**Industrial Training and Industrial Project Report**

**(Industrial Training CSE-410)**

**On**

**IOT Based Smart Doors for Monitoring Face Mask Detection**

**At**

**Bdtask Limited, Dhaka**

**logo**

This project report submitted to the department of Computer Science and Engineering, University Of Creative Technology Chittagong in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering.

**SUBMITTED BY**

Jobayer Habib Afsan

Student ID: 170321016

BL- Student ID-202101-IOT-001**017**(BdTask**)**

School of Science and Engineering.

## Department of Computer Science and Engineering, 2nd Batch

**SUPERVISED BY**

## M.M.Musharaf Hussain

## Head and Associate Professor

## Department of Computer Science & Engineering

****

**School of Science and Engineering**

**University of Creative Technology Chittagong (UCTC)**

**June 2022**

**DECLARATION**

This project report is submitted to the department of Computer Science and Engineering, University of Creative and Technology Chittagong in partial fulfillment of the requirements for the degree of Bachelor of Science. So, we hereby, declare that this report is based on the surveys found by us and our original work, that has not been submitted anywhere for any award. Materials of work found by other researchers are mentioned with due reference. All the contents provided here are totally based on our own effort dedicated for the completion of the project.

Signature of authors

Student Name: Jobayer Habib Afsan

Student ID: 170321016

Department of CSE 2nd Batch

Email: [afsan.uct@gmail.com](mailto:afsan.uct@gmail.com)

**ACKNOWLEDGEMENT**

It is our privilege to express our sincerest regards to our Industrial Training Supervisor, Atiqul Islam for his valuable inputs, guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project. We would like to take this opportunity to express our gratitude to our Honorable project supervisor Mr. M. M. Musharaf Hussain sir (The Head & Associate Professor of UCTC) for providing a wonderful opportunity to do this project. The opportunity helped us improve our research skills and helped us learn new things. Thank you for allowing us to do this project and providing invaluable guidance throughout the project. Without your continuous support, patience, enthusiasm, and immense knowledge, we would not be able to finish our project.

**Declaration of Completion of Training**

Date: 2-June-2022

To,

The

Head, Department of CSE

University of Creative Technology, Chittagong (UCTC)

Dear Honorable Sir,

This is to inform you that, the student **Name: Jobayer Habib Afsan,**

**UCTC, Student ID: 170321016,** has completed Industrial Training under the IOT section at Bdtask Ltd, both online and offline mode.

He conducts the Industrial Training and Project work, 6 months and three classes (each class -3 hours) in a week.

So, in this status we have provided him Industrial Training and Project work completion certificates.

