AGE VERIFICATION SYSTEM FOR TWO-WAY SAFETY ENABLED INTELLIGENT FUGITIVE GAS EMISSION ALERT SYSTEM IN PYTHON

A PROJECT REPORT

Submitted by

CHAMARTHI SANDEEP VARMA	**[REGISTER NO:211419104044]**
CHARALA HEMANTH	**[REGISTER NO:211419104045]**
DHULIPALA RAMGOPAL	**[REGISTER NO:211419104215]**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report "**AGE VERIFICATION SYSTEM FOR TWO-WAY SAFETY ENABLED INTELLIGENT FUGITIVE GAS EMISSION ALERT SYSTEM IN PYTHON**" is the bonafide work of "**CHAMARTHI SANDEEPVARMA(211419104044),CHARALAHEMATH(211419104045),DHULIPALA RAMGOPAL(211419104215)**" who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mrs.R.SALINI , M.TECH,
SUPERVISOR
ASSISTANT PROFESSOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the End Semester Project Viva-Voce Examination held on...**12/04/2023**.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We **CHAMARTHI SANDEEP VARMA(211419104044), CHARALA HEMANTH(211419104045)**, **DHULIPALA RAMGOPAL(211419104215)** hereby declare that this project report titled "**AGE VERIFICATION SYSTEM FOR TWO-WAY SAFETY ENABLED INTELLIGENT FUGITIVE GAS EMISSION ALERT SYSTEM IN PYTHON**" , under the guidance of **R.SALINI,M.TECH,** is the original work done by us and we have not plagiaruized or submitted to any other degree in any university by us.

CHAMARTHI SANDEEP VARMA

CHARALA HEMANTH

DHULIPALA RAMGOPAL

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYA RAJESWARI, Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D**. who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.**, for the support extended throughout the project.

We would like to thank our guide **Mrs.R.SALINI,M.TECH** , and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

CHAMARTHI SANDEEP VARMA

CHARALA HEMANTH

DHULIPALA RAMGOPAL

ABSTRACT

The newest technology now include smart embedded systems at its heart, and IoT-based smart embedded systems are the hottest topic of study. In this research, we suggest an Internet of Things-based smart stove. A cooker mishap might happen at any time. So, we are developing a stove with two-way safety capabilities, a child lock system, and gas leak detecting capabilities. The intelligent cooker will make an effort to ensure protection and will use real-time video streaming to determine age. Our main concern is making sure a kid can't turn the cooker on.Also, the cooker has a gas monitoring alarm that can guarantee safety. A petrol detection module and an ARDUINO UNO are being and a hardware implementation buzzer. A deep learning object identification method is also being used for system execution. As our stove is Internet of Things (IoT) based, it can monitor safety both remotely and manually to try to prevent accidents.Leaks are a common occurrence in small-scale companies and residential appliances nowadays. If you do not recognise and make changes soon away, it is quite dangerous to your life.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1 PROBLEM STATEMENT

Smart embedded systems have become a core component in the latest technologies, and IoT based smart embedded system is the trendiest field in the research area. In our research, we are proposing an IoT based smart stove. Any accident might occur at any time from a stove. So we are designing a two-way safety enabled stove with a child lock system and gas leakage detection feature. The intelligent stove will try to ensure safety and will detect age from real-time video streaming. Our main focus is a child would not be able to turn the stove on.As well as, the stove can *entitle* safety via gas detection alarm. We are using an ARDUINO UNO and Gas Detection Module with a buzzer for the hardware implementation. Also, we are applying a deep Learning object detection algorithm for the system execution. Since our stove is IoT-based, the stove is ensuring safety remotely as well as manually which will try to prevent accidental occurrences.

Leaks are a common occurrence in small-scale companies and residential appliances nowadays. If you do not recognise and make changes soon away, it is quite dangerous to your life. The aim of our project to prevent the gas leakage accident from the child at home and automatically monitoring and controlling the system. This system ensure the gas leakage and low gas through an alert system.

CHAPTER 2

LITERATURE SURVEY

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Researchers have focused a lot of effort on the crucial issue of identifying and alerting systems for traffic accidents. During the past few decades, many different accident detection and alerting techniques have been suggested.[1]Leakage detection is essential to prevent the loss of these energy resources because pipelines are one of the primary means of transporting hydrocarbon fluids and gasses. In this article, based on research in field data, pipeline leakage as a rise in the mean value of the flow rate differential between the inlet and the outlet sensors, where the increased value is unknown and susceptible to change.

Finally, utilizing kernel density estimates and the estimated cumulative distribution function (CDF) of the data under steady-state circumstances, an adaptive filter is suggested. The suggested filter has a superior performance in tiny leaks in comparison with various benchmarks.[2]The goal of the suggested approach is to protect property and live things from fire disasters. A complete device has also been developed to prevent this scenario by quickly cutting off the power supply, protecting people and property within a short amount of time. The primary characteristics of this paradigm are that the monitoring equipment may function from a remote location and (ii) it gathers real-world data that is sent to the cloud for storage. But, utilizing smart devices, such as smartphones, laptops, and computers, acquired data may be examined from numerous places, allowing for the creation and execution of real-time choices.[3]This essay's goal is to examine the operation, design, and potential of an automatic gas leak detection system that also notifies the user by sending a message via a GSM module, sounds an alarm with a buzzer to inform others around, and controls gas leaking (by exhaust functionalities).

The promotion of such automatic gas leak protection systems based on gas sensors that can quickly identify gas leaks like LPG, smoke, etc. is the other goal.[4]To find LPG gas, semiconductor sensors are utilised. It makes use of a MQ6 semiconductor sensor. The MQ-6 gas sensor's sensitive component is SnO2, which has a reduced conductivity in clean air. The sensor conductivity rises along with the growing gas concentration when the target flammable gas is

present. The MQ6 gas sensor responds to natural gas and has great sensitivity to propane, butane, and LPG. The sensor is versatile and inexpensive, and it may be then used detect a variety of flammable gasses, including methane.[5]We suggest an Internet of Things-based smart stove. A cooker mishap might happen at any time. So, we are creating a cooker with two-way safety capabilities, a kid lock system, and a gas leak detecting function. The smart stove will make an effort to maintain safety and use live video streaming to determine age. Our major concern is making sure a kid can't turn the cooker on. Also, the cooker has a gas detection alarm that can guarantee safety. The hardware implementation consists of a Raspberry Pi and a Gas Detector Module with a buzzer.[6]An intelligent method for utilizing IoT to track LP Gas leakage is presented. A smart electronic system is created that uses a web server to monitor the presence of butane, natural gas, LP gas, temperature, humidity, and heat index. If a petrol leak is discovered or any measured parameters surpass the threshold value, this system can sound an alert and send an alarming SMS to the appropriate authorities.[7]Microcontroller and a processor are features of Arduino. Analog input pins and digital I/O pins are both present on Arduino.

The Arduino project began in 2005 as a course for students at the Interaction Design Institute Ivrea in Italy. Its objective was to provide experts a low-cost and straightforward approach to make devices that interact with their surroundings using different sensors. This suggested work has a wide range of applications. With this suggested effort, it is feasible to recognise the various objectives with a single device.[8]To create foreign object recognition and production state detection data sets and monitoring models to detect the production plan of tobacco cabinet doors in real-time and the safety detection of foreign substances in cigarette cabinets based on deep learning with the Internet of Things environment.[9] [10]In order to maintain the confidentiality of data, this paper discusses secure data transmission and identifying changes that occurred during the exchange process. IoT devices must limit their computing to basic operations due to significant power and size constraints. The small processing devices could not be used for complex computation or additional overheads. Smart IoT-based home automation systems are quite popular, but they need to be well protected against outside intrusions and security concerns. Using change identification techniques, IoT device security is therefore of utmost significance. There is discussion of certain emerging trends and current security issues with IoT-based applications.

