SMART KITCHEN AND FOOD SPOIL DETECTION USING OPEN CV AND IOT TECHNOLOGY

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

S.SHYAM SUNDAR (REG NO:39110945)
S.SRINIVAAS (REG NO:39110971)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119

APRIL-2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **S.Shyam Sundar (39110945)** and **S.Srinivaas (39110971)** who carried out the Project Phase-2 entitled **“SMART KITCHEN AND FOOD SPOIL DETECTION USING OPEN CV AND IOT TECHNOLOGY”** under my supervision from JAN 2023 to APRIL 2023.

Internal Guide
Ms .V. DHARANI, M.E.

Head of the Department
Dr. L. LAKSHMANAN, M.E., Ph.D.

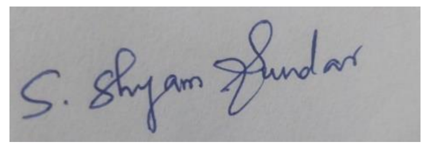
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

Submitted for Viva voce Examination held on 20.04.2023

InternalExaminer **ExternalExaminer**

DECLARATION

I, **S.Shyam sundar (Reg no:39110945),**hereby declare that the Project Phase-2 Report entitled **"SMART KITCHEN AND FOOD SPOIL DETECTION USING OPEN CV AND IOT TECHNOLOGY"** done by me under the guidance of **Ms. V. DHARANI, M.E.,** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.



DATE:20.04.2023

PLACE:Chennai **SIGNATURE OF THECANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr.L.Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.V.Dharani M.E.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This objectives to focus on the exclusive components of the IoT and its position within the consciousness of shrewd kitchens and meals waste. Apart from this, there has been a unique coverage of Smart Kitchen. Descriptions of various appliances and their applications inside the clever kitchen are indexed. In latest days, kitchen injuries have accelerated in business kitchens and in domestic kitchens. These accidents can be averted via IoT technology consisting of tracking of the whole kitchen from far flung places. To carry out this studies, it is crucial to hate and mourn. In hardware aspect, gasoline sensor, flame sensor, shipment mobile, exhaust fan are used. On the software aspect, cloud packages and incorporated messaging are used. All these sensors will be included into the Raspberry pi board for statistics transfer to the cloud. Additionally, the message logs may be generated the use of the same methods.

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 PURPOSE

We proposed the design and construction of an IoT based Smart Kitchen System are indicated here different types of sensors were used to detect gas leakages management system in a kitchen. Its outputs are then interfaced with a Raspberry pi MCU controller programmed in python language we can get this and much more safety feature that can be integrated with the automation system includes temperature sensor. The gas leaking situation Using IOT we can detect in the remote areas. Continues monitoring of services in kitchen is performed by this system.

1.2 PROJECT SCOPE

- This paper proposes an edge IoT & machine learning based approach for food quality monitoring.
- To monitor the kitchen from LPG gas detection using IOT.
- For smell monitoring in the smart home environment, with particular attention paid to food spoilage.
- Devoted to the problem of classification of food products by quality based on their color.

1.3 EXISTING SYSTEM

- There are numerous techniques available now a days in order to detect any gas leakage or the spilling of gas from the cylinders.
- The basic principle behind the technique is that the change in the concentration of LPG is detected associated it activates an audio visual alarm once it exceeds an explicit threshold worth. Further, it sends another message through radiofrequency system to the receiver module.
- Hence the receiver module can be of a mobile unit that will be placed anywhere among the premises of the house therefore the alarm area unit typically detected and detected at the place where gas flows.

- The arduino reads the voltage from the detector and uses it to calculate the modification in concentration.
- The gas detector is sensitive to many gases and actually gas kind cannot be determined. Instead, during the work, it is assumed that the gas sensing element has the same or the identical sensitivity for LPG and CH_4, it may be thought about a sound assumption.

1.4 PROPOSED SYSTEM

- We proposed the design and construction of an IoT based Smart Kitchen System are indicated here different types of sensors were used to detect gas leakages management system in a kitchen.
- Its outputs are then interfaced with an Raspberry pi MCU controller programmed in python language
- We can get this and much more safety feature that can be integrated with the automation system includes temperature sensor.
- The gas leaking situation Using IOT we can detect in the remote areas.
- Continues monitoring of services in kitchen is performed by this system.

1.5 BENEFIT OF PROPOSED SYSTEM

- Self-controlled and operating facility
- Good working system
- Suitable for handicapped peoples
- Portability and mobility

1.6 ARCHITECTURE OF THE PROPOSED SYSTEM

The Architecture of the optical character recognition system on a grid infrastructure consists of the three main components. They are:-

- Camera
- Python
- Output Interface

1.7 PROBLEM STATEMENT:

- We proposed the design and construction of an IoT based Smart Kitchen System are indicated Here different types of sensors were used to detect gas leakages management system in a kitchen.

- Its outputs are then interfaced with an Raspberry pi MCU controller programmed in python language

- We can get this and much more safety feature that can be integrated with the automation system includes temperature sensor.

- the gas leaking situation Using IOT we can detect in the remote areas.

- Continues monitoring of services in kitchen is performed by this system.

CHAPTER-2

2. LITERATURE REVIEW

1. Post harvesting process of fruit and vegetables is completed in several steps: washing, sorting, grading, packing,storage and transporting. Automatic fruit quality inspection system (2016)- blob detection technique is used for the detection of the rotten part, colour based segmentation and size detection technique is also used to identify the rotten parts basedon thresholding and binary images. Result will be accurately 90.3%. A Fruits quality management system based on image processing (2017) Analysing the fruits quality which is based on colour, shape and size should be done by non-destructive techniques because these are delicate material.it is also helpful in planning, packing, transport and marketing operations.

DISADVANTAGES

Indian in particular, we can't afford cost of today's fruit processing facilities. Its physical appearance affects its value in the market, so it is important to observe proper handling of fruits after harvesting.

MATERIAL

- Opencv
- Python
- Image Processing
- Web Camera

DISCUSSION

Texture, Colour and Size are the important parameters for fruit quality identification. Computer vision and image processing techniques have been found increasingly useful in the fruit industry, especially for applications in quality detection. Research in this area indicates the feasibility of using computer vision systems to improve product quality

The use of computer vision for the inspection of fruits has increased during recent years. The market constantly requires higher quality products and consequently, additional features have been developed to enhance computer vision inspection systems. Computer application in agriculture and food industries has been applied in the areas of inspection of fresh products. It indicates whether the fruit is good or bad based on the quality of the fruit. Exporting of fresh fruit is increased day to day from India. People are very conscious about their health; they prefer only fresh, good quality fruit. Texture, Colour and Size are the important parameters for fruit quality identification. The colour recognition is very important process in ripeness detection. The ripeness detection is external quality factor. But texture is also very important. Because of texture defected fruit can be recognized. Texture analysis detects the nonuniformity of fruit outer surface. The size is also important parameter. It clearly seen parameter all customer select fruit based on size.In this project the features required are colour, texture and size. To get exact feature pre-processing is done on acquired image. The main aim of image processing is an improvement of image so that unwanted distortions are suppressed and enhance image features which are important for further processing. The basic steps of preprocessing are first convert RGB image to gray scale image. Then image histogram equalization is applied on gray image. This helps in adjusting image intensities in order to enhance contrast. Remove noise with filter, here we use median filter for removing noise here laplacian is used for edge detection as it highlights the region with rapid intensity change. So this enhanced, noise free, filtered image is ready for further processing. In order to fulfil the consumer's desire and socio - economic requirement, fruit quality evaluation becomes very important now a day. Quality of produce encompasses sensory properties (appearance, texture, taste and aroma), nutritive values, chemical constituents, mechanical properties, functional properties and defects. All the fruits have limited shelf life during which it goes through structural and chemical changes. The ripeness detection is external quality factor. But texture is also very important. Because of texture defected fruit can be recognized. Texture analysis detects the non-uniformity of fruit outer surface. The size is also important parameter. It clearly seen parameter all customer select fruit based on size.

2. Meat freshness level is an important factor to determine meat quality for consumption. In this research, a sensor system has been designed to identify the freshness level of meat in fast, precise and non-destructive manners. The system is implemented into a Raspberry Pi equipped by gas and color sensors as the freshness identifier tools to replace the human olfaction and vision in determining a fresh meat. Pattern recognition powered by a neural network is used to identify the meat's freshness. The neural network inputs are the odors sensed by the gas sensor array of MQ-136, MQ-137, TGS 2620 and Red, Green, Blue values sensed by TCS 3200 color sensor. Three levels of freshness have been tested, such as fresh meat, half-rotten meat, and rotten meat. The usage of the three gas sensors and one color sensor of the system is capable to acquire a distinct pattern for the three categories of freshness. The freshness identification of the meat has a high percentage of success up to 80%. The errors are caused by the small different of the pattern sensed by sensors for halfrotten meat and rotten meat; these two kinds of meat fortunately are not consumable. Thus, it may conclude that the system has 100% success degree to identify fresh meat and non-fresh meat. The implementation of the system is expected to replace the traditional measurement by the human senses (i.e. nose and eyes) to obtain equal measurement as different human examiner acquires different result, and to eliminate the impact of bacteria or virus from meats to examiner. It may also replace measurement system using chemical substances so the tested meat will be still consumable.

DESIGN IMPLEMENTATION

The system design can be categorized into two parts which are hardware and software designs. The hardware design covers the circuit implementation. The software design covers the program embedded in Raspberry pi and Arduino microcontroller. The software system consists of ADC reader program and color sensor on Arduino and neural network program using Lazarus on Raspberry Pi. The block diagram of the overall system is illustrated in Figure 3. From the block diagram, it can be seen that the system identifies meat sample using electronic nose and color sensors placed in the sensor chamber. The electronic nose consists of an array of three different gas sensors: MQ-136, MQ137 and TGS 2602. The three gas sensors are exposed to the odor emitted by the meat tested. The gas sensors then respond to the aroma of the meat by generating different voltage signals depending on the level of freshness of the tested meat. The voltage values of the three sensors in the electronic nose are read through 10-bit ADC of the Arduino microcontroller. While the color sensor will respond to the color of the meat by taking the RGB value of the meat color tested. Signal data in the form voltages of three gas sensors and the RGB values of the color sensor will then be sent to Raspberry Pi via serial communication. The data received will be separated and classified by Lazarus software to become the inputs of the neural network. After the data are received, the data will be processed offline first before the online detection process done.

Hardware Design

The hardware system consists of electronic nose module, Arduino Uno microcontroller, Raspberry Pi, USB to TTL, color sensor, and sensor room/chamber construction. The chamber test has 10cm x 10cm x 10cm in size, as illustrated. The gas sensors are placed on the top of the chamber and the meat sample is placed on the bottom side of the chamber. Hardware design is begun with the electronic nose module in which three gas sensors are combined into a single PCB forming an array sensor. Then, the connection between the sensor array and microcontroller is set up. Accordingly, the hardware design includes the design of the chamber sensor. Figure 5 depicts the photograph of the array sensor of three gas sensors set up in a single PCB. The NH3 and H2S gasses are produced when a meat becomes rotten . In order to measure the gas concentration when meat sample experiences decomposition, we can use gas sensors that are capable of measuring the concentration of these gases. This research is utilizing gas sensors of MQ-136 and MQ-137 that can detect H2S and NH3 respectively. For quality measurement purpose, we also add a TGS 2602 gas sensor because it has high sensitivity with low concentration odor such as ammonia and H2S. It also has high sensitivity to VOC gas with low concentration like toluene from wood furnishing process.

As mentioned before, gas sensors have different resistance values when exposed to different types and concentrations of gases. The correlation between the value of the sensor resistance to the type and the gas concentration can be used to obtain information about the level of freshness of the meat. In theory, the fresh meat produces a different sensor response with meat that has begun to rot. By using the principle of basic voltage divider on each gas sensor, the freshness of the meat can be known as illustrated in Figure 6. Table 4 lists the application characteristic of the gas sensors. The value of load resistance (RL) is chosen to be fixed, while the sensor resistance value varies depending on the reaction to the gas type and concentration. We used RL of 10 kΩ for MQ-137 and TGS 2602, and 22 kΩ for MQ-136. The sensor responses exposed to clean air are summarized in Table 5. Theoretically, the magnitude of the voltage value at the load resistance can be calculated as in equation VL is the voltage at the load resistor that is connected to the A/D Converter, RL is the load resistance, RS is the resistance of the sensor and VC is the input voltage sensor which is 5 volts. TCS 3200 color sensor is used to retrieve data of the color characteristics of the meat under test. There are four color photodiode filters that are red, green, blue, and clear. The usages of these filter type are controlled by digital input “low” and “high” in the input pins of TCS3200 sensor chip which has been shown in Table 2. The output of this sensor is a square wave whose frequency will vary with the color detected by the photodiode. The frequency output can be scaled by the two input pins with digital input “low” and “high” as summarized in Table 3. In the design, to get the RGB value, the color sensor outputs are connected to the A/D converter pins of Arduino microcontroller. Therefore, it requires wiring connection between color sensors and Arduino.

Software Design

The software design includes programming of Arduino microcontroller, design of GUI and Neural Network in Raspberry. The GUI and NN designs are programmed using Lazarus software in Raspbian Operating System (OS), where the GUI display is an aid to run the process of identifying the freshness of meat. GUI view consists of two parts. The first part is the GUI view for the training process of the neural network as shown in Figure 8. In the GUI view for the first part shows several clickable buttons to run the training process. The first step in the training process is to enter the value of the training gain, momentum update and Minimum Error on each box that has been provided. After the parameters of the training process is completed then the training process can be started by clicking TRAIN button on the GUI. The graphic shown in this view serves to display the current error value against the number of iterations that have been passed. After the training process is complete, then the last weight and bias can be displayed by clicking the SHOW button. The value of weight and bias of training results can also be stored in the form of text data that is ".txt" in the folder that has been determined by the user. The second part of GUI display is an online user display interface of the process of reading data from gas sensors and color sensors and the process of identifying freshness of meat. To perform serial communication correctly, the first step is to fill the selection box from com port by choosing a path to use for serial communication. After filling the serial communication lines, the next step is to open the communication path that has been created by clicking the OPEN button. To close the serial communication path, click the CLOSE button. If the required serial communication parameters are in place, the serial data sent by Arduino will be received by the GUI immediately after the OPEN button is clicked. Information about the data from each sensor will be displayed in each box. The group box timer as shown in the GUI will start counting for 60 seconds when the START button is clicked. When the 60th second, the identification process will take place and the freshness of the sample meat will be shown by the GUI display. Figure 10 shows a serial data communication in Raspberry from Arduino The signal from the three gas sensors and the RGB values of the color sensor that has been processed by the A/D converter of Arduino microcontroller will be sent to Raspberry Pi in one data line via serial communication. Data delivery format that can separate and mark the process of data retrieval is required. Here, the marks A, B, C, D, E, and F are used to mark Red, Green, Blue, MQ-136, MQ-137, and TGS-2602 respectively. In example, data received on Raspberry Pi will have the format as shown in Figure 11. Later this data will be broken down by using a software program on the Lazarus in order to generate a single data for each information from three gas sensors and RGB of the color sensor. The six data derived from data acquisition by Arduino and sent to Raspberry Pi via serial communication are used for identification process by neural network. The structure of the neural network can be seen in Figure 11. The neural network has 6 inputs derived from three gas sensors, and the RGB value. There are 3 layers composed of 2 hidden layer and 1 output layer. Both hidden layer consists of 4 neurons and the output layer consists of 2 neurons. The combination of two values of the neuron at the output layer will be the target of the neural network training. The fresh meat, half-rotten meat, and rotten meat are defined by the value of 00, 01, and 11, respectively. The first step in designing a neural network program is the initialization of weight and bias. In the program line, the weight and bias values are determined randomly using a random number generator. The random syntax of the Lazarus program will generate a random number between 0 and 1. To get a random value between -1 and 1 then the random value is multiplied by 2 and then subtracted by 1. For forward propagation process is done by multiplying the inputs of the three gas sensors and the three color characteristics of the color sensor to each weight corresponding to the neuron. Then the bias will be added to each neuron.