Thanks & Regards

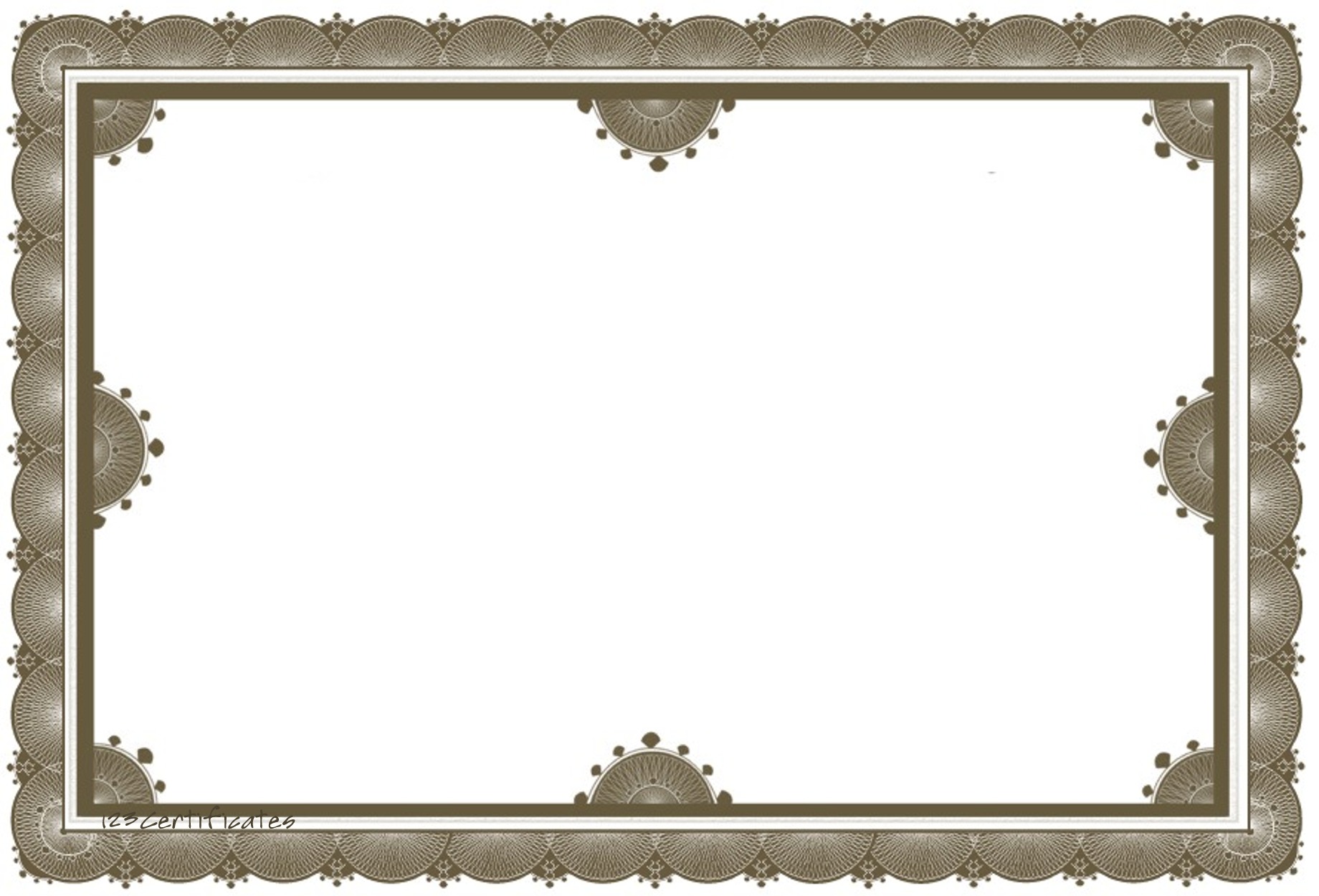


Md. Sabbir Hussain

Business Development Executive

Bdtask Ltd.



**Certificate of Completion**

# ***Certificate of Achievement***

This certificate is presented

To

## Jobayer Habib Afsan

UCTC-Student ID-170321016; BL- Student ID-202101-IOT-001**017**

*Student of Bachelor of Science in* ***Computer Science and Engineering,*** *Department of Computer Science**and Engineering, University of Creative Technology Chittagong, for his excellent performance**of Industrial Attachment & Training in Internet of Things (IOT)* ***and*** *which has completed**on 2 June, 2022 at Industrial Attachment & Training Section in Bdtask Limited.*



Md. Sabbir Hussain

Business Development Executive

Industrial Attachment & Training Section

### Bdtask Limited, Dhaka

Contact No: +8801958069665

Email ID: [sabbirbdtask@gmail.com](mailto:sabbirbdtask@gmail.com)



### Atiqul Islam

Co- Supervisor

IOT Expert, Researcher & Trainer

Industrial Attachment & Training Section

### Bdtask Limited, Dhaka

Contact No: +8801303365505

**ABSTRACT**

COVID 19 pandemic is causing a global health epidemic. The most powerful safety tool is wearing a face mask in public places and everywhere else. The COVID 19 outbreak forced governments around the world to implement lockdowns to deter virus transmission. According to survey reports, wearing a face mask at public places reduces the risk of transmission significantly. In this paper, an IoT-enabled smart door that uses a machine learning model for monitoring face mask detection. The proposed model can be used for any shopping mall, hotel, apartment entrance, etc. As an outcome a cost-effective and reliable method of using AI and sensors to build a healthy environment. Evaluation of the proposed framework is done by the Face Mask Detection algorithm using the TensorFlow software library. This proposed system can detect the users from COVID 19 by enabling the Internet of Things (IoT) technology.

**Keywords:** COVID-19, Face mask detection, Machine learning, Raspberry Pi, TensorFlow, CNN.

**CONTENTS**

**DECLARATION……………………………………………………...………………………i**

**ACKNOWLEDGEMENT……………………………………………..…………………….ii**

**ABSTRACT………………………………………………………………………………….iii**

1. **INTRODUCTION…………………………………………………………………………1**
2. **PROPOSED APPROACH………………………………………………………………..2**
   1. Software Requirements……………………………………………......…………………..2
      1. TensorFlow…………………………………………………………….….……………..2
      2. PuTTY……………………………………………………………………….…………..2
      3. VNC Viewer..……………………………………………………………………………3
   2. Hardware Requirements………………………..………………………………………….3
      1. Raspberry Pi………………………………………………………………..……………4
      2. R pi's Cam (Raspberry Pi Camera) ……………………………………………………..5
      3. IR Sensor…………………………………………………………………………….…..5
      4. Servo Motor……………………………………………………………………………...6
3. **METHODOLOGY………………………………………………………………………...7**
   1. Facemask Detection……………………………………………………………………….7
      1. Convolution Neural Networks (CNN) Algorithm……………………………………….7
      2. Dataset Collection……………………………………………………………………….8
   2. System Overview………………………………………………………………………….9
4. **RESULTS AND DISCUSSION…………………………………………………………11**
   1. System Requirements…………………………………………………………………….11
   2. Quantitative Analysis with different test case……………………………………………11
   3. Qualitative analysis with different algorithms…………………………………………...12
   4. Comparative analysis with existing papers………………………………………………12
5. **CONCLUSION…………………………………………………………………………...14**
6. **REFERENCES……………………………………………………………………….......15**
7. **INTRODUCTION**