2.2 COMPARISON TABLE

TITLE	MERITS	DEMERITS	FUTURE ENHANCEMENT
Adaptive Naive Bayes Classifier Based Filter Using Kernel Density Estimation for Pipeline Leakage Detection	Better performance in small leaks in comparison with different benchmarks.	Overall, while the proposed approach to fire prevention has several potential benefits, it's important to consider these potential drawbacks and develop strategies to mitigate them.	Improvement in drawbacks and develop strategies.
Smart IoT System for Automatic Detection and Protection from Indoor Hazards: An Experimental Study	Increase the speed and effectiveness of hazard mitigation efforts.	Does not provide details on the specific sensors and relays used in the proposed system,	Can provide specific details on sensors.
Global System for Mobile Communication based Automatic Alarm System for Disaster Management	The system is designed to automatically detect and alert users to gas leaks, potentially reducing the risk of accidents and damage.	The effectiveness of the system in different environments and scenarios may need further testing and validation.	Improve in further Testing and validation.

Gas leakage detection and controlling using IOT sensor	Intelligent and internet-based solution for gas safety monitoring.	while the MQ6 gas sensor is a versatile and inexpensive option for detecting LPG gas,	Improve in potential limitations.
.IoT Based Two Way Safety Enabled Intelligent Stove with Age Verification Using Machine Learning	The pros of this research are that it addresses an important safety concern in households and proposes a solution utilizing the latest technologies.	Potential limitations in the accuracy of the age detection feature and the possibility of false alarms in the gas leakage detection feature.	Accuracy can be increased.
Smart Evacuation Alert System using IoT	Real-time monitoring High accuracy Automation	Limited applicability Cost Data privacy and security	Can be improved in Data privacy.
The Safety Status and Foreign Object Detection Model of Tobacco Cabinets Based on Deep Learning with IoT-Cloud	The proposed objective and methodology are clear and well-defined, indicating a structured approach to the research problem.	Does not mention any results or findings from the research work, which may make it difficult to assess the impact and significance .	Should be provided experimental results.

CHAPTER 3
SYSTEM ANALYSIS

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In this existing system, the gas leakage and low gas detection are identify through smell by the person. And also anybody can access the gas stove in their home.Researchers have focused a lot of effort on the crucial issue of identifying and alerting systems for traffic accidents. During the past few decades, many different accident detection and alerting techniques have been suggested.Leakage detection is essential to prevent the loss of these energy resources because pipelines are one of the primary means of transporting hydrocarbon fluids and gasses. In this article, based on research in field data, pipeline leakage as a rise in the mean value of the flow rate differential between the inlet and the outlet sensors, where the increased value is unknown and susceptible to change.The promotion of such automatic gas leak protection systems based on gas sensors that can quickly identify gas leaks like LPG, smoke, etc. is the other goal.

The small processing devices could not be used for complex computation or additional overheads. Smart IoT-based home automation systems are quite popular, but they need to be well protected against outside intrusions and security concerns. Using change identification techniques, IoT device security is therefore of utmost significance. There is discussion of certain emerging trends and current security issues with IoT-based applications.

DISADVANTAGE

- It creates fire accident.
- Not safe.
- Child easily access the dangerous products
- Age is not detected
- Alert buzzers not used

3.2 PROPOSED SYSTEM

The proposed model is to build a machine learning model that is capable of classifying the age of the person.In this project child don't access the gas stove and we are used age detection depend upon the gas valve will be open. The dataset given from the python machine learning. The machine learning is generally build to tackle these type of complicated task. It takes more amount of time to analyse these type of data manually. The machine learning can be used to classify the text whether the person is an adult/child, by using the previous data and make them to understand the pattern and improve the accuracy of the model by adjusting parameters and use that model as the classification model.This method also find the gas in that cylinder and send the information to the consumer.In proposed method, here detecting age and ON/OFF the solenoid valve depends on age using python deep learning.Here, giving buzzer alert for gas leakage and information passed through the information.

ADVANTAGE

- More Safe by using age detection.
- Child didn't access the gas stove.
- Age detection provide more security.

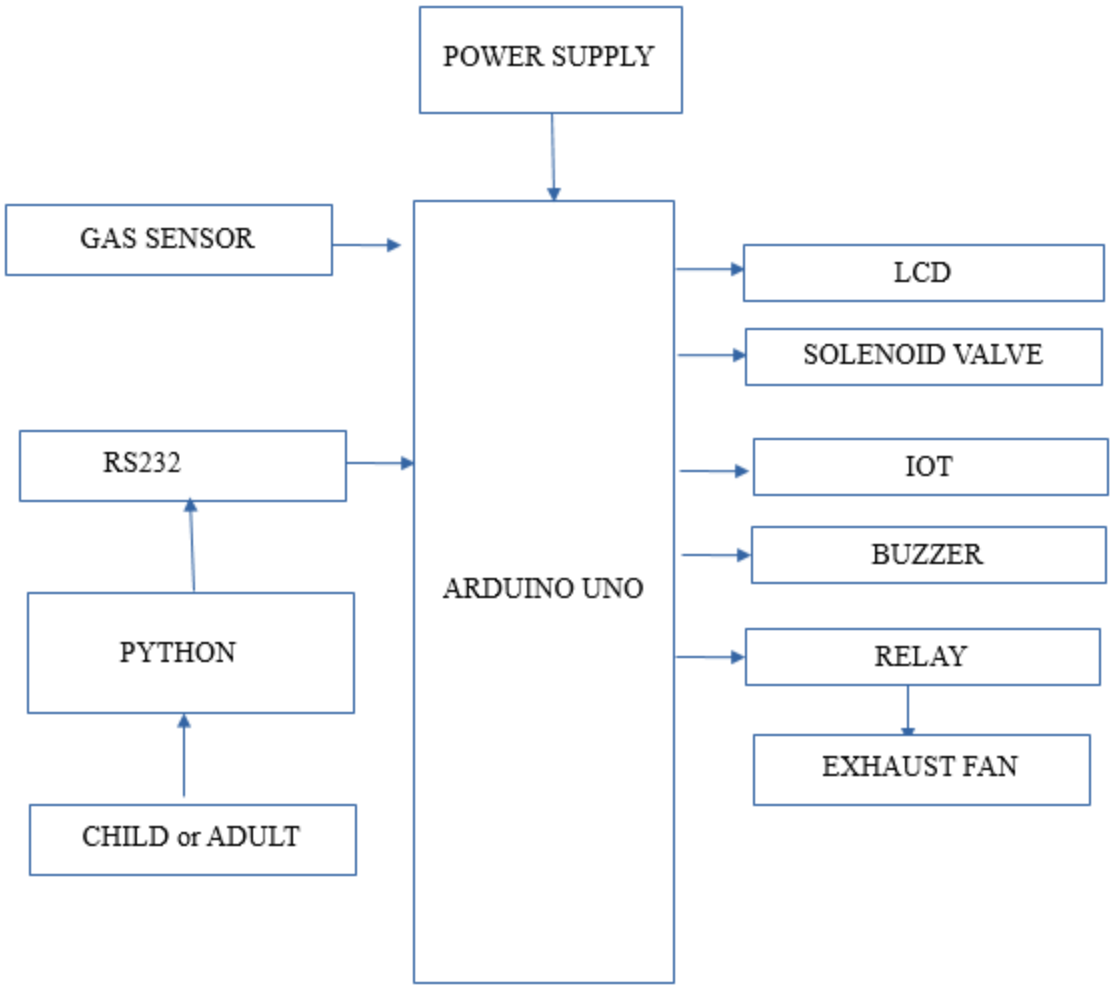


Figure 3.2 working of proposed model

3.3. Feasibility study

Data Wrangling

Data Manipulation/Data wrangling the data will be loaded, checked for accuracy, and then cleaned and trimmed for analysis in this portion of the report. Be sure to carefully document the cleaning decisions and provide justification.

Data collection

In order to monitor the functioning of devices linked to the Internet of Things, sensors are used in IoT data collecting. By gathering and transmitting real-time data that can be saved and accessed at any time, the sensors keep tabs on the state of the IoT network.

