Journal of Computing and Applied Informatics (JoCAI) Vol. 6, No. 2, 2022 | 125 – 137

DATA SCIENCE
Journal of Computing and Applied Informatics

Detection of the Use of Mask to Prevent the Spread of COVID-19 Using SVM, Haar Cascade Classifier, and Robot Arm

Andini Pratiwi[1], Erna Budhiarti Nababan[2], Amalia[3]

[1,2,3] Faculty Of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia

Abstract. Supervision is needed to hold up the case spread of COVID-19's growth rate by implementing health protocols such as using masks, especially for people who have not or still have problems wearing masks. In this research, the system utilizes robotic power to identify whether visitors are wearing masks and automatically distribute masks if the user is detected as not wearing a mask. The user face detection process uses the Haar Cascade Classifier algorithm to crop the image, focus on the face area, and SVM (Support Vector Machine) to classify users who wear masks or not. For the user who is detected as not wearing masks, myCobot-Pi, with the support of a suction pump, will distribute masks to users. Using myCobot-Pi as a raspberry pi based robotic arm allows the application of the system on minimal specifications and size devices. Through trials by taking 41 examples of detection cases, 29 cases were found that managed to detect the correct use of masks. In addition, in this study, we use a PP sheet plastic protector to replace the mask packaging because it can be carried by the suction pump properly.

Keywords: Mask, COVID-19, MyCobot-Pi, SVM, Haar Cascade Classifier, Raspberry Pi

Received 29 July 2022 | Revised 09 August 2022 | Accepted 09 August 2022

1 Introduction

COVID-19 (Corona Virus Disease 2019) is a type of infectious disease through the proliferation of variants of the coronavirus. This virus has spread and has been known to infect humans since December 2019. This disease has been categorized as a pandemic. The rapid and widespread disease gets special attention, and solutions are always intensively sought to overcome this disease. This disease infects the human respiratory tract. Most of the spread occurs through droplets, direct contact with

objects contaminated with the virus, and can even spread from the air.

*Corresponding author at: Faculty Of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia

E-mail address: ernabrn@usu.ac.id

ISSN: 2580-6769 | e-ISSN: 2580-829X | DOI: https://doi.org/10.32734/jocai.v6.i2-9289
Journal Homepage: https://jocai.usu.ac.id.

There are several health protocols and recommendations by the Indonesian government to tackle the growth of the spread of this disease. The health protocols are wearing masks, physical distancing, social distancing, washing hands before and after activities, and strengthening the body's immunity, such as getting a complete vaccine. Moreover, consciously self-isolate if someone feels they have symptoms of COVID-19 or has just had physical contact with a patient diagnosed as positive for COVID-19 or have themselves tested positive for COVID-19 [1].

A mask can prevent the transmission of COVID-19 disease because it can protect people who have not been infected with COVID-19 from people who are already infected but without symptoms. The use of a mask also can prevent the droplet transmission process [2]. However, people still do not obey wearing masks, especially when doing activities outside the home and meeting many people. Although now has been an easing of the rules for the use of masks, wearing masks, especially in public transportation or in crowds, must still be implemented. Therefore, we need a monitoring system to check whether people were wearing a mask or not on a massive scale, for example, at attendance or checking at the entrance of a place.

In reality, there are still gaps in efforts to make it effective. The possibility of gaps that occur, such as visitors who are missed from manual checking when the supervisor is not in place, and there is still human contact between visitors and supervisors, especially when distributing masks to visitors who do not wear masks. Robotic assistance is needed to reduce human contact and detect the use of masks in the face detection process using machine learning algorithms SVM (Support Vector Machines) and Haar Cascade Classifier.

2 Related Works

Several kinds of research have been carried out related to the provision of user needs automatically and the selection of the distribution of masks. In their study, Sanjaya et al. [3] successfully detected people's faces using the MobileNetv2 image classification method for 25 cities in Indonesia and concluded that the five cities were the highest and the five lowest cities in the application of the use of masks in these cities. The detection process is integrated with surveillance cameras to check people who are not wearing masks. By getting reference ideas regarding the use of machine learning and surveillance cameras for the face mask detection process from the previous research above, we try to use a different type of camera, the myCobot Camera Flange, and different methods.