The coronavirus disease, or COVID-19, which originated primarily in Wuhan, China, has rapidly spread worldwide. Face masks and sanitizers are the most effective ways to minimize transmission. When it comes to reducing disease transmission, this has shown good results. Fever, sore throat, tiredness, loss of taste and smell, and nasal congestion are all common symptoms of coronavirus infection. The majority of the time, it is transmitted indirectly through surfaces. The incubation period can be very long, ranging from 10 to14 days in extreme cases, and the virus can attack directly (from one individual to other individuals) by respiratory droplets [2]. Governments implemented a variety of protection and safety initiatives to reduce disease transmission, including social distancing, mandatory indoor mask-wearing, quarantine, restricting citizens' traveling within state boundaries and abroad, self-isolation, and the exclusion and cancellation of big social occasions and meetings [10]. From work activities to social relationships, all kinds of sports activities, as well as off-screen and on-screen entertainment have all been affected due to this COVID-19 pandemic [4]. Individuals with high body temperature are not to be permitted to enter public places because they are at a high risk of infection and spreading the virus; wearing a mask is essential. At the entrances to any city, workplaces, malls, and hospital gates, temperature and mask checks are also necessary. As a result, a smart entry device that automatically detects a mask at the door opening system is developed.

1. **PROPOSED APPROACH**

**2.1. Software Requirements**

* + 1. **TensorFlow**

TensorFlow is a machine learning software library that is open source and free. It was created to perform large numerical computations without regard for deep learning. This TensorFlow can be used for a variety of activities, but it is primarily focused on deep neural network inference and training. TensorFlow also supports traditional machine learning. Google's TensorFlow is a Python library that allows for quick numerical computation. Deep learning models are either generated directly using TensorFlow, which is also a base library, or they are created to simplify the process by using wrapper libraries built on top of TensorFlow. TensorFlow enables the creation of dataflow graphs and structures to determine how the data flows through the graph by receiving inputs as a multi-dimensional tensor array. It allows building a flow chart for these inputs which is carried out on the one end and is performed on the other.

**2.1.2. PuTTY**

PuTTY is an open-source software that provides both serial console and software transfer for network files. . It supports a wide range of network protocols, such as SCP, Telnet, SSH, rlogin, and a crude socket connection which can also be connected to a serial port. PuTTy is a terminal emulator which enables users to access the Raspberry Pi command-line interface from any laptop or desktop device. SSH (secure shell) is used for this, which opens a terminal window on the laptop or device that can be used to send commands to the Raspberry Pi and retrieve data from it before sending it to the computer. Putty itself has the main Ikey file format, known as ppk. Raspberry Pis are commonly used as stand-alone, lightweight network computers. Raspberry Pi is wired to the same local network as the remote computer. On the Raspberry Pi, SSH is enabled which is supported by default in some Raspberry Pi distributions, but it can be configured again later using the Raspi-config tools.

**2.1.3. VNC Viewer**

Virtual Network Computing (VNC) is a graphical desktop sharing application that lets us monitor the desktop interface of one machine with another computer or mobile device remotely. The VNC viewer transmits to the VNC server with a mouse, keyboard, or touch case, receiving updates back on the display. Working directly on the Raspberry Pi is not always convenient. You may also want to include a remote control from another device to work on it. VNC uses Real VNC, which is used with the Raspberry OS. It comprises VNC Viewer, which allows users to remotely access a Raspberry with desktop, and a VNC server enables to monitor the Raspberry Pi remotely. It must be enabled first before using the VNC server. The VNC server provides the users with wireless monitoring to the Raspberry graphical desktop, which enables communication. However, the VNC server can be used to access the graphic remote if the Raspberry is headless and doesn't have a graphic screen.

**2.2. Hardware Requirements**

**2.2.1. Raspberry Pi**

The Raspberry Pi is a low-cost tiny computer that connects to a computer monitor or television and operates with a regular keyboard and mouse as shown in Fig 1 [8]. It is a handy little gadget that focuses on teaching people of all ages about scripting languages like Scratch and Python. It can perform all the functions of a desktop computer, such as internet surfing and viewing greater-definition clip, worksheets, and playing games. It has been used in several digital devices, including tweeting birdhouses, music machines, and detectors, as well as weather stations and infrared cameras since it is capable of interacting with the outside environment.

