EPiC Series in Computing
Volume 81, 2022, Pages 111–120
Proceedings of 11th International Congress on Advanced Applied Informatics

Smart Early Screening System for COVID-19 Spreading Prevention

Annisa Istiqomah Arrahmah[1] and Surya Ramadhan[2]
[1] Bina Nusantara University Bandung Campus, Jakarta, Indonesia
[2] Institut Teknologi Bandung, Bandung, Indonesia
annisa.arrahmah@binus.ac.id, surya_ramadhan@students.itb.ac.id

Abstract

Since the end of 2019, a respiratory disease called COVID-19 caused by SARS-COV2 has been spread around the world. The disease has similar symptoms with influenza. The common symptoms are cough, fever and fatigue. The human-to human transmission occurs primarily through droplets spread by coughing or sneezing from infected people directly and indirectly. In this paper a system based on embedded devices that can be used to help prevent the spread of COVID-19 between humans through face mask detection and a contactless thermal sensor is proposed. CNN based deep learning for the facemask detection and IR thermal sensor for non-contact human temperature measurement are used. The system is implemented locally on the Raspberry Pi platform. The training result shows the accuracy of the face mask detector is higher than 90% and stable after epoch 2. The thermal sensor shows the stable input with 0.25 deviation.

1 Introduction

COVID-19 is a respiratory disease caused by SARS-COV2 virus (a member of the coronavirus family) that infected humans for the first time in Wuhan, Hubei Province, China at the end of 2019 [1]. According to CSSE (Center for Systems Science and Engineering) at Johns Hopkins University, until November 2021, there are 254.562.905 global cases with 5.118.632 global deaths [2]. This spreading is more massive than the previous disease caused by the coronavirus family with daily increase cases around 2 % and 4.2% mortality rate [3]. The characteristic of COVID-19 disease is reportedly similar with the earlier disease caused by beta-coronavirus. The symptoms can be variated; severe, mild and asymptomatic. Common symptoms include dry cough, fever, fatigue, sore throat and even diarrhea. In some cases, loss of smell or pneumonia may occur. Whereas the symptoms occur approximately 5.2 days of incubation period after the virus infection. Human to human transmission happens commonly from direct contact through the droplet of infected people by sneezing or coughing. In some cases, the transmission can occur indirectly through a surface that has droplet trails of the infected people. Starting from 11 March 2020, World Health Organization (WHO) has categorized the COVID-19 as the global pandemic. At the end of 2020, COVID-19 vaccine has been successfully developed by several countries with different levels of efficacy. Although at the beginning of 2021 the vaccines are starting to be distributed around the world, the distribution is still uneven through region and group of ages. Thus, preventing the spread of this disease is still important.

The World Health Organization, as the global health organization, has issued regulations to flatten the COVID-19 spreading curve. This regulation consists of limiting the social gathering, avoiding close contact (social distancing), cleaning the hands regularly, obligatory to wear a face mask, measuring the body temperature regularly, and self-isolation (quarantine) for the infected people. This regulation should be executed by all countries around the world in daily life and eventually change the daily routine and activities of human life.

This paper describes the development of a cost-effective smart early screening system for COVID-19. The system is implemented in Raspberry Pi V3 embedded system. The aim of this system is to help personnel and organizations prevent the spread of coronavirus pandemic based on the regulation stated by WHO. The system focuses on early screening by detecting the infected people based on the common symptoms of COVID19, i.e. fever, using contactless thermal sensor. Another purpose of this system is to restrain the spread of the coronavirus by screening the face mask obligatory in the public space using face mask detection based on computer vision. The face mask detection uses convolutional neural network (CNN). The computation for the system is done locally in Raspberry Pi.

2 Basic Theories

2.1 Convolutional Neural Network (CNN)

One example of a feedforward neural network is convolutional neural network. CNN is mimicking the architecture and functionality of neurons in a human brain [4]. In CNN, the neurons are split into a 3-dimensional structure. Each neuron examines a small feature or part of a two-dimensional image. The output of CNN is a likelihood probability for a specific feature from a certain class [5]. CNN common layers can be seen in Figure 1. ImageNet is a part of a computer vision project based on CNN deep learning.

