Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 201 (2022) 478–486

Procedia Computer Science

www.elsevier.com/locate/procedia

The 5th International Conference on Emerging Data and Industry 4.0 (EDI40)
March 22-25, 2022, Porto, Portugal

Smart Door for COVID Restricted Areas

Daniyar Syrlybayev[a], Nurlan Nauryz[a], Aidana Seisekulova[a], Kaiyrbek Yerzhanov[a], Md. Hazrat Ali[a,]*

[a]*Mechanical and Aerospace Egnieering Departement, School of Engineering and Digital Sciences, Nazrbayev University, 53 Kabanbay Batyr ave., Nur-Sultan 01000, Kazakhstan*

Abstract

During the Covid pandemic, the world has faced inevitable changes. In Kazakhstan, people receive color statuses depending on the vaccination status, which can be checked through mobile phones. Depending on this colour, different restrictions are imposed. This status, along with the presence of a facemask, is checked in public places. However, security workers may overlook or ignore visitors without facemasks or dangerous status. To address these issues, the following paper presents a method for automating the checking procedure of the visitors at the entrance to any building. The automatic door opener is programmed using Python and OpenCV image processing library. This door opener can identify the green color from the display of the mobile phone placed in front of the camera as well as the presence of the facemask on the human face. The door is opener sends signals to the Arduino microcontroller, which actuates the motor to open the door if all the above-mentioned conditions are satisfied. The main feature of this project is that a build-in face recognizer was used to classify the human faces on masked and non-masked. Despite this, such a method shows good accuracy of 99% in identifying non-masked faces and 77% if a masked face test dataset is used. This project has the potential benefit of automating the checkup process, avoiding the crowds' formation in front of the door, and facilitating the work of security guards.

Keywords: Image Processing; Color Recognition; Facemask Recognition;

* Corresponding author. Tel.:+77172706145
E-mail address: md.ali@nu.edu.kz

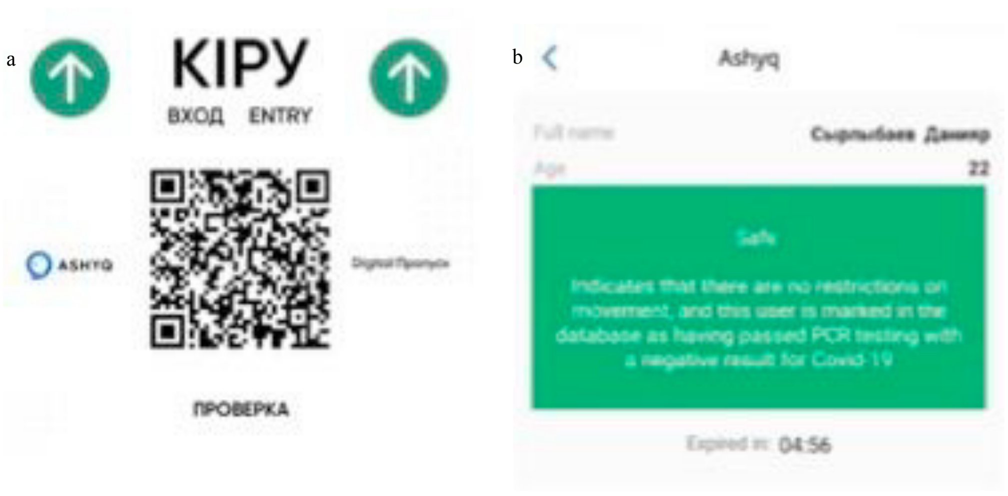
1. Introduction

Humanity has encountered various issues due to the COVID pandemic. Protecting the health of the population is the paramount goal. Social distancing and wearing facemasks have become precautionary rules worldwide, including in Kazakhstan. In the Republic of Kazakhstan, only vaccinated citizens can access malls and shopping centers. The control over the vaccination status is performed using the Ashyq program. The main principle is that people scan the QR code shown at the entrance and show the result to the security guard. Additionally, security checks the presence of the facemask on the guest. People can enter the building if they have green status in Ashyq and have a facemask present. Only people with green status can enter the building. The example of the QR code and positive response of the Ashyq program is shown in Figure 1. Nevertheless, the security may overlook the intruders, especially when there are many visitors.

In this paper, a novel automatic door opening system is proposed. This system detects the positive status on the vaccination software from the guest's phone display, recognizes the presence of his or her facemask from the camera, and actuates the motor which can control the door using Python and OpenCV library. The main feature of this method is that it relies purely on OpenCV's build-in face recognizer to classify the human face as masked and non-masked. The system can control the turnstile or sliding door, potentially eliminating the need for human labor and offering efficient security.