Preprocessing

Data preprocessing changes the data into a format that can be processed in data mining, machine learning, and other data science tasks more quickly and efficiently. Typically, the methods are applied in the early phases of machine learning.

3.4 PROJECT REQUIREMENTS

1.SOFTWARE REQUIREMENTS

- ARDUINO IDE
- EMBEDDED C
- JUPITER PYTHON

2.HARDWARE REQUIREMENTS

- ARDUINO UNO
- POWER SUPPLY
- MQ-2 GAS SENSOR
- LCD

- SOLENOID VALVE
- BUZZER
- RELAY
- EXHAUST
- IOT
- RS232

CHAPTER 4

SYSTEM DESIGN

CHAPTER-4

SYSTEM DESIGN

4.1 UML DIAGRAMS

4.1.1 USE CASE DIAGRAM

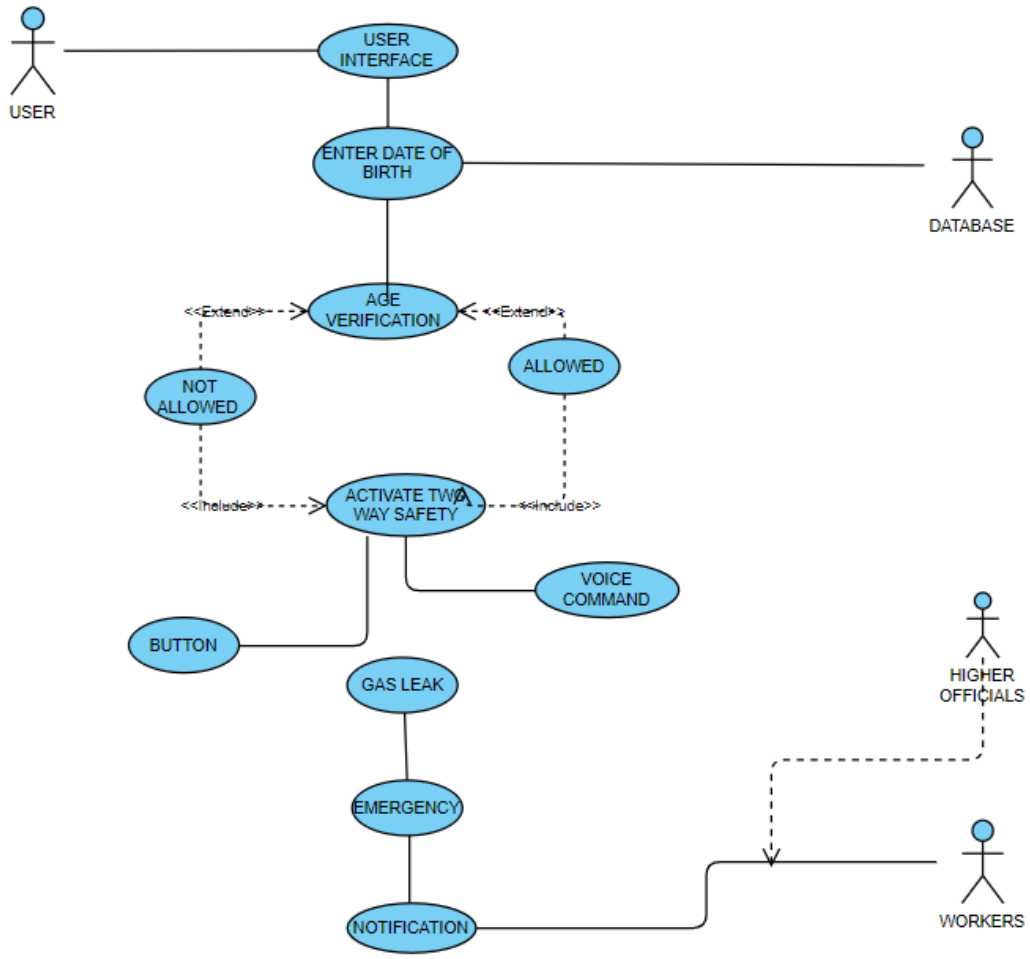


Figure 4.1 Use case Diagram

The above use case diagram contains users who give input as that detects the age of the users, that will be stored in the database and gives alert notifications to workers or other higher officials if there is any gas leakage.

4.1.2 ACTIVITY DIAGRAM

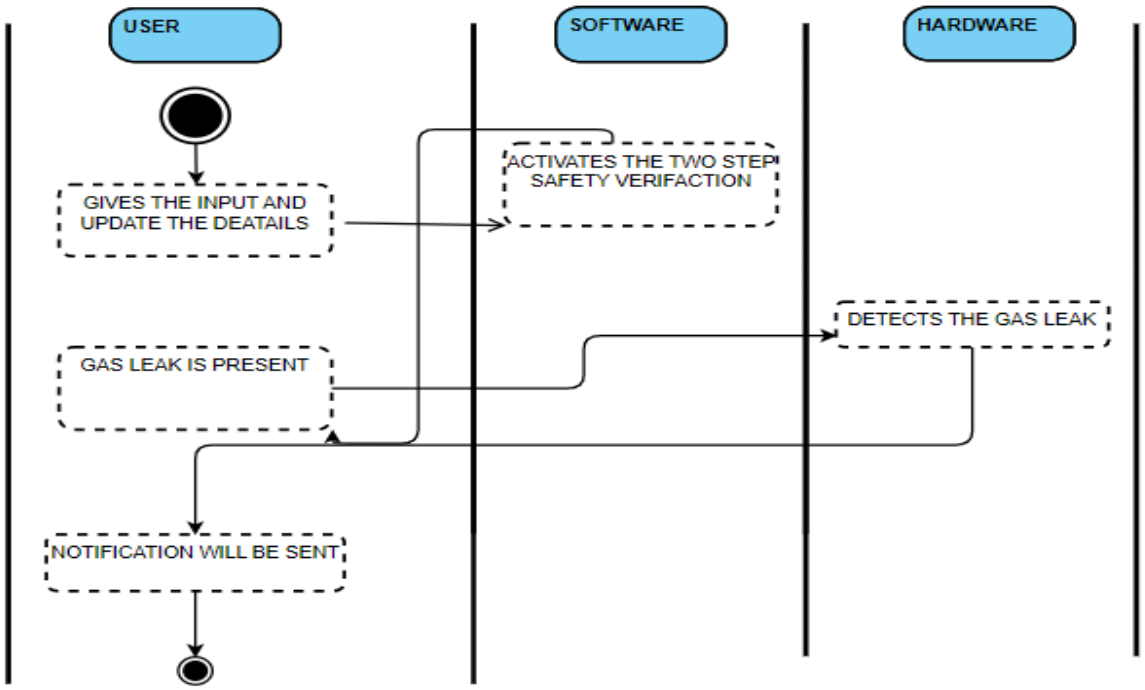
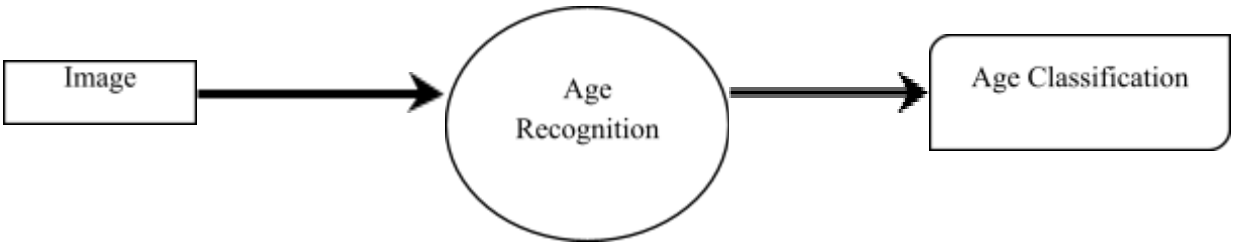


Figure 4.2 Activity diagram

This activity diagram represents the entire activities that are covered from the user giving input to the software that verifies the age of the person and detects any gas leakage occurs with the help of hardware.