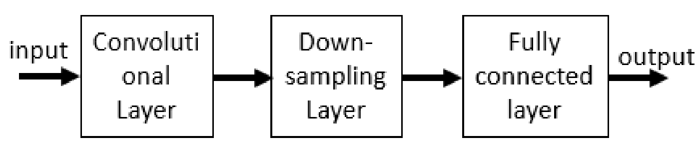

Figure 1: Common layers in CNN

2.2 Mobile Net V2 Architecture

Mobile NetV2, refinement of mobile NetV1 is a CNN based architecture with highly effective performance for numerous tasks introduced by Google [6]. This architecture is specified for neural network implementation in a limited resource or environment such as an embedded system. It is stated that the architecture is intended for computer vision applications in a mobile device. The architecture uses a layer module with linear bottleneck in an inverted residual. The important block on this architecture is depth-wise separable convolution, replacing convolutional layers because of its inexpensive computation. Figure 2 represents the main building block of this module. The implementation is available in the TensorFlow-Slim model library.

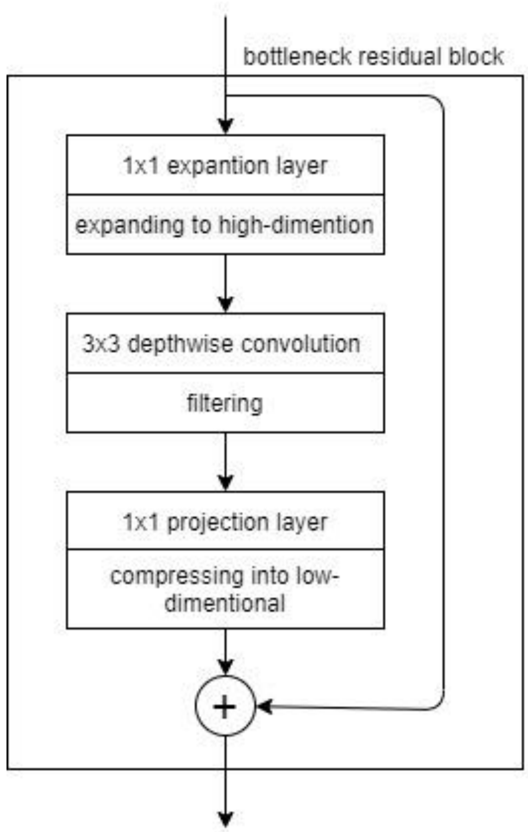

Figure 2: Main block in Mobile Net V2 architecture

2.3 HOG and SVM based Face Detection

Histogram of Oriented Gradient (HOG) is a feature descriptor used in Computer Vision proposed by Dalal and Triggs originally applied for human detection [7]. This technique inspects the shape and the structure of an object by counting the occurrence of gradient orientation in the region of interest in the input image. The HOG method can also be utilized for face detection by inspecting the basic contour of the human face. The method of the face detection using HOG is summarized in Figure 3. At the first step, the input image is converted into a grayscale image. Then, the input is divided into grids with 16x16 pixels for each grid. For each grid, the gradient point for the major direction is calculated and the strongest one is chosen, resulting the HOG representation of the image. To find the face in the image, the collected HOG descriptors are compared with the known HOG pattern that was extracted from face data training using the SVM classifier. Thus, the face localization is obtained.