In other research, Henderi et al. [4] utilized a camera connected to Sipeed Maix Bit. The study can efficiently implement the Python programming language and image classification using the CNN (Convolutional Neural Network) method, which can detect whether people wear masks by displaying the results on the LCD screen. Concerning the research above, in addition, to classifying users wearing a mask or not, we develop the system with additional actions. The system will deliver a mask to users who are detected not wearing masks.

In addition to the combination of the use of CNN and MobileNetv2, through research conducted by Marwansyah et al. [5] regarding the detection of the use of face masks through the use of machine learning, we can also use the Haar Cascade Classifier algorithm to assist the face cropping process. It aims to get an image that is part of the face only. From this study, we use it as a reference for using the Haar Cascade Classifier in the pre-process, precisely in the dataset collection process.

The previous research by Jie et al. [6] utilized a 6-DOF robotic arm with a Blynk wireless controller interface that can detect object colors and take sorting actions to pick up and place objects into certain positions according to their color. The robot arm exists as an embodiment of industrial needs. By adopting the same principle through a 6-DOF robotic arm, we will try to use it for the mask distribution process as the output action of the face mask detection process by changing the use of the gripper to a pump for optimizing object retrieval, as well as utilizing the SVM algorithm assisted by the Haar Cascade Classifier, instead of using the methods that already available in the Blynk app.

Lastly, in the research conducted by Tahtawi et al. [7], the 3-DOF robot arm is made in a miniature version adapting its development process in the laboratory, completing the task of carrying and placing objects based on the given coordinates. With the inverse kinematics method, a trial was conducted to obtain the results of an average error between the given target coordinates and the actual movement of 3.7%. This research can be used as a reference regarding inaccuracies between calculations and computations. However, this study is only used as a reference for point tolerance in defining the position of the robot arm.

3 Methodology

Figure 1 shows the general architecture of the proposed system.

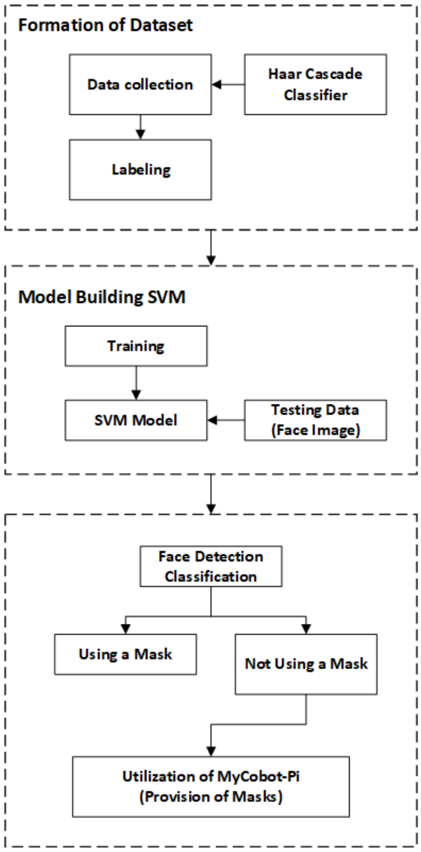


Figure 1 General Architecture

3.1 Formation of Dataset

A. Data Collection

Forming the dataset begins with receiving the input of image data. The image data is the image of the user's face which will later be grouped into two types, the face image when wearing a mask and the face image without wearing a mask. For the first data collection, 200 images of users using masks were taken, and then 200 images of users without masks. Each of which was saved in a different file. All the photos were taken using a camera by running a data capture program in python. Face detection was taken using the references of Haar Cascade Classifier, utilizing the "CascadeClassifier" and "detectMultiScale" functions from the python library openCV to precisely capture images with focus frame only in the face area. The area for the face is resized to 50 x 50 pixels. The faces were captured by detecting the dark and light pixel areas as the average intensity of the area (Figure 2).

Figure 2 Face Area Detection with Haar Cascade Classifier

In the slicing process of cutting the face area of a frame, the calculation of the x and y coordinates starts from the top left (Figure 3). The x and y coordinate values start from the top left corner of the face detected area box, then add the w and h values or by definition through rows and columns, rows are y: y+h and columns are x: x+h.