4.2 DATA FLOW DIAGRAM

LEVEL 0



LEVEL 1

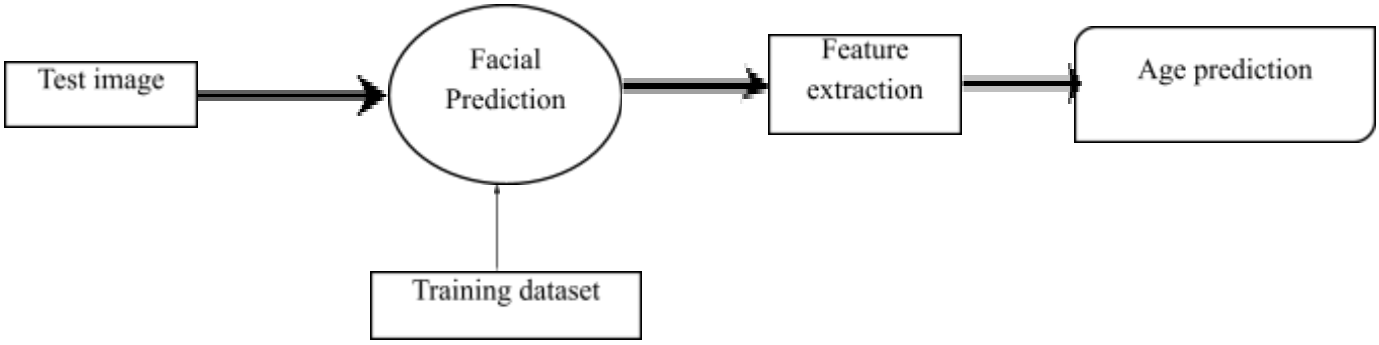


Figure 4.3 Data flow Diagram

The Level 0 DFD diagram shows that input will be given and the age is verified by the software and the tells whether the person is allowed are not.

CHAPTER 5

SYSTEM ARCHITECTURE

CHAPTER-5

SYSTEM ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

The ARDUINO UNO is used in this proposed method to detect the abnormality and control for safety. The LCD is used to display the current execution process. The python image processing is used to detect the person age. Solenoid valve is used to ON and OFF the valve and leaking the gas. MQ6 gas sensor is used to detect the gas leakage and if any gas leakage buzzer will alert. The buzzer is used to provide the audio alert. IOT is used to update the sensor in the webpage.

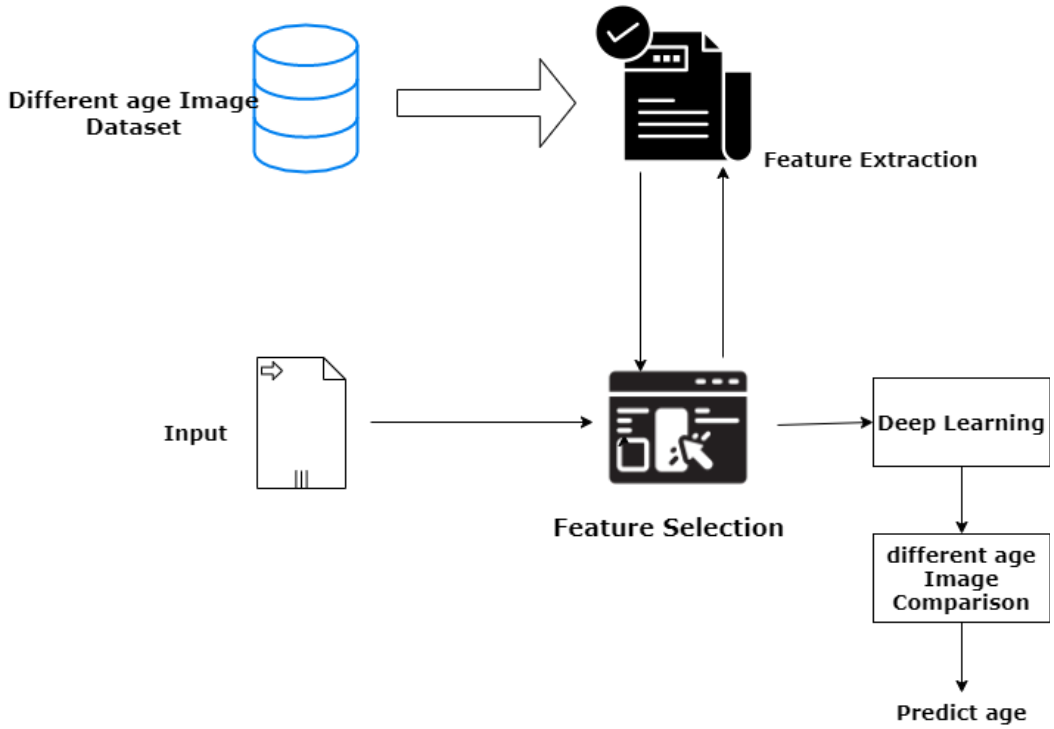


Figure 5.1 System Architecture

CHAPTER 6
SYSTEM IMPLEMENTATION

CHAPTER-6
SYSTEM IMPLEMENTATION

6.1 MODULE DESCRIPTION

LIST OF MODULES

- IMPORT THE GIVEN IMAGE FROM DATASET
- TO TRAIN THE MODULE BY GIVEN IMAGE DATASET
- WORKING PROCESS OF LAYERS IN CNN MODEL
- AGE CLASSIFICATION PREDICTION

6.1.1 IMPORT THE GIVEN IMAGE FROM DATASET

We have to import our data set using keras preprocessing image data generator function also we create size, rescale, range, zoom range, horizontal flip. Then we import our image dataset from folder through the data generator function. Here we set train, test, and validation also we set target size, batch size and class-mode from this function we have to train using our own created network by adding layers of CNN.

6.1.2 TO TRAIN THE MODULE BY GIVEN IMAGE DATASET

To train our dataset using classifier and fit generator function also we make training steps per epoch's then total number of epochs, validation data and validation steps using this data we can train our dataset.

6.1.3 WORKING PROCESS OF LAYERS IN CNN MODEL

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

6.1.4.AGE CLASSIFICATION PREDICTION

We give input image using Keras preprocessing package. That input Image converted into array value using pillow and image to array function package. We have already classified age image dataset. It classifies what are the age is. Then we have to predict our age using predict function.

The age recognition method is based on a two-channel architecture that is able to recognize classification of age s. The age images are used as the input into the inception layer of the CNN. The Training phase involves the feature extraction and classification using convolution neural networks

6.2 HARDWARE DESCRIPTION

6.2.1 ARDUINO UNO

The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

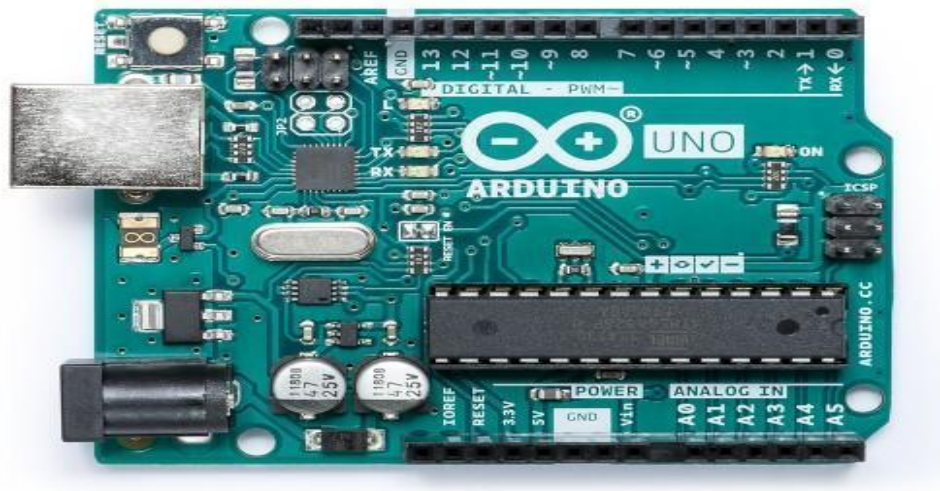


Figure 6.4 Arduino

A microcontroller board called Arduino Uno is based on the ATmega328P. (datasheet). It contains a 16 MHz quartz crystal, 6 analogue inputs, 14 digital input/output pins (of which 6 may be used as PWM outputs), a USB port, a power connector, an ICSP header, and a reset button. It comes with everything needed to support the microcontroller; to get started, just plug in a USB cable, an AC-to-DC converter, or a battery. You can experiment with your UNO without being very concerned that you'll make a mistake; in the worst case, you can replace the chip for a few dollars and start over.

