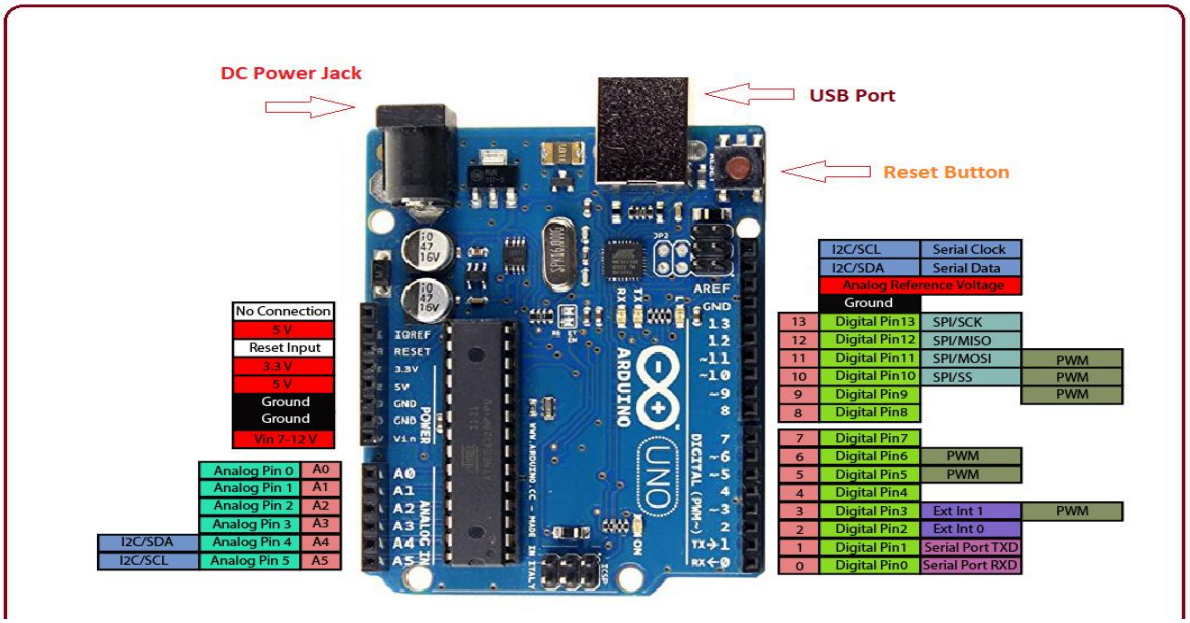

Figure 14: Arduino Uno Pin Diagram

3.1.2 Ultrasonic Sensor

The **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Figure 15: How Ultrasonic Sensor Works?

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor [16].

Figure 16: HC-SR04 Ultrasonic Sensor & Pin Diagram

3.1.3 Temperature & Humidity Sensor

The **DHT11** is a commonly used **Temperature and humidity sensor.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers [17].

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you.

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

The DHT11 measures temperature with a surface mounted NTC temperature sensor (thermistor) built into the unit [18].

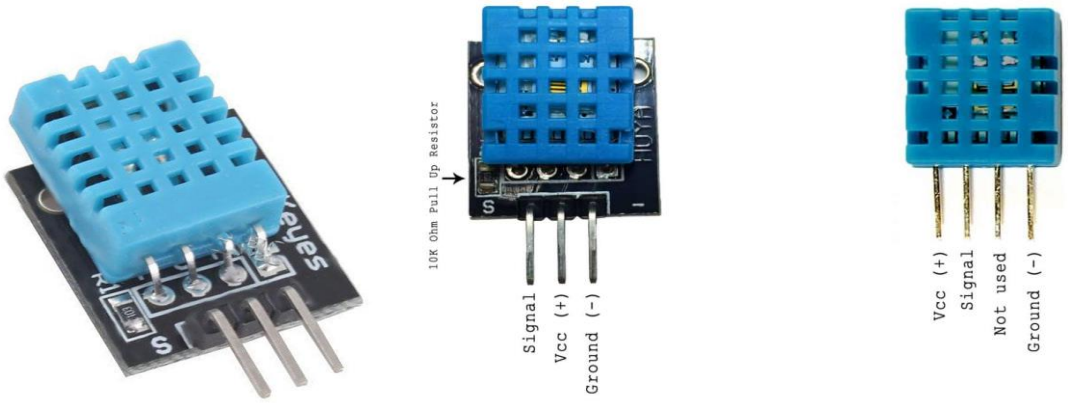

Figure 17: DHT11 Temperature & Humidity Sensor & Pin Diagram

3.1.4 Vibration Sensor

The **SW-420 Vibration Module**, which can work from 3.3V to the 5V. The sensor uses LM393 comparator to detect the vibration over a threshold point and provide digital data, Logic Low or Logic High, 0 or 1. During normal operation, the sensor provides Logic Low and when the vibration is detected, the sensor provides Logic High**.** There are three peripherals available in the module, two LEDs, one for the Power state and other for the sensor's output. Additionally, a potentiometer is available which can be further used to control the threshold point of the vibration. In this project, we will use 5V to power the module [19].

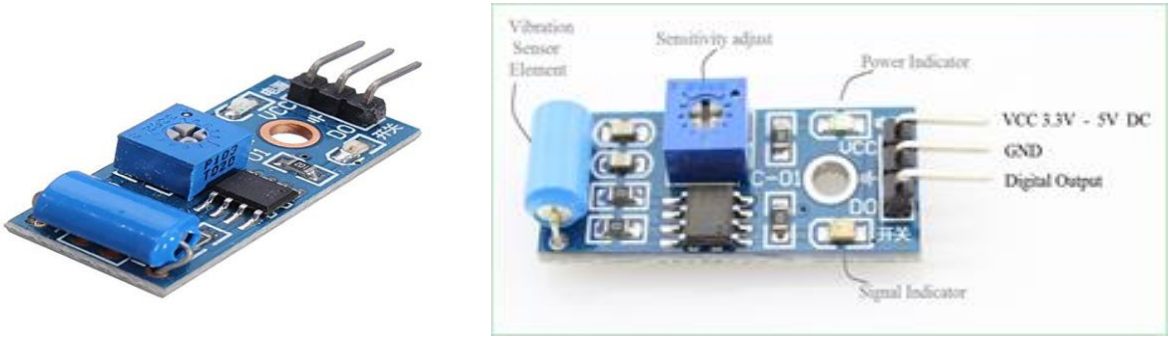

Figure 18: SW-420 Vibration Sensor & Pin Diagram

3.1.5 Flex Sensor

A **Flex Sensor** is a kind of sensor which is used to measure the amount of defection otherwise bending. The designing of this sensor can be done by using materials like plastic and carbon. The carbon surface is arranged on a plastic strip as this strip is turned aside then the sensor's

resistance will be changed. Thus, it is also named a bend sensor. As its varying resistance can be directly proportional to the quantity of turn thus it can also be employed like a goniometer.

These sensors are classified into two types based on its size namely 2.2-inch flex sensor & 4.5-inch flex sensor. The size, as well as the resistance of these sensors, is dissimilar except the working principle. Therefor the suitable size can be preferred based on the necessity. Here this article discusses an overview of 2.2-inch flex-sensor. This type of sensor is used in various applications like computer interface, rehabilitation, servo motor control, security system, music interface, intensity control, and wherever the consumer needs to modify the resistance throughout bending.

This sensor works on the bending strip principle which means whenever the strip is twisted then its resistance will be changed. This can be measured with the help of any controller. This sensor works similar to a variable resistance because when it twists then the resistance will be changed. The resistance change can depend on the linearity of the surface because the resistance will be dissimilar when it is level. When the sensor is twisted 450 then the resistance would be dissimilar. Similarly, when this senor is twisted to 900 then the resistance would be dissimilar. These three are the flex sensor's bending conditions. According to these three cases, the resistance will be normal in the first case, the resistance will be double as contrasted with the first case, and the resistance will be four-time when compared with the first case. So the resistance will be increased when the angle is increased [20].

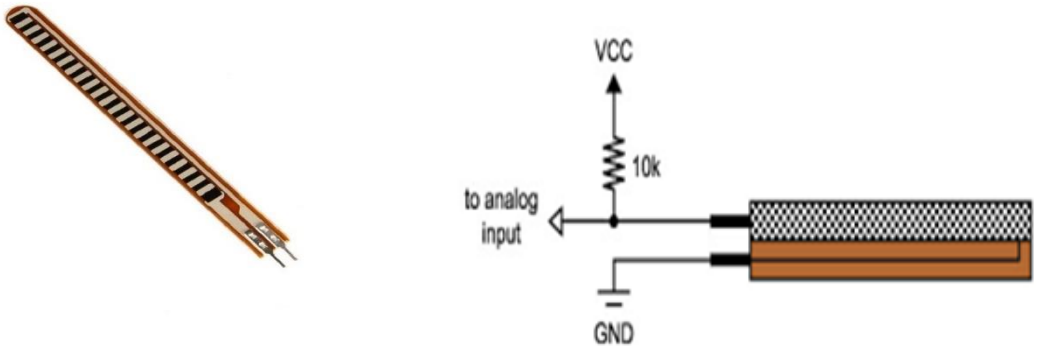

Figure 19: Flex Sensor & Pin Diagram

3.1.6 Load Cell

A **Load Cell** is a transducer that is used to create an electrical signal whose magnitude is directly proportional to the force being measured. The various types of load cells include hydraulic load cells, pneumatic load cells and strain gauge load cells.

Strain gauge load cells convert the load acting on them into electrical signals. The gauges themselves are bonded onto a beam or structural member that deforms when weight is applied.

The gauges are mounted in a differential bridge to enhance measurement accuracy. When weight is applied, the strain changes the electrical resistance of the gauges in proportion to the load [21].