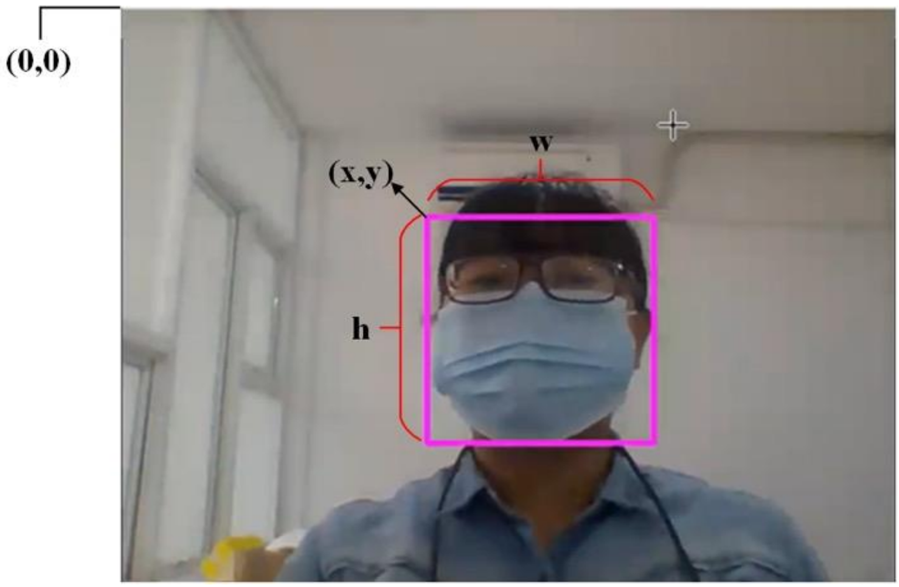


Figure 3 Positioning in Slicing the Facial Area

B. Labeling

The labeling process is carried out manually. Label 1 for faces without masks dan label 0 for faces that wear the mask.

3.2 Model Building SVM

A. Training

Image data (face images without masks or using masks) are read in the form of the pixel array measuring 50x50 pixels (Figure 4). Each pixel consists of an array of RGB values representing each pixel's color. The array size will be 50 x 50 x 3 for one image. The RGB value limit is 0 to 255.

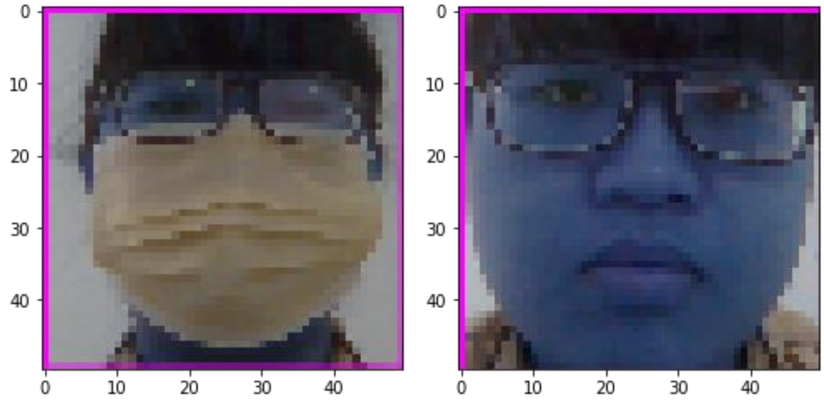


Figure 4 Face Dataset Sample Using Mask (left) and Without mask (right)

There needs to be a simplification process for the array model fitting process to reshape. For example, one of them becomes (200, 50 * 50 * 3), which results in a shape (200, 7500) for each dataset file.

The training process begins with separating training and testing data using the "train_test_split" function, which is imported from the python sci-kit learn library in the model selection section. The training data is taken as much as 75% or 300 data and leaves 100 data for testing data. Testing data will later be used in the modeling process, with its use without data labels in the testing process to ensure that the model can function with the type of data that we will try. Then do the fitting process between x_train and y_train.