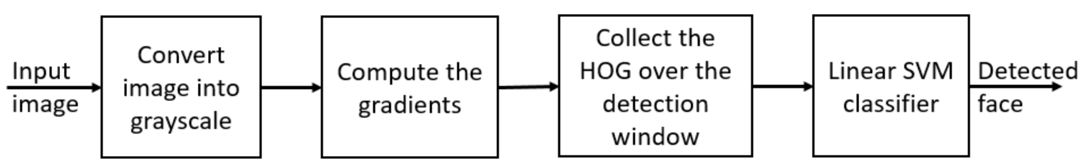

Figure 3: Face detection with HOG and SVM

3 Related Works

There are several works, related to this paper and another paper with specific reasons helping COVID-19 prevention by utilizing information technology. In [8], several significant applications using artificial intelligence and IoT are collected and stated. In [9] – [11] face mask detection is done using several deep learning algorithms. In [9] an IoT system is introduced, but the face mask classifier was done on the server side. In [9] – [11] IoT based system are also introduced to monitor and detect earlier COVID-19 symptoms using mobile applications. In [12] – [13], the author only focuses on face mask detection implementation. Our main focus of this system is to help prevent the spread of COVID-19 by utilizing embedded system platforms, therefore the cost can be reduced. This system is also inspired from the previous work.

4 Design and Implementation

As stated in the earlier section, the smart early screening system uses Raspberry Pi V3 as the main processing board. A Raspberry Pi V3 Camera Module is used as an input for the face mask detection. IR thermal sensor is applied for the contactless temperature measurement. The overview of the system is shown in Figure 4. The system is deployed in front of the portal; indoor or outdoor. While a passenger is going through the portal, the system will screen the body temperature and give a warning if the person's body temperature is higher than normal, indicating the person has a fever. The system also screens the passengers face mask status using a Raspberry Pi camera. The face mask is detected using real-time computer vision. The output is displayed through LCD and checked by the portal security guard. Both of the computation is done locally in the Raspberry Pi.

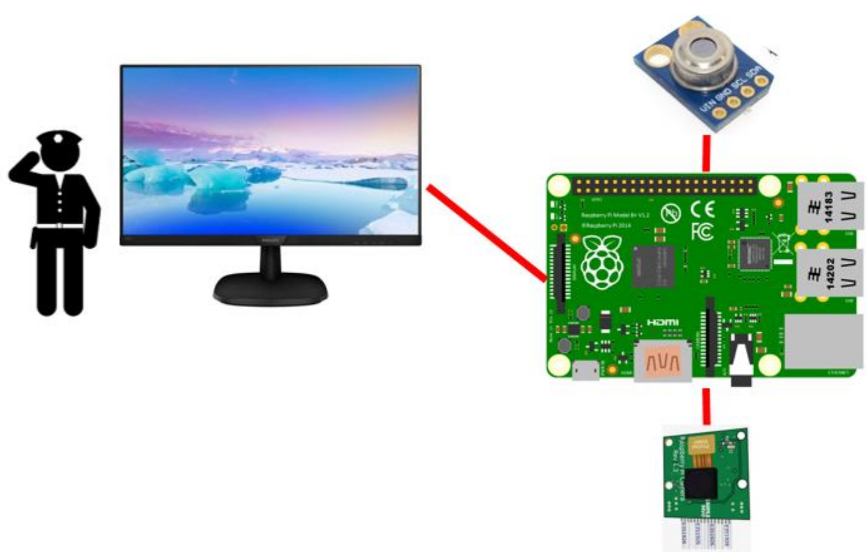

Figure 4: Overview of the screening system

4.1 Contactless Human Body Temperature Measurement

In this paper, MLX90614 infrared temperature sensor is used for non-contact human body temperature. The sensor communicates to the Raspberry board through the I2C interface. The schematic of the human body temperature measurement can be seen in Figure 5. Two pull up 4.7 KΩ resistor are connected to I2C bus. Each on SDA (Serial Data) and SCL (Serial clock) signals pin, respectively. A 0.1 µF capacitor is connected between source and ground to stabilize the input and output voltage.