6.2.2 POWER SUPPLY

This section describes how to generate +5V DC power supply.The power supply section is the important one. It should deliver constant output regulated power supply for successful working of the project. A 0-12V/1 mA transformer is used for this purpose. The primary of this

transformer is connected in to main supply through on/off switch& fuse for protecting from overload and short circuit protection. The secondary is connected to the diodes to convert 12V AC to 12V DC voltage. And filtered by the capacitors, which is further regulated to +5v, by using IC 7805.

6.2.3 BUZZER

An inexpensive yet effective part to include sound characteristics in our project or system is a buzzer. Because of its 2-pin structure's modest size and ease of usage on PCBs, Perf Boards, and other surfaces, this part is employed in many electronic applications.Buzzers come in two different categories and are widely accessible. The buzzer that is being demonstrated here is a straightforward device that, when activated, emits a continuous beeeeeeppp sound.

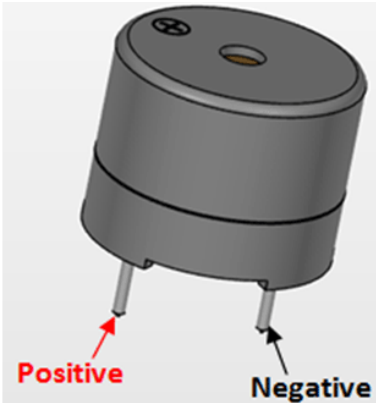


Figure 6.5 Active Passive Buzzer

A premade buzzer, which will be bigger than this and emit a beep, is another sort. Beep. Beep. Due to an inbuilt oscillating circuit, it produces sound. But, the one displayed here is the most popular since it can be modified with the aid of additional circuits for easy in our application.

6.2.4 LCD

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.

6.2.5 RS (REGISTER SELECT):

A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register.

6.2.6 GAS SENSOR INTERFACE WITH ARDUINO

Figure 6.6 Gas sensor

Sensitive material of MQ-2 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electro circuit, Convert change of conductivity to correspond output signal of gas concentration. MQ-2 gas sensor has high sensitivity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

6.2.7 RELAY:

Relays are the primary protection as well as switching devices in most of the control processes or equipment. All the relays respond to one or more electrical quantities like voltage

or current such that they open or close the contacts or circuits. A relay is a switching device as it works to isolate or change the state of an electric circuit from one state to another.

6.2.8 ELECTRIC SOLENOID VALVE

Figure 6.7 Solenoid Valve

- 12V DC Solenoid Water Air Valve Switch (Normally Closed) 1/2 controls the flow of fluid (liquid or air) and acts as a valve between high-pressure fluid!. This liquid valve would make a great addition to your robotic gardening project. There are two (Nominal NPT) outlets. Normally, the valve is closed.
- When a 12V DC supply is applied to the two terminals, the valve opens and water can push through.
- The valve works with the solenoid coil which operates electronically with DC 12 volt supply. As it is a normally closed assembly, it opens the flow of fluids as soon as it is powered ON and stops/blocks the flow when the supply voltage removed.

6.2.9 INTERNET OF THINGS

The **internet of things** (**IoT**) is the network of physical devices, vehicles, buildings and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. In 2013 the Global Standards Initiative on Internet of Things (IoT-GSI) defined the IoT as "the infrastructure of the information society. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic

benefit. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

6.3 SOFTWARE REQUIREMENTS

6.3.1 EMBEDDED C

Embedded C is most popular programming language in software field for developing electronic gadgets. Each processor used in electronic system is associated with embedded software.Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.Let's see the block diagram representation of embedded system programming:

The Embedded C code written in above block diagram is used for blinking the LED connected with Port0 of microcontroller.

In embedded system programming C code is preferred over other language. Due to the following reasons:

- Easy to understand
- High Reliability
- Portability
- Scalability

6.3.2 SYSTEM:

System is a way of working, organizing or performing one or many tasks according to a fixed set of rules, program or plan.It is an arrangement in which all the unit combined to perform a work together by following certain set of rules in real time computation. It can also be defined as a way of working, organizing or doing one or many tasks according to a fixed plan.

An Embedded System is a system that has software embedded into computer-hardware, which makes a system dedicated for a variety of application or specific part of an application or product or part of a larger system.An embedded system can be a small independent system or a large combinational system. It is a microcontroller-based control system used to perform a specific task of operation.

An embedded system is a combination of three major components:

- **Hardware**
- **Application software**
- **RealTimeOperatingsystem(RTOS):**

. 6.3.3 ARDUINO SOFTWARE IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

CHAPTER 7
PERFORMANCE EVALUATION

CHAPTER 7

PERFORMANCE EVALUATION

7.1 ACCURACY OF TRAIN AND TEST DATASET

Figure 7.1 Screenshot of accuracy of test and train dataset

CHAPTER 8
CONCLUSION

CHAPTER 8

CONCLUSION

8.1 CONCLUSION AND FUTURE ENHANCEMENT

In this project, a research to classify facial ages over static facial images using deep learning techniques was developed. This is a complex problem that has already been approached several times with different techniques. While good results have been achieved using feature engineering, this project focused on feature learning, which is one of DL promises. While feature engineering is not necessary, image pre-processing boosts classification accuracy. Hence, it reduces noise on the input data. Nowadays, facial age detection software includes the use of feature engineering. A solution totally based on feature learning does not seem close yet because of a major limitation. Thus, age classification could be achieved by means of deep learning techniques.This device can be introduces compactly so that it can be easy to carry.Further this alert system can be monitored through AI system.

APPENDICES

A1.SOURCE CODE

A.1.1 HARDWARE COMPONENTS CODE

ARDUINO

```
#include "gas.h"
#define relay 7
#define relay_2 2
const int buzzerPin = 5;
void iot_send(String s);
#include <LiquidCrystal.h>
int rs = 13, re = 12, d4 = 11 , d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, re, d4, d5, d6, d7);
int gas ;
char a;
void setup() {
  Serial.begin(9600);
  pinMode(gas, INPUT);
  pinMode(relay_2, OUTPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH);
  digitalWrite(relay_2, HIGH);
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("hello,world");
 // lcd.clear();
```

```
}

void loop() {
  gas = analogRead(A1);
  iot_send("@" + String(gas) + "#");
  if (gas < 700) {
    tone(buzzerPin, 1000);
    digitalWrite(relay_2, LOW);
    Serial.println(gas);
    Serial.println("gas lassing");
    lcd.setCursor(0, 0);
    lcd.print("GAS LOSSING");
    delay(1000);
  }
  else   {
    digitalWrite(relay_2, HIGH);
    Serial.println("gas not lassing");
    Serial.println(gas);
    delay(1000);
    lcd.setCursor(0, 0);
    lcd.print("GAS NOT LOSSING");
  }
  while (Serial.available() > 0)
    // Serial.print(gas);

  {
```

```
a = Serial.read();
  Serial.println(a);


  if (a == 'B')
  {
    tone(buzzerPin, 1000);
    digitalWrite(relay, HIGH);
    Serial.println("GAS NOT FLOWING,but entered  BABY ");
     iot_send("* CHILD ENTERED #");
    lcd.setCursor(0, 0);
    lcd.print("GAS off");
    lcd.setCursor(0, 1);
    lcd.print("GAS off,BABY");
  }


  if (a == 'A')
  {
    noTone(buzzerPin);
    digitalWrite(relay, LOW);
    Serial.println("GAS FLOWING ");
  iot_send("*  ADAELT ENTERED #");
    lcd.setCursor(0, 0);
    lcd.print("GAS NOT LOSSING");
    lcd.setCursor(3, 1);
    lcd.print("GAS FLOW,18+");
  }
```

```
  }
}
void iot_send(String s)
{
  for (int i = 0; i < s.length(); i++)
  {
    Serial.write(s[i]);
  }
  delay(3000);
}
```