Conclusion

Electronic nose system on a single board computer that has been developed showed its success in identifying the tested meats. A semiconductor gas sensor array of TGS2602, MQ137 and MQ138 as olfaction sensor and TCS3200 as RGB vision sensor can identify the odor and colors of different freshness of meat. Baseline method has been used to produce a clear output differences generated by gas sensors reacted to meat freshness level. The identification process with the neural network have been working very well, in which the electronic nose system is able to recognize the pattern of each freshness with success rate of 100% for fresh meat and non-fresh meat (i.e. half rotten and rotten meat). The system may be expected to replace the traditional ways by human and chemical techniques.

CHAPTER 3

PROPOSED SYSTEM

3.1 INTRODUCTION :

We proposed the design and construction of an IoT based Smart Kitchen System are indicated Here different types of sensors were used to detect gas leakages management system in a kitchen.

Its outputs are then interfaced with an Raspberry pi MCU controller programmed in python language.

We can get this and much more safety feature that can be integrated with the automation system includes temperature sensor.

the gas leaking situation Using IOT we can detect in the remote areas.

Continues monitoring of services in kitchen is performed by this system.

3.2 BLOCK DIAGRAM:

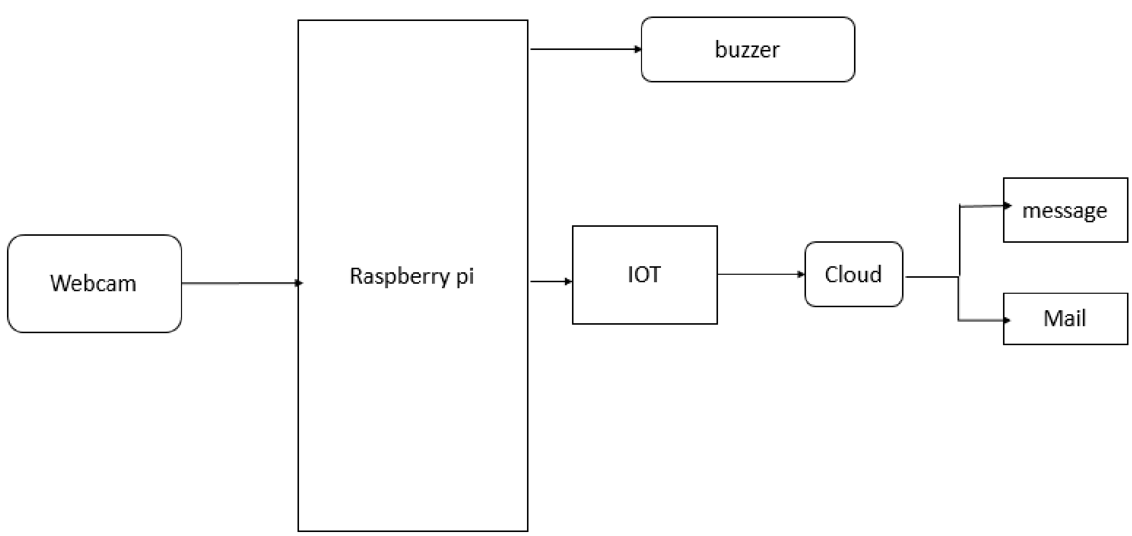


Fig: 3.1 Block diagram

Hardware

- Raspberry Pi
- Web Camera
- Gas sensor
- PIR sensor
- Monitor

Software

- Raspian OS
- Python
- Open CV

PROBLEM DEFINITION :

The main purpose of this study is to forecast the market situation of products with short shelf life. For this purpose, deep learning_x0002_based convolutional neural network models are formed from images collected by digital cameras and labeled by researchers according to aspects of the images. Deep learning model performance is evaluated using F1 score and classification accuracy.

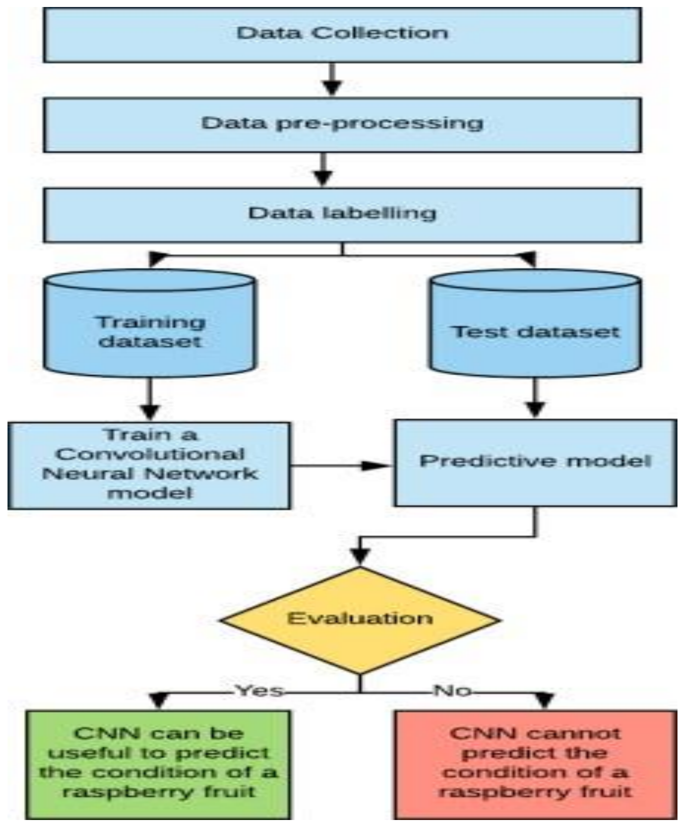


Fig:3.2 Problem definition

CHAPTER 4

HARDWARE DESCRIPTION

4.1 RASPBERRY PI

Raspberry Pi seems to be new in the world and many people really don't know what the Raspberry Pi is. Raspberry Pi can be defined as a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you would expect a desktop computer to do, from browsing the internet and playing high definition video, to making spreadsheets, word-processing, and playing games. It is great bonding with Arduino and can do a lot with Arduino.

4.2 Exploring the Raspberry Pi Board

There are two models of Raspberry Pi, model A and model B. These two are bit similar with few advance features on model B compared to model A. Model B has 512 MB RAM, two USB port where as Model A has 256 MB RAM and just a USB port. Besides, Model B has Ethernet port while Model A does not.

The overview of the Raspberry Pi Model B is shown in Figure

.

Fig:4.1 Raspberry Pi ModelB

Figure : Raspberry Pi Model B(Sanders, 2013)

Different components of the Pi are named in the Figure and brief description on each component is given in following sections.

4.2.1 SD Card Slot

Raspberry Pi doesn't have the real hard drive as in laptop and computer, SD card is taken as solid state drive (SSD) which is used to install operating system and all others software and store everything. This card is needed to insert into the slot for using the Raspberry Pi.

SD card may be 2GB, 4GB or 16GB.

4.2.2 Micro USB Power

The power port is a 5V micro-USB input and supply should be exactly 5v as it doesn't have onboard power regulator. So, power supply shouldn't exceed than 5V.

4.2.3 HDMI Out

This output port is used to connect the Raspberry Pi with a monitor via HDMI (High Definition Multimedia Interface). Hence, any screen or TV can be connected to it which consists of HDMI port.

4.2.4 Ethernet and USB port

Both the Ethernet port and USB port on Model B are supplied via the onboard LAN9512 chip. It is a high-speed USB 2.0 hub with a 10/100 Ethernet controller (Donat, 2014).

USB ports are used to connect the inputs (keyboard, mouse). Almost everything that can connect to computer via USB also can connect with Raspberry Pi.

4.2.5 RCA Video Out and Audio Out

Audio and RCA video jacks are present on the board for audio and video out.

The Raspberry Pi does support sound over its HDMI output, but there is a standard 3.5-mm audio jack to plug in headphones but USM mikes may work or not.

For video, the RCA jack sends video to any connected RCA video device.

4.2.6 GPIO Headers(Pins)

GPIO pins stands for general purpose of input output pins. These pins are used to connect any number of physical extensions with the Raspberry Pi. Raspberry Pi has pre-installed libraries that allow us to access the pins using programming languages like C, C++ or python.

4.2.7 Chips (Broadcom)

The most important component in a Raspberry Pi is chip that is Broadcom which is placed at the middle of the board. The chip consists of ARM11 processor running at 700 MHz and a Videocore4 GPU and can be over clocked to at least 800 MHz without a problem.

4.3 Raspberry Pi 2 Model B

Recently, Raspberry Pi 2 Model B has been lunched recently which Broadcom BCM2836 ARM Cortex-A7 Quad Core Processor has powered Single Board Computer running at

900MHz, 1GB RAM and 4 Quad USB ports. It is the advanced version of Model B and is 6 times faster than Model B Raspberry Pi. In addition, it has combined 4-pole jack for connecting your stereo audio out and composite video out and advanced power management. Figure 4.2 shows the top view of the board with labels of some important components (raspberrypi.org, 2015a).

Fig:4.2 Raspberry Pi 2 Model B

4.4 Hardware Required for Raspberry Pi

Raspberry Pi can't start alone, it needs many others peripherals (hardware). There is brief description of the hardware requirements in the following section (Bates, 2014).

4.4.1 Power Supply

As mentioned already in above theory portion, Raspberry Pi needs 5V power supply. If supply exceeds 5V then it can't guaranteed to work properly. And the power supply also need to supply at least 500 milliamps (mA), and preferably more like 1 amp (A). If the supply is 500 mA or less, it is likely to have the mal-function of keyboard and mouse. It is not good idea to power the Raspberry Pi from USB port of computer and hub as they mostly provide current less than required. Hence, the Raspberry Pi requires a Micro-USB connection which is capable of supplying at least 700 mA (or 0.7 A) at 5V.

4.4.2 Storage

A separate hardware is required for the storage purpose in Raspberry Pi. For this, SD card is used, mostly 4 GB and 8 GB if needed. The operating system and all files are stored in the card. We can buy blank SD card and install operating system or buy a pre-installed one.

4.4.3 Input

External keyboard and mouse are required to provide input to the Raspberry Pi. No any additional software is needed to use the keyboard and mouse.

4.4.4 Monitor

We can use monitor or TV with HDMI port or DVI inputs as the screen for the Raspberry Pi. For DVI inputs, HDMI-to-DVI converters are required which can be finding easily in a market. Monitor is most important for the Raspberry Pi as it is the only way to see what we have done on it.

4.4.5 Network

As in laptop or computer, we can access to internet and network in Raspberry Pi as well. For that, we can use wired Ethernet connection which is easier option or Wi-Fi module to access Wi-Fi in the Raspberry Pi.

A power supply (sometimes known as a power supply unit or PSU) is a device or system that supplies electrical or other types of energy to an output load or group of loads. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

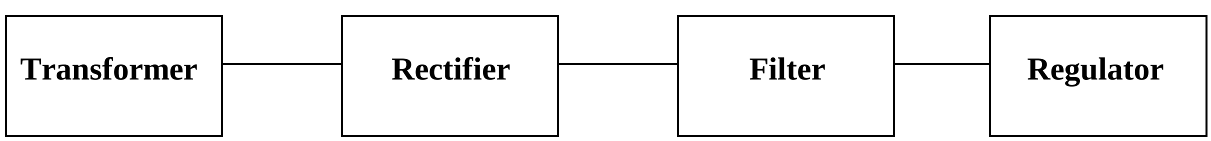


Fig:4.3 Block diagram of power supply

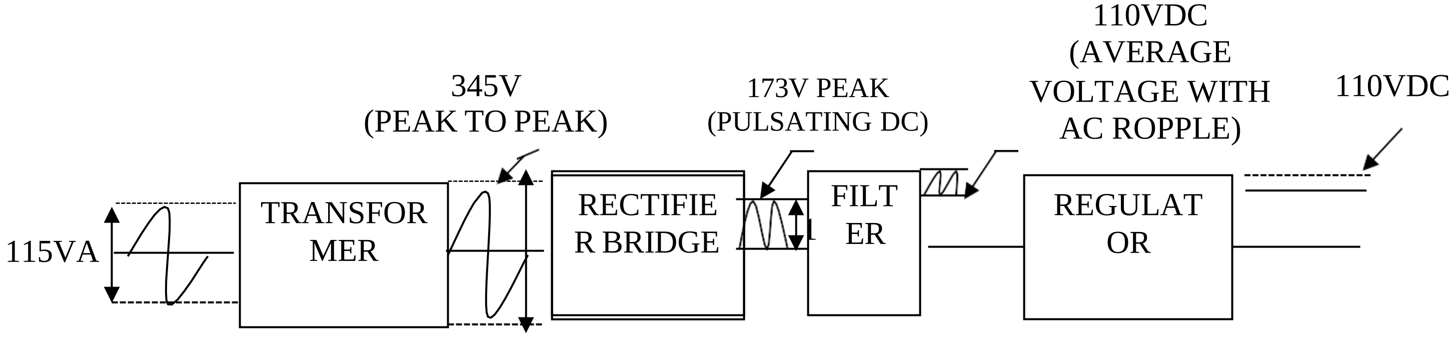


Fig 4.4 Processing of power supply

The transformer steps up or steps down the input line voltage and isolates the power supply from the power line. The RECTIFIER section converts the alternating current input signal to a pulsating direct current. However, as you proceed in this chapter you will learn that pulsating dc is not desirable. For this reason a FILTER section is used to convert pulsating dc to a purer, more desirable form of dc voltage.

The final section, the REGULATOR, does just what the name implies. It maintains the output of the power supply at a constant level in spite of large changes in load current or input line voltages. Now that you know what each section does, let's trace an ac signal through the power supply. At this point you need to see

how this signal is altered within each section of the power supply. Later on in the chapter you will see how these changes take place. An input signal of 115 volts ac is applied to the primary of the transformer.

The transformer is a step-up transformer with a turns ratio of 1:3. You can calculate the output for this transformer by multiplying the input voltage by the ratio of turns in the primary to the ratio of turns in the secondary; therefore, 115 volts ac ´ 3 = 345 volts ac (peak-to- peak) at the output. Because each diode in the rectifier section conducts for 180 degrees of the 360-degree input, the output of the rectifier will be one-half, or approximately 173 volts of pulsating dc. The filter section, a network of resistors, capacitors, or inductors, controls the rise and fall time of the varying signal; consequently, the signal remains at a more constant dc level. You will see the filter process more clearly in the discussion of the actual filter circuits. The output of the filter is a signal of 110 volts dc, with ac ripple riding on the dc. The reason for the lower voltage (average voltage) will be explained later in this chapter. The regulator maintains its output at a constant 110-volt dc level, which is used by the electronic equipment (more commonly called the load).

4.5.1 Simple 5V power supply for digital circuits

- Brief description of operation: Gives out well-regulated +5V output, output current capability of 100 mA
- Circuit protection: Built-in overheating protection shuts down output when regulator IC gets too hot
- Circuit complexity: Very simple and easy to build
- Circuit performance: Very stable +5V output voltage, reliable operation
- Availability of components: Easy to get, uses only very common basic components
- Design testing: Based on datasheet example circuit, I have used this circuit successfully as part of many electronics projects
- Applications: Part of electronics devices, small laboratory power supply
- Power supply voltage: Unregulated DC 8-18V power supply
- Power supply current: Needed output current + 5 mA
- Component costs: Few dollars for the electronics components + the input transformer cost

4.6 WEBCAM:-

A **webcam** is a video camera which feeds its images in real time to a computer or computer network, often via USB, Ethernet or Wi-Fi. Their most popular use is the establishment of video links, permitting computers to act as videophones or videoconference stations. This common use as a video camera for the World Wide Web gave the webcam its name. Other popular uses include security surveillance and computer vision. Webcams are known for their low manufacturing cost and flexibility, making them the lowest cost form of video telephony. They have also become a source of security and privacy issues, as some built-in webcams can be remotely activated via spyware.