B. SVM Model

The modeling utilizes the workings of the SVM algorithm. The working concept of SVM (Support Vector Machine) is to separate the inputs into different classes using the best hyperplane (Figure 5).[8]

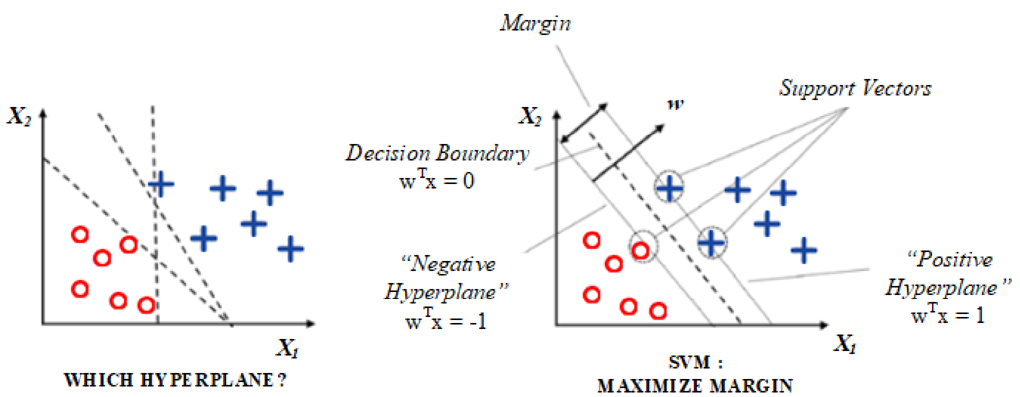


Figure 5 Determination of Hyperplane to Separate Positive Class and Negative Class

As a simple example shown in Figure 5 (right), the hyperplane is determined by looking at the distance between the outermost data, which is closest to the opposing class (support vector) to the boundary line as circled in the figure. Determination of this hyperplane also considers the margin with the most significant possible distance between classes. In determining the hyperplane to find the minimum point value, the min τ(w) equation and the following equation limits are used.

$$\min_{\vec{w}} \tau(w) = \frac{1}{2}||\vec{w}||^2 \quad (1)$$

$$y_i(w.x_i + b) \geq 1;\ i = 1,2,3,\ldots,N \quad (2)$$

The value of w adjusts the number of data classes with each weight, the value of i for each x and y data is adjusted to the number of data classes, and the value of b is the bias value.

In some cases, the hyperplane cannot separate the two classes ideally, which causes the constraints and optimization in equation (1) to not be fulfilled. These cases are usually marked by data points that are not in the supposed class area and are actually in the opponent's class area. For that, we need a soft margin technique to overcome this problem. The soft margin technique is characterized by adding a variable slack ξi, whose value is calculated from the distance of the wrong data area to the margin formed from the class support vector. By adding the slack variable to the terms of equation (2), it becomes y_i (w.x_i+b)+ξi ≥1, and equation (1) is adjusted as follows.

$$min_w = \tau(w) = \frac{1}{2||w||^2} + C\sum_{i=1}^{l}\varepsilon_1 \quad (3)$$

with C as the parameter to set the trade-off between classification error and margin. Determination of the value of C is determined by trying several values and evaluating the accuracy of the effects of the tested values. The greater the value of C, the greater the possibility of a penalty for the classification error.

A kernel function is implemented for data that cannot be determined linearly (Non-Linear SVM). The kernel is used to reduce the magnitude of the computational value. The kernel is used to perform computations at a higher dimension by transforming data into kernel space. Commonly used kernel functions are Linear, Polynomial, and Radial. One of them, the kernel, which is the default SVC kernel in the library, is RBF (Radial Basis Function).

The modeling in this research is by calling the SVC (Support Vector Classification) model function by default, using the Python Sci-kit Learn SVM library. To check the model, we can take advantage of the testing process by calling the "predict" function on the test data without contacting the label, only using the data stored in the "x_test" variable. Now we have got the matching result label from "x_test", which is stored in the "y_pred" variable. To find out only the accuracy for validating the model, you can use the "accuracy_score" function by comparing the contents of the data array between "y_test", which is the actual label of the "x_test" data with the contents of the "y_pred" data which is the result of the label from the model testing process. By default, this matching process involves checking per class, either by OVO (One VS One) feature or OVR (One VS Rest) for each feature.

By default, SVC will use an OVO checking method by comparing one feature at a time, which in this case is each face image data without a mask and one using a mask. However, the function calls, especially the "decision_function_shape" function in the "predict", have transformed technical OVO into OVR for the classification section. Also, non-linear SVM is RBF (Radial Basis Function). This is suitable for 3-dimensional data where in the classification process, it isn't easy to classify the data only by dividing it using a plane if the SVM is linear. Hence, it is more flexible for the classification process of the shape of the image data. RBF is also recommended by default in functions and is suitable for processing non-text data such as images.