2. Literature Review

A smart algorithm for color detection and tracking from the mobile phone screen and an image recognition system is needed to implement this project. A color detection system is needed to identify the status color of the visitor. The second check involves the presence of the face mask on the human face, which requires face detection and features recognition algorithms. Thus, the focus of the current review is on the face and color recognition tracking algorithms that can be used in this project.



Ashyq

Full name Сырлыбаев Данияр

Age 22

Safe

Indicates that there are no restrictions on movement, and this user is marked in the database as having passed PCR testing with a negative result for Covid-19

Expired in: 04:56

Fig. 1. Examples of a) The QR code to be scanned by the guest b) Positive Status in Ashyq software

Goyal et al. [1] discussed various face detection algorithms in OpenCV that apply to different conditions. Algorithms for static images like face extraction from the photos with noncolored and coloured backgrounds were summarized. Moreover, the processing algorithms for the moving faces, such as face recognition by detection of the blinking eyes and Haar Cascade Algorithms, were discussed. It was emphasized that the Haar detection method is currently the most effective one. One of the works where the Haar face detection method was utilized was made by Chen and Sang [2]. They implemented the face mask recognition algorithm using computer vision and neural network-based Gaussian Mixture Model. This AI-based model was trained to evaluate the probability of whether the face image is normal (only face) or abnormal (in case if human wears glasses or facemasks). The model decides whether a human face image is normal or abnormal based on the threshold. Haar classifiers were used to detect the proper mask-wearing in the study of Cabani et al. [3]

Liu et al. [4] developed a computer vision-based Android OS app to identify the colour of the reagent and classify them as positive and negative depending on whether they are applicable or not. An OpenCV runs on Java and Weka neural network library were used to recognize the colour. The main algorithm consists of image acquisition followed by reagent tubed edge detection and colour recognition. The colour recognition is done from the bottom edge pixel and based on a neural network algorithm developed in previous works. This algorithm shows 92% of efficiency in the discrimination of the negative reagent. Zhang et al. [5] developed a robotic arm that picks up the round yellow objects from the conveyor and places them into the container. An OpenCV-based code is written to accomplish this task. First, it reads the image from the camera and converts it to Hue, Saturation, Value (HSV) system. It allows specifying the range of colours accepted by the code instead of a single RGB value. Based on the HSV image, the histogram equalization is implemented to convert the image to its black and white equivalent so that the object that needs to be picked up is white and the rest of the image is black. A similar study with the control of the robotic arm using a computer vision algorithm was performed by Szabo et al. [6]. In their study, they used the Raspberry PI platform with preinstalled OpenCV library to control the robot. Another similar study was performed by Jia et al. [7]. In their study, they developed a sorting robot that puts the yellow and silver objects into corresponding bins. The experimental verification showed the efficiency of this algorithm. Zhang et al. [8] developed a computer vision system that can recognize the cherry in the video and be used in the cherry collecting robot. By reducing the noise, they wrote a code that extracts every pixel's RGB value of the image and finds the red colour. Using the morphological transformations and thresholding, the image was converted to its black and white equivalent, where the white area corresponded to the red cherry in the usual RGB channel. Based on this last threshold image, the edge of the cherry is detected using the Canny Edge detector. Cai et al. [9] proposed the colour recognition algorithm in OpenCV. In their study, the image is converted to its hue-saturation-value equivalent. In these conversions, ranges of hue, saturation, and value are defined for several colours, and based on these ranges, the colour is classified. The corresponding HSV code ranges for several images are summarized and can be used for colour detection in the current project. However, the following method fails to classify the colour in low brightness conditions. Suryawibawa [10] developed a software application capable of reading the images of the leaves from the camera and comparing them with the database. The application initially acquires the image of the leaf and then preprocesses it to resize, remove noise and convert to a binary image. Afterward, using the canny edge detection, the image veins are extracted. By employing OBR algorithms within the OpenCV, matching the features from the latter image is performed against the image in the database using the City Block Distance method.

The project presented in this paper was also based on the OpenCV algorithms and neural network training.

