International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST2183123

Auto Temperature and Mask Scanning Entry System

Rubana Khan[1], Kshitij Rangari[2], Chetan Asutkar[2], Payal Kapse[2], Tejaswi Vaidya[2], Chinmay Raut[2], Bhagyashree Shambharkar[2]

[1]Assistant Professor, Department of Computer Technology, Priyadarshini College of Engineering, Nagpur, Maharashtra, India

[2]BE Scholar, Department of Computer Technology, Priyadarshini College of Engineering, Nagpur, Maharashtra, India

Article Info
Volume 8, Issue 3
Page Number : 557-562

Publication Issue
May-June-2021

Article History
Accepted : 03 June 2021
Published : 08 June 2021

ABSTRACT

Now that many shops, offices and institutions are re-opening again after the Corona lockdown, many businesses are faced with the need to provide the best possible protection for their staff and customers. Face masks and body temperature checks play an important part in the protection effort.

While this is already done routinely and at a large scale at airports or railway stations, many businesses and institutions are struggling to meet the challenge. Face mask monitoring often requires additional staff resources. At the same time, body temperature checks by staff come with certain risks in terms of hygiene and data privacy.

In this project, we introduce an affordable IoT-based solution aiming to increase COVID-19 indoor safety, covering several relevant aspects: 1) contactless temperature sensing 2) mask detection. Contactless temperature sensing subsystem relies on Arduino Uno using infrared sensor or thermal camera, while mask detection and social distancing check are performed by leveraging computer vision techniques on camera- equipped computer.

Keywords : IR Temperature Sensor, Jumping Cable, Power Supply, Arduino

I. INTRODUCTION

Since the last days of the previous year, the occurrence of novel infectious flu-alike respiratory disease COVID-19 caused by SARS-Cov-2 virus (also known as coronavirus) has affected almost every aspect of people's lives globally. First, it was discovered in China, but spread quickly to other continents in just few weeks. According to [1], until July 11th, 2020, the total number of identified cases was 12,653,451, while taking 563,517 lives worldwide. Common symptoms of coronavirus disease include fever, tiredness, sore throat, nasal congestion [2], loss of taste and smell [3]. In most cases, it is transmitted directly (person to person) through respiratory droplets, but also indirectly via surfaces [4, 5]. Incubation period could be quite long and varies (between 14 and 27 days in extreme cases) [6, 7]. Furthermore, even asymptomatic persons (almost 45% of cases) can spread the disease [7] making the situation even worse. Therefore, the usage of face masks and sanitizers has shown positive results when it comes to disease spread reduction [8].

However, the crucial problem is the lack of approved vaccine and medication [9]. Due to these facts, many protection and safety measures were taken by governments in order to reduce the disease spread, such as obligatory indoor mask wearing, Social distancing, quarantine, self-isolation, limiting citizens' movement within country boarders and abroad, often together with prohibition and cancellation of huge public events and gatherings [10]. Despite the fact that the pandemic seemed weaker at some points, most of safety regulations are still applied due to unstable situation. From workplace behavior to social relations, sport and entertainment, coronavirus disease poses many changes to our everyday routine, habits and activities. In this paper, cost- effective IoT-based system aiming to help organizations respect the COVID-19 safety rules and guidelines in order to reduce the disease spread is presented. We focus on most common indoor measures - people with high body temperature should stay at home, wearing mask is obligatory and distance between persons should be at least 1.5-2 meters. For the first scenario, Arduino Uno microcontroller1 board with contactless temperature sensor is used, while we rely on Raspberry Pi2 single-board computer equipped with camera making use of computer vision techniques for other two scenarios. We decided to use these devices due to their small size and affordability There are several existing works that contain some of the elements relevant to the work presented in this paper. However, to the best of our knowledge, there is no such solution covering all these aspects together to achieve this goal while allowing execution on low-cost IoT devices at the same time. In [17], a dataset for masked face recognition is introduced and its application by different algorithms in context of campus and enterprise coronavirus prevention discussed. Moreover, in [18], a high-accuracy method for facial mask detection using semantic segmentation based on fully convolutional networks, gradient descent and binomial cross entropy was presented. However, performance-wis , it is too heavy for low-power IoT devices, such as Raspberry Pi. On the other side, in [19], a state model-based solution for face mask detection relying on Viola-Jones algorithm in context of ATM center security was described. When it comes to temperature sensing, there are several variants of Arduino-based solutions. In [20], Arduino was used for real-time temperature visualization using MATLAB. However, the used sensor does not allow contactless temperature sensing. Moreover, in [21], a similar system incorporating the usage of smartphones for remote temperature monitoring using Arduino Uno was presented. The system architecture presented in this paper is inspired by our previous work on remote smart grid monitoring anomaly, power consumption monitoring and relay protection using IoT devices (smartphones and Arduino Uno) [16] and video surveillance system relying on Raspberry Pi single board computers and Edge servers [22]. Our main goal is to provide a comprehensive solution for COVID-19 safety monitoring which relies on IoT devices as much as possible in order to be affordable at the same time.