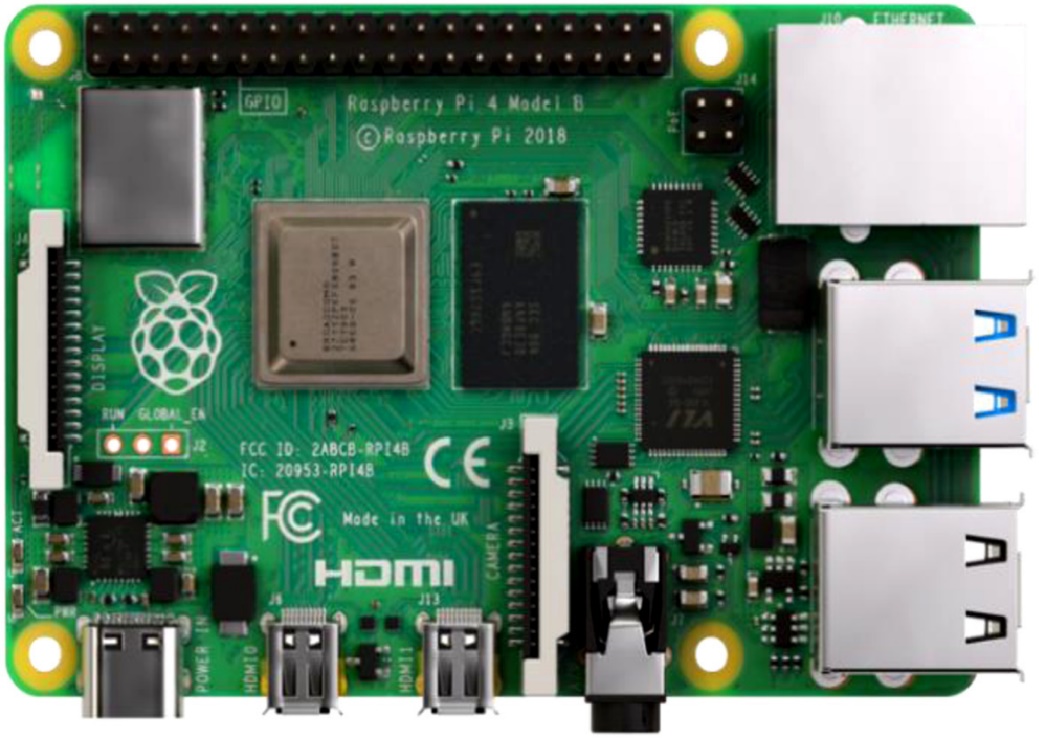


Figure 1. Raspberry Pi

It has a 1.2-GHz quad-core chipset BCM2387 with a GPU support of a dual-core and a video core multimedia co-processor and the GPU, which includes dual core multimedia co-processor, including a Bluetooth 4,1 (Bluetooth and Bluetooth Classic). With Bluetooth Low Energy (BLE) and BCM43143 Wi-Fi, the Raspberry Pi 3 offers an up-grade towards a new main processor and improved networking. Furthermore, the power management of Raspberry 3 has been improved, with an upgraded power supply with 3.5 Amps that can handle more powerful external USB devices. The built-in USB ports of Raspberry Pi 3 provide sufficient connectivity to link the mouse or anything else to the RPi. Most Raspberry Pi system chips can be overclocked to 800 MHz, and some can be overclocked to 1000 MHz. It is reported that the Raspberry Pi 2 can be similarly overclocked, even reaching 1500 MHz in extreme cases (without all safety features and overvoltage restrictions). On Linux distributions, you can use the program command to run "sudo raspiconfig" to perform boot overclocking without breaking the warranty. In certain instances, the Pi will automatically deactivate overclocking when the chip temperature reaches 85°C to cancel the automated settings on overclock and overclocking (that will cancel warranty). A cooler can be used to prevent overheating of Raspberry pi.

**2.2.2. R pi's Cam (Raspberry Pi Camera)**

An 8-megapixel sensor Pi Camera of Raspberry is used in this project. This camera module consists of 1080p30, 720p60, and 640 × 480p90 video support and support resolution of 3270 × 2444 pixels resolution. Fig 2 shows a Raspberry Pi camera module. It has a fixed lens and a Sony IMX219 image sensor that was specifically made for the R Pi as an add-on board. The Pi module is linked to the RPi through one of the board's little ports on the top part, and it also makes use of the specialized CSI gui, which is specifically made for camera connectivity.

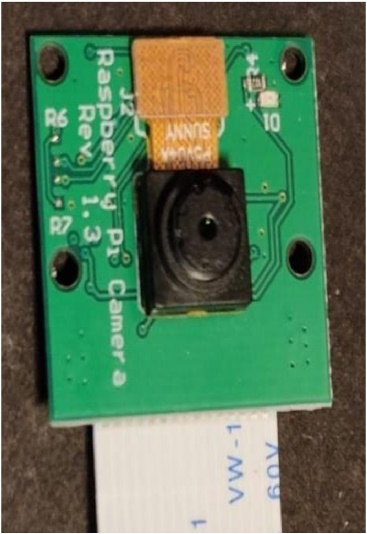


Figure 2. Raspberry Pi Camera

**2.2.3. IR Sensor**

Infrared sensors are used to count and monitor the number of people who enter and leave the room. The IR sensor's operating voltage is 5VDC, and the I/O pins are 5V and 3.3V compatible. It comes with a variety of options. Fig 3 depicts an IR Sensor that features a built ambient light sensor and a mounting hole, as well as an adjustable sensing range of up to 20cm.

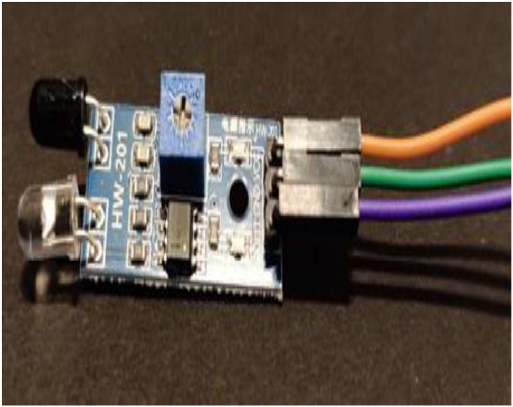


Figure 3. IR Sensor

**2.2.4. Servo Motor**

A servo motor is used to demonstrate the opening and closing of the main door. Fig 4 shows a diagram of a Servo Motor [7] that produces velocity and torque based on the voltage and the amount of current supplied. It also works as a part of a closed-loop system providing velocity and torque as commanded from the servo controller with a feedback device to close the loop.