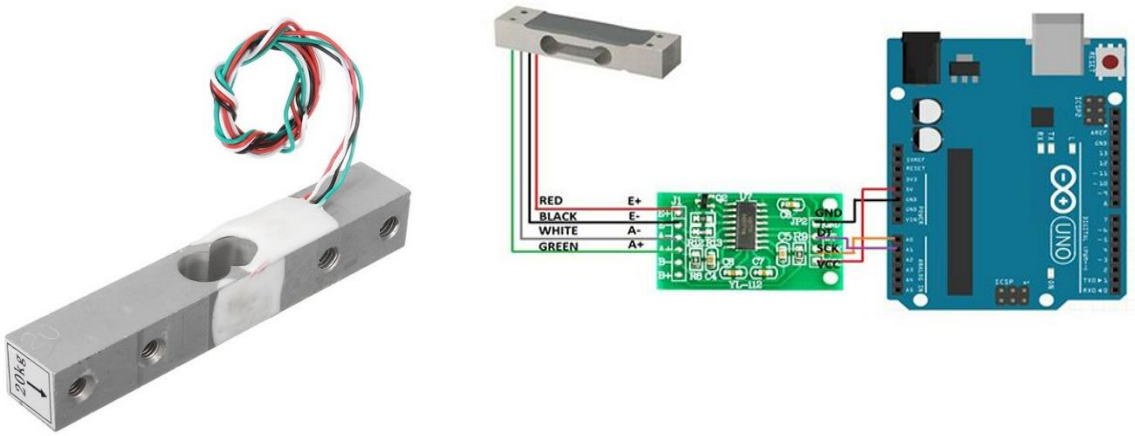

Figure 20: Load Cell & It's Interfacing using HX711 Load Cell Amplifier

3.1.7 Camera Module

OV7670 Camera Module is a FIFO camera Module available from different Manufacturers with different pin Configurations. The OV7670 provides full frame, windowed 8-bit images in a wide range of formats. The image array is capable of operating at up to 30 frames per second (fps) in VGA. The OV7670 includes

- Image Sensor Array (of about 656 x 488 pixels)
- Timing Generator
- Analog Signal Processor
- A/D Converters
- Test Pattern Generator
- Digital Signal Processor (DSP)
- Image Scaler
- Digital Video Port
- LED and Strobe Flash Control Output

The OV7670 image sensor is controlled using Serial Camera Control Bus (SCCB) which is an I2C interface (SIOC, SIOD) with a maximum clock frequency of 400KHz [22].

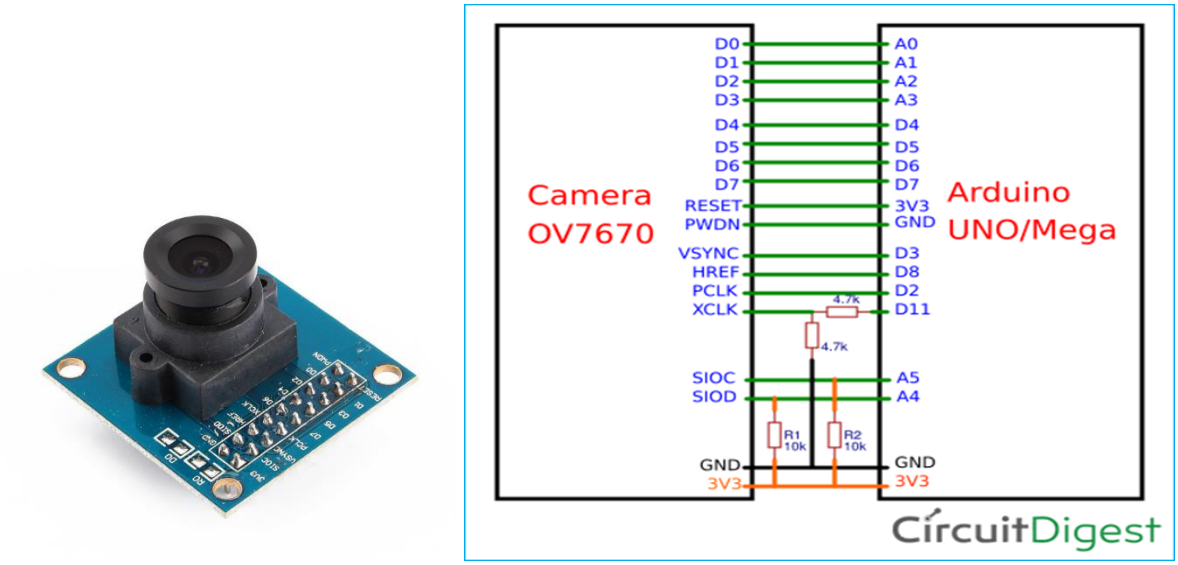

Figure 21: OV7670 Camera Module & It's Interfacing with Arduino Uno

3.1.8 Infrared Sensor

IR Sensor is a multipurpose infrared sensor which can be used for obstacle sensing, colour detection, fire detection, line sensing, etc and also as an encoder sensor. The sensor provides a digital output.

The sensor outputs a logic one(+5V) at the digital output when an object is placed in front of the sensor and a logic zero(0V), when there is no object in front of the sensor. An onboard LED is used to indicate the presence of an object. This digital output can be directly connected to an Arduino, Raspberry Pi, AVR, PIC, 8051 or any other microcontroller to read the sensor output.

IR sensors are highly susceptible to ambient light and the IR sensor on this sensor is suitably covered to reduce effect of ambient light on the sensor. The sensor has a maximum range of around 40-50 cm indoors and around 15-20 cm outdoors.

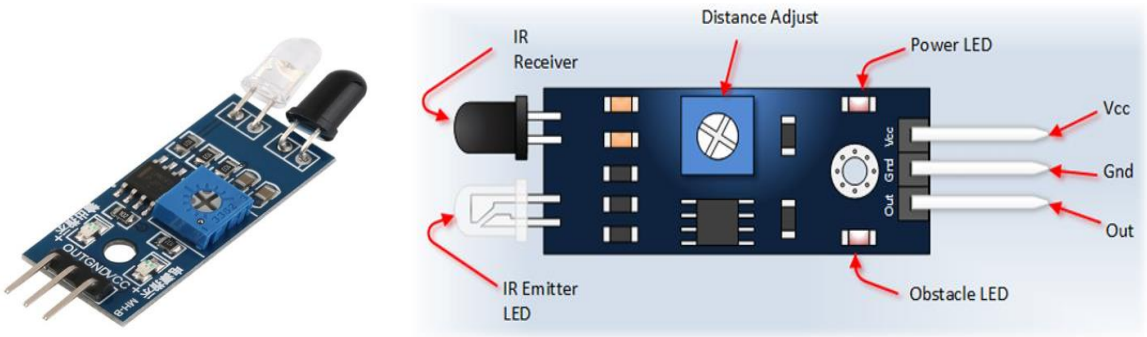

Figure 22: IR Sensor & Pin Diagram

3.1.9 Node MCU

Node MCU is an open source LUA based firmware developed for ESP8266 wi-fi chip. By exploring functionality with ESP8266 chip, Node MCU firmware comes with ESP8266 Development board/kit i.e. Node MCU Development board.

Since Node MCU is open source platform, their hardware design is open for edit/modify/build.

Node MCU Dev Kit/board consist of ESP8266 wi-fi enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 Wi-Fi Module. Node MCU Dev Kit has Arduino like Analog (i.e. A0) and Digital (D0-D8) pins on its board.

It supports serial communication protocols i.e. UART, SPI, I2C etc.

Using such serial protocols, we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc [37].

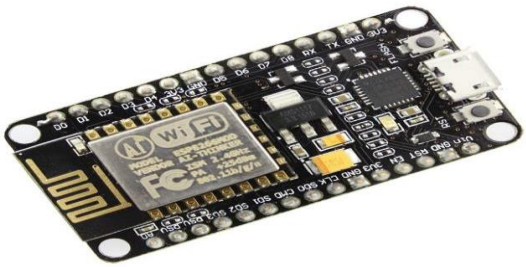

Figure 23: Node MCU

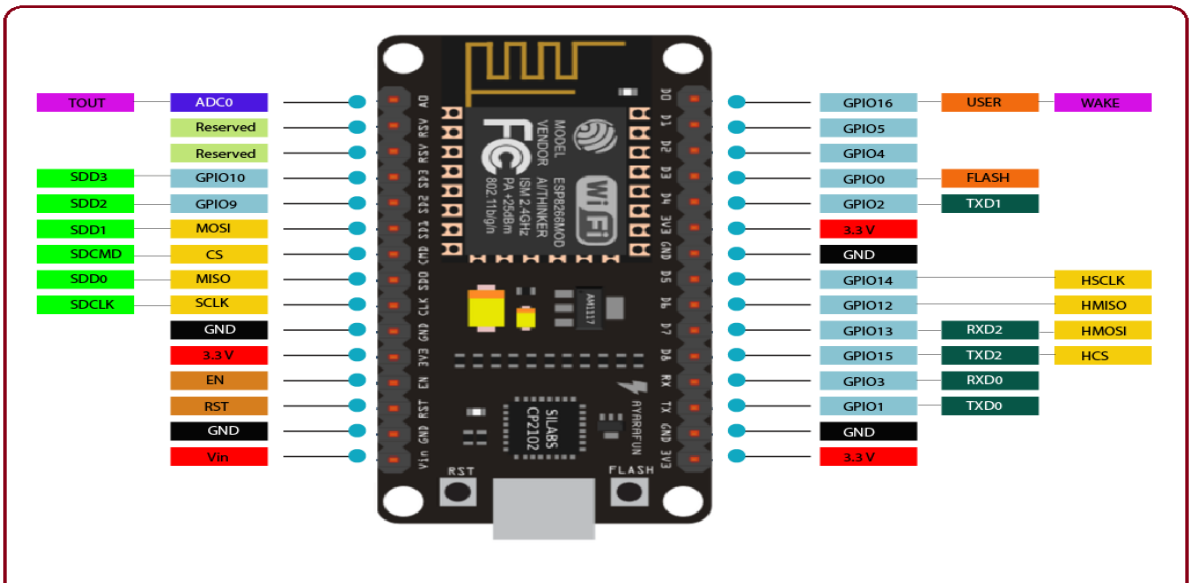

Figure 24: Node MCU Pin Diagram

3.1.10 Buzzer:

A Buzzer is a mechanical, electromechanical, magnetic, electromagnetic, electro-acoustic or piezoelectric audio signalling device. A piezo electric buzzer can be driven by an oscillating electronic circuit or other audio signal source. The items Listed here are for Hobby Electronics /DIY activities Only [29].