4.6.1 LENS

Webcams typically include a lens, an image sensor, support electronics, and may also include a microphone for sound. Various lenses are available, the most common in consumer-grade webcams being a plastic lens that can be screwed in and out to focus the camera. Fixed focus lenses, which have no provision for adjustment, are also available. As a camera system's depth of field is greater for small image formats and is greater for lenses with a large f-number (small aperture), the systems used in webcams have a sufficiently large depth of field that the use of a fixed focus lens does not impact image sharpness to a great extent.

Image sensors can be CMOS or CCD, the former being dominant for low-cost cameras, but CCD cameras do not necessarily outperform CMOS-based cameras in the low cost price range. Most consumer webcams are capable of providing VGA resolution video at a frame rate of 30 frames per second. Many newer devices can produce video in multi-megapixel resolutions, and a few can run at high frame rates such as the PlayStation Eye, which can produce 320×240 video at 120 frames per second.

Support electronics read the image from the sensor and transmit it to the host computer. The camera pictured to the right, for example, uses a Sonix SN9C101 to transmit its image over USB. Typically, each frame is transmitted uncompressed in RGB or YUV or compressed as JPEG. Some cameras, such as mobile phone cameras, use a CMOS sensor with supporting electronics "on die", i.e. the sensor and the support electronics are built on a single silicochip to save space and manufacturing costs.

4.6.2 PORTABILITY:

Most webcams feature built-in microphones to make video calling and videoconferencing more convenient. The USB video device class (UVC) specification allows for interconnectivity of webcams to computers without the need for proprietary device drivers. Microsoft Windows XP SP2, Linux and Mac OS X (since October 2005) have UVC support built in and do not require extra device drivers, although they are often installed to add additional features.

4.6.3 FEATURES (LOGITECH WEBCAM C100):-

- Plug-and-play setup (UVC)
- Video capture: Up to 640 x 480 pixels
- Photos: Up to 1.3 megapixels (software enhanced)
- Frame rate: Up to 30 frames per second (with recommended system)
- Hi-Speed USB 2.0 certified
- Fixed focus
- Universal clip fits notebooks, LCD or CRT monitor

Fig: 4.5 Webcam

4.7 Image Processing on Raspberry Pi

Today image processing are used in various techniques, this paper presents the implementation of image processing operations on Raspberry Pi. The Raspberry Pi is a basic embedded system and being a low cost a single board computer used to reduce the complexity of systems in real time applications. This platform is mainly based on python. Raspberry pi consist of Camera slot Interface (CSI) to interface the raspberry pi camera. Here, the Dark and Low contrast images captured by using the Raspberry Pi camera module are enhanced in order to identify the particular region of image. This concept is used in the real time application of MAV, The MAVs are basically used to capture images and videos through the Raspberry pi camera module. Because of its credit card sized (small) and less weight in the design. However, the image captured by MAVs will consist of unwanted things due to atmospheric conditions; hence it is necessary to remove noise present in the MAVs images.

4.8 INTRODUCTION

The image processing is a form of signal processing where the input is an image, photograph or video frame, the output of an image processing may be either an image or a video frame or a set of characteristics or parameters related to the image. The acquisition of digital image usually suffers from undesirable camera shakes and due to unstable random camera motions. Hence image enhancement algorithms are required to remove these unwanted camera shakes. This image processing concepts are implemented in Raspberry pi in the application of MAV. The Raspberry Pi is a basic embedded system having a credit card-sized single board computers developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SOC) which includes an ARM1176JZF-S Core (ARM V6K)700 MHz CPU processor, Broadcom Video Core IV GPU having 17 pins, 3.5W of power, and 512 MB of RAM memory. The Raspberry Pi system has Secure SD card reader (models A and B) or Micro SD card reader (models A+ and B+) sockets for boot media and persistent storage. The system provides Debian Linux operating system

Raspbian image for download. Python is used as main programming language for raspberry pi. A micro air vehicle (MAV) is a remote-controlled, Unmanned Aircraft Vehicle (UAV) significantly smaller than typical UAVs that have a size restriction. UAV is an aircraft without a human pilot. Its flight is controlled either autonomously on board computers or by the remote control of a pilot on the ground or in another vehicle. By having a Raspberry Pi camera module available on a MAV the efficiency of this air vehicle increases and new fields of applications are available. It is needed in military Operations, in which targets have to be identified. Such identification is often done by a human on ground, to reduce the probability of mistakes. But a Raspberry Pi camera module is also helpful if a MAV shall autonomously fly through an arch.

4.9 BASIC CONCEPT OF IMAGE PROCESSING

In general, any digital image processing algorithm consists of three stages: input, processor and output. In the input stage image is captured by a camera. It sent to a particular system to focus on a pixel of image that's gives, its output as a processed image.

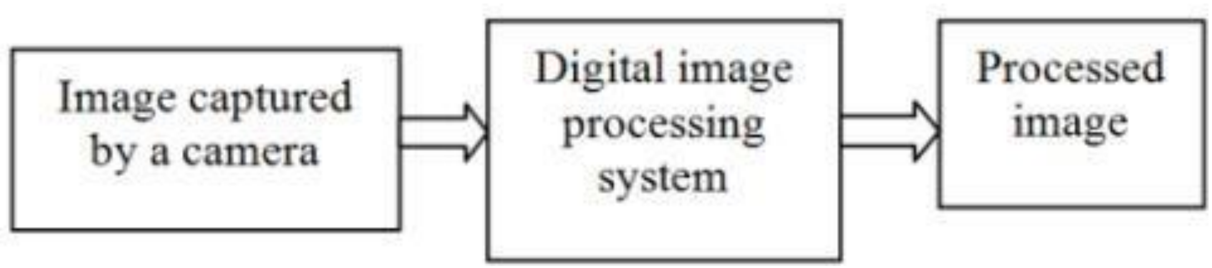


Fig:4.6 General Block diagram of image processing

4.10 SYSTEM HARDWARE DESIGN

The Raspberry Pi board is the central module of the whole embedded image capturing and processing system as given in fig. 2. Its main parts include: main processing chip unit, memory, power supply HDMI Out i.e VGA display, Ethernet port, and USB ports.

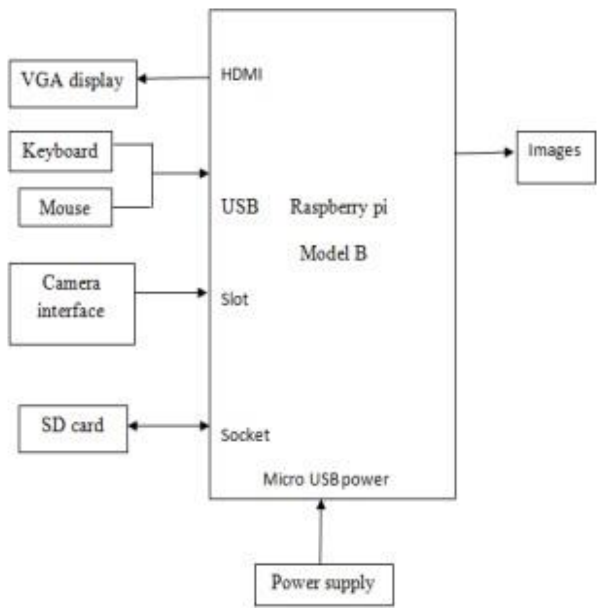


Fig: 4.7 System Block Diagram

Gas sensor

Gas sensor module

In current technology scenario, monitoring of gases produced is very important. From home appliances such as air conditioners to electric chimneys and safety systems at industries monitoring of gases is very crucial. **Gas sensors** are very important part of such systems. Small like a nose, gas sensors spontaneously react to the gas present, thus keeping the system updated about any alterations that occur in the concentration of molecules at gaseous state.

Gas sensors are available in wide specifications depending on the sensitivity levels, type of gas to be sensed, physical dimensions and numerous other factors. This Insight covers a **methane gas sensor** that can sense gases such as ammonia which might get produced from methane. When a gas interacts with this sensor, it is first ionized into its constituents and is then adsorbed by the sensing element. This adsorption creates a potential difference on the element which is conveyed to the processor unit through output pins in form of current. What is this sensing element? Is it kept in some chamber or is kept exposed? How does it get current and how it is taken out? Let’s find out in this Insight!!!

Fig:4.8 Gas sensor module (i)

The **gas sensor module** consists of a steel exoskeleton under which a sensing element is housed. This sensing element is subjected to current through connecting leads. This current is known as heating current through it, the gases coming close to the sensing element get ionized and are absorbed by the sensing element. This changes the resistance of the sensing element which alters the value of the current going out of it.

Fig:4.9 Gas sensor module (ii)

Image shows externals of a standard gas sensor module: a steel mesh, copper clamping ring and connecting leads. The top part is a stainless steel mesh which takes care of the following:

Filtering out the suspended particles so that only gaseous elements are able to pass to insides of the sensor.

- ✓ Protecting the insides of the sensor.
- ✓ Exhibits an anti-explosion network that keeps the sensor module intact at high temperatures and gas pressures.

In order to manage above listed functions efficiently, the steel mesh is made into two layers. The mesh is bound to rest of the body via a copper plated clamping ring.

Fig: 4.10 Gas sensor module (iii)

1)The connecting leads of the sensor are thick so that sensor can be connected firmly to the circuit and sufficient amount of heat gets conducted to the inside part. They are casted from copper and have tin plating over them. Four of the six leads (A, B, C, D) are for signal fetching while two (1, 2) are used to provide sufficient heat to the sensing element.

The pins are placed on a Bakelite base which is a good insulator and provides firm gripping to the connecting leads of the sensor.

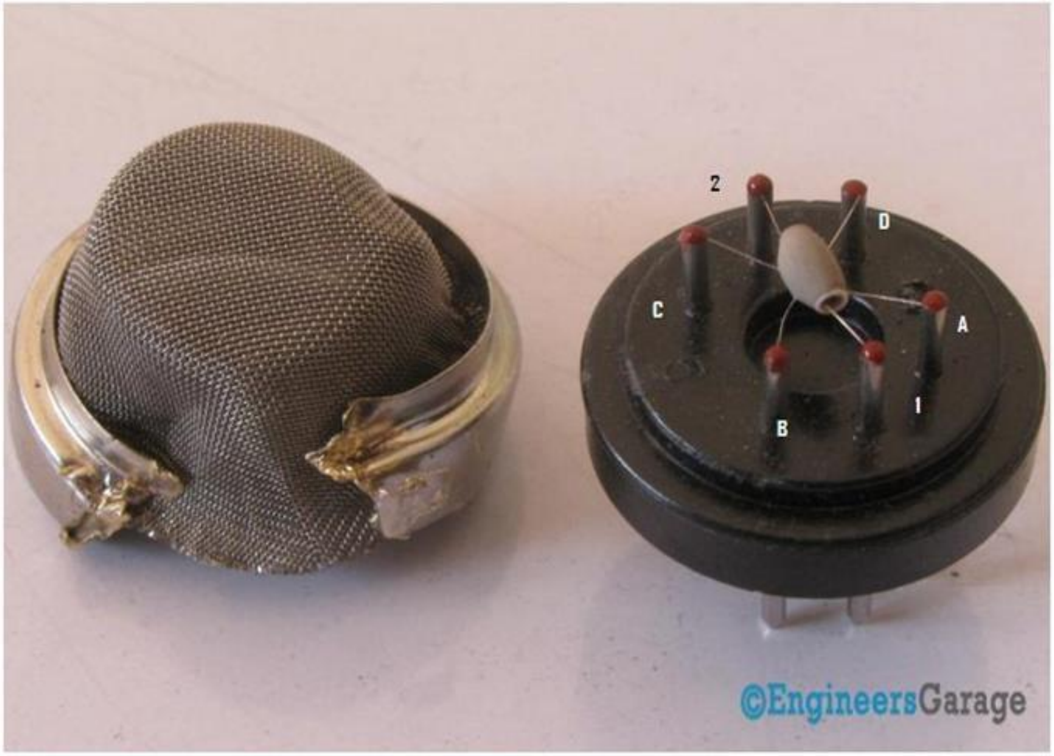


Fig: 4.11 Gas sensor module (iv)

The top of the gas sensor is removed off to see the internals parts of the sensor: sensing element and connection wiring. The hexapod structure is constituted by the sensing element and six connecting legs that extend beyond the Bakelite base.

Fig: 4.12 Aluminium Oxide based ceramic and has a coating of tin oxide

Image shows the hollow sensing element which is made up from Aluminium Oxide based ceramic and has a coating of tin oxide. Using a ceramic substrate increases the heating efficiency and tin oxide, being sensitive towards adsorbing desired gas' components (in this case methane and its products) suffices as sensing coating.

The leads responsible for heating the sensing element are connected through Nickel-Chromium, well known conductive alloy. Leads responsible for output signals are connected using platinum wires which convey small changes in the current that passes through the sensing element. The platinum wires are connected to the body of the sensing element while Nickel-Chromium wires pass through its hollow structure.

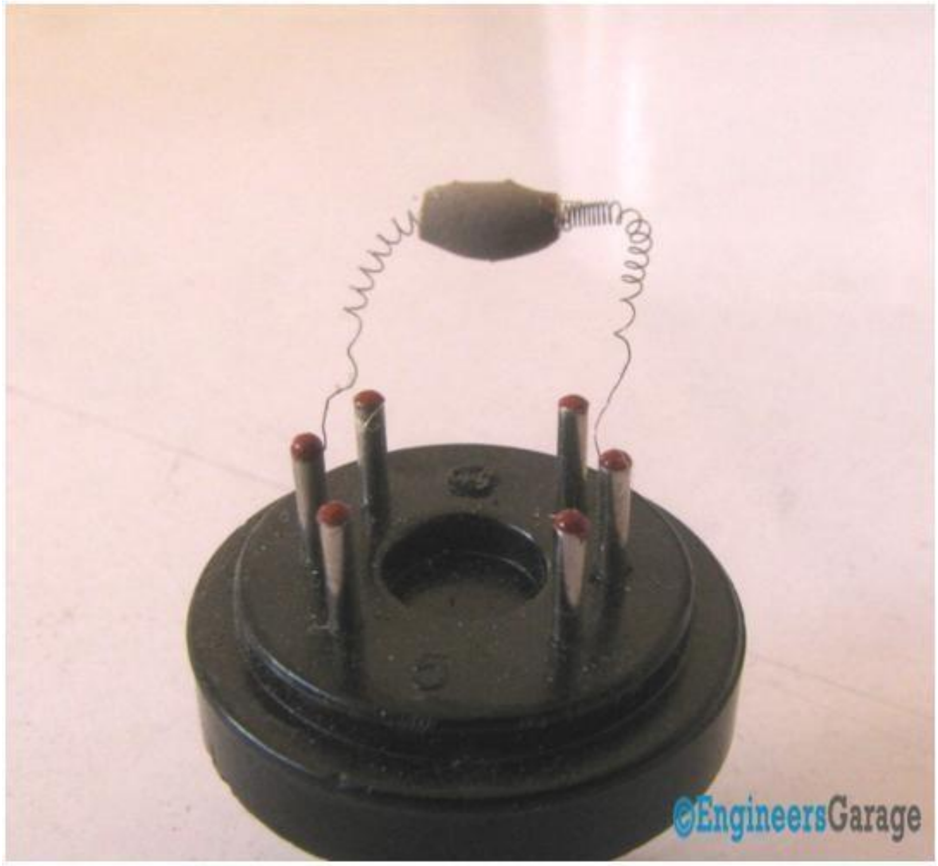


Fig:4.13 Hollow ceramic

While other wires are attached to the outer body of the element, Nickel-Chromium wires are placed inside the element in a spring shaped. Image shows coiled part of the wire which is placed on the inside of the hollow ceramic.