3.3 Evaluation

The model that has been made will be used to detect faces that do not use masks (Figure 6). The MyCobot-Pi 6-DOF arm robot is also used as a driving robot. It uses a Raspberry Pi 4 processor equipped with other components such as the myCobot-Flat Base as a base for gluing the robot that is placed on a surface so that it doesn't move easily. MyCobot - Camera Flange is a camera to receive image input data, and myCobot – Suction Pump is a component that is specifically used in the output process (sucking objects).

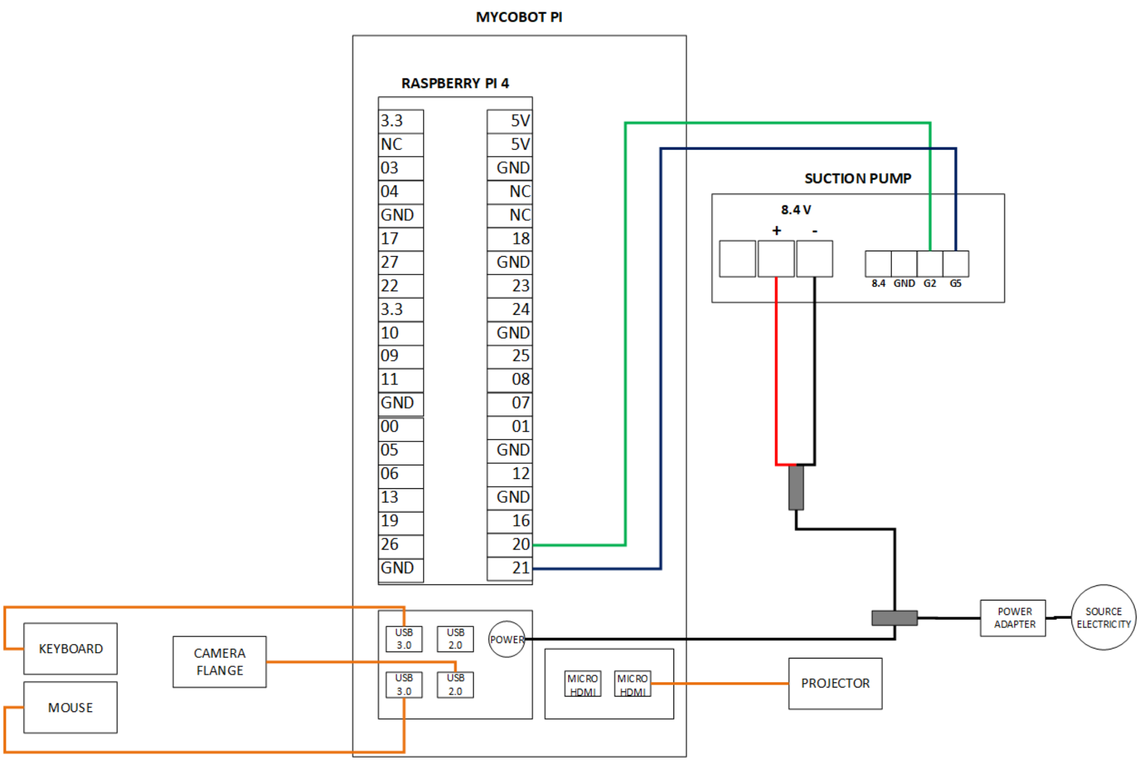


Figure 6 Tool Circuit Schematic

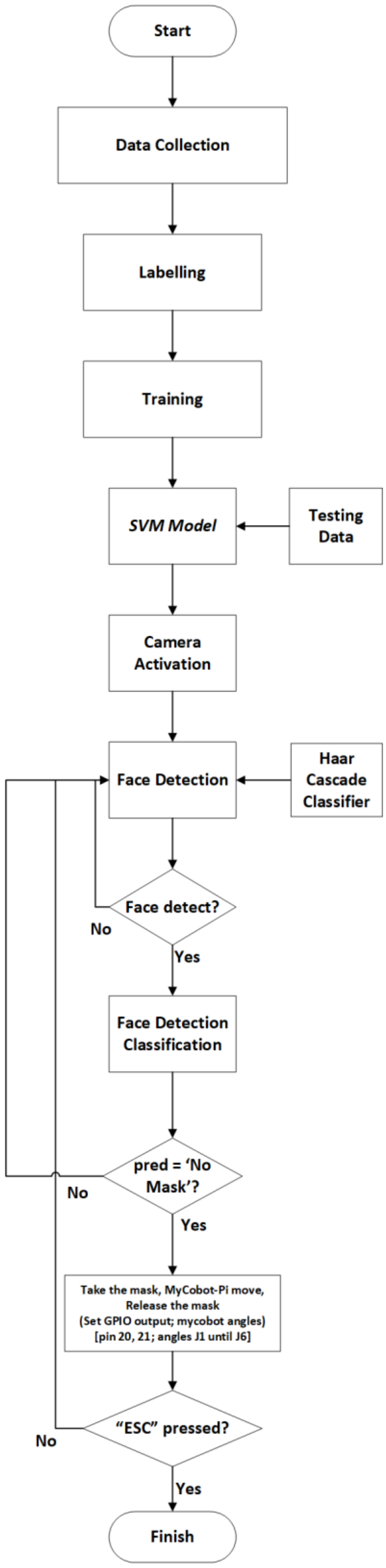


Figure 7 Mask Detection Flowchart

Figure 7 shows the mask detection flowchart. By getting the detection result in the form of a face without a mask, myCobot-Pi will move to the object's position. Then the suction pump will set the GPIO output to carry out the object suction process. After a while, the robot will move towards the specified position (towards the user). Then, the object sucked in and carried by the robot will be released so that it can be accepted by users who are detected not wearing the mask. The program will continue to repeat in the face detection section until the user presses the "ESC" key on the keyboard when the "result" frame is active to complete the program.

4 Result and Discussion

Model testing was carried out on HP laptop devices with Windows 10 Enterprise operating system specifications, Intel(R) Core (TM) i5-6200U processor; CPU 64-bit @2.30GHz 2.40GHz. Testing is done by running the SVM model program in python. By calling the "accuracy_score" function from the python sci-kit learn metrics library by comparing the testing label and the model prediction label, the accuracy value is 0.98 from a maximum value of 1 (Figure 8).