Figure 25: Buzzer

3.1.11 3.7 Volt 18650 Series Rechargeable Batteries

An 18650 battery is a cell that's 18mm x 65mm in size. The name, 18650, refers exclusively to the size of the lithium-ion battery cell, but there can be minor variations even here. The 18650 has become the new gold standard for replaceable and rechargeable batteries.

They offer the performance of a lithium-ion cell, a capacity in the range of 1800mAh to around 3500mAh, and an output of 3.7 volts. They're used in a huge range of devices from laptops to laser pointers, and camera accessories like gimbals and sliders [30].

Figure 26: 3.7 V 18650 Series Rechargeable Batteries

3.1.12 Software Components:

- **Arduino IDE:** The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
  This software can be used with any Arduino board [28].

- **Jupyter Notebook:** The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more [31].

- **Pytessaract:** Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images.

  Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file [32].

- **OpenCV:** OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products [33].

- **Python:** Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse [34].

- **Thingspeak:** ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics [35].

3.2 Proposed Mechanism

3.2.1 Description of Hardware Assembling

Step 1: Place Arduino Uno & Node MCU on the Bridge structure.

Step 2: Assemble HC-SR04 Ultrasonic Sensor & DHT11 Temperature & Humidity Sensor on the Bridge Structure.

Step 3: Mount SW-420 Vibration Sensor & Flex Sensor on the Structure of the Bridge.

Step 4: Place Load Cell & Infrared Sensor on the Bridge Structure.

Step 5: Mount OV7670 Camera Module on the Structure of the Bridge.

Step 6: Connect OV7670 Camera Module with Arduino by using Jumper wires & rest of the components with Node MCU.

Step 7: Attach Batteries to supply power to the system.

3.2.2 Description of Vehicle Number Plate Detection

Step 1: Image Acquisition

Step 2: RGB to Grayscale Conversion

Step 3: Noise Removal by Iterative Bilateral Filtering

Step 4: Contrast Enhancement by Using Histogram Equalization

Step 5: Image Thresholding

Step 6: Edge Detection by Canny Operator

Step 7: Candidate Plate Area Detection by Contour Detection

Step 8: Actual Number Plate Area Extraction

Step 9: Enhancement of Extracted Plate Region by Histogram Equalization

Step 10: Number Plate Character Recognition by OCR

Step 1: Image Acquisition

The general aim of Image Acquisition is to transform an optical image (Real World Data) into an array of numerical data which could be later manipulated on a computer, before any video or image processing can commence an image must be captured by camera and converted into a manageable entity. The Image Acquisition process consists of three steps:

1. Optical system which focuses the energy
2. Energy reflected from the object of interest
3. A sensor which measure the amount of energy.

Image Acquisition is achieved by suitable camera. We use different cameras for different application. If we need an x-ray image, we use a camera (film) that is sensitive to x-ray. If we want infrared image, we use camera which are sensitive to infrared radiation. For normal images (family pictures etc) we use cameras which are sensitive to visual spectrum. Image Acquisition is the first step in any image processing system [12].

Figure 27: Original Images

Step 2: RGB to Grayscale Conversion

Then the Original Image (RGB Image) is converted into Grayscale Image.

Figure 28: Grayscale Converted Images

Step 3: Noise Removal by Iterative Bilateral Filtering

A **Bilateral Filter** is used for smoothening images and reducing noise, while preserving edges. It explains an approach using the averaging filter, while this article provides one using a median filter. However, these convolutions often result in a loss of important edge information,

since they blur out everything, irrespective of it being noise or an edge. To counter this problem, the non-linear bilateral filter was introduced [23].

The bilateral filter can be formulated as follows:

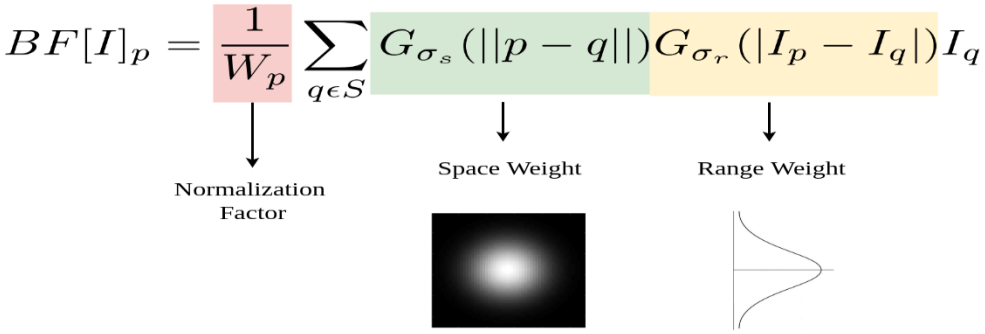

Figure 29: Formula of Bilateral Filter

Where,

σ_s : The size of the neighbourhood

σ_r : Minimum amplitude of an edge. It ensures that only those pixels with intensity values similar to that of the central pixel are considered for blurring while sharp intensity changes are maintained. The smaller the value, The sharper the edge. It tends to infinity.

I_q : Intensity of pixel q

$G\sigma_s$ (|| p-q ||) : Normalized Gaussian Function

Figure 30: Noise Removed Images

Step 4: Contrast Enhancement by Using Histogram Equalization

The **Histogram Equalization** is an approach to enhance a given image. The approach is to design a transformation T(.) such that the gray values in the output is uniformly distributed in

[0,1]. Let us assume for the moment that the input image to be enhanced has continuous gray values, with r=0 representing Black & r=1 representing White. We need to design a gray value transformation s=T(r) based on the histogram of the input image, which will enhance the image [38].

To implement Histogram Equalization, For images with discrete gray values, compute:

$$p_{in}(r_k) = \frac{n_k}{n} \quad 0 \le r_k \le 1 \quad 0 \le k \le L-1$$

L: Total number of gray levels

n_k: Number of pixels with gray value r_k

n: Total number of pixels in the image

Then, Based on CDF (Cumulative Distribution Function) , compute the discrete version of the previous transformation:

$$s_k = T(r_k) = \sum_{j=0}^{k} p_{in}(r_j) \qquad 0 \le k \le L-1$$

Figure 31: After Histogram Equalization

Step 5: Image Thresholding

Thresholding is a technique in OpenCV, which is the assignment of pixel values in relation to the threshold value provided. In thresholding, each pixel value is compared with the threshold

value. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value (generally 255). Thresholding is a very popular segmentation technique, used for separating an object considered as a foreground from its background. A threshold is a value which has two regions on its either side i.e. below the threshold or above the threshold [25].

Otsu Thresholding

In Otsu Thresholding, A value of the threshold isn't chosen but is determined automatically. A Bimodal image (two distinct image values) is considered. The Histogram generated contains two peaks. So, a generic condition would be to choose a threshold value that lies in the middle of both the histogram peak values [13].

Here, the image is segmented into two light and dark regions T0 and T1, where region T0 is a set of intensity level from 0 to t or in set notation T 0 = {0, 1, ..., t} & region T 1 = {t, t + 1, ..., l − 1, l} where t is the threshold value, l is the image maximum gray level (for instance 256). T0 and T1 can be assigned to object and background or vice versa (object not necessarily always occupies the light region). Otsu's thresholding method scans all the possible thresholding values and calculates the minimum value for the pixel levels each side of the threshold. The goal is to find the threshold value with the minimum entropy for sum of foreground and background. Otsu's method determines the threshold value based on the statistical information of the image where for a chosen threshold value t the variance of clusters T0 and T1 can be computed. The optimal threshold value is calculated by minimizing the sum of the weighted group variances, where the weights are the probability of the respective groups.

Given: P(i) as the histogram probabilities of the observed gray value i=1….l

$$\mathbf{P(i)=[\ number\{\ (r,\ c)\ |\ image(r,\ c)=i\}]/\ (R,C)}$$

Where,

r = Index for row of the image

c = Index for column of the image

R = No. of rows of the image

C = No. of columns of the image

Now,

$w_b(t)$ = Weight of class T0 with intensity value from 0 to t

$\mu_b(t)$ = Mean of class T0 with intensity value from 0 to t

$\sigma_b^2(t)$ = Variance of class T0 with intensity value from 0 to t

$w_f(t)$ = Weight of class T1 with intensity value from t+1 to 1

$\mu_f(t)$ = Mean of class T1 with intensity value from t+1 to 1

$\sigma_f^2(t)$ = Variance of class T1 with intensity value from t+1 to 1

σ_w^2 = The weighted sum of group variances

The best threshold value t^* is the value with minimum within class variance. The within class variance defines as following,

$$\boldsymbol{\sigma_w^2 = w_b(t) * \sigma_b^2(t) + w_f(t) * \sigma_f^2(t)}$$

where,

$w_b(t) = \sum_{i=1}^{t} P(i)$

$w_f(t) = \sum_{i=t+1}^{l} P(i)$

$\mu_b(t) = \sum_{i=1}^{t} i * P(i) / w_b(t)$

$\mu_f(t) = \sum_{i=t+1}^{l} i * P(i) / w_f(t)$

$\sigma_b^2(t) = \sum_{i=1}^{t} (i - \mu_b(t))^2 * P(i) / w_b(t)$

$\sigma_f^2(t) = \sum_{i=t+1}^{l} (i - \mu_f(t))^2 * P(i) / w_f(t)$

Figure 32: Images After Thresholding

Step 6: Edge Detection by Canny Operator

The **Canny Edge Detector** is an edge detection operator that uses a multistage algorithm to detect a wide range of edges in images [26].