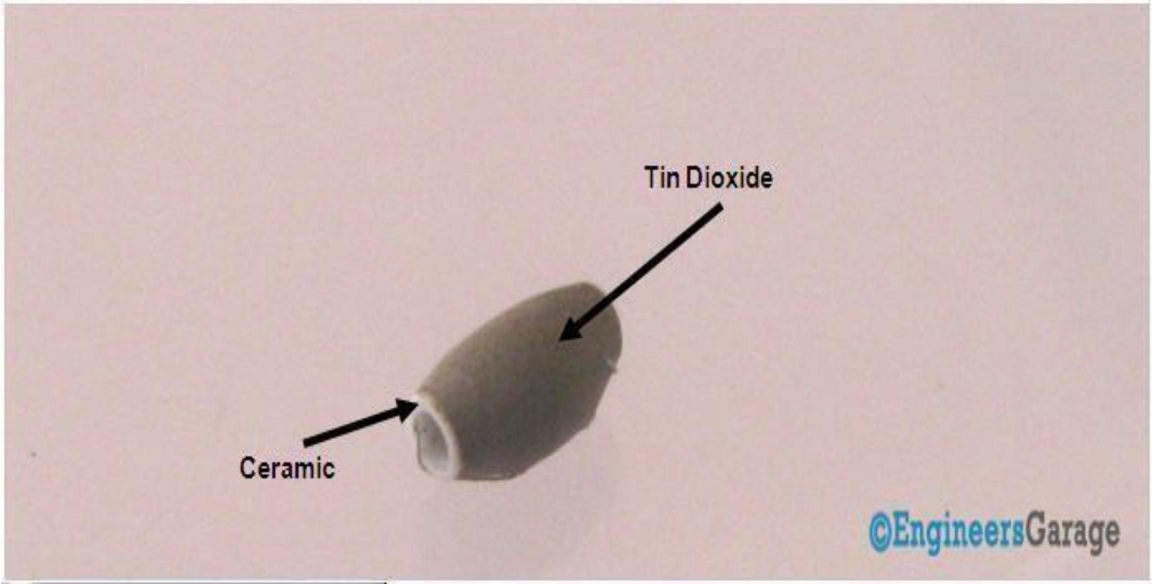


Fig:4.14 Ceramic with tin dioxide

Image shows the ceramic with tin dioxide on the top coating that has good adsorbing property. Any gas to be monitored has specific temperature at which it ionizes. The task of the sensor is to work at the desired temperature so that gas molecules get ionized. Through Nickel-chromium wire, the ceramic region of the sensing element is subjected to heating current. The heat is radiated by the element in the nearby region where gases interact with it and get ionized. Once, ionized, they are absorbed by the tin dioxide. Adsorbed molecules change the resistance of the tin dioxide layer. This changes the current flowing through the sensing element and is conveyed through the output leads to the unit that controls the **working of the gas sensor**.

CHAPTER 5

5. SOFTWARE DESCRIPTION

5.1 Start up with Raspberry Pi

5.1.1 Installation of operating system on Raspberry Pi

Raspberry Pi is a small computer; hence operating system (OS) should be installed. As the Raspberry doesn't have hard drive, OS is installed in the external memory. For that, memory card (SD card) is used for the installation of operating system and all the required software and supporting files are stored in the same SD card.

There are different types of operating system but we preferred to talk about NOOBS (New Out of the Box Software) as it is suitable for the beginners. We can either buy a preinstalled SD card or empty SD card. In pre-installed SD card, NOOBS is already copied and ready to boot. The detail description of installation of operating system in a blank SD card in step wise is given in Appendix B.

5.1.2 Use of laptop screen, keyboard and mouse for Raspberry Pi

If you don't have suitable screen for the Raspberry Pi, we can use the monitor of a laptop. Besides, we can use keyboard and mouse of laptop as well for Raspberry Pi. For that, two software are needed to install in a laptop which screen is going to use for Raspberry Pi. Installation and configuration of the software is discussed briefly in the following sections. Xming

This is the first software need to be installed which can be downloaded from the link, Download_Xming and install it in the laptop. After completion of installation, run the application called 'XLaunch' and verify that the settings are as shown in three following figures.

Finally, click ”Next” button, it goes to last configuration dialogue box where ”Finish” button is needed to click for the completion of the setting. After Completion of configuration, double click the application named as”Xming”.

5.2 PuTTY

This is the primary software need to be installed. It can be downloaded in the provided link as follow, Download Putty. As, it is downloaded, it needs to be installed following some few normal steps of installation. For Configuration, double click the icon of Putty after the completion of installation and enter the IP address of Raspberry Pi as shown in Figure 12-4.

Finally, click on “open” button and as a result; window will be displayed as shown in Figure 5.1.

Fig :5.1 PuTTy login windows

In this window, username and password of the Raspberry Pi should be entered. The default username for a Raspberry Pi is pi. Press enter after entering correct username and password and you will see the windows where you should enter the text ”lesson” as shown in Figure5.1.

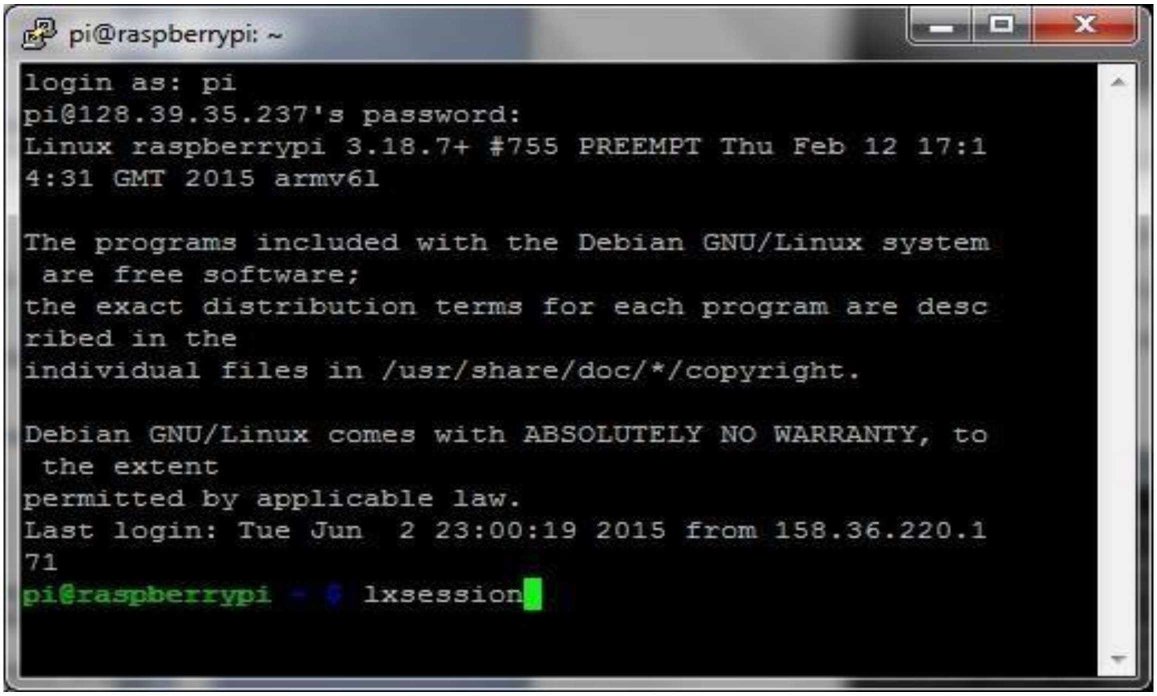


Fig 5.2: PuTTy success windows

Then, you will be able to see the desktop of Raspberry Pi in the laptop as shown in Figure:5.2 and mouse and keyboard can be used for the Raspberry Pi.

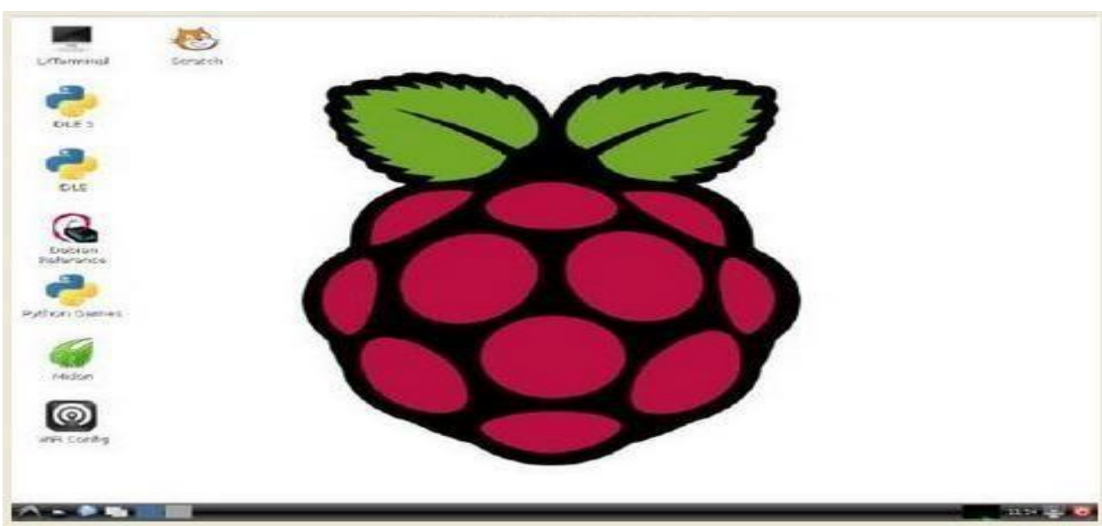


Fig 5.3: Raspberry Pi Desktop on your Laptop Display

5.3 Installation of required applications on Raspberry Pi

There are many applications that are needed to install in the Raspberry Pi for the completion of the thesis. For data logging, MySQL apache5 and php my admin are needed to install whereas for the web-page development, PHP is needed to install. Web page is used for the monitoring and managing purpose

Raspbian OS

Raspberry Pi OS (formerly Raspbian) is a Debian-based operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. The first version of Raspbian was created by Mike Thompson and Peter Green as an independent project The initial released build was completed on July 15, 2012'

Raspberry Pi OS is highly optimized for the Raspberry Pi line of compact single-board computers with ARM CPUs. It runs on every Raspberry Pi except the Pico microcontroller. Raspberry Pi OS uses a modified LXDE as its desktop environment with the Openbox stacking window manager, along with a unique theme. The distribution is shipped with a copy of the algebra program Wolfram Mathematica and a version of Minecraft called Minecraft: Pi Edition (note that Minecraft: Pi Edition is no longer installed as of the Debian bullseye update) as well as a lightweight version of the Chromium web browser.

User Interface

Raspberry Pi OS's desktop environment, PIXEL, looks similar to many common desktops, such as macOS, Microsoft Windows, and is based on LXDE. The menu bar is positioned at the top and contains an application menu and shortcuts to Terminal, Chromium, and File Manager. On the right is a Bluetooth menu, a Wi-Fi menu, volume control, and a digital clock.

Components

PCManFM is a file browser allowing quick access to all areas of the computer, and was redesigned in the first Raspberry Pi OS Buster release (2019-06-20).

Raspberry Pi OS originally used Epiphany as the web browser, but switched to Chromium with the launch of its redesigned desktop.

Raspberry Pi OS comes with many beginner IDEs, such as Thonny Python IDE, Mu Editor, and Greenfoot. It also ships with educational software like Scratch and Bookshelf.

Version Size

The Raspberry Pi documentation recommends at least a 4GB microSD card for Raspberry Pi OS Lite, an 8GB microSD card for Raspberry Pi OS, and a 16GB microSD card for Raspberry Pi OS Full.[19] The image files themselves are 1,875MB, 3,980MB, and 8,603MB respectively.

Python

What is Python?

Python: Dynamic programming language which supports several different programing paradigms:

- Procedural programming
- Object oriented programming
- Functional programming

Standard: Python byte code is executed in the Python interpreter (similar to Java) → platform independent code

Why Python?

Extremely versatile language Website development, data analysis, server maintenance, numerical analysis,

- Syntax is clear, easy to read and learn (almost pseudo code)

- Common language
- Intuitive object oriented programming
- Full modularity, hierarchical packages
- Comprehensive standard library for many tasks
- Big community Simply extendable via C/C++, wrapping of C/C++ libraries
- Focus: Programming speed

History

Start implementation in December 1989 by Guido van Rossum (CWI)

- 16.10.2000: Python 2.0
- Unicode support
- Garbage collector
- Development process more community oriented
- 3.12.2008: Python 3.0
- Not 100% backwards compatible 2007 & 2010 most popular programming language (TIOBE Index)
- Recommendation for scientific programming (Nature News, NPG, 2015)
- Current version: Python 3.9.2
- Python2 is out of support

Is Python fast enough?

- For user programs: Python is fast enough!
- Most parts of Python are written in C
- For compute intensive algorithms: Fortran, C, C++ might be better
- Performance-critical parts can be re-implemented in C/C++ if necessary
- First analyses, then optimize!

Conclusion

Python is a fully-fledged programming language. Any code you download from github, the Python Package Index, or anywhere else, can be malicious and would almost certainly go unnoticed by most people's firewalls.

OpenCV

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. Let’s start the chapter by defining the term "Computer Vision"

Computer Vision

Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields:

- Image Processing: It focuses on image manipulation.
- Pattern Recognition: It explains various techniques to classify patterns.
- Photogrammetry: It is concerned with obtaining accurate measurements from images.

Computer Vision Vs Image Processing

Image processing deals with image-to-image transformation. The input and output of image processing are both images.

Computer vision is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

Applications of Computer Vision

Here we have listed down some of major domains where Computer Vision is heavily used.

Robotics Application

• Localization – Determine robot location automatically

• Navigation

• Obstacles avoidance

• Assembly (peg-in-hole, welding, painting)

• Manipulation (e.g. PUMA robot manipulator)

• Human Robot Interaction (HRI): Intelligent robotics to interact with and serve people

Medicine Application

Classification and detection (e.g. lesion or cells classification and tumor detection)

• 2D/3D segmentation

• 3D human organ reconstruction (MRI or ultrasound)

• Vision-guided robotics surgery

Industrial Automation Application

• Industrial inspection (defect detection)

• Assembly

• Barcode and package label reading

• Object sorting

• Document understanding (e.g. OCR)

Security Application

• Biometrics (iris, finger print, face recognition)

- Surveillance ─ Detecting certain suspicious activities or behaviors Transportation

Application

- Autonomous vehicle
- Safety, e.g., driver vigilance monitoring

Features of OpenCV Library

Using OpenCV library, you can –

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it. OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.

This tutorial explains the concepts of OpenCV with examples using Java bindings.

OpenCV Library Modules

Following are the main library modules of the OpenCV library.

Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array Mat, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name org.opencv.core.

Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.imgproc.

Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name org.opencv.video.

Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name org.opencv.videoio.

calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name org.opencv.calib3d.

features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name org.opencv.features2d.

Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.objdetect.

Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, org.opencv.imgcodecs and org.opencv.videoio.

OpenCV — Storing Images

To capture an image, we use devices like cameras and scanners. These devices record numerical values of the image (Ex: pixel values). OpenCV is a library which processes the digital images, therefore we need to store these images for processing. The Mat class of OpenCV library is used to store the values of an image. It represents an n-dimensional array and is used to store image data of grayscale or color images, voxel volumes, vector fields, point clouds, tensors, histograms, etc. This class comprises of two data parts: the header and a pointer

- Header: Contains information like size, method used for storing, and the address of the matrix (constant in size).
- Pointer: Stores the pixel values of the image (Keeps on varying).

The Mat Class

The OpenCV Java library provides this class with the same name (Mat) within the package org.opencv.core.

Constructors

The Mat class of OpenCV Java library has various constructors, using which you can construct the Mat object.