II. RELATED WORK

To solve these problems, we are working on that automates the process of Temperature Checkup by using Facial Land marking & Contactless IR Temperature Sensor and Mask Detection using Deep Learning Neural Network.

This system not only detects human temperature but also scans persons wearing face mask or not. When a passer without wearing face mask is approaching to the camera sensor, display shows the body temperature and sounds "ATTENTION, MASK NOT DETECTED" warning to remind the person detected at the same time. Of course the setting of "Wearing mask or not " can be based on officer's preference to make an adjustment. If a potential person is close to fever temperature and exceeds the specific temperature is detected, camera will make a quick response and will sound "WARNING, TEMPERATURE OUT OF RANGE" to inform officer by alarm message.

Moreover, there is another powerful function cooperating with body temperature detection, "Face Recognition with wearing face mask". Through the wide angle of lens, face captured by the camera will be instantly processing face matching with internal database. Whether wearing face mask, camera can still recognize to identify the personnel identities.
Here is a video demonstration divided into 3 parts regarding on body temperature detection with face recognition, face mask detection and over heat with smart detection below.

In these tough COVID-19 times, wouldn't it be satisfying to do something related to it? I decided to build a very simple and basic Convolutional Neural Network (CNN) model using TensorFlow with Keras library and OpenCV to detect if you are wearing a face mask to protect yourself. Interesting! Isn't it?
For building this model, I will be using the face mask dataset provided by Prajna Bhandary. It consists of about 1,376 images with 690 images containing people with face masks and 686 images containing people without face masks.

I am going to use these images to build a CNN model using TensorFlow to detect if you are wearing a face mask by using the webcam of your PC. Additionally, you can also use your phone's camera to do the same!

Step 1: Data Visualization
In the first step, let us visualize the total number of images in our dataset in both categories. We can see that there are 690 images in the 'yes' class and 686 images in the 'no' class.

The number of images with facemask labelled 'yes': 690 The number of images with facemask labelled 'no': 686

Step 2: Data Augmentation In the next step, we augment our dataset to include more number of images for our training. In this step of data augmentation, we rotate and flip each of the images in our dataset. We see that, after data augmentation, we have a total of 2751 images with 1380 images in the 'yes' class and '1371' images in the 'no' class.
Number of examples: 2751 Percentage of positive examples: 50.163576881134134%, number of pos examples: 1380 Percentage of negative examples: 49.836423118865866%, number of neg examples: 1371

Step 3: Splitting the data In this step, we split our data into the training set which will contain the images on which the CNN model will be trained and the test set with the images on which our model will be tested.
In this, we take split_size =0.8, which means that 80% of the total images will go to the training set and the remaining 20% of the images will go to the test set.
The number of images with facemask in the training set labelled 'yes': 1104 The number of images with facemask in the test set labelled 'yes': 276

The number of images without facemask in the training set labelled 'no': 1096 The number of images without facemask in the test set labelled 'no': 275

After splitting, we see that the desired percentage of images have been distributed to both the training set and the test set as mentioned above

Step 4: Building the Model
In the next step, we build our Sequential CNN model with various layers such as Conv2D, MaxPooling2D, Flatten, Dropout and Dense. In the last Dense layer, we use the 'softmax' function to output a vector that gives the probability of each of the two classes.

Here, we use the 'adam' optimizer and 'binary_crossentropy' as our loss function as there are only two classes. Additionally, you can even use the MobileNetV2 for better accuracy.

Step 5: Pre-Training the CNN model
After building our model, let us create the 'train_generator' and 'validation_generator' to fit them to our model in the next step. We see that there are a total of 2200 images in the training set and 551 images in the test set.
Found 2200 images belonging to 2 classes. Found 551 images belonging to 2 classes.