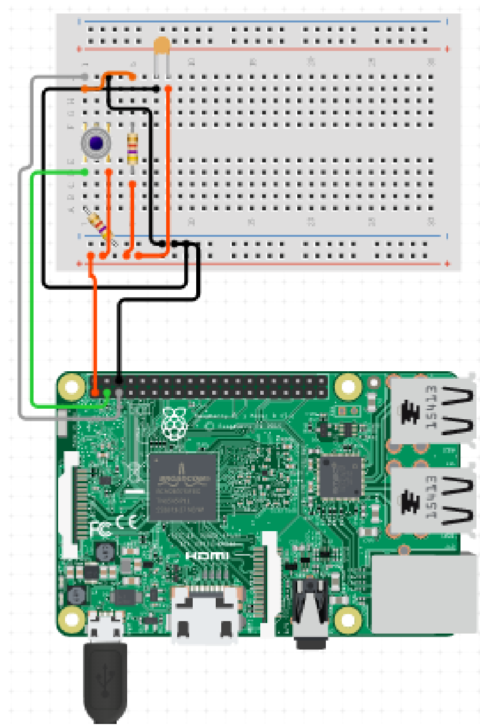

Figure 5: Schematic of contactless body temperature measurement

4.2 Face Mask Data preparation

The face mask dataset is collected manually by searching the face image through the internet, then an artificial mask is added to the face image using computer vision. Thus, 'no mask' dataset and artificial 'with mask' dataset are obtained. The process of making the 'with mask' dataset can be seen in Figure 6. The process is done by utilizing the facial landmark library dlib from OpenCV. By facial land-marking, the line between chin and nose can be pointed out. Then, three different face mask images are placed based on this line. 500 face images for 'no mask' dataset and 500 'with mask' dataset is created using this algorithm. The input face mask images vary from male to female, with vail, with mustache and bread, long and short hair, with glasses, and etc. By applying this, the accuracy of the classifier is increasing. Data augmentation is also implemented to randomly zoom, rotate, shifting and flipping the original dataset to add more training data.

4.3 Face Mask Classifier Modeling

Applying face mask detection in Raspberry Pi requires two stages of implementation. The first one is training and validating the face mask data using the dataset that is already labeled. The next stage is implementing or testing the face mask detection. Because the testing process is done in a constrained environment, an effective architecture should be used to reduce the run time. MobileNetV2 convolutional neural network architecture available in Keras/Tensorflow is used to train the face mask classifier in Python language.

Figure 7 shows the training process of the face mask classifier. First, the Mobile V2 Architecture in Keras/Tensorflow is tuned using pre-trained ImageNet weights, therefore the face detection based on CNN can be implemented effectively in Raspberry Pi V3. Scikit-learn is used to label a class, segment a dataset and show the classification report in probability. For the deep learning hyperparameter, the learning rate is $1e^{-4}$, the number of training epochs is 20 and the batch size is 32. The input data image is preprocessed before inserted to the training module. 80% dataset for training and 20% dataset for validating is used. Evaluation of the face mask classifier model is done in the form of accuracy and loss metrics.

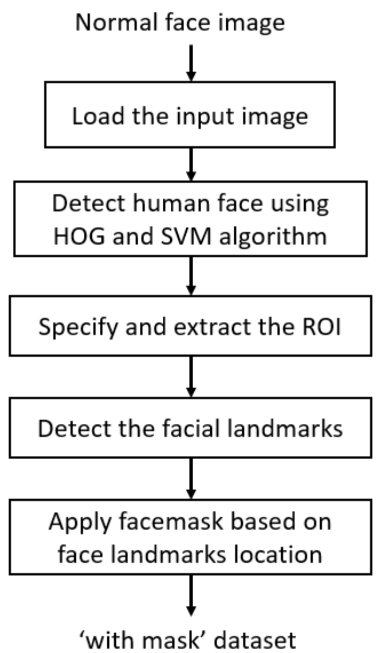

Figure 6: Face mask data preparation

As mentioned before, MobileNet architecture is used while performing deep learning. Pre-trained ImageNet weights are used for the head of the network. Then, a fully connected head is constructed and added to the old head. During backpropagation, the weights of the base layer are not updated and the weights of the head layer are updated.