1. This is the default constructor with no parameters in most cases. We use this to constructor to create an empty matrix and pass this to other OpenCV methods. 2. Mat(int rows, int cols, int type) This constructor accepts three parameters of integer type representing the number of rows and columns in a 2D array and the type of the array (that is to be used to store data). 3. Mat(int rows, int cols, int type, Scalar s) Including the parameters of the previous one, this constructor additionally accepts an object of the class Scalar as parameter. 4. Mat(Size size, int type) 3. OpenCV — Storing Images

CHAPTER 6

RESULTS AND CONCLUSION

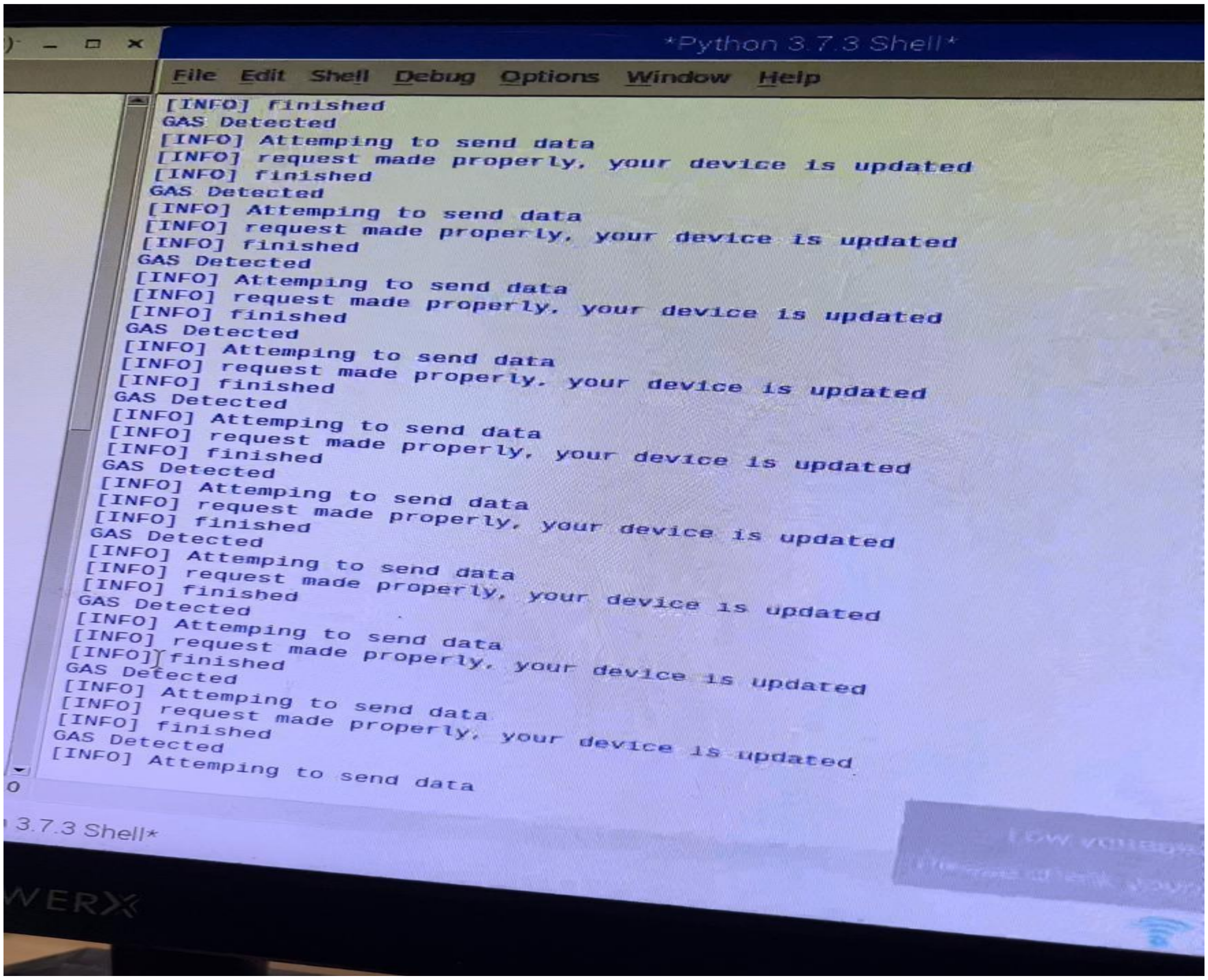


Fig :6.1 spoilage detection output (i)

Fig :6.2 spoilage detection output (ii)

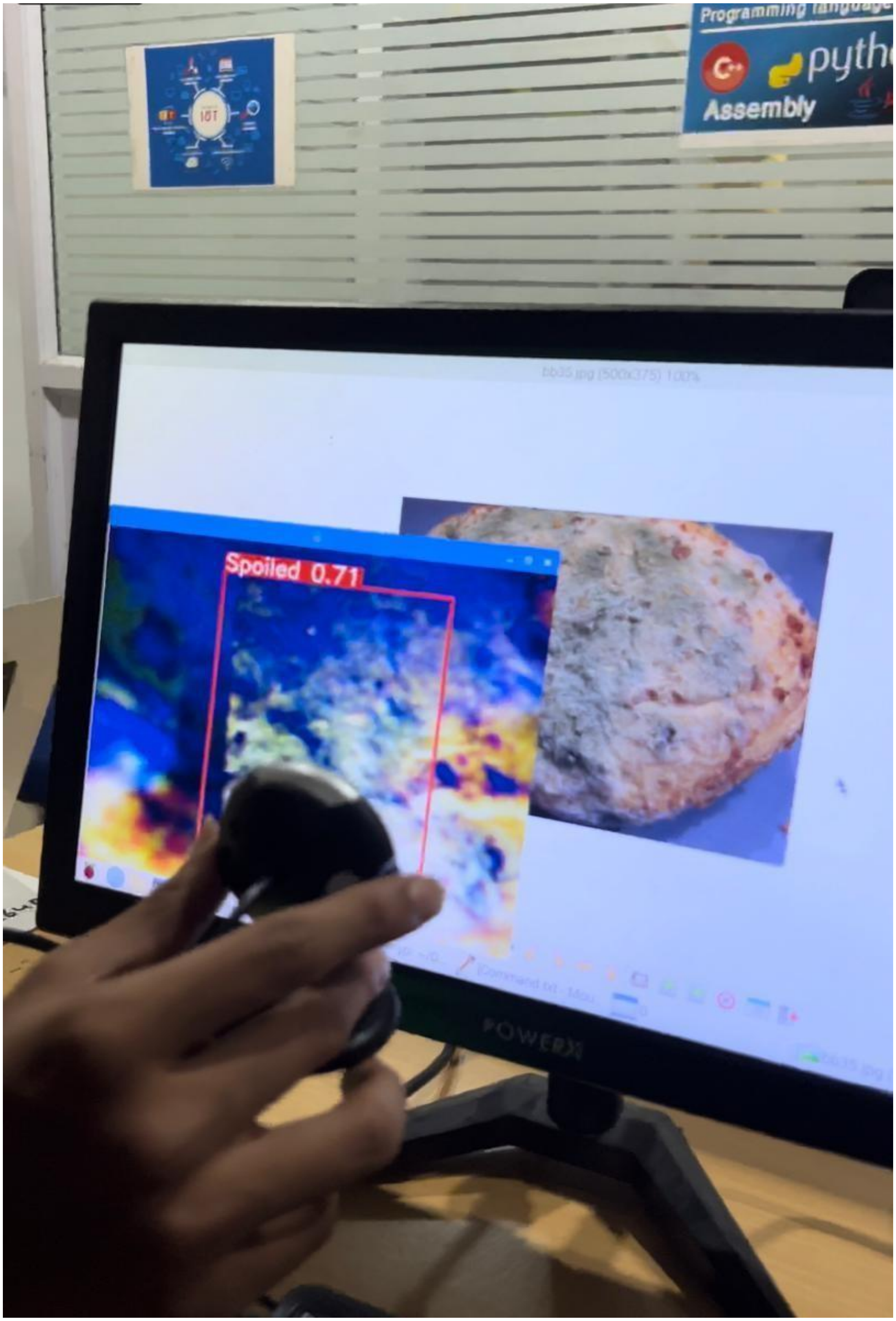


Fig: 6.3 spoilage detection output (iii)

6.2 CONCLUSION

In this project, the identification of normal and defective foods based on quality using OPENCV/PYTHON is successfully done with accuracy. The use of image processing for identifying the quality can be applied not only to any particular fruit. We can also apply this method to identify quality of vegetables with more accuracy. Thus, this will enable the technology to be applied in many products. To replace manual inspection of food, computer vision system is used which provide authentic, equitable and non-destructive rating. The image processing is done by software OpenCv using a language python. The software is divided into two parts first one is for image analysis and other is for controlling hardware based on image processing results. The system is operated in two different scenarios in first the image is captured with camera the all the image processing is done in the control module. All the process are shown on monitor and then based on decision taken by control module. The conveyor assembly is operated.

REFERENCES

[1] X. Chen and A. L. Yuille, “Detecting and reading text in natural scenes,” in Proc. Comput. Vision Pattern Rec-ognit., 2004, vol. 2, pp. II-366–II-373.

[2] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, “Text extraction and document image seg-mentation using matched wavelets and MRF model,” IEEE Trans Image Process., vol. 16, no. 8, pp. 2117– 2128,Aug. 2007.

[3] K. Kim, K. Jung, and J. Kim, “Texture-based ap-proach for text detection in images using support vec-tor machines and continuously adaptive mean shift al-gorithm,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12,pp. 1631–1639, Dec. 2003.

[4] N. Giudice and G. Legge, “Blind navigation and the role of technology,”in The Engineering Handbook of Smart Technology for Aging, Disability,and Indepen-dence, A. A. Helal, M. Mokhtari, and B. Abdulrazak, Eds. Hoboken, NJ, USA: Wiley, 2008.

[5] World Health Organization. (2009). 10 facts about blindness and visual impairment.

[6] Advance Data Reports from the National Health In-terview Survey (2008).

[7] International Workshop on Camera-Based Docu-ment Analysis and Recognition (CBDAR 2005, 2007, 2009, 2011).

[8] X. Chen, J. Yang, J. Zhang, and A. Waibel, “Auto-matic detection and recognition of signs from natural scenes,” IEEE Trans. Image Process.,vol. 13, no. 1, pp. 87–99, Jan. 2004.

[9] D. Dakopoulos and N. G. Bourbakis, “Wearable ob-stacle avoidance electronic travel aids for blind: A sur-vey,” IEEE Trans. Syst., Man, Cybern.,vol. 40, no. 1, pp. 25–35, Jan. 2010.

[10] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in Proc. Comput. Vision Pattern Recognit.,2010, pp. 2963–2970.

[11] Y. Freund and R. Schapire, “Experiments with a new boosting algorithm,”in Proc. Int. Conf. Machine Learning, 1996, pp. 148–156.

[13] An overview of the Tesseract OCR (optical charac-ter recognition) engine, and its possible enhancement for use in Wales in a pre-competitive research stage Prepared by the Language Technologies Unit (Canol-fan Bedwyr), Bangor University April 2008.

[14] A. Shahab, F. Shafait, and A. Dengel, “ICDAR 2011 robust reading competition:ICDAR Robust Reading Competition Challenge 2: Reading text in scene imag-es,” in Proc. Int. Conf. Document Anal. Recognit., 2011, pp. 1491–1496.

[15] KReader Mobile User Guide, knfb Reading Technol-ogy Inc. (2008).

[16] S. M. Lucas, “ICDAR 2005 text locating competi-tion results,” in Proc.Int. Conf. Document Anal. Recog-nit., 2005, vol. 1, pp. 80–84.

APPENDIX

A. SOURCE CODE

GAS MOTOR :

```
import sys
import numpy as np
import glob
import time
import RPi.GPIO as GPIO
import os
import board
import math
import requests

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO_GAS = 26
GPIO_Relay = 5
motor = 0
gas = 0

GPIO.setup(GPIO_GAS, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(GPIO_Relay, GPIO.OUT)
GPIO.output(GPIO_Relay, True)

TOKEN = "BBFF-cGbcWz965p1L6HJuQOCmCXGZVJM5tC"
DEVICE_LABEL = "smartkitchen"
VARIABLE_LABEL_1 = "gas"
VARIABLE_LABEL_2 = "motor"
```

```
def build_payload(variable_1, variable_2,gas, motor):

    payload = {variable_1: gas,
               variable_2: motor}

    return payload

def post_request(payload):
    # Creates the headers for the HTTP requests
    url = "http://industrial.api.ubidots.com"
    url = "{}/api/v1.6/devices/{}".format(url, DEVICE_LABEL)
    headers = {"X-Auth-Token": TOKEN, "Content-Type": "application/json"}

    # Makes the HTTP requests
    status = 400
    attempts = 0
    while status >= 400 and attempts <= 5:
        req = requests.post(url=url, headers=headers, json=payload)
        status = req.status_code
        attempts += 1
        time.sleep(1)

    # Processes results
    if status >= 400:
        print("[ERROR] Could not send data after 5 attempts, please check \
            your token credentials and internet connection")
        return False

    print("[INFO] request made properly, your device is updated")
    return True
```

```
if__name__== '__main__':
    print ("Processing...")
    while True:
        try:
            if GPIO.input(GPIO_GAS) == GPIO.LOW:
                print('GAS Detected')
                GPIO.output(GPIO_Relay, False)
                motor = 1
                gas = 1
            else:
                print('GAS Not Detected')
                GPIO.output(GPIO_Relay, True)
                motor = 0
                gas = 0
            time.sleep(0.5)
            payload = build_payload(VARIABLE_LABEL_1,VARIABLE_LABEL_2,gas,
motor)
            print("[INFO] Attemping to send data")
            post_request(payload)
            print("[INFO] finished")

        except KeyboardInterrupt:
            cv2.destroyAllWindows()
            cap.release()
            sys.exit()
```

Detect.py :

```
        # YOLOv5 🚀 by Ultralytics, GPL-3.0 license
"""
```

Run YOLOv5 detection inference on images, videos, directories, globs, YouTube, webcam, streams, etc.