NODE MCU

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include<SoftwareSerial.h>
SoftwareSerial ss(D1, D2);//rx tx
/************************ WiFi Access Point *********************************/
String tdata, hdata, bpdata, f1data, f2data;
int count = 0, count1 = 0, count2 = 0, count3 = 0,  count4 = 0, uvc = 0;
#define WLAN_SSID       "M0527"
#define WLAN_PASS       "12345678"

/************************ Adafruit.io Setup *********************************/

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883                   // use 8883 for SSL
```

```
#define AIO_USERNAME    "M0527"
#define AIO_KEY         "aio_rMOY40WKeESRHnhZgPRmV4R3tXKq"

/************ Global State (you don't need to change this!) ******************/

// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;
// or... use WiFiFlientSecure for SSL
//WiFiClientSecure client;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

/****************************** Feeds ***************************************/

// Setup a feed called 'photocell' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish covid1 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/info");
Adafruit_MQTT_Publish hb1 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/gas_value");
Adafruit_MQTT_Publish spo21 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/");
Adafruit_MQTT_Publish temp1 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/");
Adafruit_MQTT_Publish speed1 = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/");

Adafruit_MQTT_Subscribe mode0 = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME
"/feeds/mode0");
Adafruit_MQTT_Subscribe fan = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/fan");
```

```
//Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

/************************** Sketch Code ************************************/


// Bug workaround for Arduino 1.6.6, it seems to need a function declaration

// for some reason (only affects ESP8266, likely an arduino-builder bug).

void MQTT_connect();

char tdata1[50];

char hdata1[5];

char bpdata1[10];

char f1data1[5];

char f2data1[5];

void setup() {

  Serial.begin(9600);

  ss.begin(9600);

  pinMode(BUILTIN_LED, OUTPUT);

  digitalWrite(BUILTIN_LED, HIGH);

  pinMode(D1, OUTPUT);

  delay(10);

  digitalWrite(D4, LOW);

  pinMode(D4, OUTPUT);


  Serial.println(F("Adafruit MQTT"));

  Serial.println(); Serial.println();

  Serial.print("Connecting to ");

  Serial.println(WLAN_SSID);


  WiFi.begin(WLAN_SSID, WLAN_PASS);
```

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

  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());

  mqtt.subscribe(&mode0);
  mqtt.subscribe(&fan);
}

void loop() {

  MQTT_connect();
  count = 0;
  count1 = 0;
  count2 = 0;
  count3 = 0;
  count4 = 0;
  tdata = "";
  hdata = "";
  bpdata = "";
  f1data = "";
  f2data = "";
  while (Serial.available() > 0)
  {
```

```
char c = Serial.read();
// Serial.print(c);
if (c == '*')
{
  while (Serial.available() > 0)
  {
    char t = Serial.read();
    if (count == 50)
    {
      break;
    }
    if (t == '#') {
      break;
    }
    tdata = tdata + t;
    count++;
  }
}
if (c == '@')
{
  while (Serial.available() > 0)
  {
    char h = Serial.read();
    if (count1 == 1)
    {
      break;
    }
    if (h == '#') {
```

```
      break;
    }
    hdata += h;
    count1++;
  }
}
if (c == '$')
{
  while (Serial.available() > 0)
  {
    char u = Serial.read();
    if (count2 == 4)
    {
      break;
    }
    if (u == '#') {
      break;
    }
    bpdata += u;
    count2++;
  }
}
if (c == '^')
{
  while (Serial.available() > 0)
  {
    char v = Serial.read();
    if (count3 == 3)
```

```
      {
        break;
      }
      if (v == '#') {
        break;
      }
      f1data += v;
      count3++;
    }
  }
  if (c == '&')
  {
    while (Serial.available() > 0)
    {
      char l = Serial.read();
      if (count3 == 3)
      {
        break;
      }
      if (l == '#') {
        break;
      }
      f2data += l;
      count4++;
    }
  }
}
```

```
if ( tdata.length() > 0 )
{
  for (int i = 0; i < 50; i++) {
    tdata1[i] = tdata[i];
  }

  Serial.println(tdata1);
  covid1.publish(tdata1);
  digitalWrite(D4, HIGH);
  delay(1000);
  digitalWrite(D4, LOW);
  tdata = "";

  delay(2000);
}

if ( hdata.length() > 0 )
{
  for (int i = 0; i < 1; i++) {
    hdata1[i] = hdata[i];
  }
  Serial.println(hdata1);
  hb1.publish(hdata1);
  digitalWrite(D4, HIGH);
  delay(1000);
  digitalWrite(D4, LOW);
  hdata = "";
  delay(2000);
```

```
}
if ( bpdata.length () > 0 )
{
  for (int i = 0; i < 5; i++) {
    bpdata1[i] = bpdata[i];
  }

  Serial.println(bpdata1);
  spo21.publish(bpdata1);
  digitalWrite(D4, HIGH);
  delay(1000);
  digitalWrite(D4, LOW);
  bpdata = "";
  delay(2000);
}
if ( f1data.length () > 0 )
{
  for (int i = 0; i < 5; i++) {
    f1data1[i] = f1data[i];
  }

  Serial.println(f1data1);
  temp1.publish(f1data1);
  digitalWrite(D4, HIGH);
  delay(1000);
  digitalWrite(D4, LOW);
  f1data = "";
  delay(2000);
```

```
}
if ( f2data.length () > 0 )
{
  for (int i = 0; i < 5; i++) {
    f2data1[i] = f2data[i];
  }
  Serial.println(f2data );
  speed1.publish(f2data1);
  digitalWrite(D4, HIGH);
  delay(1000);
  digitalWrite(D4, LOW);
  f1data = "";
  delay(2000);
}
tdata = "";
hdata = "";
bpdata = "";
f1data = "";
f2data = "";
count  = 0;
count1 = 0;
count2 = 0;
count3 = 0;
count4 = 0;
delay(100); // ThingSpeak will only accept updates every 15 seconds.
```

```
Adafruit_MQTT_Subscribe *subscription;
while ((subscription = mqtt.readSubscription(5000))) {
  if (subscription == &mode0) {
    Serial.print(F("Gotm: "));
  }

  if (strcmp((char *)mode0.lastread, "AUTO") == 0) {
    Serial.write("AUTO");
    ss.write('A');
  }

  if (strcmp((char *)mode0.lastread, "MANUAL") == 0) {
    Serial.println("MANUAL");
    ss.write('M');
  }
//}

//while ((subscription = mqtt.readSubscription(5000))) {
  if (subscription == &fan) {
    Serial.print(F("Gotf: "));
  }
  if (strcmp((char *)fan.lastread, "ON") == 0) {
    Serial.write("ON");
    ss.write('X');
  }

  if (strcmp((char *)fan.lastread, "OFF") == 0) {
    Serial.println("OFF");
```

```
      ss.write('Y');
    }
  }
}

// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
  int8_t ret;

  // Stop if already connected.
  if (mqtt.connected()) {
    return;
  }

  Serial.print("Connecting to MQTT... ");

  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000);  // wait 5 seconds
    retries--;
    if (retries == 0) {
      // basically die and wait for WDT to reset me
      while (1);
    }
```

```
  }

  Serial.println("MQTT Connected!");

}
```