The algorithm consists of five different steps. They are:

i. Apply Gaussian filter to smooth the image in order to remove the noise.
ii. Find the intensity gradients of the image.
iii. Apply non-maximum suppression to get rid of spurious response to edge detection.
iv. Apply double threshold to determine potential edges.

v. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak & not connected to strong edges.

i. **Noise Reduction by Gaussian Filter**

The equation for a Gaussian filter kernel of size (2k+1) * (2k+1) is given by,

$$\mathbf{H_{ij} = 1/2\pi\sigma^2 \ \ exp(- [(i-(k+1))^2 + (j-(k+1))^2] / 2\sigma^2)}$$

Where, 1<= i,j <= (2k+1)

ii. **Gradient Calculation**

Edges correspond to a change of pixels' intensity. To detect it, the easiest way is to apply filters that highlight this intensity change in both directions: horizontal (x) and vertical (y)

When the image is smoothed, the derivatives ***Ix*** and ***Iy*** w.r.t. ***x*** and ***y*** are calculated. It can be implemented by convolving ***I*** with Sobel kernels ***Kx*** and ***Ky***, respectively:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Sobel filters for both direction (horizontal and vertical)

Then, the magnitude ***G*** and the slope ***θ*** of the gradient are calculated as follow:

$$|G| = \sqrt{I_x^2 + I_y^2},$$
$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right)$$

Gradient intensity and Edge direction

iii. Non-Maximum Suppression

The final image should have thin edges. Thus, we must perform non-maximum suppression to thin out the edges.

The principle is simple: the algorithm goes through all the points on the gradient intensity matrix and finds the pixels with the maximum value in the edge directions.

Steps are:

- Create a matrix initialized to 0 of the same size of the original gradient intensity matrix;

- Identify the edge direction based on the angle value from the angle matrix;
- Check if the pixel in the same direction has a higher intensity than the pixel that is currently processed;
- Return the image processed with the non-max suppression algorithm.

iv. Double Threshold

The double threshold step aims at identifying 3 kinds of pixels: strong, weak, and non-relevant:

- Strong pixels are pixels that have an intensity so high that we are sure they contribute to the final edge.
- Weak pixels are pixels that have an intensity value that is not enough to be considered as strong ones, but yet not small enough to be considered as non-relevant for the edge detection.
- Other pixels are considered as non-relevant for the edge. Now you can see what the double thresholds holds for:
    - High threshold is used to identify the strong pixels (intensity higher than the high threshold)
    - Low threshold is used to identify the non-relevant pixels (intensity lower than the low threshold)
    - All pixels having intensity between both thresholds are flagged as weak and the Hysteresis mechanism (next step) will help us identify the ones that could be considered as strong and the ones that are considered as non-relevant.

v. Edge Tracking by Hysteresis

Based on the threshold results, the hysteresis consists of transforming weak pixels into strong ones, if and only if at least one of the pixels around the one being processed is a strong one, as described below:

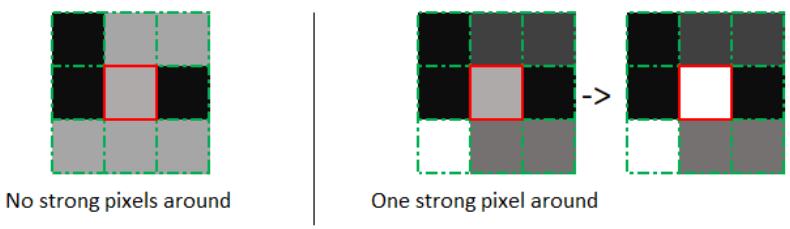

Figure 33: Edge Tracking by Hysteresis

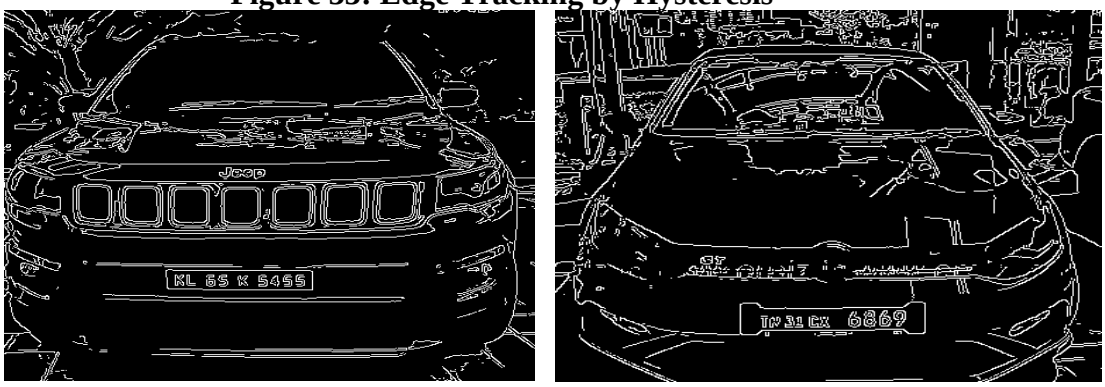

Figure 34: Images After Canny Edge Detection

Step 7: Candidate Plate Area Detection by Contour Detection

Contours are defined as the line joining all the points along the boundary of an image that are having the same intensity. Contours come handy in shape analysis, finding the size of the object of interest, and object detection [27].

OpenCV has findContour() function that helps in extracting the contours from the image. It works best on binary images, so we should first apply thresholding techniques, Sobel edges, etc. There are three essential arguments in cv2.findContours() function. First one is source image, second is contour retrieval mode, third is contour approximation method and it outputs the image, contours, and hierarchy. '*contours*' is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x, y) coordinates of boundary points of the object.

Figure 35: Bounding Box on Contours

Step 8: Actual Number Plate Area Extraction

The license plate is extracted using either a shape analysis or a colour analysis method. In the General License Plate has in form of a rectangular shape. Thus, algorithms look for geometrical shapes of a rectangular proportion. In India, most license plates are white or yellow, and therefore can also use colour analysis. Before you find the rectangle in an image, the image must be in a binary image or the edges of the image should be detected. Then you should find and connect to the relevant rectangular corners. Finally, the areas connected to the box are connected and all rectangular areas of interest are extracted.

Figure 36: Actual Number Plate Area Extractions

Step 9: Enhancement of Extracted Plate Region by Histogram Equalization

Finally, the masked contour is enhanced by Histogram Equalization to extract the Number Plate.

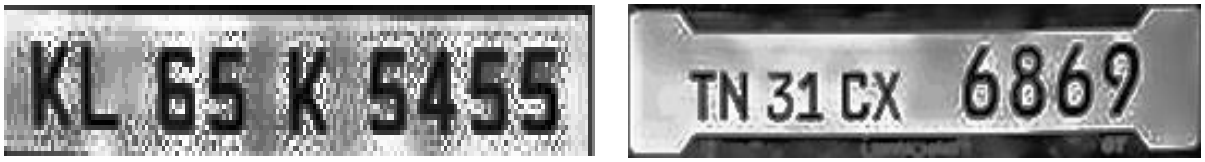

Figure 37: Enhancing the Extracted Plate Regions

Step 10: Number Plate Character Recognition by OCR

The optical character recognition is a recognition method in which the input is an image and the output is string of character. OCR is a process which separates the different characters from each other taken from an image. Template matching is one of the approaches of OCR. The cropped image is compared with the template data stored in database. OCR automatically identifies and recognizes the characters without any indirect input. The characters on the number plate have uniform fonts then the OCR for number plate recognition is less complex as compared to other methods.

Figure 38: Number Plate Character Recognitions

3.3 System Designs

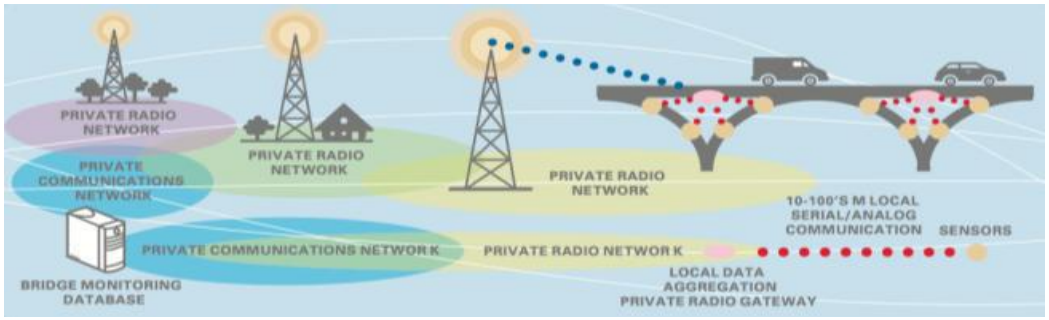

Figure 39: Private Communication in the Proposed Model

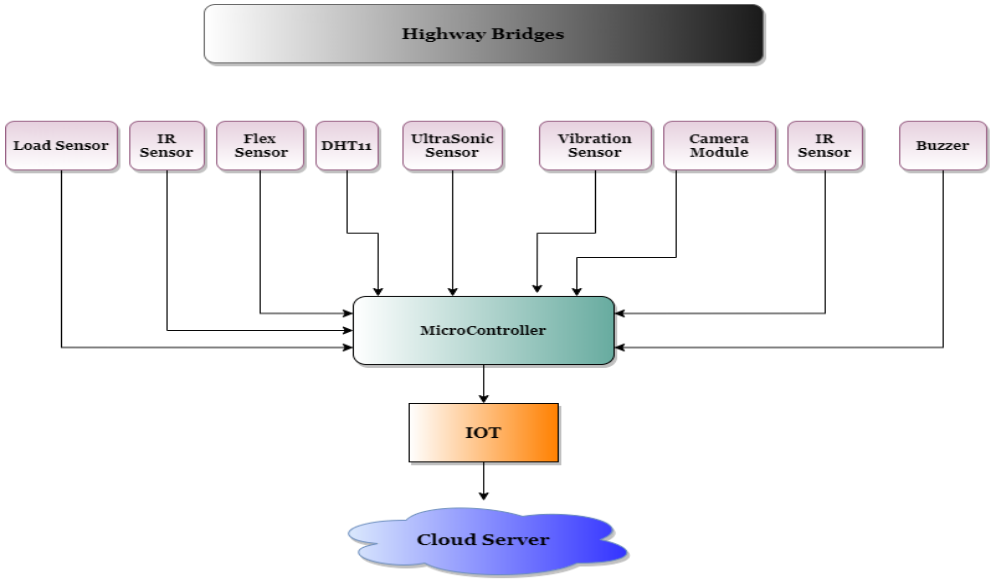

Figure 40: Block Diagram of Automatic Real-Time Bridge Structural Health Monitoring System