Figure 4. Servo Motor

1. **METHODOLOGY**
   1. **Face mask detection**
      1. **Convolution Neural Networks (CNN) Algorithm**

In this paper, a deep learning algorithm is used to identify face mask recognition and, Convolution Neural Networks (CNN) classification. A CNN is a form of artificial neural network that is specifically built to interpret pixel input and is mainly used for image recognition and analysis, in which each layer applies to a different set of filters. Around 100’s to 1000’s of filters is combined to give a final result and then the obtained output is sent to the next layer in this neural network. Evaluation of the proposed framework is done by the face mask detection algorithm using the TensorFlow software library as shown in Fig 5. The Mask detector model is trained by using Keras and TensorFlow. The steps involved in the algorithm is given below.

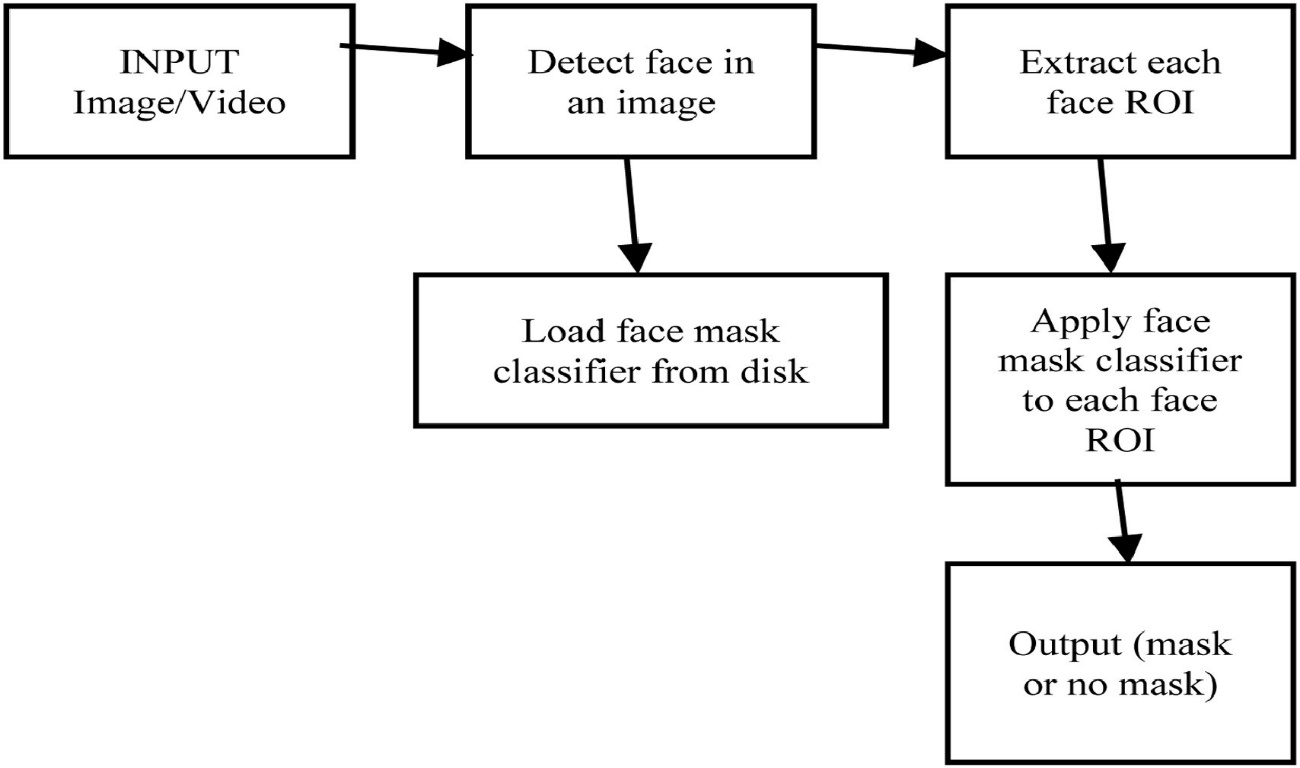


Figure 5. Face Mask Detection

STEP 1: DATASET COLLECTION

STEP 2: PRE-PROCESSING

STEP 3: SPLITTING

STEP 4: TRAINING

STEP 5: TESTING/EVALUATION

According to the above-mentioned algorithm, all the required dataset and components for building the network is collected from various categories. Once the initial dataset is ready, the next step is to train and test the set. This test dataset is used only in evaluating the performance of the network. Next training should be done, so the neural network learns to identify different categories in the given labels. Finally, the dataset should be evaluated and compared with the ground-truth labels.