A.1.2 SOFTWARE COMPONENTS CODE

M1.ipynb

To build a model for training and testing:

```
import os
import numpy as np # linear algebra
import matplotlib.pyplot as plt


# Dl framwork - tensorflow, keras a backend
import tensorflow as tf
import tensorflow.keras.backend as K
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
from tensorflow.keras.layers import Conv2D, SeparableConv2D, MaxPool2D, LeakyReLU, Activation
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from IPython.display import display
from os import listdir
from os.path import isfile, join
from PIL import Image
import glob
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense


import warnings
warnings.filterwarnings('ignore')

dir_name_train_6to20 = 'Dataset/train/6-20'
dir_name_train_25to30 = 'Dataset/train/25-30'
dir_name_train_42to48 = 'Dataset/train/42-48'
dir_name_train_60to98 = 'Dataset/train/60-98'

def plot_images(item_dir, n=6):
    all_item_dir = os.listdir(item_dir)
```

```
    item_files = [os.path.join(item_dir, file) for file in all_item_dir][:n]

    plt.figure(figsize=(80, 40))
    for idx, img_path in enumerate(item_files):
        plt.subplot(7, n, idx+1)
        img = plt.imread(img_path)
        plt.imshow(img, cmap='gray')
        plt.axis('off')

    plt.tight_layout()


def Images_details_Print_data(data, path):
    print(" ====== Images in: ", path)
    for k, v in data.items():
        print("%s:\t%s" % (k, v))

def Images_details(path):
    files = [f for f in glob.glob(path + "**/*.*", recursive=True)]
    data = {}
    data['images_count'] = len(files)
    data['min_width'] = 10**100  # No image will be bigger than that
    data['max_width'] = 0
    data['min_height'] = 10**100  # No image will be bigger than that
    data['max_height'] = 0


    for f in files:
        im = Image.open(f)
        width, height = im.size
        data['min_width'] = min(width, data['min_width'])
        data['max_width'] = max(width, data['max_height'])
        data['min_height'] = min(height, data['min_height'])
        data['max_height'] = max(height, data['max_height'])

    Images_details_Print_data(data, path)


print("")
print("Trainned data for 6-20:")
print("")
Images_details(dir_name_train_6to20)
print("")
plot_images(dir_name_train_6to20, 10)

print("")
print("Trainned data for 25-30:")
print("")
Images_details(dir_name_train_25to30)
print("")
plot_images(dir_name_train_25to30, 10)
```

```
print("")
print("Trainned data for 42-48:")
print("")
Images_details(dir_name_train_42to48)
print("")
plot_images(dir_name_train_42to48, 10)

print("")
print("Trainned data for 60-98:")
print("")
Images_details(dir_name_train_60to98)
print("")
plot_images(dir_name_train_60to98, 10)


Classifier=Sequential()
Classifier.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
Classifier.add(MaxPooling2D(pool_size=(2,2)))
Classifier.add(Flatten())
Classifier.add(Dense(38, activation='relu'))


Classifier.add(Dense(4, activation='softmax'))
Classifier.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'])


train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=Tr
ue)
test_datagen=ImageDataGenerator(rescale=1./255)


training_set=train_datagen.flow_from_directory('dataset/Train',target_size=(128,128),batch_size=32,clas
s_mode='categorical')
test_set=test_datagen.flow_from_directory('dataset/Test',target_size=(128,128),batch_size=32,class_mod
e='categorical')


img_dims = 150
epochs = 10
batch_size = 32


#### Fitting the model
history = Classifier.fit_generator(
        training_set, steps_per_epoch=training_set.samples // batch_size,
        epochs=epochs,
        validation_data=test_set,validation_steps=test_set.samples // batch_size)


def graph():
   #Plot training & validation accuracy values
```

```
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()

    # Plot training & validation loss values
    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()
graph()
```

M2.ipynb

Module 2:

```
# Dl framwork - tensorflow, keras a backend
import tensorflow as tf

import tensorflow.keras.backend as K

from tensorflow.keras.models import Model

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Input

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import Conv2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Dropout

from tensorflow.keras.layers import LeakyReLU

from tensorflow.keras.layers import Activation

from tensorflow.keras.optimizers import Adam
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.callbacks import ModelCheckpoint

from tensorflow.keras.callbacks import ReduceLROnPlateau

from tensorflow.keras.callbacks import EarlyStopping

import warnings
warnings.filterwarnings('ignore')


model = Sequential()
# 1st Convolutional Layer
model.add(Conv2D(filters=96, input_shape=(224,224,3), kernel_size=(11,11), strides=(4,4), padding='valid'))
model.add(Activation('relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))
# 2nd Convolutional Layer
model.add(Conv2D(filters=256, kernel_size=(11,11), strides=(1,1), padding='valid'))
model.add(Activation('relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))
# 3rd Convolutional Layer
model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='valid'))
model.add(Activation('relu'))
# 4th Convolutional Layer
model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='valid'))
model.add(Activation('relu'))
# 5th Convolutional Layer
model.add(Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), padding='valid'))
model.add(Activation('relu'))
# Max Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))
# Passing it to a Fully Connected layer
model.add(Flatten())
# 1st Fully Connected Layer
model.add(Dense(4096, input_shape=(224*224*3,)))
model.add(Activation('relu'))
# Add Dropout to prevent overfitting
model.add(Dropout(0.4))
# 2nd Fully Connected Layer
model.add(Dense(4096))
model.add(Activation('relu'))
# Add Dropout
model.add(Dropout(0.4))
# 3rd Fully Connected Layer
model.add(Dense(1000))
model.add(Activation('relu'))
# Add Dropout
model.add(Dropout(0.4))
# Output Layer
model.add(Dense(4))
model.add(Activation('softmax'))
```

```
model.summary()

# Compile the model
model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy'])


train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)


training_set=train_datagen.flow_from_directory('dataset/Train',target_size=(224,224),batch_size=32,class_mode='
categorical')
test_set=test_datagen.flow_from_directory('dataset/Test',target_size=(224,224),batch_size=32,class_mode='categor
ical')


img_dims = 150
epochs = 1
batch_size = 32


#### Fitting the model
history = model.fit(
        training_set, steps_per_epoch=training_set.samples // batch_size,
        epochs=epochs,
        validation_data=test_set,validation_steps=test_set.samples // batch_size)


import matplotlib.pyplot as plt


def graph():
   #Plot training & validation accuracy values
   plt.plot(history.history['accuracy'])
   plt.plot(history.history['val_accuracy'])
   plt.title('Model accuracy')
   plt.ylabel('Accuracy')
   plt.xlabel('Epoch')
   plt.legend(['Train', 'Test'], loc='upper left')
   plt.show()

   # Plot training & validation loss values
   plt.plot(history.history['loss'])
   plt.plot(history.history['val_loss'])
   plt.title('Model loss')
   plt.ylabel('Loss')
   plt.xlabel('Epoch')
   plt.legend(['Train', 'Test'], loc='upper left')
   plt.show()
graph()


print("[INFO] Calculating model accuracy")
scores = model.evaluate(test_set)
print(f"Test Accuracy: {scores[1]*100}")
```

M3.iypnb

```
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Flatten

from tensorflow.keras.layers import Dense

import warnings
warnings.filterwarnings('ignore')

Classifier=Sequential()

Classifier.add(Convolution2D(32,3,3,input_shape=(225,225,3),activation='relu'))
Classifier.add(MaxPooling2D(pool_size=(2,2)))
Classifier.add(Convolution2D(128,3,3,activation='relu'))
Classifier.add(MaxPooling2D(pool_size=(2,2)))
Classifier.add(Flatten())
Classifier.add(Dense(256, activation='relu'))
Classifier.add(Dense(4, activation='softmax'))

Classifier.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'])
Classifier.summary()

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

test_datagen=ImageDataGenerator(rescale=1./255)

training_set=train_datagen.flow_from_directory('dataset/Train',target_size=(225,225),batch_size=32,class_mode='
categorical')

test_set=test_datagen.flow_from_directory('dataset/Test',target_size=(225,225),batch_size=32,class_mode='categor
ical')

from IPython.display import display
```

```
img_dims = 150
epochs = 60
batch_size = 32


Classifier.fit_generator( training_set, steps_per_epoch=training_set.samples // batch_size,
        epochs=epochs,
        validation_data=test_set,validation_steps=test_set.samples // batch_size)



import h5py


Classifier.save('e.h5')


from keras.models import load_model


model=load_model('e.h5')


import numpy as np


from tensorflow.keras.preprocessing import image
test_image=image.load_img('c5.jpg',target_size=(225,225))


import matplotlib.pyplot as plt
img = plt.imshow(test_image)


test_image=image.img_to_array(test_image)


test_image=np.expand_dims(test_image,axis=0)


result=model.predict(test_image)


result


prediction = result[0]


classes=training_set.class_indices


classes
```

```
prediction=list(prediction)


prediction


classes=['25-30','42-48','6-20','60-98']
    output = zip(classes, prediction)
    output = dict(output)
    if output['25-30'] == 1.0:
        a="25-30"
    elif output['42-48'] == 1.0:
        a="42-48"
    elif output['6-20'] == 1.0:
        a="6-20"
    elif output['60-98'] == 1.0:
        a="60-98"
```