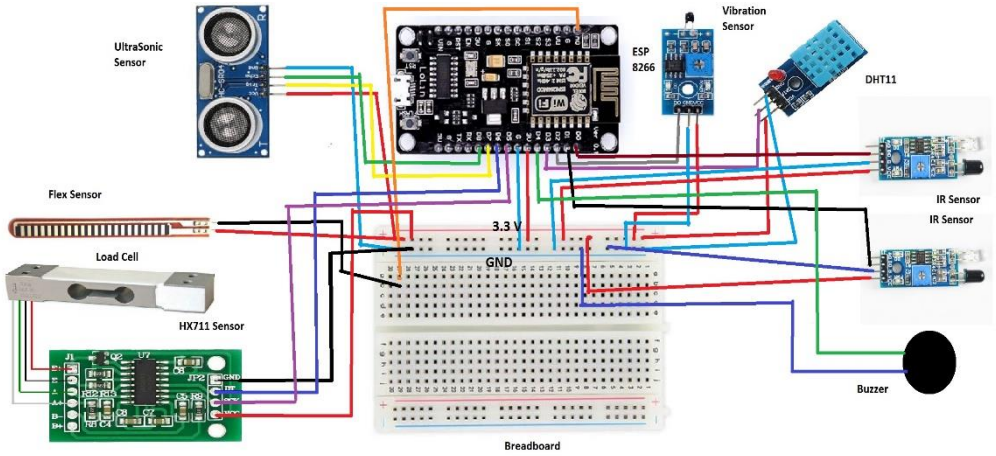

Figure 41: Circuit Diagram of Automatic Real-Time Bridge Structural Health Monitoring System

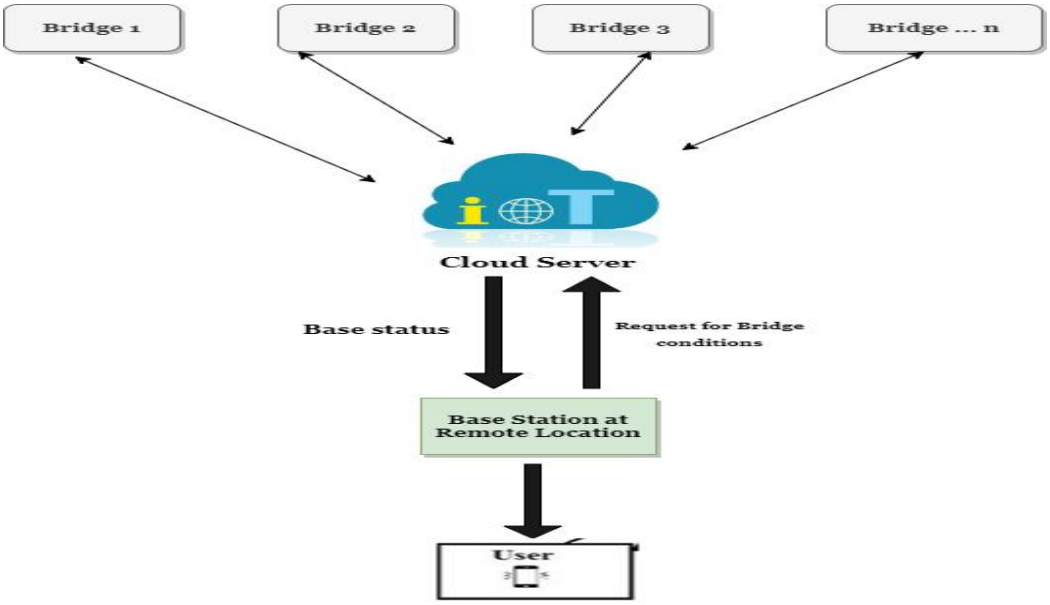

Figure 42: System Architecture of Automatic Real-Time Bridge Structural Health Monitoring System

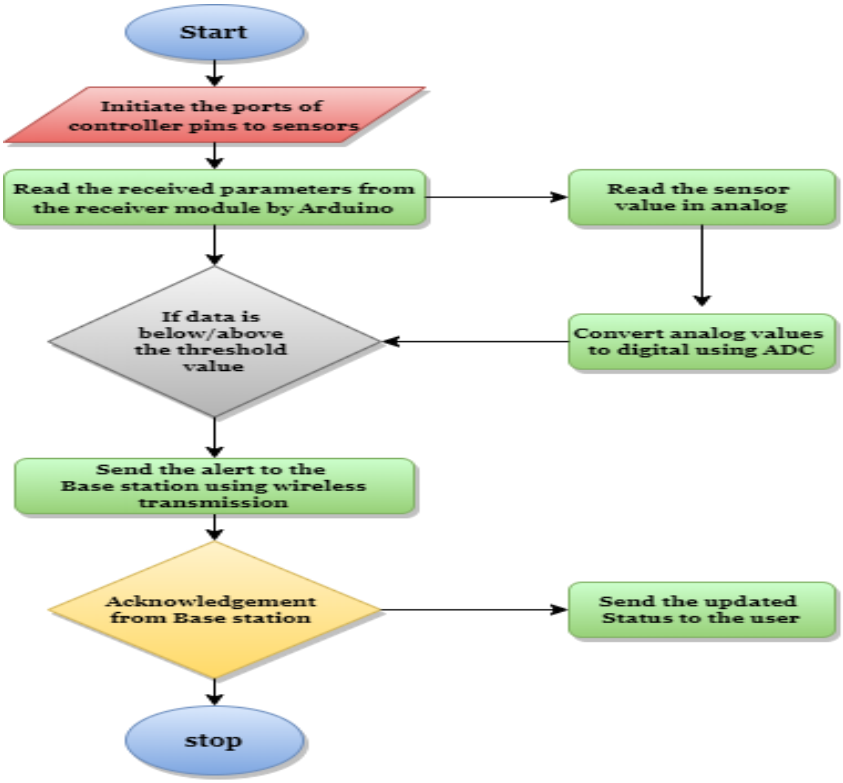

Figure 43: Flow Chart of the Working Principle of Automatic Real-Time Bridge Structural Health Monitoring System

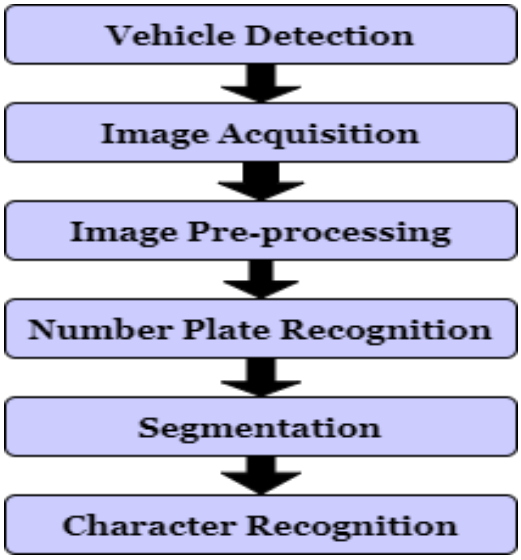

Figure 44: Block Diagram of the Working Principle of Vehicle Number Plate Detection

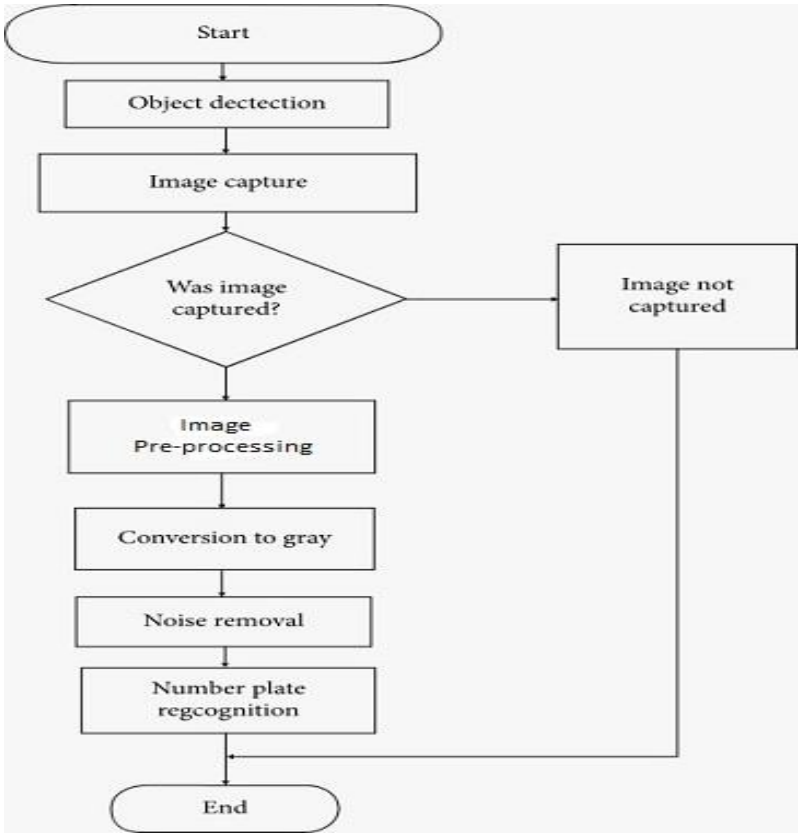

Figure 45: Flow Chart of the of the Working Principle of Vehicle Number Plate Detection