* + 1. **Dataset Collection**

The images used for training and testing the model were obtained from the internet. The dataset used in this project was designed by Prajna Bhandary. This dataset contains 1,376 photos divided into two classes: 690 images with masks and 686 images without masks as shown in Fig 6. To create this dataset, they took regular photographs of people's faces and then used a custom-designed computer vision Python script to apply face masks to the pictures, yielding an artificial dataset. Facial landmarks allow the users to instantly infer the position of facial components such as the eyes, nose, eyebrows, mouth, and jawline. Then, using facial landmarks, the dataset of faces wearing masks can be created. To determine the bounding box region of a face in an image, start with an image of an individual who's not wearing a face mask and then apply face detection. It can capture the face Region of Interest (ROI) after determining where the face is now in the picture, and then utilize facial landmarks to detect the position of mouth, eyes, nose, and other features. Initially, an image of a mask is required, which will be put to the face automatically utilizing facial landmarks (particularly, the regions around the mouth and chin) to determine where the mask should be placed. After that, the mask is scaled and twisted before being fitted to the face, and the process is repeated for each of the input images, yielding an artificial face mask dataset as shown in Fig. 6.



Figure 6. People with and without mask

The face is captured and the blob is constructed from the image that depicts people with and without wearing masks [1]. This blob is passed via network to achieve face detection from the extracted blob and the trust (i.e., probability) is also associated with extracted detection. The weak detection is filtered to ensure that the confidence (probability) is more than min degree of reliability so that face ROI (Region of Interest) is extracted and switched to RGB format from BGR format and it is reformatted to 4 × 224, and then pre-processing is done, Now extracted face is sent via the mask detection model to detect the face sent is wearing a proper face mask or without a face mask. So, the bounding box and text are drawn and probability is included in the label. Finally, a white box which is known as a rectangle bounding box appears with a label as a frame on the output screen.

* 1. **System Overview**

Fig 7 shows the overview of the connection structures that make up the solution. Any person attempting to enter the building should first pass through infrared sensors, which are used to track and manage the individual count of people entering the room and later exiting. Body temperature is tested only when the people's total count inside a room is less than the given limit. The MLX90614 body temperature sensor is used for this purpose. If the person's body temperature is too high, the door will not open; if the person's temperature is average, the door will open and proceed to the next level, i.e., mask detection. The Raspberry Pi single-board computer with Raspberry Pi Camera is used for this function. If an individual wearing a mask is detected, the door will be opened. If the individual is discovered without a mask, the door will not open. To ensure the guidelines and safety for indoor workers during this COVID-19, this IoT solution based comes into action.

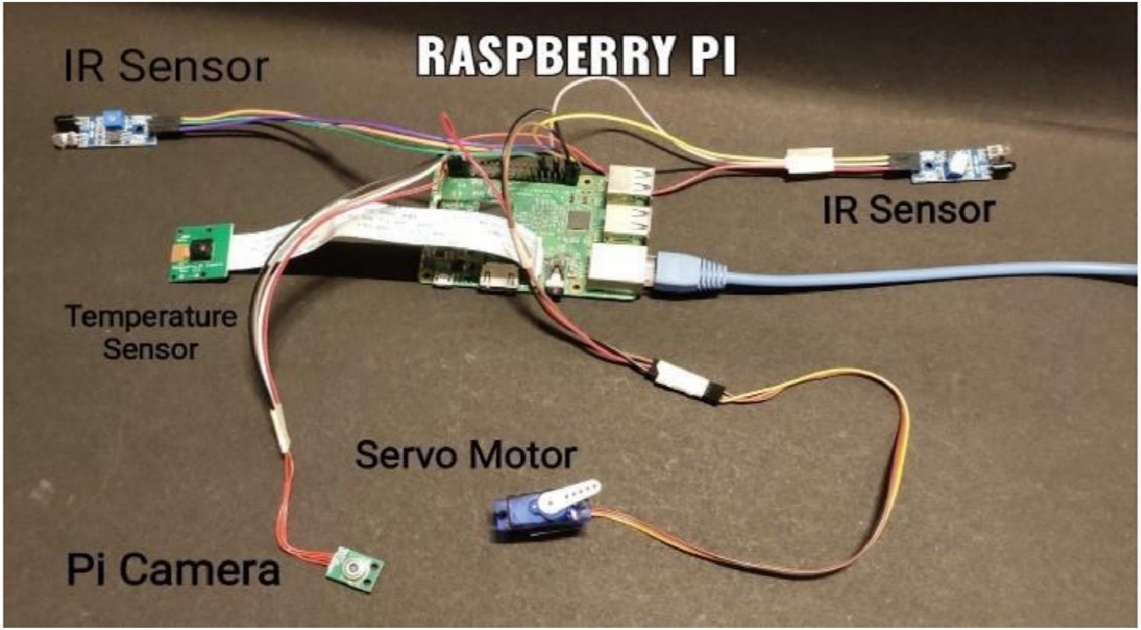


Figure 7. System Overview

1. **RESULTS AND DISCUSSION**
   1. **System Requirement**

The training was carried out on a computer running the 62-bit Windows10 operating system and equipped with an Intel ® CoreTM i5-8265U CPU running at 1.60GHz and 8 GB of RAM. Python 3.9 is being used as the application development language. The model was developed and trained using Keras as the backend and the Tensor-flow platform. To generate mask detector model input dataset and fine-tune MobileNetV2 is accepted using the training python script. Implementation of face mask detection in Raspberry Pi can be done using the mask detector model.