Step 6: Training the CNN model
This step is the main step where we fit our images in the training set and the test set to our Sequential model we built using keras library. I have trained the model for 30 epochs (iterations). However, we can train for more number of epochs to attain higher accuracy lest there occurs over-fitting.
history = model.fit_generator(train_generator, epochs=30, validation data=validation_generator, callbacks=[checkpoint]) >>Epoch 30/30 220/220 [==============================] - 231s 1s/step - loss: 0.0368 - acc: 0.9886 - val_loss: 0.1072 - val_acc: 0.9619

We see that after the 30th epoch, our model has an accuracy of 98.86% with the training set and an accuracy of 96.19% with the test set. This implies that it is well trained without any over-fitting.

Step 7: Labeling the Information
After building the model, we label two probabilities for our results. ['0' as 'without mask' and '1' as 'with mask']. I am also setting the boundary rectangle color using the RGB values.['RED' for 'without mask' and 'GREEN' for 'with mask]
labels_dict={0:'without_mask',1:'with_mask'}
color_dict={0:(0,0,255),1:(0,255,0)}
Step 8: Importing the Face detection Program

After this, we intend to use it to detect if we are wearing a face mask using our PC's webcam. For this, first, we need to implement face detection. In this, I am using the Haar Feature-based Cascade Classifiers for detecting the features of the face.
face_clsfr=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

This cascade classifier is designed by OpenCV to detect the frontal face by training thousands of images. The .xml file for the same needs to be downloaded and used in detecting the face. I have uploaded the file in my GitHub repository.

Step 9: Detecting the Faces with and without Masks
In the last step, we use the OpenCV library to run an infinite loop to use our web camera in which we detect the face using the Cascade Classifier. The code webcam = cv2.VideoCapture(0)denotes the usage of webcam.

The model will predict the possibility of each of the two classes ([without_mask, with_mask]). Based on which probability is higher, the label will be chosen and displayed around our faces.

For evaluation, the following, the following devices were Raspberry Pi 3 (RPi 3) and Arduino Uno sensor and thermal camera. In Table I, the performance evaluation for various scena settings are given. The first column denote scenario (mask detection, distance check, temperature measurement). The second c hardware configuration used in the experi third column is the frame size expressed a horizontal multiplied by vertical pixels. Fu column shows performance results achiev configuration expressed as number of proc or measurements per second (mps). Final