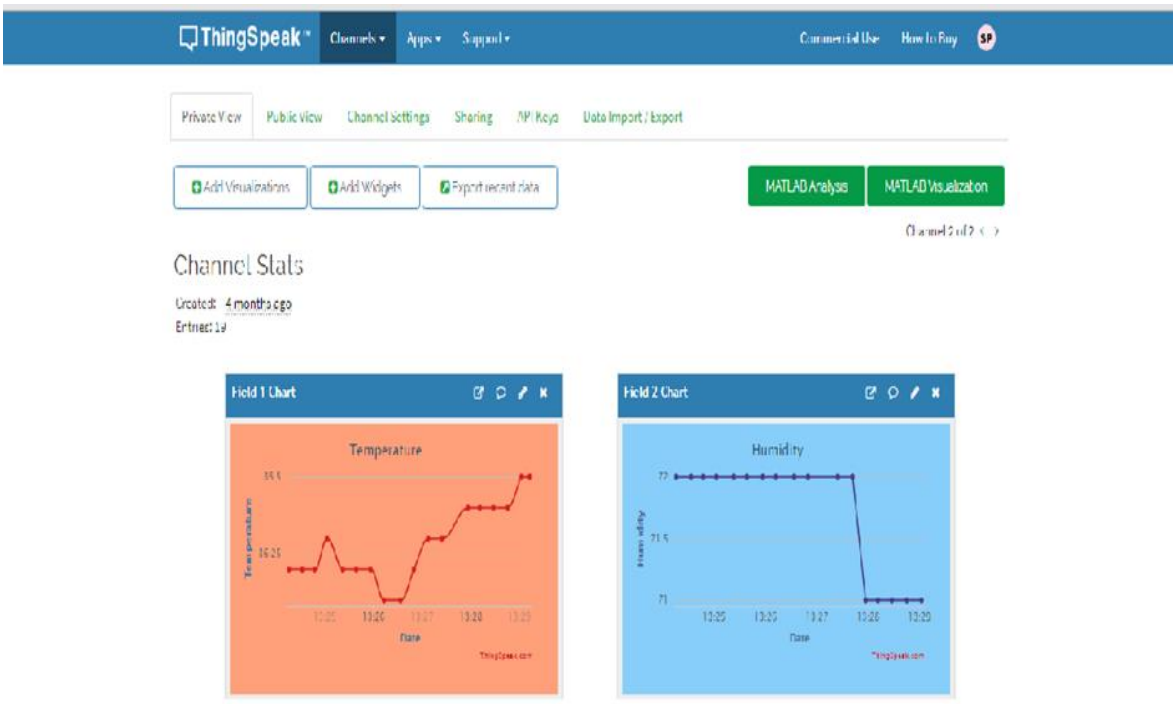

Figure 46: Displaying & Analysing Different Parameters in a Visual Interface for Real-Time Monitoring

Figure 47: Displaying & Analysing Different Parameters in a Visual Interface for Real-Time Monitoring

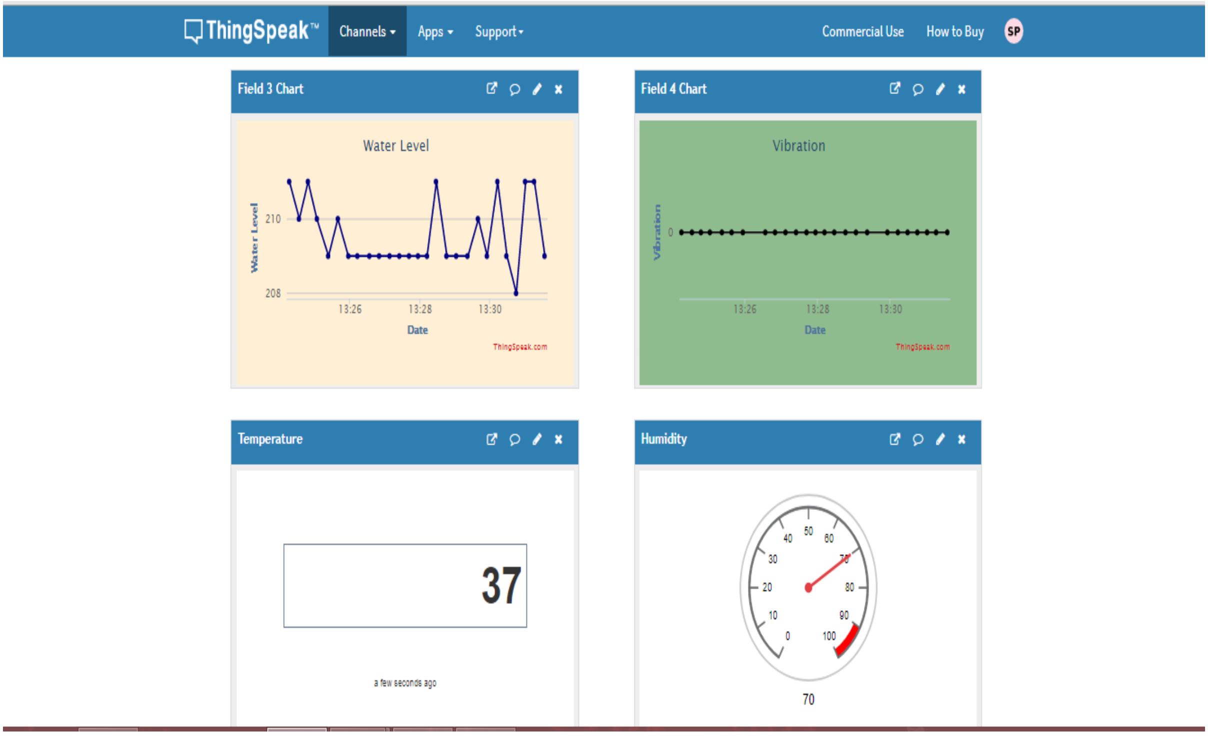

Figure 48: Displaying & Analysing Different Parameters in a Visual Interface for Real-Time Monitoring

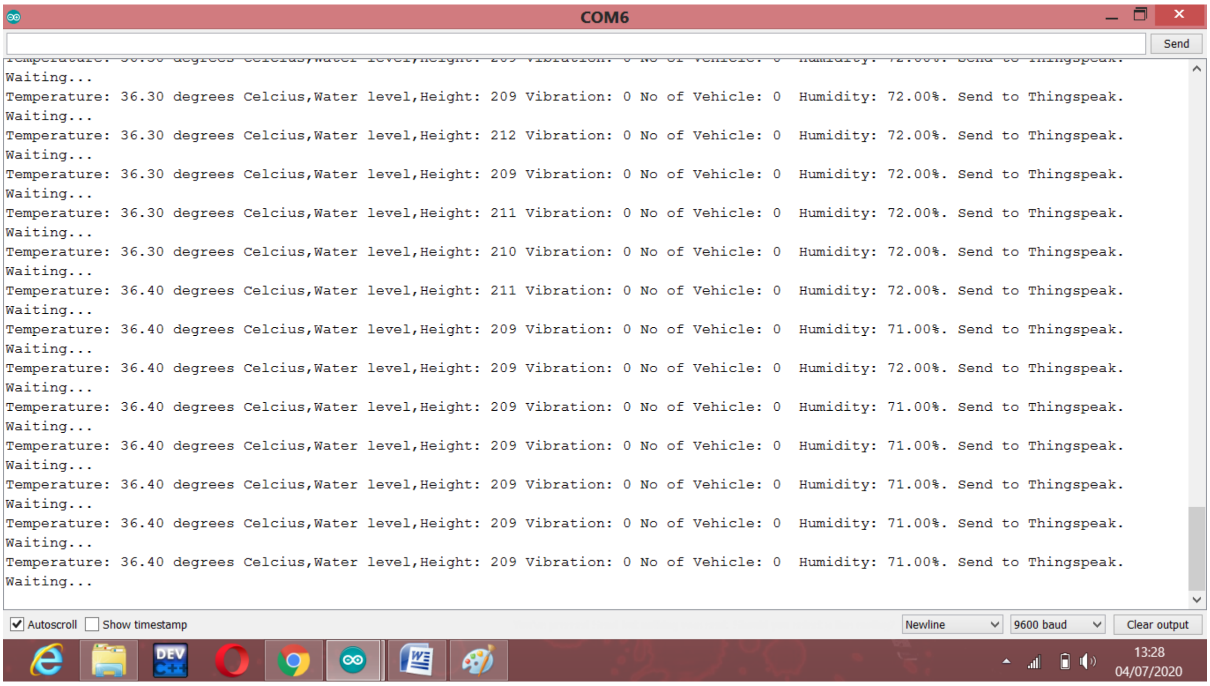

Figure 49: Transferring the Parameter Values Wirelessly from the Visual Interface to the Arduino IDE

Chapter 4: Coding

4.1 Code for Automatic Real-Time Bridge Structural Health Monitoring System

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include "HX711.h"

HX711 scale(12, 14);

float calibration_factor = 2230;
float units;
float ounces;
const int buzzerPin = 2;

const int flexPin = A0;

String apiKey = "AULUV71GF2VCKFIZ";
const char *ssid =  "xtz";
const char *pass =  "nopass123";
const char* server = "api.thingspeak.com";

#define DHTPIN 0

int ob1 = 16;
int ob2 = 5;
const int trigP = 13;
const int echoP = 15;
int vs =4;

long duration,m;
int d,cm;
int c=0;

DHT dht(DHTPIN, DHT11);

int readPing(){
  delay(70);
  digitalWrite(trigP,LOW);   delayMicroseconds(5);
  digitalWrite(trigP,HIGH);  delayMicroseconds(10);
  digitalWrite(trigP,LOW);
  duration = pulseIn(echoP,HIGH);

  cm=(duration/2)/29.1;
  if (cm==0){
    cm=250;
  }
  if (cm>=250)
  {cm=250;}
  return cm;
}
```

```
float weight() {

  scale.set_scale(calibration_factor);