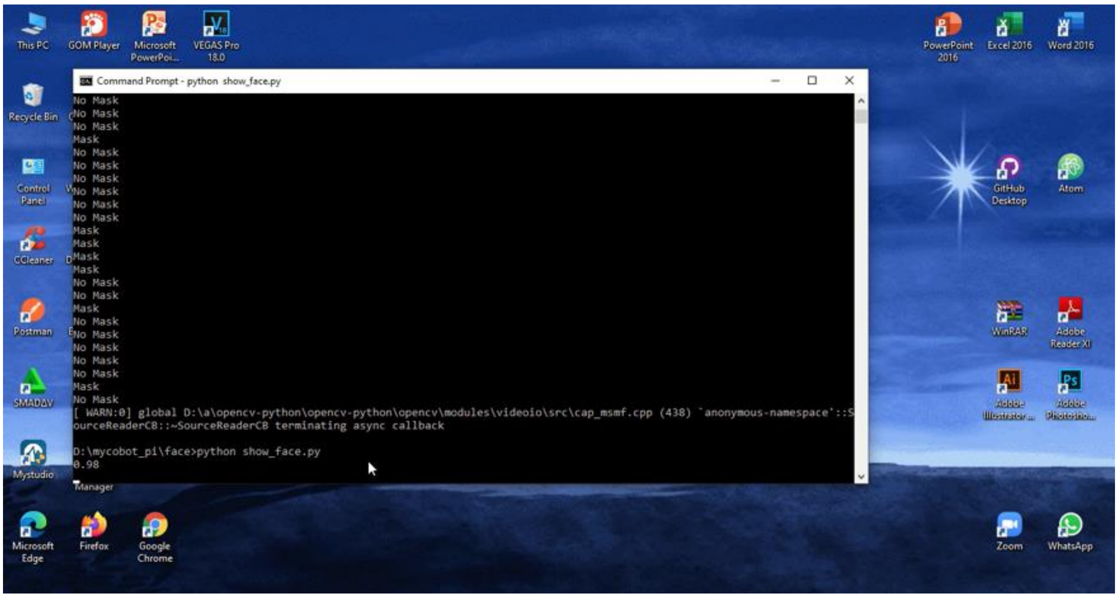
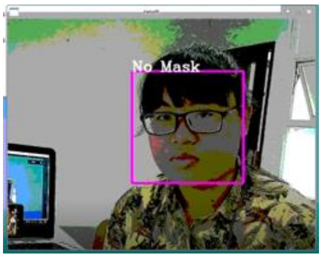
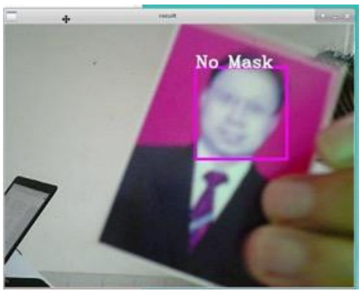
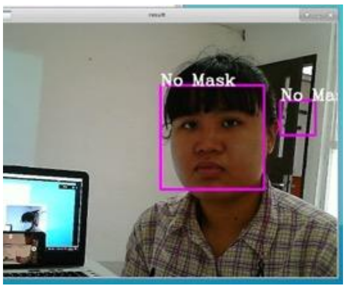
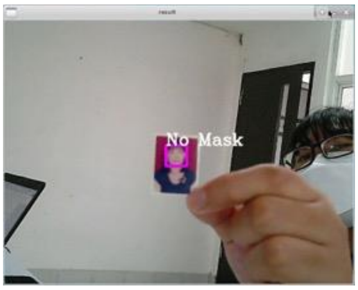
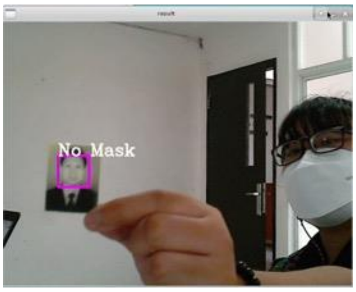
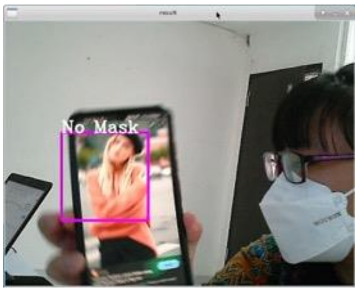


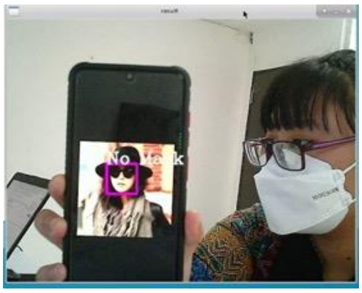
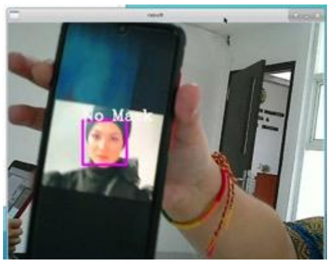
Figure 8 Accuracy Test Result

Furthermore, several tests were carried out to detect the use of masks. The tests included those that were carried out on faces of objects without masks and wearing glasses, faces without masks and not wearing glasses, faces wearing hats, and objects with faces wearing hijab. Table 1 shows some test samples that correctly show face detection's success or failure on each type of object.

Table 1 Sample Test Results for the Use of Masks

No	Results	Explanation
1		They are detecting faces of people without masks and wearing glasses incorrectly because areas other than the face (clothes) are seen as faces using masks.

2		Detects faces of people without masks and wearing glasses correctly because it successfully sees faces without masks.
3		Detects the face of a man without a mask and wearing glasses (through a photo) precisely because it has successfully detected it as a face without a mask.
4		Detects faces of people without masks and not wearing glasses correctly because it successfully detects faces that do not wear masks.
5		Detects a woman's face without a mask and glasses correctly because it has succeeded in catching a face without a mask (in the photo).
6		Detects a man's face without a mask and glasses correctly because it successfully detects a face without a mask (in the photo).
7		Detect faces of people without a mask and wearing a hat correctly because it has successfully detected it as a face without a mask.

8		Detects faces of people without a mask, wearing dark glasses, and wearing a hat correctly because it has successfully noticed it as a face without a mask.
9		Detects faces of people without a mask and wearing a hijab correctly because it has successfully seen it as a face without a mask.