In this work, a Raspberry Pi 3 Model V is used. It is a low-cost, compact gadget that plugs into a computer. The Raspberry Pi is connected to the laptop via a LAN cable. The VCC of the first IR sensor has a connection with the 2nd pin on R pi, the GND pin has a connection with the 34th pin of the R pi, and the output is connected to the 40th pin of the R pi, which is a general-purpose I/O pin commonly known as GPIO pin. VCC of the second IR sensor has a connection with the 17th pin, GND to the 13th pin, and out pin to the R pi's 38th GPIO pin. The servo motor's VCC has a connection with 5V input which is the 2nd pin, next GND pin is connected to pin number 39 and the signal pin to pin number 37 which is the GPIO pin on R pi's. The MLX90614 temperature sensor's VCC pin has a connection with 1st pin on R pi, its GND has a connection to pin number 16 on R pi, and the HCL and HDL pins of the sensor are connected to the 2nd and the 3rd pins of the R pi. The R pi camera is attached to the R pi's camera module port.

* 1. **QUANTITATIVE ANALYSIS WITH DIFFERENT TEST CASE**

Optimization techniques are approaches for lowering training losses by changing the properties of neural networks such as weights and learning rate. As optimization elements were introduced in the analysis, Fig 13 depicts validity accuracy diagrams in relation to consistency and loss of validity as contrasted to loss of training. On the one hand, using more hidden layers provides a deeper analytical model, while on the other hand, each extra layer adds complexity to computing. In addition, increasing the number of neurons in each layer will increase processing costs. To enhance the number of data samples, zoom, pre-processing, shear, and other image augmentation features are frequently used. When these parameters are used, images with these qualities are generated during deep learning model training. Image samples generated using image augmentation enhanced the rate of existing data samples by almost 3x to 4x. However, this cannot be employed because the model will be strongly biased and will fail to generalize properly. The term epoch refers to the entire set of conceivable inputs. As in the case of calculating model weights after each epoch, the weights are re-adjusted and tested against the same dataset's subsequent cycle simulation (called next epoch). When this is run, the entire training data set is presumed to be in the main memory. Because it is not practical to retain the complete dataset in main memory at different periods for larger datasets. The epoch (dataset) is partitioned into batches, and each batch is loaded into the main memory and run in a sequential manner, with the findings totaled up and finally interpreted as an epoch output.

**4.3. Qualitative analysis with different algorithms**

The fundamental advantage of CNN over its predecessors would be that it automatically detects significant features without the need for human interference. As a result, CNN would be an excellent answer to computer vision and picture categorization challenges. To utilize another approach, first features from images should be created and then feed those features into a classification technique such as SVM, KNN, logistic regression, and so on. When compared to CNN, these algorithms learn less.

* 1. **Comparative analysis with existing papers**

A review of the literature finds that none of the previously published research attempted to incorporate all of the aforementioned criteria. Investigated mask detection and social distance recognition [2], but this system cannot be implemented on a Raspberry Pi due to the high processing capacity. Based on fully convolutional networks, the researchers developed a high-accuracy technique for detecting facial masks [13, 15]. However, it was not implemented on the Raspberry Pi. During the training of the CNN model, very satisfactory validation accuracy was obtained through many experiments and has a recorded accuracy of 99 percent to batch sizes fixed to 32 and 20 iterations for epochs as shown in table 1. As indicated in Fig 13 the performance testing results from visual representation through accuracy and loss. Fig 14 shows the test results on the performance of the model in detecting a person wearing a facemask with a rate of 98.55%. The face will be bounded by the green-colored rectangular box if the mask is detected. Fig 15 displays the test results on the performance of the model in detecting a person without wearing a facemask with a rate of 100%. The face will be bounded by red colored rectangular box if the mask is not detected. To check the temperature measurement system's accuracy and dependability, a mercurial thermometer is used to measure body temperature which is then compared to the temperature value of the system measure. Table 4 shows the results of the experiment. Experiment results show that the absolute measurement error is less than 0.1 C, which equals the medical body temperature monitor.

1. **CONCLUSION**

New developments and the availability of smart technologies force to the creation of new models, which will help meet the needs of developing countries. In this work, an IoT-enabled smart door is developed to monitor body temperature and detect face masks that can enhance public safety. This will help to reduce manpower while also providing an extra layer of protection against the spread of Covid-19 infection. The model uses a real-time deep learning system using Raspberry pi to detect face masks, and temperature detection as well as monitor the count of people present at any given time. The device performs excellently when it comes to temperature measurement and mask detection, the trained model was able to achieve a result of 97 percent. The test results demonstrate a high level of accuracy in detecting people wearing and not wearing facemasks, as well as it also generates alarms monitored and recorded. Furthermore, there are numerous techniques to enhance performance to improve results. Future development will include improving the accuracy of these steps, using a combination of various features, and improving performance, as well as producing a mobile app with a user friendly interface for monitoring. As a result, authorities will be able to take immediate action following pandemic safety standards.