  Serial.print("Reading: ");
  units = scale.get_units(), 10;
  if (units < 0)
  {
    units = 0.00;
  }
  ounces = units * 0.035274;
  Serial.print(units);
  Serial.print(" grams");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();

  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 1;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 1;
  }
  return units;
}

WiFiClient client;

void setup()
{
       delay(10);
       dht.begin();
       Serial.println("Connecting to ");
       Serial.println(ssid);
       WiFi.begin(ssid, pass);

      while (WiFi.status() != WL_CONNECTED)
     {
           delay(500);
           Serial.print(".");
     }
      Serial.println("");
      Serial.println("WiFi connected");

 pinMode(trigP, OUTPUT);
 pinMode(echoP, INPUT);
```

```
 pinMode(vs, INPUT);
 pinMode(ob1, INPUT);
 pinMode(ob2, INPUT);
 pinMode(buzzerPin,OUTPUT);
 Serial.begin(9600);

  Serial.println("HX711 calibration sketch");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");

  scale.set_scale();
  scale.tare();

  long zero_factor = scale.read_average();
  Serial.print("Zero factor: ");
  Serial.println(zero_factor);

}



void loop()
{
     float h = dht.readHumidity();
     float t = dht.readTemperature();
     int f;

  d = readPing();
  float wt = weight();
  m=digitalRead(vs);
  f = analogRead(flexPin);
  Serial.println(f);
  delay(100);
   if(digitalRead(ob1)==LOW){
            c++;
             }
   if(digitalRead(ob2)==LOW){
   c--;
     }
     if(c<0){
      c=0; }

            if (isnan(h) || isnan(t) || isnan(d)|| isnan(m)|| isnan(c)|| isnan(f)|| isnan(wt))
              {
                 Serial.println("Failed to read from DHT sensor!");
                  return;
              }
```

```
            if (client.connect(server,80))
          {
               String postStr = apiKey;
               postStr +="&field1=";
               postStr += String(t);
               postStr +="&field2=";
               postStr += String(h);
               postStr +="&field3=";
               postStr += String(d);
               postStr +="&field4=";
               postStr += String(m);
               postStr +="&field5=";
               postStr += String(c);
               postStr +="&field6=";
               postStr += String(f);
               postStr +="&field7=";
               postStr += String(wt);
               postStr += "\r\n\r\n";

               client.print("POST /update HTTP/1.1\n");
               client.print("Host: api.thingspeak.com\n");
               client.print("Connection: close\n");
               client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
               client.print("Content-Type: application/x-www-form-urlencoded\n");
               client.print("Content-Length: ");
               client.print(postStr.length());
               client.print("\n\n");
               client.print(postStr);

               Serial.print("Temperature: ");
               Serial.print(t);
               Serial.print(" degrees Celcius,Water level,Height: ");
               Serial.print(d);
                Serial.print(" Vibration: ");
               Serial.print(m);
               Serial.print(" No of Vehicle: ");
               Serial.print(c);
               Serial.print("  Humidity: ");
               Serial.print(h);
               Serial.print(" Bending: ");
               Serial.print(f);
               Serial.print(" Bridge Load: ");
               Serial.print(wt);
               Serial.println("%. Send to Thingspeak.");
            }
   client.stop();
   Serial.println("Waiting...");

if ( t > 60 || d > 20 || m > 0 || c > 10 || f > 20 || wt > 20)
{
```

```
      digitalWrite(buzzerPin,HIGH);
      Serial.println("Bridge is in Danger...");
      Serial.println("Alert has been send to base station...");
      Serial.println("Close the bridge...");

    }
  else{
    digitalWrite(buzzerPin,LOW);
      Serial.println("Bridge is Safe...");
      Serial.println("Everything is fine...");
      Serial.println("Data send to the base station...");
  }

 delay(200);
}
```

4.2 Code for Vehicle Number Plate Detection

```
import numpy as np
import cv2
import imutils
import pytesseract
from PIL import Image
pytesseract.pytesseract.tesseract_cmd = 'C:/Program Files/Tesseract-OCR/tesseract.exe'

# Read the image file
image = cv2.imread('D:\car_1.jpg')

# Resize the image - change width to 500
image = imutils.resize(image, width=500)
#some image might not perform perfect in this particular width in that case we might change
the width of image

# Display the original image
cv2.imshow("Original Image", image)
cv2.waitKey(0)

# RGB to Gray scale conversion
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("1. Grayscale Conversion", gray)
cv2.waitKey(0)

# Noise removal with iterative bilateral filter(removes noise while preserving edges)
noise_removal = cv2.bilateralFilter(gray, 11, 17, 17)
cv2.imshow("2. Bilateral Filter", noise_removal)
cv2.waitKey(0)
```

```
# Apply Histogram equalisation
equal_hist= cv2.equalizeHist(noise_removal)
cv2.imshow("3. After Histogram equalisation",equal_hist)
cv2.waitKey(0)

# Find Threshold  of the image
ret,thresh_image = cv2.threshold(equal_hist,0,255,cv2.THRESH_OTSU)
cv2.imshow("4. Image after Thresholding",thresh_image)
cv2.waitKey(0)

# Find Edges of the Noise removal
canny_edged = cv2.Canny(noise_removal,170,200)
cv2.imshow("5.  Canny Edges", canny_edged)
cv2.waitKey(0)

# Find contours based on Edges
cnts, new = cv2.findContours(canny_edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)

# Create copy of original image to draw all contours
img1 = image.copy()
cv2.drawContours(img1, cnts, -1, (0,255,0), 3)
cv2.imshow("6.  All Contours", img1)
cv2.waitKey(0)

#sort contours based on their area keeping minimum required area as '30' (anything smaller than this will not be considered)
cnts=sorted(cnts, key = cv2.contourArea, reverse = True)[:30]
NumberPlateCnt = None #we currently have no Number plate contour

# Top 30 Contours
img2 = image.copy()
cv2.drawContours(img2, cnts, -1, (0,255,0), 3)
cv2.imshow("7.  Top 30 Contours", img2)
cv2.waitKey(0)

# loop over our contours to find the best possible approximate contour of number plate

c = 0
for c in cnts:
      peri = cv2.arcLength(c, True)
      approx = cv2.approxPolyDP(c, 0.02 * peri, True)
      if len(approx) == 4:  # Select the contour with 4 corners
         NumberPlateCnt = approx #This is our approx Number Plate Contour

         # Crop those contours and store it in Cropped Images folder
         x, y, w, h = cv2.boundingRect(c)
         new_img = noise_removal [y:y + h, x:x + w] #Create new image
cv2.imwrite("plate.jpg", new_img)#Store new image
         break;
```

```
# Drawing the selected contour on the original image
cv2.drawContours(image, [NumberPlateCnt], -1, (0,255,0), 3)
cv2.imshow("9. Final Image With Number Plate Detected", image)
cv2.waitKey(0)

cv2.imshow("10. Extracted Number Plate ",new_img)
cv2.waitKey(0)

# Apply Histogram equalisation into extracted number plate
eq_hist_num = cv2.equalizeHist(new_img)
cv2.imshow("11. Extracted Number Plate(Histogram equalisation)",eq_hist_num)
cv2.waitKey(0)

# Use tesseract to covert image into string
cv2.imwrite('temp.png', new_img)
text = pytesseract.image_to_string(Image.open('temp.png'))
print("Number is :", text)