In addition to the sample test results described in Table 1, several other tests were carried out, which included for each group of different types of object conditions, 17 cases for face detection of objects without masks and wearing glasses, 14 cases for face detection without masks and without glasses, 6 cases for face detection wearing a hat, and 4 cases for face detection wearing a hijab. From each of these test categories, for 14 cases of face detection without a mask and wearing glasses, there were only 4 cases that failed to detect faces correctly. Next, for the 14 cases of face detection without a mask and glasses, 7 cases were not detected correctly. Furthermore, for the detection of face samples wearing hats, out of 6 cases, only 1 case sample did not detect it correctly. Finally, for samples of faces wearing hijab, out of 4 cases with different facial conditions, all of them managed to catch faces without masks on faces wearing the hijab.

5 Conclusion

There are still failures in several experiments even though using the same situation and image object because the model is not perfectly accurate. In this experiment, we concluded the SVM model assisted by the Haar Cascade Classifier could be used for the classification of real-time mask detection images. The images were captured using myCobot Camera Flange. Of 41 tested cases, 29 cases were successfully detected. Detection can also work without gender restrictions because it can be processed to detect both male and female faces. In addition, it is also reinforced by the results of model accuracy that are run on HP brand laptops with the Windows 10 Enterprise operating system, Intel(R) Core(TM) i5-6200U processor; 64-bit CPU @2.30GHz 2.40 GHz is 0.98 or 98%. For further research, researchers can try a 6-DOF robotic arm that has higher specifications in model processing and detection to minimize detection failures. This research does not include lighting settings in face detection, so it can be a suggestion for future research.

MyCobot Suction Pump, which was a part of myCobot-Pi, can successfully be used to lift mask packaged in a PP protector plastic sheet, which has a non-slippery surface and can be used as a substitute for mask packaging.

REFERENCES

[1] H. Hasma, M. Musfirah, and R. Rusmalawati, "Penerapan Kebijakan Protokol Kesehatan dalam Pencegahan COVID-19," *J. Ilm. Kesehat. Sandi Husada*, vol. 10, no. 2, pp. 356–363, 2021, doi: 10.35816/jiskh.v10i2.620.

[2] D. H. Liyanage, P. S. Adikari, K. Pathirathna, and W. Dias, "Directives on Community Use of Face Masks During COVID-19 Pandemic: a Commentary Article," *J. Adm. Kesehat. Indones.*, vol. 9, no. 1, p. 82, 2021, doi: 10.20473/jaki.v9i1.2021.82-94.

[3] S. A. Sanjaya and S. A. Rakhmawan, "How Can Machine Learning Help the Authorities?," *Semin. Nas. Off. Stat.*, vol. 2020, no. 1, pp. 370–376, 2021, doi: 10.34123/semnasoffstat.v2020i1.497.

[4] Henderi, A. S. Rafika, H. L. H. Spits Warnar, and M. A. Saputra, "An Application of Mask Detector for Prevent COVID-19 in Public Services Area," *J. Phys. Conf. Ser.*, vol. 1641, no. 1, pp. 0–6, 2020, doi: 10.1088/1742-6596/1641/1/012063.

[5] R. Marwansyah and A. Mulyani, "Implementasi Face Mask Detection Untuk Menghambat Laju Penyebaran COVID-19 Berbasis Machine Learning," *J. Inf. Syst. Applied, Manag. Account. Res.*, vol. 5, no. 2, p. 413, 2021, doi: 10.52362/jisamar.v5i2.399.

[6] L. W. Jie, T. P. Sen, N. M. A. Ghani, and M. F. Abas, "Automatic control of color sorting and Pick/Place of a 6-DOF Robot Arm," *J. Eur. des Syst. Autom.*, vol. 54, no. 3, pp. 435–443, 2021, doi: 10.18280/JESA.540306.

[7] A. R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale robot arm design with pick and place mission based on inverse kinematics," *J. Robot. Control*, vol. 2, no. 6, pp. 469–475, 2021, doi: 10.18196/jrc.26124.

[8] D. I. Pushpita Anna Octaviani, Yuciana Wilandari, "Penerapan Metode SVM Pada Data Akreditasi Sekolah Dasar Di Kabupaten Magelang," *J. Gaussian*, vol. 3, no. 8, pp. 811–820, 2014.