4.4 Face Mask Classifier Implementation

Implementing the face mask classifier for face mask detection is done using OpenCV, Tensorflow, Keras and Numpy libraries. A video stream as an input is converted into image sequence to process inference in every image frame to perform real-time detection. Each frame is evaluated to detect a face and classify the face between mask and no mask face label. The process is shown in Figure 8.

At the first stage, a preprocessing is done to normalize the size of the input image. Next, a pre-trained Deep Neural Network (DNN) based model, built-in module in OpenCV, is used to detect and localize the face contour in the image. As a result, a bounding box is created around the face and claimed as the region of interest (ROI) for the face mask classifier. A face mask classifier model, obtained from the modeling process in the training stage, is applied to the ROI to classify the face between mask and no mask class.

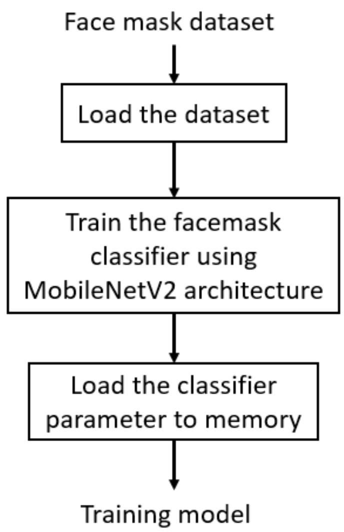

Figure 7: Face mask classifier training stage

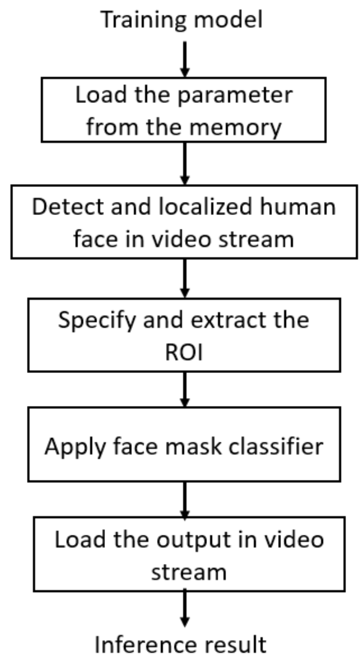

Figure 8: Face mask classifier implementation stage

5 Results and Discussion

5.1 Contactless Body Temperature Measurement

Figure 9 shows the graphic of human body temperature measured using the IR thermal sensor in t times. From the graphic, it can be seen that the temperature is stable enough with the deviation 0.25. Figure 12 shows the output of the body temperature sensor implementation on the system.

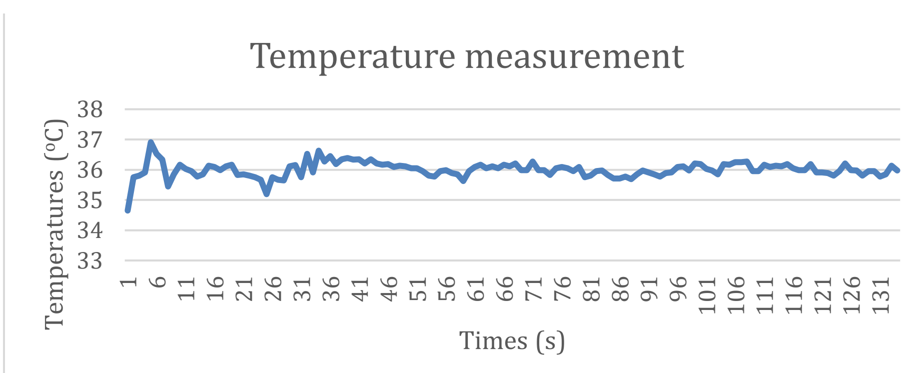

Figure 9: Temperature measurement

5.2 Face Mask Data Set

For the dataset, the ‘with mask dataset is artificially made. Figure 10 shows the example of the ‘no mask’ dataset and artificial ‘with mask’ dataset after using the facial landmark algorithm. The training is done in the computer to reduce the time computation, then the model is imported to the Raspberry Pi V3.