cv2.waitKey(0) #Wait for user input before closing the images displayed
```

Chapter 5: System Testing

This system is designed for condition monitoring of diff parameters like temp, weight, vibrations , bending angle of Highway bridges and also monitor current status from remote location using IOT technology. The Vibration sensor senses real time vibrations and these vibrations are in form of analog signal. Latter on these vibrations i.e. analog signals are converted into digital signals by using inbuilt ADC of microcontroller. In the proposed system, there are two node (i) transmitter node and (ii) base station. Here base station acts as receiver and ESP8266 is the communication media between various transmitter node.

The Bridge Monitoring System was designed and the hardware for the same was built. The above are the output obtained.

1. When the system is powered up, there will be a display of "Bridge Monitoring System" on ThingSpeak which located in base station. Also the information is sent to the base station via ESP8266.
2. Then the ESP8266 initialization takes place and it searches for the signal. As the initialization is completed, there will be a constant monitoring of flex sensor, IR Sensor, load Sensor, water level sensor and the digital output value is displayed in the ThingSpeak.
3. There is an IR sensor at the entry of the bridge, which detects the vehicle and gives the incremented count of the number of vehicles on the bridge. If the number of vehicles exceeds the threshold value (i.e. 5), the gate is closed and alert message will be sent to base station. There is an IR sensor at the exit of the bridge, which detects the vehicle and gives the decremented count of the number of vehicles on the bridge. The gate is opened if the number of vehicles is equal to 3.
4. There is a vibrator sensor at the bottom of the bridge, which detects the earthquakes (heavy vibrations). The gate is closed when there are heavy vibrations. the information is sent to the monitoring house.
5. There is a flex sensor beneath the bridge, which detects the cracks and bending. The gate is closed when the threshold value of flex exceeds. The information is sent to the monitoring house via wireless communication.
6. There is a load sensor at the entry of the bridge, which detects the load on the bridge. The gate is closed when there are heavy loads and the alert is sent to the base station via wireless communication.

After successful assemblage of the hardware parts and the configuration of the software components, the functionality of the system was tested on a built model. The IR sensor was used to detect the moving cars. Once detected, the camera was triggered to capture the image of the plate number. The system was tested on a simulated prototype road. Cars were made to pass on the bridge with varying speeds. Out of the cars that passed. The sensors were programmed such that when a fast car or heavy loaded car approaches then via load sensor we can automatically trigger the USB camera to take pictures of the car as it is assumed to be over speeding. The camera was placed in such a way that the rear of the car will be captured so that its plate number can be captured and verified with an existing database. After this was complete, the captured image was made to undergo image processing and optical character recognition. This is made possible by the OpenCV codes written with Python. Once the text was extracted, it was displayed on a web page.

The results obtained are as shown below.

Figure 50: Captured Image

As seen from the above screenshot, the module was able to capture different plate numbers of cars which was deemed to have crossed the set speed threshold, and it was able to tag each one of them with a unique number. This serves as ways of differentiating each vehicle from the other so it can easily be traced on the database. The saving sequence could also be observed from the screenshots shown below.

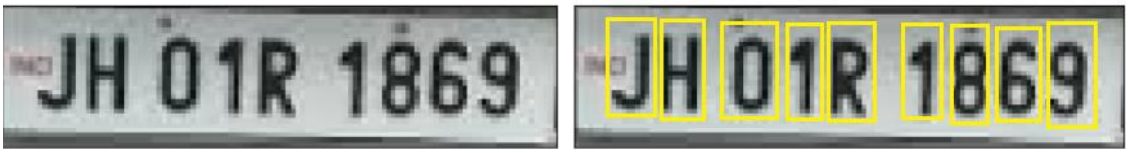

Figure 51: Saving Sequence

As shown in above Figure, the text received from the camera is in the threshold form, i.e., it is susceptible to noise and other kinds of interference. Then we have to perform filtering on the image by removing the noise components. The image is made available to the web page developed for the system, which is accessible to the database that holds the accurate information of the car owner.

Here we have discussed the different components and modules used by us to monitor the bridge condition. Such a system will help to control the dynamic parameters of the bridge for preventing it from the disaster which can save the many lives and also wealth. This system is unique in its ability to monitor the bridge environment, transmit the environmental data through wireless communication and send alerts to the bridge management staff in real time for prompt reactions. This system can enable 24x7 bridge safety management as well as prompt and appropriate Responses to emergency incidents. The system continuously monitors the bridge parameter value and judges whether the bridge is safe or not for traveling. In case the parameter values are beyond the threshold values then an alert sound is given to the people. This implementation is greatly useful to provide safety for the human.

Chapter 6: Conclusion

To measure the level of success of our project, we had to examine our specified objectives to determine whether they were met by our final product. We found that most of our major objectives were indeed satisfied, as shown in the following sentences, by our bridge structural health monitoring system. Here we have discussed the different methods used by the researchers to monitor the bridge condition. Such a system will help to control the dynamic parameters of the bridge for preventing it from the disaster which can save the many lives and also wealth. This system is unique in its ability to monitor the bridge environment, transmit the environmental data through wireless communication and send alerts to the bridge management staff in real time for prompt reactions. This system can enable 24x7 bridge safety management as well as prompt and appropriate Responses to emergency incidents. The system continuously monitors the bridge parameter value and judges whether the bridge is safe or not for traveling. In case the parameter values are beyond the threshold values then an alert sound is given to the people. This implementation is greatly useful to provide safety for the human.

Two of our objectives were not completely satisfied by our project. One is extracting vehicle owner information from the detected vehicle number plate. However, hopefully this objective we can fulfil into the future. Second is Reliability. As Our automatic real-time bridge structural health monitoring system is made of cheap components So, the reliability is uncertain.

Overall, our project was definitely a success. Only two of our minor objectives were not satisfied, and these do not affect the system's capability to function properly. There were also several other minor problems such as loose connections, and these are discussed more fully in the recommendations for future work. These problems were also fairly minor, and while it was disappointing to see any problems occur, these were fairly insignificant to the overall success and workability of our project.We learned a great deal about working on a major engineering project through this experience. Other than learning much about the specific design and manufacturing elements that this particular project required, there are several overriding concepts that we have taken from this project are:

- Allow more time than you think you need for each step of the process. Things take longer than expected, and this impacted us particularly in the construction phase of our project. Machining & making circuit boards are just several areas where our project construction was more time-consuming than anticipated.
- Organize and plan a specific construction process. When there are many different construction aspects that all need to be done, it is important to think carefully and plan what needs to be done first, second etc. This helps to avoid problems where one process is completed before another one that needed to be done first.
- Test early, so that changes can be made and problems can be solved. Once again, things do not get finished as quickly as expected, so testing early gives time to correct problems that had not necessarily been anticipated.

These principles can be applied to many different kinds of projects and construction processes, and this project has shown us how important planning and organization are when taking on any type of major project. A realistic timetable for completion, a proper order of processes, and planning on various types of failures occurring are things that are easy to overlook, but doing so could cause a project to fail or never get completed. As engineers, we have been trained to design, analyse, and build, and this project has shown us that these three principles must be combined with two other crucial ones: planning and organization.

Chapter 7: Future Scope

The system continuously monitors the bridge parameter value and judges whether the bridge is safe or not for traveling. In case the parameter values are beyond the threshold values then an alert sound is given to the people. This implementation is greatly useful to provide safety for the human.

In the existing application, we have tried to incorporate all the necessary and secure application that will be of beneficial use to the citizens and their surrounding environment. In the future we can improve our project by adding these features,

- The system could be implanted that could along with detection also notify to the municipal services about the future life of the bridge as to how long the bridge will sustain.

- The existing system could also be built using a solar panel around it so it uses solar energy and sustains for a longer period of time.

- In future we can add alert auto- barrier when there are signs of collapsing the bridge. This system will help to reduce big disasters in future. This system can save the lives of many people.

- In addition, we can add various algorithms for damage identification purpose in structures of bridges will be proposed using machine learning and graph analysis.

- Scanning number plate sometimes goes unsuccessful by using the shape analysis method to detect exact area of the plate. Future extension of this work is to develop Detecting number plate characters during night times. It’s sometime work inefficient in night images.

- The processing time to make the plate recognition is low, and the proposed system showed a good performance in a real environment. However, it was possible to verify that the character recognition is sensitive to the environment illumination, and to improve the accuracy, new experiments will be done considering other types of filters as well as new deep learning methods will be implemented in the future works.

Chapter 8: References

1. Ms. Arohi, D. Sonawane, Ms. Pooja, P. Vichare, Mr. Shubham S. Patil and Mr. Nitin, P. Chavande, “Bridge Monitoring System Using IOT”, Journal of Advances in Electrical Devices Volume 3, Issue 2, 2018

2. Varsha Kusal, Amrita Argade, Sanika Chiplunkar, Rohini Kumbhar, Swati A. Khodke, “Bridge Monitoring and Alert Generation System Using IOT”, International Journal of Advance Research, Ideas and Innovations in Technology, 2017

3. Amro Al-Radaideh, A. R. Al-Ali1, Salwa Bheiry, Sameer Alawnah, “A Wireless Sensor Network Monitoring System for Highway Bridges”, IEEE, International Conference on Electrical and Information Technologies ICEIT, 2015.

4. Snehal Sonawane, Nikita Bhadane, Sayali Zope, Ashitosh Pangavhane and V. S. Tidake, “Design of Bridge Monitoring System based on IoT”, MVP Journal of Engineering Sciences, June 2018

5. Jin-Lian Lee, Yaw-Yauan Tyan, Ming-Hui Wen1, Yun-Wu Wu, “Development of an IoT-based Bridge Safety Monitoring System”, IEEE, International Conference on Applied System Innovation, 2017

6. Gaurav Agrawal, Yogesh Jadhav, Sreeranjini Nair, Anurag Kumar & Prof. Sinu Nambiar, “IOT Based Bridge Safety Monitoring System”, International Journal for Research in Applied Science & Engineering Technology (IJRASET**),** April 2019

7. Shachi P., Manjunatha S., “Automatic Bridge Health Monitoring System Using Wireless Sensors”, International Journal of Science and Research (IJSR), 2015

8. Ashwini R, Sneha Shivanand Mesta, Varsha A U, Ravichandran G, Haritha K Sivaraman, “Bridge Monitoring System Using Wireless Networks”, IJARIIE, 2017

9. K.M. Sajjad, “Automatic License Plate Recognition using Python and OpenCV”

10. Pratiksha Jain, Neha Chopra & Vaishali Gupta, “Automatic License Plate Recognition using OpenCV”, International Journal of Computer Applications Technology and Research, 2014

11. Tella Pavani & DVR Mohan, “Number Plate Recognition by using open CV- Python”, International Research Journal of Engineering and Technology (IRJET), March 2019

12. Vikas Kumar Mishra, Shobhit Kumar & Neeraj Shukla, “Image Acquisition and Techniques to Perform Image Acquisition”

13. Jamileh Youse, “Image Binarization using Otsu Thresholding Algorithm”

14. https://en.wikipedia.org/wiki/Bridge

15. https://en.wikipedia.org/wiki/List_of_bridge_failures#2000%E2%80%93present

16. https://components101.com/ultrasonic-sensor-working-pinout-datasheet

17. https://components101.com/dht11-temperature-sensor

18. https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/

19. https://circuitdigest.com/microcontroller-projects/arduino-sw-420-vibration-sensor-module-interfacing

20. https://www.elprocus.com/flex-sensor-working-and-its-applications/

21. https://robu.in/product/weighing-load-cell-sensor-20kg-electronic-yzc-133-wires/?gclid=EAIaIQobChMI756Pu5KQ6AIVDaqWCh139gZOEAQYAyABEgLwyPD_BwE

22. https://circuitdigest.com/microcontroller-projects/how-to-use-ov7670-camera-module-with-arduino

23. https://www.geeksforgeeks.org/python-bilateral-filtering/

24. https://www.geeksforgeeks.org/python-morphological-operations-in-image-processing-opening-set-1/

25. https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-1-simple-thresholding/

26. https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

27. https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/

28. https://www.arduino.cc/en/main/software

29. https://www.instructables.com/id/How-to-use-a-Buzzer-Arduino-Tutorial/

30. https://www.makeuseof.com/tag/18650-battery/

31. https://jupyter.org/

32. https://pypi.org/

33. https://opencv.org/about/

34. https://www.python.org/doc/essays/blurb/

35. https://thingspeak.com/pages/learn_more

36. Introduction to Embedded Systems Book- Shibu K V

37. Arduino Cookbook, 2nd Edition, by. Michael Margolis

38. Digital Image Processing, 4th Edition by Rafael. C. Gonzalez & Richard. E. Woods