Usage - sources:
    $ python detect.py --weights yolov5s.pt --source 0   # webcam
                                                    img.jpg   # image
                                                    vid.mp4   # video
                                                    screen   # screenshot
                                                    path/   # directory
                                                    list.txt   # list of images
                                                    list.streams   # list of streams
                                                    'path/*.jpg'   # glob
                                                    'https://youtu.be/Zgi9g1ksQHc' # YouTube
                                                    'rtsp://example.com/media.mp4'   # RTSP, RTMP,
HTTP stream

Usage - formats:
    $ python detect.py --weights yolov5s.pt   # PyTorch
                                 yolov5s.torchscript   # TorchScript
                                 yolov5s.onnx   # ONNX Runtime or OpenCV DNN with --
dnn
                                 yolov5s_openvino_model   # OpenVINO
                                 yolov5s.engine   # TensorRT
                                 yolov5s.mlmodel   # CoreML (macOS-only)
                                 yolov5s_saved_model   # TensorFlow SavedModel
                                 yolov5s.pb   # TensorFlow GraphDef
                                 yolov5s.tflite   # TensorFlow Lite
                                 yolov5s_edgetpu.tflite   # TensorFlow Edge TPU
                                 yolov5s_paddle_model   # PaddlePaddle
"""
import argparse
import os
import platform

```
import sys
from pathlib import Path

import torch

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

from models.common import DetectMultiBackend
from utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages,
LoadScreenshots, LoadStreams
from utils.general import (LOGGER, Profile, check_file, check_img_size,
check_imshow, check_requirements, colorstr, cv2,
                           increment_path, non_max_suppression, print_args, scale_boxes,
strip_optimizer, xyxy2xywh)
from utils.plots import Annotator, colors, save_one_box
from utils.torch_utils import select_device, smart_inference_mode


@smart_inference_mode()
def run(
        weights=ROOT / 'yolov5s.pt', # model path or triton URL
        source=ROOT / 'data/images',  # file/dir/URL/glob/screen/0(webcam)
        data=ROOT / 'data/coco128.yaml',  # dataset.yaml path
        imgsz=(240, 240), # inference size (height, width)
        conf_thres=0.25, # confidence threshold
        iou_thres=0.45, # NMS IOU threshold
        max_det=1000,  # maximum detections per image
```

```
        device='',  # cuda device, i.e. 0 or 0,1,2,3 or cpu
        view_img=False, # show results
        save_txt=False,  # save results to *.txt
        save_conf=False, # save confidences in --save-txt labels
        save_crop=False, # save cropped prediction boxes
        nosave=False, # do not save images/videos
        classes=None, # filter by class: --class 0, or --class 0 2 3
        agnostic_nms=False, # class-agnostic NMS
        augment=False, #  augmented  inference
        visualize=False,  # visualize features
        update=False,  # update all models
        project=ROOT / 'runs/detect',  # save results to project/name
        name='exp',  # save results to project/name
        exist_ok=False,  # existing project/name ok, do not increment
        line_thickness=3, # bounding box thickness (pixels)
        hide_labels=False, # hide labels
        hide_conf=False,  # hide confidences
        half=False,  # use FP16 half-precision inference
        dnn=False,  # use OpenCV DNN for ONNX inference
        vid_stride=1,  # video frame-rate stride
):
    source = str(source)
    save_img = not nosave and not source.endswith('.txt')  # save inference images
    is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
    is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://', 'https://'))
    webcam =  source.isnumeric()  or  source.endswith('.streams')  or  (is_url  and  not
is_file)
    screenshot = source.lower().startswith('screen')
    if is_url and is_file:
        source = check_file(source)  # download
```

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

    # Load model
    device = select_device(device)
    model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data,
fp16=half)
    stride, names, pt = model.stride, model.names, model.pt
    imgsz = check_img_size(imgsz, s=stride)  # check image size

    # Dataloader
    bs = 1  # batch_size
    if webcam:
        view_img = check_imshow(warn=True)
        dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt,
vid_stride=vid_stride)
        bs = len(dataset)
    elif screenshot:
        dataset = LoadScreenshots(source, img_size=imgsz, stride=stride, auto=pt)
    else:
        dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt,
vid_stride=vid_stride)
    vid_path, vid_writer = [None] * bs, [None] * bs

    # Run inference
    model.warmup(imgsz=(1 if pt or model.triton else bs, 3, *imgsz)) # warmup
    seen, windows, dt = 0, [], (Profile(), Profile(), Profile())
    for path, im, im0s, vid_cap, s in dataset:
```

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

    # Inference
    with dt[1]:
        visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if visualize
else False
        pred = model(im, augment=augment, visualize=visualize)

    # NMS
    with dt[2]:
        pred = non_max_suppression(pred, conf_thres, iou_thres, classes,
agnostic_nms, max_det=max_det)

    # Second-stage classifier (optional)
    # pred = utils.general.apply_classifier(pred, classifier_model, im, im0s)

    # Process predictions
    for i, det in enumerate(pred): # per image
        seen += 1
        if webcam:  # batch_size >= 1
            p, im0, frame = path[i], im0s[i].copy(), dataset.count
            s += f'{i}: '
        else:
            p, im0, frame = path, im0s.copy(), getattr(dataset, 'frame', 0)

        p = Path(p)  # to Path
```

```
            save_path = str(save_dir / p.name)  # im.jpg
            txt_path = str(save_dir / 'labels' / p.stem) + ('' if dataset.mode == 'image' else
f'_{frame}')  # im.txt
            s += '%gx%g ' % im.shape[2:]  # print string
            gn = torch.tensor(im0.shape)[[1, 0, 1, 0]]  # normalization gain whwh
            imc = im0.copy() if save_crop else im0 # for save_crop
            annotator = Annotator(im0, line_width=line_thickness, example=str(names))
            if len(det):
                # Rescale boxes from img_size to im0 size
                det[:, :4] = scale_boxes(im.shape[2:], det[:, :4], im0.shape).round()

                # Print results
                for c in det[:, 5].unique():
                    n = (det[:, 5] == c).sum()  # detections per class
                    s += f"{n} {names[int(c)]}{'s' * (n > 1)}, "  # add to string

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

                    if save_img or save_crop or view_img:  # Add bbox to image
                        c = int(cls)  # integer class
                        label = None if hide_labels else (names[c] if hide_conf else f'{names[c]}
{conf:.2f}')
                        annotator.box_label(xyxy, label, color=colors(c, True))
                    if save_crop:
```

```
                    save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] /
f'{p.stem}.jpg', BGR=True)

            # Stream results
            im0 = annotator.result()
            if view_img:
                if platform.system() == 'Linux' and p not in windows:
                    windows.append(p)
                    cv2.namedWindow(str(p),          cv2.WINDOW_NORMAL          |
cv2.WINDOW_KEEPRATIO)  # allow window resize (Linux)
                    cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])
                cv2.imshow(str(p), im0)
                cv2.waitKey(1)  # 1 millisecond

            # Save results (image with detections)
            '''if save_img:
                if dataset.mode == 'image':
                    cv2.imwrite(save_path, im0)
                else:  # 'video' or 'stream'
                    if vid_path[i] != save_path:  # new video
                        vid_path[i] = save_path
                        if isinstance(vid_writer[i], cv2.VideoWriter):
                            vid_writer[i].release()  # release previous video writer
                        if vid_cap:  # video
                            fps = vid_cap.get(cv2.CAP_PROP_FPS)
                            w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
                            h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
                        else:  # stream
                            fps, w, h = 30, im0.shape[1], im0.shape[0]
                        save_path = str(Path(save_path).with_suffix('.mp4'))  # force *.mp4
suffix on results videos
```

```
                vid_writer[i] = cv2.VideoWriter(save_path, cv2.VideoWriter_fourcc(*'mp4v'), fps, (w, h))
            vid_writer[i].write(im0)'''

        # Print time (inference-only)
        LOGGER.info(f"{s}{'' if len(det) else '(no detections), '}{dt[1].dt * 1E3:.1f}ms")

    # Print results
    t = tuple(x.t / seen * 1E3 for x in dt)  # speeds per image
    LOGGER.info(f'Speed: %.1fms pre-process, %.1fms inference, %.1fms NMS per image at shape {(1, 3, *imgsz)}' % t)
    if save_txt or save_img:
        s = f"\n{len(list(save_dir.glob('labels/*.txt')))} labels saved to {save_dir / 'labels'}" if save_txt else ''
        LOGGER.info(f"Results saved to {colorstr('bold', save_dir)}{s}")
    if update:
        strip_optimizer(weights[0])  # update model (to fix SourceChangeWarning)


def parse_opt():
    parser = argparse.ArgumentParser()
    parser.add_argument('--weights', nargs='+', type=str, default=ROOT / 'yolov5s.pt', help='model path or triton URL')
    parser.add_argument('--source', type=str, default=ROOT / 'data/images', help='file/dir/URL/glob/screen/0(webcam)')
    parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='(optional) dataset.yaml path')
    parser.add_argument('--imgsz', '--img', '--img-size', nargs='+', type=int, default=[640], help='inference size h,w')
    parser.add_argument('--conf-thres', type=float, default=0.25, help='confidence threshold')
```

parser.add_argument('--iou-thres', type=float, default=0.45, help='NMS IoU threshold')

parser.add_argument('--max-det', type=int, default=1000, help='maximum detections per image')

parser.add_argument('--device', default='', help='cuda device, i.e. 0 or 0,1,2,3 or cpu')

parser.add_argument('--view-img', action='store_true', help='show results')

parser.add_argument('--save-txt', action='store_true', help='save results to *.txt')

parser.add_argument('--save-conf', action='store_true', help='save confidences in --save-txt labels')

parser.add_argument('--save-crop', action='store_true', help='save cropped prediction boxes')

parser.add_argument('--nosave', action='store_true', help='do not save images/videos')

parser.add_argument('--classes', nargs='+', type=int, help='filter by class: --classes 0, or --classes 0 2 3')

parser.add_argument('--agnostic-nms', action='store_true', help='class-agnostic NMS')

parser.add_argument('--augment', action='store_true', help='augmented inference')

parser.add_argument('--visualize', action='store_true', help='visualize features')

parser.add_argument('--update', action='store_true', help='update all models')

parser.add_argument('--project', default=ROOT / 'runs/detect', help='save results to project/name')

parser.add_argument('--name', default='exp', help='save results to project/name')

parser.add_argument('--exist-ok', action='store_true', help='existing project/name ok, do not increment')

parser.add_argument('--line-thickness', default=3, type=int, help='bounding box thickness (pixels)')

parser.add_argument('--hide-labels', default=False, action='store_true', help='hide labels')

```
    parser.add_argument('--hide-conf', default=False, action='store_true', help='hide confidences')
    parser.add_argument('--half', action='store_true', help='use FP16 half-precision inference')
    parser.add_argument('--dnn', action='store_true', help='use OpenCV DNN for ONNX inference')
    parser.add_argument('--vid-stride', type=int, default=1, help='video frame-rate stride')
    opt = parser.parse_args()
    opt.imgsz *= 2 if len(opt.imgsz) == 1 else 1  # expand
    print_args(vars(opt))
    return opt


def main(opt):
    check_requirements(exclude=('tensorboard', 'thop'))
    run(**vars(opt))
if __name__ == "__main__":
    opt = parse_opt()
    main(opt)
```

B. RESEARCH PAPER

Smart Food Spoil Detection and Kitchen Using Open CV and IOT Technology

Shyam Sundar S
Student
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
shyamthenu@gmail.com

Srinivaas S
Student
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
srinivaas341@gmail.com

Dharani V
Assistant Professor
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, India.
vdharani.cse@sathyabama.ac.in

ABSTRACT: - This objective to focus on the exclusive components of the IoT and its position within the consciousness of shrewd kitchens and meals waste. Apart from this, there has been a unique coverage of Smart Kitchen. Descriptions of various appliances and their applications inside the clever kitchen are indexed. In latest days, kitchen injuries have accelerated in business kitchens and in domestic kitchens. These accidents can be averted via IoT technology consisting of tracking of the whole kitchen from far flung places. To carry out this studies, it is crucial to hate and mourn. In hardware aspect, gasoline sensor, flame sensor, shipment mobile, exhaust fan are used. On the software aspect, cloud packages and incorporated messaging are used. All these sensors will be included into the Raspberry pi board for statistics transfer to the cloud. Additionally, the message logs may be generated the use of the same methods.

Keywords: Deep Learning, IOT, Image segmentation

INTRODUCTION

Food waste, from the dew to the prey, is one of the largest worldwide issues, the amount of meals misplaced and wasted in line with yrs is 40-50% for roots, culmination and veggies. This article proposes an IoT and system mastering-primarily based approach to meals pleasant tracking. The need for this form of gadget is to correctly classify veggies and culmination into three classes, clean, semisparkling and spoiled, in addition to being established in environmental conditions, whether they favor or now not using an professional algorithm advanced below a particular environmental information fashionable. It is a completely unique machine that uses the IoT edge and its software program is programmable so that it may be used to save any type of food virtually via filling in some files in the settings segment of the system.' Graphic interface.

OBJECTIVE

This article proposes an IoT and system gaining knowledge of-based totally approach to meals nice tracking. To monitor the kitchen from LPG Gas is detected via IOT Observing robust odors in the home environment, with precise interest to food deliver. Dedicated to the question of digesting food via first-rate in keeping with color.

SCOPE

This report proposes an IoT and system mastering technique to meals satisfactory tracking. To screen the cooking of LPG gasoline the usage of IoT To display odors within the clever domestic surroundings, with unique attention to food donation. Dedicated to the hassle of digesting meals by means of high-quality in keeping with colour.

DESIGN AND METHODOLOGY

This chapter describes the experimental approach pursued in this study to answer the research question. Experiments are performed in stages, starting with problem understanding, data collection, data cleaning, modeling, and subsequent evaluation. A digital camera is used to collect data in the form of images. The Python programming language and Tensorflow library are used to create the model. The main purpose of this study is to build a model that predicts the clinical status of images collected from raspberry baskets over time. Illustrated outline of the experiment

LITERATURE SURVEY

Post-harvest processing of fruits and greens takes place in several degrees: washing, sorting, grading, packaging, storage and transport. Automatic Fruit Quality Inspection System (2016) - Spot detection technique is used for corrupt detection, colour based segmentation and size detection approach also are used to identify corrupt components in threshold and binary photographs. It will be exactly 90.Three%. Fruit quality management system based on image processing (2017) Fruit first-class analysis based on color, form and size need to be performed via non-negative strategies as this is a delicate cloth. It is likewise useful for organizing, packaging, transportation and marketing activities.

MATERIAL

- Opencv
- Python
- ImageProcessing
- Web camera

DISCUSSION

Texture, color and length are important parameters for figuring out excellent merchandise. Computer imaginative and prescient and photo processing strategies have validated more and more beneficial inside the product industry, especially for satisfactory-touchy programs. Research on this region suggests the opportunity to apply pc imaginative and prescient systems to improve paintings fine. The use of laptop imaginative and prescient for product inspection has grown in latest years. The market is continuously worrying better great merchandise and that is why incremental improvements were made to improve computerized inspection structures. A laptop software for the rural and meals industries has been implemented to the sphere inspection of clean produce. Whether the fruit is right or horrific relies upon on the nice of the fruit. The export of fresh produce from India is growing every day. People are very fitness conscious; they choose handiest fresh, famous fruit. Texture, shade and size are crucial parameters for figuring out first-rate merchandise. Color popularity is a completely crucial process in coming across adulthood. Maturity exposure is an external satisfactory thing. Yes, but CNN itself. Due to the dearth of texture, the fruit may be identified. Texture analysis exhibits nonuniformity of the fruit's outer floor. Size is likewise an important parameter. The parameter honestly noticed all the clients selected in step with their size. The traits required in this challenge are shade, texture and size. To gain correct capabilities, preprocessing is executed on the acquired photograph. The predominant motive of photograph processing is photograph enhancement to take away unwanted distortions and enhance picture functions which are essential for in addition processing. The fundamental preprocessing steps are to first convert the RGB picture to a grayscale photograph. Then the My photograph equation is carried out to the gray image. This facilitates to adjust photo intensities to enhance assessment. Remove the noise filter, here an average filter to dispose of the noise right here laplacian is used for area detection because it highlights the area of speedy brightness trade. Thus, this more advantageous, noise-loose photograph is filtered and equipped for in addition processing. In order to meet the preference and demands of the socio-economic pain, the assessment of the first-rate of the goods is very essential nowadays. Product nice consists of sensory properties (appearance, texture, taste and aroma), dietary values, chemical elements, mechanical residences, purposeful properties and defects. All products have a constrained shelf life in the course of which they go through structural and chemical changes. Maturity publicity is an outside nice thing. Yes, but CNN itself. Due to the shortage of texture, the fruit can be diagnosed. Texture evaluation exhibits non-uniformity of the fruit's outer floor. Size is also an essential parameter. The parameter makes it clean that each one clients pick out consistent with their length.

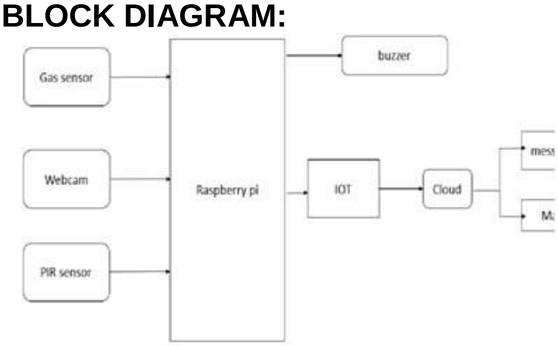
2. The freshness stage of meat is an important component in figuring out the first-class of meat intended for intake. In this studies, a sensor system changed into established to perceive the freshness of subject foods in a quick, accurate and non-negative manner. The device became implemented on a Raspberry Pi prepared with gasoline and colour sensors as weed identity equipment to replace human scent and imaginative and prescient in determining fresh meat. An instance

of popularity primarily based on neural networks is used to apprehend the newness of meat. The inputs to the neural community are the odors detected by means of the MQ-136, MQ-137, TGS 2620 community fuel sensors and the pink, inexperienced, and blue colour sensors detected by the TCS 3200. Three stages of novelty have been skilled, along with sparkling meat, semi-green meat, and spoiled meat. Using three sensors of the gasoline gadget and one colour sensor can accumulate a wonderful shape for all three kinds of novelty. Weeding meat has a excessive fulfillment fee of up to eighty%. Errors are because of a small difference in the pattern detected via the sensors for the gas.