A2 SAMPLE SCREENS

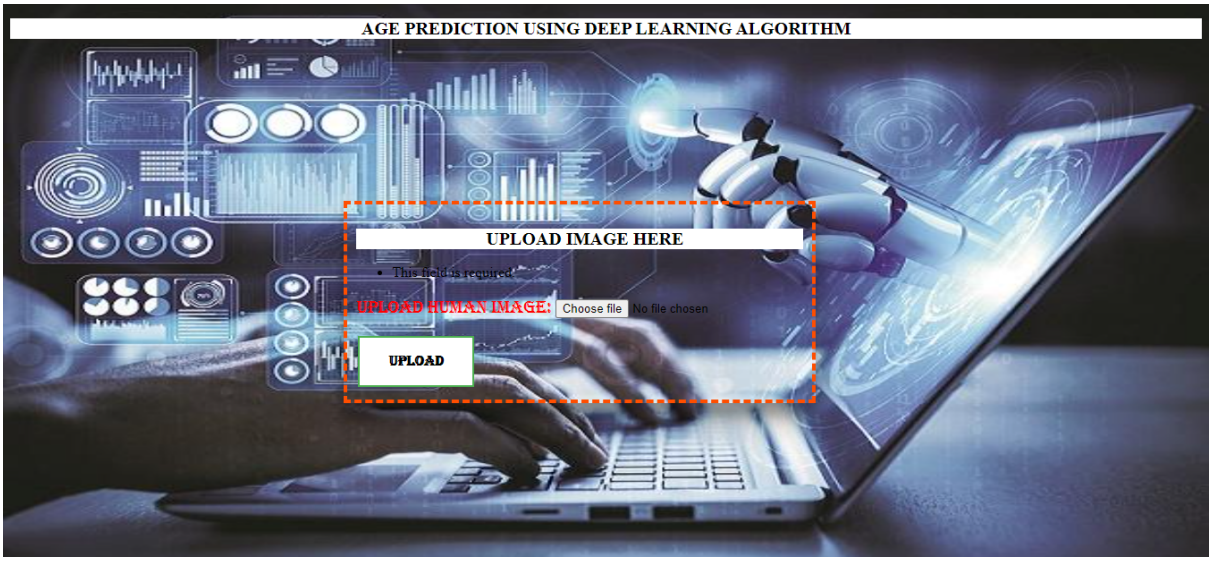


Figure A2.1 Screenshot of image upload page

Result Go Back!!!

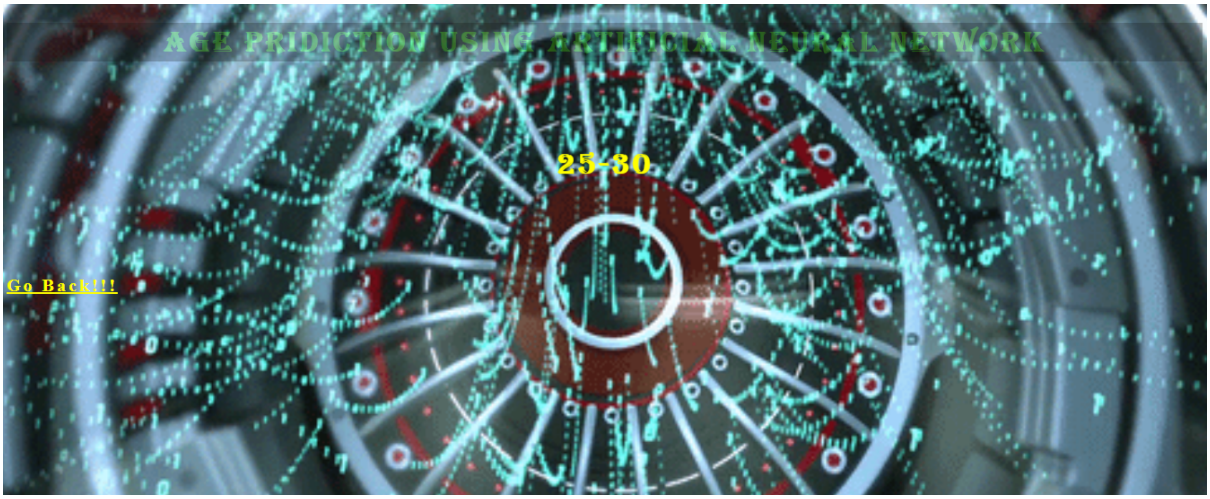


Figure A2.2 Screenshot of Result

REFERENCES :

[1] Iman Amini , Yindi Jing , Senior Member, IEEE, and Tongwen Chen"Adaptive Naive Bayes Classifier Based Filter Using Kernel Density Estimation for Pipeline Leakage Detection"naive Bayes classification and the estimated cumulative distribution function (CDF) Volume: 31, Issue: 1, January 2023.

[2] R. Islam, M. I. Hossain, M. S. Rahman,S. Kabir, M. S. R. Sohan, A. Shufian"Smart IoT System for Automatic Detection and Protection from Indoor Hazards: An Experimental Study" in 18 Sep 2022.

[3] Vibhu Singh,Karan,Vansh Panwar,Mandeep Kaur, Nitin Rakesh,Parma Nand "Global System for Mobile Communication based Automatic Alarm System for Disaster Management" using sensors in 25 March 2022.

[4] Mrs. B. Sudha Madhuri1 , G. Priyanka2 , V.M .Vaishnavi3 , T.S.R Ananya4 , N.Meghana "Gas leakage detection and controlling using IOT sensor" Vol. 11, Issue 6, June 2022.

[5] Mariha Afroz; Nazia Hasan; Md. Ishan Arefin Hossain"IoT Based Two Way Safety Enabled Intelligent Stove with Age Verification Using Machine Learning" in Aug 2021.

[6] G.A Senthil; P Suganthi; R. Prabha; M. Madhumathi; S. Prabhu; S Sridevi"An Enhanced Smart Intelligent Detecting and Alerting System for Industrial Gas Leakage using IoT in Sensor Network" this approach will aid in recognizing per user natural gas consumption. on 14 March 2023.

[7] Yellamma Pachipala,Ch. Kavya,R. Veera Dinesh"Smart Evacuation Alert System using IoT " using flame sensors on 18 March 2022.

[8] Chengyuan Wang; Junli Zhao; Zengchen Yu; Shuxuan Xie; Xiaofei Ji; Zhibo Wan "The Safety Status and Foreign Object Detection Model of Tobacco Cabinets Based on Deep Learning with IoT-Cloud" using deep learning with IOT on 28 March 2022.

[9] Homero J. Velasteguí,Acurio M. Santiago"IoT-based Security Alarm Protocol"using vulnerability detection techniques on 05 January 2022.

[10] Jatin Arora; Monika Sethi; Saravjeet Singh"Change Identification and Notification Framework for IoT Enabled Smart Security System" based smart appliances for controlling the operations of home devices raises an issue of data security on 13-14 October 2022.