Figure 10: Example of the ‘with mask’ dataset with three different masks

5.3 Face Mask Detection

Figure 11 shows the testing output of the face mask system. From the figure, it can be seen that the face mask detection works correctly. Figure 11 shows the evaluation result of the face mask classifier training process with 80% data for training and 20% data for validation.

Figure 11: The example of face mask detection with body temperature output

Figure 12: Accuracy and training loss evaluation

Based on the result, the accuracy of the face mask classifier model is higher than 90% and becomes stable between data train and validation after epoch 5. Meanwhile, the training and validation loss are less than 5%, and become stable at epoch 7.

6 Conclusion

According to the characteristics of COVID-19 pandemic, several regulations were made by the government to prevent the spreading of COVID-19 pandemic. An early screening system based on the recommended regulation using IR thermal sensor for the contactless temperature measurement and face mask detection are developed. The thermal sensor is shown to have a stable input. The face mask detection also has been implemented in the Raspberry Pi using a face detector and face mask classifier. The training result shows the accuracy is higher than 90% and stable after epoch 2.

References

[1] H. Lu, C. W. Stratton, and Y. W. Tang, “Outbreak of pneumonia of unknown etiology in Wuhan, China: The mystery and the miracle,” 2020.

[2] "*COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)*," Nov, 2021. Accessed on: Nov. 17, 2021. [Online]. Available: https://www.arcgis.com/apps/dashboards/bda7594740fd40299423467b48e9ecf6

[3] H. A. Rothan and S. N. Byrareddy, “The epidemiology and pathogenesis of coronavirus disease (COVID-19) outbreak,” J. Autoimmun., vol. 109, no. February, p. 102433, 2020, doi: 10.1016/j.jaut.2020.102433.

[4] B. Kwolek, "Face detection using convolutional neural networks and gabor filters," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 3696 LNCS, pp. 551–556, 2005, doi: 10.1007/11550822_86.

[5] L. Haoxiang and Lin, "A convolutional neural network approach for face identification," Proc. CVPR, pp. 5325–5334, 2015, [Online]. Available: http://www.cs.nyu.edu/~xd283/face2012.pdf.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[7] D., Navneet, and B. Triggs. "Histograms of oriented gradients for human detection." IEEE computer society conference on computer vision and pattern recognition (CVPR'05), pp.886-893, 2005, doi: 10.1109/CVPR.2005.177.

[8] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of things (IoT) applications to fight against COVID-19 pandemic," Diabetes Metab. Syndr. Clin. Res. Rev., vol. 14, no. 4, pp. 521–524, 2020, doi: 10.1016/j.dsx.2020.04.041.

[9] N. Petrovic and D. Kocic, "IoT-based System for COVID-19 Indoor Safety Monitoring," IcETRAN 2020, no. September, 2020, [Online]. Available: https://www.researchgate.net/publication/343231422_IoTbased_System_for_COVID-19_Indoor_Safety_Monitoring.

[10] M. Otoom, N. Otoum, M. A. Alzubaidi, Y. Etoom, and R. Banihani, "An IoT-based framework for early identification and monitoring of COVID-19 cases," Biomed. Signal Process. Control, vol. 62, no. July, p. 102149, 2020, doi: 10.1016/j.bspc.2020.102149.

[11] R. Vaishya, M. Javaid, I. H. Khan, and A. Haleem, "Artificial Intelligence (AI) applications for COVID-19 pandemic," Diabetes Metab. Syndr. Clin. Res. Rev., vol. 14, no. 4, pp. 337–339, 2020, doi: 10.1016/j.dsx.2020.04.012.

[12] A. Rosebrock, "Deep Learning for Computer Vision with Python", PyImageSearch.Inc. London.

[13] A. Rosebrock, "Raspberry Pi for Computer Vision", PyImageSearch.Inc. London