SCOPE AND LIMITATIONS:

This study focuses on classifying raspberries as fresh and rotten and estimating the number of days before blackberries spoil. This leads to implementing deep learning techniques on mainstream images. The dataset used in the experiment was created by taking images of raspberries with an RGB camera and creating images based on visual inspection. Build models using convolutional neural networks to recognize patterns in images and predict product states. The model is installed and evaluated based on the F1 score and its classification accuracy. For my experiment, I bought raspberries from different stores. Data is a shared resource and not restricted to any one person. The photos are taken in different lighting conditions so that the brightness of the photos does not affect the model's performance. The camera used to capture the image is a regular digital camera that produces RGB images. In contrast to multispectral cameras, which capture both external and internal aspects of products, digital cameras only capture external aspects of products such as colour, size, and shape. Therefore, the model is designed to predict fruit condition based on the external soil and does not take into account the internal aspects of the fruit. Image labels are then assigned based on the researcher's visual perception of the state of the fruit observed. The fruit was refrigerated throughout the experiment, so environmental factors were not taken into account. A relatively small sample of 1128 images are also available for testing. There is an imbalance in which there are more good images than bad images.

BLOCK DIAGRAM:



PURPOSE

We advise the layout and creation of an IoT based totally smart kitchen machine. Here various sensor systems are used to come across gasoline leak in the kitchen management. Its outputs are then interlinked with the Raspberry pi MCU controller programmed inside the Python language, we will attain this and lots of different protection functions that may be included into the automation gadget which includes a temperature sensor. Through IoT, we can come across Gas Leak condition in far flung places. Continuous monitoring of kitchen offerings is carried out by means of this machine.

EXISTING SYSTEM

• Many strategies are now to be had to hit upon any fuel leak or fuel glide from cylinders.

• The foremost principle of the technique is that the change within the LPG withdrawal is detected with the activation of the audio visible alarm as quickly as it exceeds the explicit threshold. In addition, it sends some other message via the device's radio frequency receiver module. • The receiver module can therefore be a mobile unit, so as to be positioned somewhere in the premises of the residence, so the alarm zone is detected and detected via the unit commonly in the region in which the steam flows.

• The Arduino reads the voltage from the detector and makes use of it to calculate the trade in voltage.

• The gasoline detector is sensitive to many gases and cannot determine the type of gas. However, in practice, assuming that the gas sensitive detail has the identical or equal sensitivity for LPG and CH4 may be taken into consideration a sturdy assumption.

PROBLEM STATEMENT • We recommend the layout and construction of an IoT-based smart kitchen machine. Here numerous sensor structures are used to locate gas leak in the kitchen control. • Its outputs are then interfaced with a Raspberry pi MCU controller programmed in the Python language

• We can reap this safety characteristic and plenty of others that can be included into the automation device such as a temperature sensor.

The popularity of the lake via IoT, we are able to detect it in far flung places. • Continuous tracking of kitchen offerings is completed on this way.

PROPOSED SYSTEM

• We propose the layout and creation of an IoT-primarily based clever kitchen system. Here various sensor systems are used to locate gas leak inside the kitchen management.

• Its outputs are then interfaced with a Raspberry pi MCU controller programmed inside the Python language • We can acquire this protection function and lots of others that can be integrated into the automation machine along with a temperature sensor.

• Status of fuel leaks Through IOT, we can stumble on in far off locations. •

Continuous tracking of kitchen services is carried out on this way.

ADVANTAGES OF PROPOSED SYSTEM

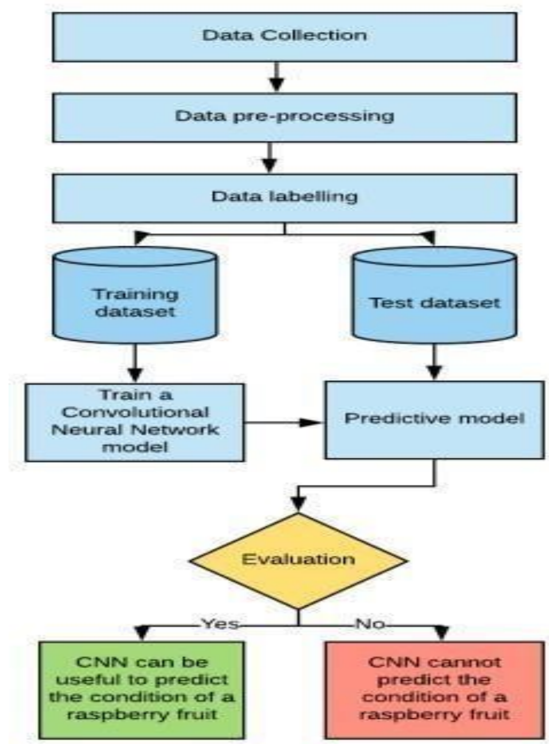
- Self-monitored and operational education
- Properly working system
- Adapted to humans with disabilities
- Portability and mobility

RESEARCH METHODOLOGY

The study is categorized as primary research because it focuses on forecasting consumer behaviour and collects data in the form of photos. An RGB camera is used to take pictures of the scraped baskets. We want to conduct a literature study of existing methods for the detection of food waste, the use of hyper spectral images in agriculture, and convolutional neural network image processing as part of the primary research. Both quantitative and empirical studies are included. By accuracy and F1 score, the CNN classifier's performance will be assessed. These findings lead to the inductive study conclusion, which is whether or not the CNN model can forecast the state of the raspberry fruit.

PROBLEM DEFINITION

The main purpose of this study is to forecast the market situation of products with short shelf life. For this purpose, deep learning based convolutional neural network models are formed from images collected by digital cameras and labelled by researchers according to aspects of the images. Deep learning model performance is evaluated using F1 score and classification accuracy.



DATA COLLECTION

You don't compulsion to construct a dataset for this experiment. A record is created by capturing the current arc. Purchase a basket of raspberries with the same expiration date from another store and store them in the refrigerator at 4 degrees Celsius. Photographs of each raspberry basket are taken by removing the basket from the refrigerator and laying it on side on a white flat surface. Photos are taken using a digital camera with a resolution of 13 megapixels. The image be of three treatment: red, green and blue. Raspberry clusters are expected to rot over time, as evidenced by fungal growth and signs of surface rot and warping of the fruit. All information is captured as images. Once the images are collected, they are pre processed and annotated to create datasets.

DATA PROCESSING AND PROFILING

All images collected from Raspberry are 4160x3120 pixels in size. The images are decomposed or subdivided into smaller dimensions to increase the number of images and also allow the model to easily identify patterns from images 18 that emphasize regions of interest within the images. A Python programming language like the OpenCV library is used to read the system image and perform the segmentation. After the images are segmented, they are analysed to check for discrepancies, background information, and image pixel quality. Images that do not meet the above criteria cannot be used for research. Once the images have been segmented and profiled, each image is labelled based on the researcher's visual interpretation of that image's fruit quality. Signs such as the presence of fungi, fruit deformation, and rot on the surface of the fruit should be considered when evaluating photographs of pterygium as good or bad. increase. Datasets are created in TF Record file format. This is a simple binary file format used for training models using the Tensor flow framework. 70% of the data is used for model training and the remaining 30% for testing the model implementation.

MODELLING

This research aims to build a model that predicts the state of red berries in order to acquire advanced skills. Convolutional neural networks are used for this purpose. The model implementation uses the Python programming language and the Tensor flow library. Results are generated on a Dublin Institute of Technology-provided cluster equipped with two 2.1 GHz 8-core Intel Xeon processors and dual Tesla K40 and K80 graphics cards. The rhombus is composed of layers of the neural tongue, layers of various whorls, layers of the largest layer, and layers of the large layer. Local correlations are somehow exploited locally to create patterns of spatial connectivity between adjacent layers of neurons (Hu, Huang, Wei, Zhang & Li, 2015). Some of them face the stage of life. It consists of predefined filters that are continuously trained. Additional information such as edges and gradients, as well as the raw input image itself, are provided to the network using this layer. It highlights the opposite and helps improve detection rates (Schmidhuber, 2015). The layers of a convolutional neural network are: I. Layer II. Stratus Group, iii. Figure 3.2 Fully Connected Layer.

Components of convolutional networks.

Convolutional Layers

Convolutional layer neurons are poorly connected to the next layer of neurons. A clone of the convolutional neuron layer is introduced to cover the entire input. This is done using filters.

- Filters - Filters are used for simultaneous training using a simple feature detector scanning the input vector. Each filter uses the same weights, so it can be considered a clone of the other filter. The width and height of the filter are defined according to your requirements. The output of the applied filter will be the feature map
- Feature Maps - Usually many filters are used in convolutional layers. Each of these filters returns or creates a unique table. The output ratio of convolutional layers depends on three parameters. Width and Height provide information about the dimensions of the film panel. Height is the number of threads in the roll layer. The slider filter window can be controlled by a module called step. This module allows you to change the overlap between the input image and the filter, so the feature maps have different sizes. Multiple levels of scrolling in one developer kit.

POLLINGLAYER

A feature map is supplied as input to a layer group. Having many filters in a convolutional layer gives many output values with many values used in the layers of the lower network. Overlay the file on the map and downsample it to reduce its size. This is done by taking a number of units and returning the sum of those units. One way to aggregate the values is to use max polish. Max-polishing computes the aggregate by taking the largest value among the units. Helps reduce overfitting

Fullyconnectedlayer

A neuron in a connectivity layer is connected to all neurons in the previous layer. Convolutional layers detect local features, while fully connected layers detect features in the global image. This layer applies a Softmax-like activation function to output the predicted class probabilities.

EVALUATION

This test uses the F1 score and accuracy to evaluate the model. The model runs on 70% of the data and 30% is used for estimation. The metrics used to evaluate the model in this classification task are: twenty one.

- Accuracy or Classification accuracy (CA),
- Precision
- Recall
- F1 score

HARDWARE DESCRIPTION RASPBERRY PI

The Raspberry Pi appears to be new to the arena and a whole lot of people don't simply realize what the

Raspberry Pi is. Raspberry Pi can be defined as a low-fee credit cardsized laptop that plugs into a computer or tv show and uses a general keyboard. And the mouse. It's a succesful little gadget that we could people of every age learn how to compute and explore programming in languages like Scratch and Python. It is capable of doing the whole lot you'll assume from a computing device pc, from browsing the Internet and playing high definition movies, to developing spreadsheets, word processing and games. It's superb to connect with Arduino and may do loads with Arduino.

Exploring The Rasbperry Pi Board

There are two fashions of Raspberry Pi, version A and version B. These fashions have few advanced capabilities in model B as compared to model A. Model B has 512MB of RAM, two USB ports whilst model A has 256MB of RAM. And only a USB port. Also, the Model B has an Air port, the A does not.
Overview of the Raspberry Pi Model B in
Figure 5-1.
Figure five-1: Raspberry Pi Model B (Sanders, 2013)
The special Pi's are named in determine five1 and a quick description of every factor is given inside the following sections.

SD Card Slot

The Raspberry Pi does not have a real hard drive like in laptops and computer systems, the SD card is taken into consideration a strong nation drive (SSD) that's used to install the running system and all other software and all storage. This card is required to be inserted into the slot as a PI.
The SD card can be 2 GB, 4 GB or sixteen GB.

MICRO USB Power

The electricity port is a 5V micro-USB enter and the electricity supply must be exactly 5v because it does not have a built in electricity controller. So the strength supply need to no longer exceed 5V.

HDMI out

This output port is used to attach the screen via HDMI Pi to the Raspberry Pi (High Definition Multimedia Interface). Therefore, any display screen or TV can be connected to it that has an HDMI port.

Ethernet and USB port

The Model B's Air port and USB port may be powered via the onboard LAN9512 chip. High pace USB 2.Zero hub with 10/a hundred Esternet controller (Donat, 2014). USB ports to attach inputs (keyboard, mouse). Almost whatever which could connect to a laptop via USB also can hook up with a Raspberry Pi.

Hardware required for Raspberry Pi

Raspberry Pi can't begin via itself, it needs many different peripherals (hardware). A quick description of the hardware necessities is given inside the next phase (Bates, 2014). **4.1 Power Supply**

As already mentioned within the theoretical component above, the Raspberry Pi desires a 5V power supply. If the power deliver exceeds 5V, it is not assured to paintings nicely. And a supply of at least 500 milliamperes (mA), and greater ideally 1 ampere (A). If the strength supply is 500mA or much less, the keyboard and mouse are probably practical. It's no longer an excellent idea to electricity the Raspberry Pi from the USB port and hub of the computer, a good way to generally deliver much less contemporary than needed. Therefore, the Raspberry Pi requires a Micro-USB connection capable of delivering as a minimum 700mA (or zero.7A) at 5V.

4.2 Storage

Separate hardware is needed to use storage on the Raspberry Pi. For this, an SD card is used, ideally four GB and 8 GB, if essential. The running system and all files are stored on paper. We should purchase a blank SD card and install the working device or buy a preset up one.

4.3 Entry

An external keyboard and mouse is needed to offer enter to the Raspberry Pi. No additional programming is required to use the keyboard and mouse.

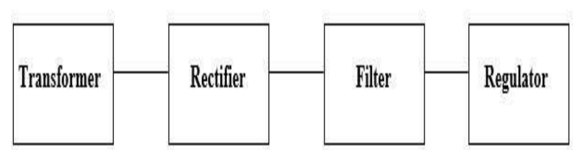
4.4 Monitor

We can use a display or TV with HDMI port or DVI inputs as a display for the Raspberry Pi. For DVI inputs, you want an HDMI to DVI converter, and it is able to be without problems located on the market. The reveal could be very important to the Raspberry Pi because it's miles the handiest way to peer what we have finished on it.

4.5 Network

Just like on a pc or pc, we also can access the Internet and community on the Raspberry Pi.
For this we will use a stressed out connection that's a simpler alternative or a Wi-Fi module to access Wi-Fi on the Raspberry PI. **4.6 Power supply**

A power supply (every now and then called a power deliver or PSU) is a device or gadget that materials electric or other styles of electricity to an output load or organization of loads. The term is most often implemented to electrical substances, less regularly to mechanical resources, and infrequently to others.



WEBCAM: -

A webcam is a video camera that transmits its photos in real time to a computer or pc network, frequently thru USB, Air or WiFi. Their maximum not unusual use is setting up video links, allowing computer systems to behave as videophone stations or videoconference stations. It is commonly used as a video camera for the World Wide Web. Other famous makes use of encompass security surveillance and laptop imaginative and prescient. Cams are recognised for their low production fee and flexibility, making them the cheapest form of video calling. They have also emerge as a supply of safety and privacy concerns, as some built in webcams may be eliminated remotely by adware.

LENS

Cams usually consist of a lens, an photograph sensor, supporting electronics, and may even consist of a microphone for

sound. There are an expansion of lenses available, the most common in customer webcams is a plastic lens, which can be bent and bent to focus the camera. Fixed focal period lenses, which have no adjustment, are also to be had. As the depth of subject of the digicam system is more for small photo codecs and greater for the quantity of lenses (small aperture) lenses, the structures in webcams have a fairly large intensity of field so that the usage of the first lens does no longer have tons impact at the photograph sharpness.

PORTABILITY

Most webcams come with constructed in microphones to make video calling and video conferencing greater convenient. The USB Video Device Class (UVC) specification permits for connecting a webcam to computers without the want for proprietary tool drivers. Microsoft Windows XP SP2, Linux and Mac OS X (as of Oct. 2005) have constructed in help for UVC and do not require additional device drivers, despite the fact that they're often mounted to decorate extra functionality.