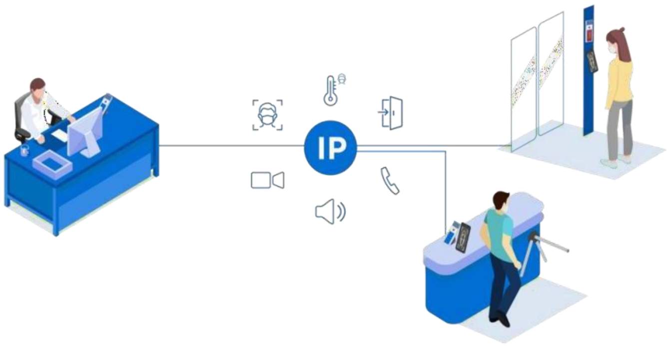

Figure 1. Automated Temperature and Face Detection

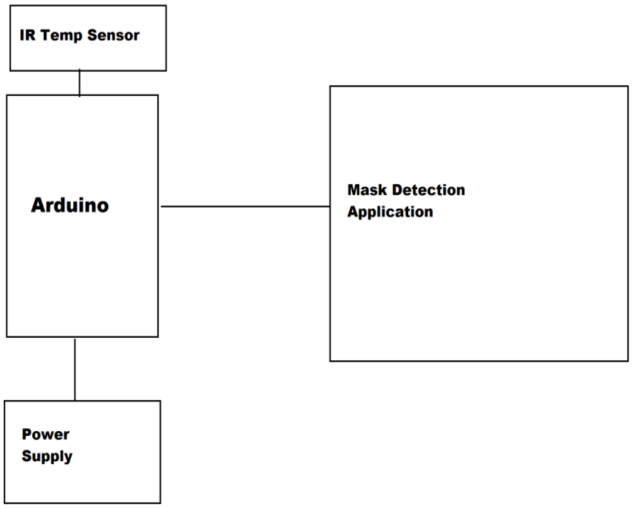

Figure 2. Block Diagram

III. CONCLUSION

According to the achieved results, the proposed solution is usable for its purpose under certain performance limitations (such as number of processed frames or measurements per second). Moreover, it relies on both open hardware and free software, being definite and desirable advantage for such systems. In future, it is planned to experiment with various deep learning and computer vision frameworks for object detection on Computer in order to achieve higher frame rate. Moreover, we would like to extend this solution with environment sensing mechanisms for adaptive building air conditioning and ventilation airborne protection in order to reduce the spread of coronavirus indoors, especially during summer. Finally, the ultimate goal is to integrate the system presented in this paper with our framework for efficient resource planning during pandemic crisis in order to enable efficient security personnel.

IV. REFERENCES

[1]. Dawson et al., "Loss of Taste and Smell as Distinguishing Symptoms of COVID-19", Clinical Infectious Diseases June 2020, pp. 1-4, 2020. https://doi.org/10.1093/cid/ciaa799[4] L. Morawska, "How can airborne transmission of COVID-19 indoors be minimized?", Environment International vol. 142, September 2020, 105832, pp. 1-7, 2020. https://doi.org/10.1016/j.envint.2020.105832[1] Coronavirus Update (Live) [online]. Available on : https://www.worldometers.info/coronavirus/, last accessed: 11/07/2020.

[2]. P. Zhai et al., "The epidemiology, diagnosis and treatment of COVID19", International Journal of Antimicrobial Agents vol. 55 issue 5, May 2020, 105955, pp. 1-13, 2020. https://doi.org/10.1016/j.ijantimicag.2020.105955

[3]. P. Dawson et al., "Loss of Taste and Smell as Distinguishing Symptoms of COVID-19", Clinical Infectious Diseases June 2020, pp. 1-4, 2020. https://doi.org/10.1093/cid/ciaa799

[4]. L. Morawska, "How can airborne transmission of COVID-19 indoors be minimized?", Environment International vol. 142, September 2020, 105832, pp. 1-7, 2020. https://doi.org/10.1016/j.envint.2020.105832

[5]. T. Galbadage, B. Peterson, R. Gunasekera, "Does COVID-19 Spread Through Droplets Alone?", Frontiers in Public Health, vol. 8, April 2020, pp. 1-4, 2020. https://doi.org/10.3389/fpubh.2020.00163[6] Coronavirus Incubation Period [online].

Available on: https://www.worldometers.info/coronavirus/coronavirusincubationperiod/ , last accessed: 11/07/2020.

[6]. D. Oran, E. Topol, Prevalence of Asymptomatic SARS-CoV-2 Infection: A Narrative Review, Annals of Internal Medicine, June 2020, pp. 1-7, 2020. https://doi.org/10.7326/M20-3012

[7]. T. Dbouk, D. Drikakis, “On respiratory droplets and face masks”, Physics of Fluids 32, 063303, pp. 1-11, 2020. https://doi.org/10.1063/5.0015044

[8]. Y. Song et al., “COVID-19 Treatment: Close to a Cure? – A Rapid Review of Pharmacotherapies for the Novel Coronavirus” [preprint], pp. 1-25, 2020. https://doi.org/10.20944/preprints202003.0378.v1

[9]. V. Balachandar et al., “COVID-19: emerging protective measures”, European Review for Medical and Pharmacological Sciences vol. 24 (6), pp. 3422-3425, 2020. https://doi.org/10.26355/eurrev_202003_20713

[10]. Open Computer Vision [Online]. Available on: https://opencv.org/ , last accessed: 07/07/2020. 8th Student Projects Conference, Niš, Serbia, pp. 21-24,

Cite this article as :

Rubana Khan, Kshitij Rangari, Chetan Asutkar, Payal Kapse, Tejaswi Vaidya, Chinmay Raut, Bhagyashree Shambharkar, " Auto Temperature and Mask Scanning Entry System", International Journal of Scientific Research in Science and Technology(IJSRST), Print ISSN : 2395-6011, Online ISSN : 2395-602X,Volume 8, Issue 3, pp.557-562, May-June-2021. Available at doi : https://doi.org/10.32628/IJSRST2183123
Journal URL : https://ijsrst.com/IJSRST2183123