**REFERENCES**

[1] A. Rosebrock,"COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning”, May 4, 2020, https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/. Google Scholar

[2] A. Hidayat, Subono, V.A. Wardhany, A.S. Nugroho, S. Hakim, M. Jhoswanda, “Designing IoT-Based Independent Pulse Oximetry Kit as an Early Detection Tool for Covid-19 Symptoms “, 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE). Google Scholar

[3] Carlo Alberto Boano Matteo Lasagni Kay Romer Tanja Lange. "Accurate Temperature Measurements for Medical Research using Body Sensor Networks". Google Scholar

[4] Cristina S..........-.C......, B-P BUTUNOI and Calin C......,” IoT Based Intelligent Building Applications in the Context of COVID-19 Pandemic”, 2020, 10.1109/ISETC50328.2020.9301124 Google Scholar

[5] F.H. Yahaya, Y.M. Yusoff, H.Z. Abidin, R.A. Rahman Development of a PIC-based wireless Sensor node utilizing XBEE Technology IEEE International Conference on Information Management and Engineering (2010) Google Scholar

[6] Gayatri Deora Ramakrishna Godhula and Dr. Vishwas Udpikar “Study of Masked Face Detection Approach in Video Analytics IEEE Conference on Advances in Signal Processing (2016) Google Scholar

[7] https://www.electronicwings.com/arduino/servo-motor-interfacing-with-arduino-uno. Google Scholar

[8] https://www.robocarstore.com/products/raspberry-pi-4-model-b-board-with-1gb-lpddr4-sdram. Google Scholar

[9] K Baskaran, Baskaran P., N. Kumaratharan, Rajaram V., “IoT Based COVID Preventive System for Work Environment”, the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) IEEE Xplore Part Number: CFP20OSV-ART; ISBN: 978-1-7281-5464-0. Google Scholar

[10] Lim, M.G., & Chuah, J.H. (2018). Durian Types Recognition Using Deep Learning Techniques. 2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC). doi:10.1109/icsgrc.2018.8657535. Google Scholar

[11] L. Tang and H. Kiourmars, Wireless Network for Health Monitoring Heart Rate and Temperature Sensor", 10.1109/ICSensT.2011.6137000. Google Scholar

[12] T. Meenpal, A. Balakrishnan, A. Verma Facial Mask Detection using Semantic Segmentation 2019 4th International Conference on Computing, Communications and Security (ICCCS) (2019), 10.1109/cccs.2019.8888092 View PDFGoogle Scholar

[13] Ana.nya Pa.ndey Mrudula, Kruthi.ka Dinesh, P. Reethika Smart Door Unlocking System Int. Res. J. Eng. Technol. (IRJET), Volume: 07 (2020) Issue: 05 May Google Scholar

[14] N. Ozkaya, S. Sagiroglu Intelligent face Mask Prediction System IEEE International Joint Conference on Neural Networks (2008) Google Scholar

[15] Petrovic, N. & Radenković, M. & Nejkovic, V.. (2020). Data-Driven Mobile Applications Based on AppSheet as Support in COVID-19 Crisis. https://www.researchgate.net/publication/343678867\_Data-Driven\_Mobile\_Applications\_Based\_on\_AppSheet\_as\_Support\_in\_COVID-19\_Crisis. Google Scholar

[16] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth,” SSDMNV2: A real-time DNN-based face mask detection system using single shot multibox detector andMobileNetV2”, Published online 2020 Dec 31. Google Scholar

[17] Rit.wik Biswas, Av.ijit Roy Real-Time Temperature Graph using MATLAB and Arduino Int. J. Eng. Res. Technol. (IJERT), 09 (05) (2020) (May 2020) https://www.ijert.org/real-time-temperature-graph-using-matlab-and-arduino Google Scholar

[18] Sam.my V. Militante, Na.nette V. Dionisio Deep Learning Implementation of Facemask and Physical Distancing Detection with Alarm Systems Vocational Education and Electrical Engineering (ICVEE) 2020 Third International Conference on (2020), pp. 1-5 https://ieeexplore.ieee.org/document/9232610 View PDFCrossRefView Record in ScopusGoogle Scholar

[19] S.S.. Vedaei, A. Fotovvat, M.R. Mohebbian, G.M.E. Rahman, K.A. Wahid, P. Babyn, H.R. Marateb, M. Mansourian, And Ramin Sami, “An IoT-Based System for Automated Health Monitoring and Surveillance in Post-Pandemic Life”, date of publication October 12, 2020. Google Scholar

[20] P. Subramani, G.B. Rajendran, J. Sengupta, R. Pérez de Prado, P.B. Divakarachari A block bi-diagonalization-based pre-coding for indoor multiple-input-multiple-output-visible light communication system Energies, 13 (13) (2020), p. 3466 View PDFCrossRefView Record in ScopusGoogle Scholar

[21] T. Nguyen, B.H. Liu, N. Nguyen, B. Dumba, J.T. Chou Smart grid vulnerability and defense analysis under cascading failure attacks IEEE Trans. Power Delivery (2021) Google Scholar