FEATURES OF WEBCAM

- Plug-and-play configuration (UVC)
- Video capture: as much as 640 x 480 pixels
- Photos: up to one.3 megapixels (software enhancement)
- Frame charge: as much as 30 frames according to 2d (with encouraged gadget)
- USB 2.0 Hi-Speed Certified
- fixed awareness
- Universal clip fits laptops, LCD or CRT monitors.

IMAGEPROCESSING AND RASPBERRY PI

Today's image processing makes use of a spread of strategies, this text gives the implementation of the photo processing operation at the Raspberry Pi. Raspberry Pi is a simple embedded gadget and being a low price single board computer to lessen gadget complexity in actual time programs. This platform is specifically based on Python. The Raspberry Pi includes a Camera Slot interface (CSI) to interface the Raspberry Pi Camera, right here, dark and coffee-first-rate snap shots captured the usage of the Raspberry Pi Camera Module are stronger to discover specific regions of the picture. This concept is used in a actual-time MAV application. MAVs are normally used to capture pics and motion pictures thru the Raspberry Pi digicam module. Due to the credit score card size (small) and much less weight in design. The image taken by way of the MAVs will include undesirable gadgets due to atmospheric situations; consequently, it is essential to eliminate the noise inside the MAV pics.

Gas sensor Module

In ultra-modern technological scenario, the tracking of the produced gases may be very critical. From family appliances which includes air conditioners to electric powered stoves and security structures in industries, fuel monitoring could be very important. Gas sensors are the maximum critical part of these structures. As small as a nostril, fuel sensors routinely react to the presence of fuel, consequently informing the system of any modifications that arise inside the retention of molecules in the gaseous state.

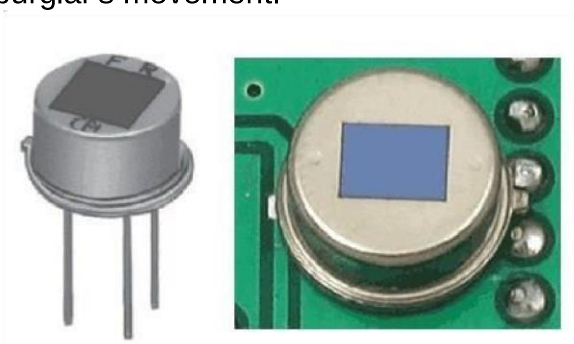
Gas sensors are available in many kinds depending on the level of sensitivity, type of gas to be detected, bodily dimensions and lots of different elements. This prospect includes a methane gasoline sensor that could locate gases like ammonia that could be constituted of methane. When the fuel enters this sensor, it is first ionized into its additives and ignited through the touchy element. This adsorption creates a capacity difference throughout the element this is pushed to the processing unit through the output pin inside the form of a contemporary.

PIR MOTOR SENSOR

A PIR detector is a movement detector that detects warmth emitted by way of a dwelling body. These are frequently fitted with security lighting fixtures that come on robotically if approached. They are very effective in improving domestic protection structures.

It is a passive sensor due to the fact, in preference to emitting light rays or microwave strength, which should be intercepted via a passing human to "odor" it, the PIR is absolutely sensitive to the infrared strength emitted through any living thing. When an interloper enters the detector's area of view, the detector "sees" a sharp increase in infrared electricity.

The PIR sensor is designed to show at the mild whilst it tactics, however does now not fight, a desk bound individual. The lighting fixtures are designed in this manner. A transferring man or woman shows a sudden change in infrared electricity, but a slower trade is emitted through a stationary body. Slower changes also are caused by sluggish fluctuations in environmental temperature. If mild were to cause these touchy adjustments extra slowly, it might act to chill the megastar at night time, in preference to triggering the burglar's movement.



Connecting PIR

Opencv

OpenCV is a cross-platform library the usage of which we can develop actual-time pc imaginative and prescient applications. It specially makes a speciality of image processing, video capture and analysis, which includes features which includes face detection and item detection. Let's begin the bankruptcy via defining the term "Computer Vision"

Computer vision

Computer vision may be described as a subject that explains the way to reconstruct, interrupt and understand a 3-d scene from its 2D pics, based on the residences of the structure that is in the scene. It offers with the modeling and reproduction of human imaginative and prescient the usage of computer software program and hardware.
Computer vision overlaps extensively with the following areas:

- Image processing: It focuses on photograph manipulation.
- Pattern Recognition: explains numerous strategies for analyzing patterns.
- Photogrammetry: This includes acquiring particular measurements from images. **OpenCV Library Features**

Through the OpenCV library you could use -

- read and write pics
- Download and shop films
- image processing (filter out, remodel)

Provide function • detection • Detect unique gadgets consisting of faces, eyes, cars, in movies or photographs. • Spend a video, this is, it carries an evaluation of the movement, cast off the history and song the things that it

contains. OpenCV first in C++. In addition to this, Python and Java bindings are provided. OpenCV works on various operating systems such as Windows, Linux, OSx, FreeBSD, NetBSD, OpenBSD, etc.
This academic explains the ideas of the usage of OpenCV with examples of Java bindings. **OpenCV Library Modules**

Here are the primary libraries of the OpenCV module library.

Basic capability

This module covers the simple records structures like Scalar, Point, Range and so on., which are used to construct OpenCV packages. In addition to these, there are also multidimensional arrays of Mats that store pics. The OpenCV Java library, whose module package is covered with the call org.Opencv.Core.

Image processing

This module includes diverse picture processing operations which includes photograph filtering, geometric photo adjustments, coloration area conversion, histograms, etc. The OpenCV Java library, whose module is protected as a package named org.Opencv.Imgproc.

I see

This module covers video analysis ideas which includes motion estimation, historical past subtraction and item tracking. The OpenCV Java library, whose module is covered as a package with the call org.Opencv.Video.

Video I/O

The improvement of the video capture module and the video code the use of the OpenCV library. The OpenCV Java library, whose module is protected as a package deal named org.Opencv.Videoio.

Caliber3d

Its module algorithms encompass simple multi-view geometry algorithms, unmarried and stereo camera calibration, object estimation, stereo matching, and three-D reconstruction of elements. The OpenCV Java library, whose module is covered as a package named org.Opencv.Calib3d.

Features2d

Its module is familiar with the principles of characteristic detection and description. The OpenCV Java library, whose module is covered as a bundlenamed org.Opencv.Features2d.

Object detection

Its module consists of the detection of objects and instances of predefined kinds along with faces, eyes, cups, people, cars, and many others. The OpenCV Java library, whose module is covered as a package deal named org.Opencv.Objdetect. Highgui
The UI interface is easy to apply with simple methods. The OpenCV Java library, whose module capability is

covered in two Extraordinary documents,specificallyorg.Opencv.Imgcodecs and org.Opencv.Videoio.

OpenCV - Image Storage

To capture an photo, we use gadgets which include cameras and scanners. These devices report the virtual values of the image (Ex: pixel values). OpenCV is a library that procedures virtual photographs, so we want to store those pix for processing. Mat class of OpenCV library to save picture values. It represents an n-dimensional array and is used to save grayscale or colour snap shots, voxel volumes, discipline vectors, scatterplots, tensors, histograms, and so forth. This type includes two statistics components: a header and a pointer.

• **Header**:

Information which includes length, technique of garage used, array cope with (constant size).

• Tip: adds pixel values to the photograph (continues to vary).

Conclusion:

In this mission, quality-based identity of everyday and faulty merchandise the usage of OPENCV/PYTHON is efficaciously and accurately executed. The use of photograph processing to become aware of first-class can be carried out not only to any fruit. We can also use this method to understand the satisfactory of greens more correctly. Therefore, this technique can be carried out in many products. For guide inspection of meals garage, a pc imaginative and prescient machine is used that gives an actual, honest and non-adverse assessment. Image processing finished with the aid of OpenCv makes use of the Python programming language. The software is divided into two elements, one for image analysis and the other for hardware manipulate that consequences in picture processing. The system works below distinctive scenarios, first the photograph is captured with the digicam, then all of the image processing is completed in the manage module. All the approaches are shown at the reveal and then consistent with the choice made by using the control module. Transporter reduced assembly.

REFERENCES

[1] X. Chen and A. L. Yuille, Text detection and analyzing in natural scenes, in Proc. Calculus Vision Pattern Recognition, 2004, vol. 2, p. II-366-II373.

[2] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi, "Text Extraction and Image Segmentation from Documents Using Paired Wavelets and MRF Model", IEEE Trans Image Process., vol. 16, no. Eight, p. 2117-2128, Sext. 2007

[3] K. Kim, K. Jung, and J. Kim, "A TextureBased Approach to Detecting Texture in Images Using Support Vector Machines

and a Continuously Adaptive Mean Shift Algorithm," IEEE Trans. Model Anal. Mach. Intel., vol. 25, no. XII, p. 1631-1639, December 2003.

[4] N. Giudice and G. Legge, "Blind navigation and the function of technology", in The Engineering Handbook of Smart Technology for the Elderly, Disabled, and Independent, A. A. Helal, M. Mokhtari and B. Abdulrazak, Ed. . Hoboken, NJ, USA: Wiley, 2008.

[5] World Health Organization. (2009). 10 records about blindness and visible impairment.

[6] National Health Survey preliminary information reports via interview (2008).

[7] International workshop on the analysis and review of files by way of digital camera (CBDAR 2005, 2007, 2009, 2011).

[8] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic Signal Detection and Recognition of Natural Scenes," IEEE Trans. Image Processing, vol. 13, no. 1, p. 87–ninety nine, January 2004.

[9] D. Dakopoulos and N. G. Bourbakis, "Escape for a stack of electronic tour aids for the blind: A sur-vey", IEEE Trans. Syst., Man, Cybern., vol. Forty, no. 1, p. 25–35, January 2010.

[10] B. Epshtein, E. Ofek, and Y. Wexler, "Texture detection in herbal scenes with stroke-width transformation", in Proc. Calculus Vision Pattern Recognition., 2010, pp. 2963-2970.

[11] Y. Freund and R. Schapire,"Experiments with a new boosting set of rules," in Proc. Int. Conf. Machine Learning, 1996, p. 148156.

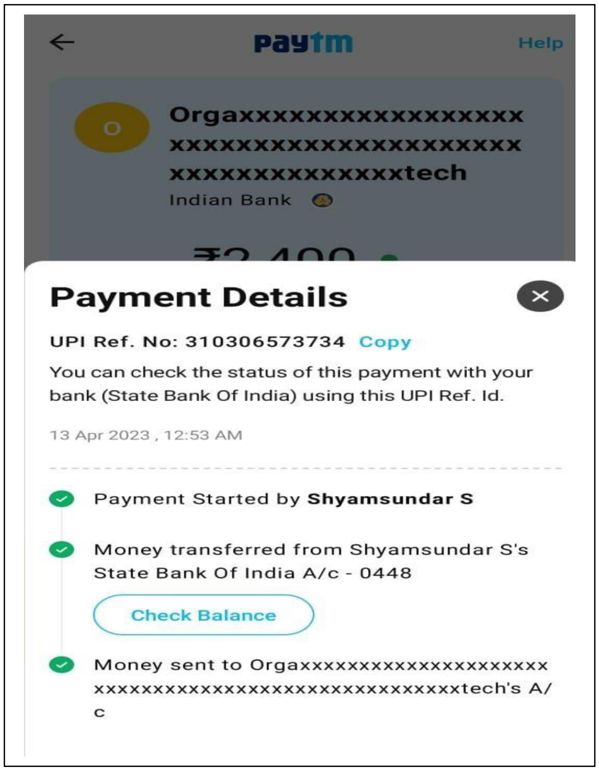
[12] An evaluate of the Tesseract OCR(optical individual popularity) system and its improvement to be used in Wales at some point of the pre-aggressive research section organized through the Language Technology Unit (Canol-fan Bedwyr), Bangor University April 2008.

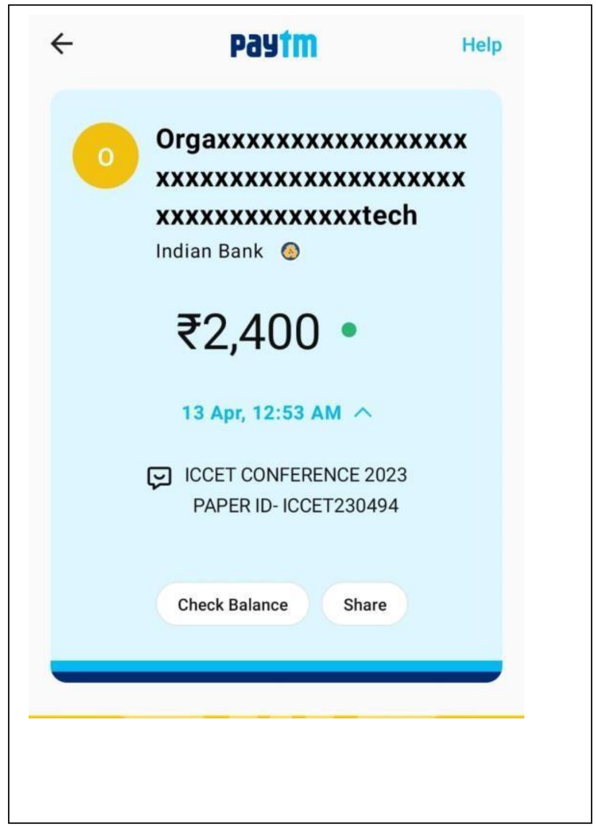
[13] A. Shahab, F. Shafait and A. Dengel, "ICDAR Robust Reading Competition 2011: ICDAR Robust Reading Competition Challenge 2: Reading Text in Scene Images", in Proc. Int. Conf. Document Anal. Review, 2011, p. 1491-1496.

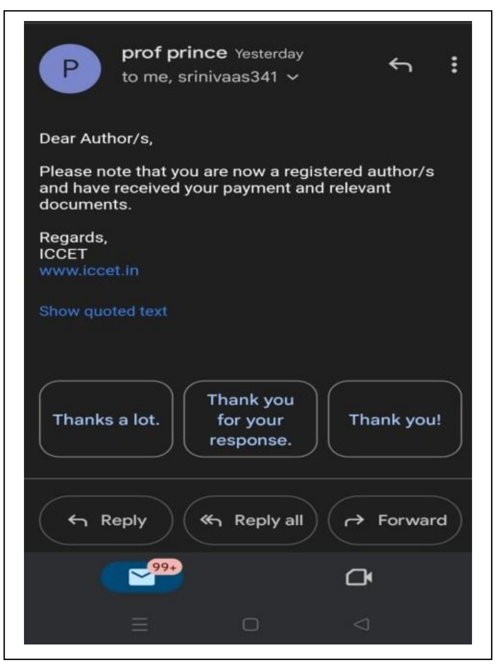
[14] KReader Mobile User Guide, knfb Reading Technology Inc. (2008).

[15] S. M. Lucas, "ICDAR 2005 text consequences of the locator competition", in Proc.Int. Conf. Document Anal. Review, 2005, vol. 1, p. 80-eighty

four.characteristic carried out in situ," Journal of immunological methods, vol. 130, no. 1, p. 111-121, 1990 .


Paytm
Help
Orgaxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxtech
Indian Bank
Payment Details
UPI Ref. No: 310306573734 Copy
You can check the status of this payment with your bank (State Bank Of India) using this UPI Ref. Id.
13 Apr 2023 , 12:53 AM
Payment Started by Shyamsundar S
Money transferred from Shyamsundar S's State Bank Of India A/c - 0448
Check Balance
Money sent to Orgaxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxtech's A/c


Paytm
Help
Orgaxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxtech
Indian Bank
₹2,400
13 Apr, 12:53 AM
ICCET CONFERENCE 2023
PAPER ID- ICCET230494
Check Balance
Share


prof prince Yesterday
to me, srinivaas341
Dear Author/s,
Please note that you are now a registered author/s and have received your payment and relevant documents.
Regards,
ICCET
www.iccet.in
Show quoted text
Thanks a lot.
Thank you for your response.
Thank you!
Reply
Reply all
Forward
99+