3. Methodology

From the discussion above, a two-stage checkup is performed by the security. In the following paper, the computer vision-based system was developed to perform this checkup. Figure 2 shows how this program operates in general. As shown, the program checks whether the person has a green status in the Ashyq software and whether he or she put on the facemask. Only in case, both results are satisfactory the motor is actuated. A Python-based open-source OpenCV library was used to develop the code. The Arduino microcontroller controlled the motor, and it was activated if all the necessary checkups were satisfied. The communication between Python and Arduino was established using the serial port. It should be noted that this motor can be used to control any opening device such as a sliding door and turnstile. The following subsections describe how each step was implemented in the developed code.

3.1 Check the status of the person

Figure 3 represents the algorithm to check a person's status. The program waits for the person's response in the idle state. The message asking to put the phone display closer to the camera is prompted in this stage. When the person puts his or her phone sufficiently close to the camera, the resulting frame is firstly preprocessed. At this stage, noise is removed by blurring the image using the Median Blur technique, and the edges are extracted using

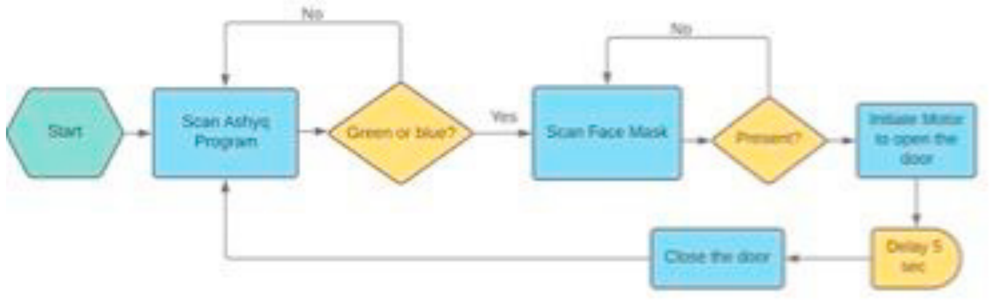


Fig. 2. The general operation of the program

Canny edge detection. This step allows to extract of only precise edges, and hence the contrasting boundaries of the display could be isolated. In the next stage, two diagonal corners of the phone are found, and the phone's area is calculated. If this area is smaller than the threshold value of 10000 pixels^2 program returns to the idle stage. Otherwise, the display image is captured and cropped to isolate it, and the green mask is applied to it. This masking technique allows viewing only a given range of colors. The target color was green, and the applied mask searched its different shades. Finally, again the largest green contour on the masked image is found, and if its value is larger than 5000 pixels^2, the second stage checkup initiates. Everything below this threshold is considered noise.

3.2 Checking the facemask presence

The artificial neural network (ANN) was trained to check whether the facemask is present on the human face. This ANN is based on the face recognizer built in the OpenCV library. It usually is used to recognize and distinguish the faces of two different people; however, it was utilized to check whether a facemask is present on the human face. The advantage of this approach is that there is no need to create custom ANN using special libraries, which simplifies the program. The 2400 images of people of different gender, race, and ethnicity with the facemask and without facemask were used to train the neural network. The examples of such images are shown in Figures 4 and 5. These images were classified, and the neural network was trained based on this classification.

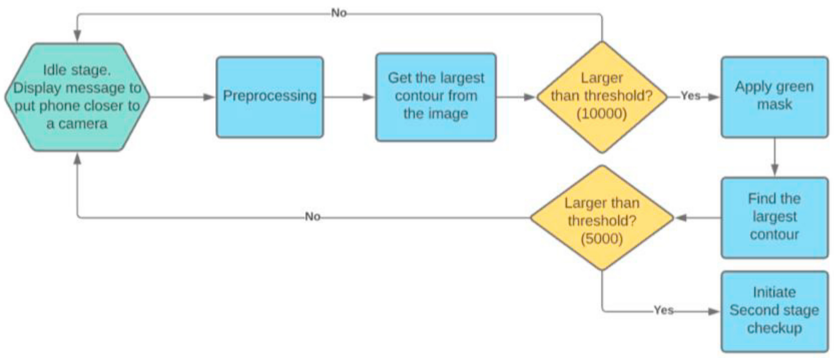

Idle stage. Display message to put phone closer to a camera
Preprocessing
Get the largest contour from the image
Larger than threshold? (10000)
Yes
No
Apply green mask
Find the largest contour
Larger than threshold? (5000)
No
Yes
Initiate Second stage checkup

Fig. 3. Algorithm of the first stage checkup

The facemask recognition algorithm is shown in Figure 6. To check whether the facemask is present on the human face, the face itself needs to be detected. For this reason, the image was first converted to a greyscale value,

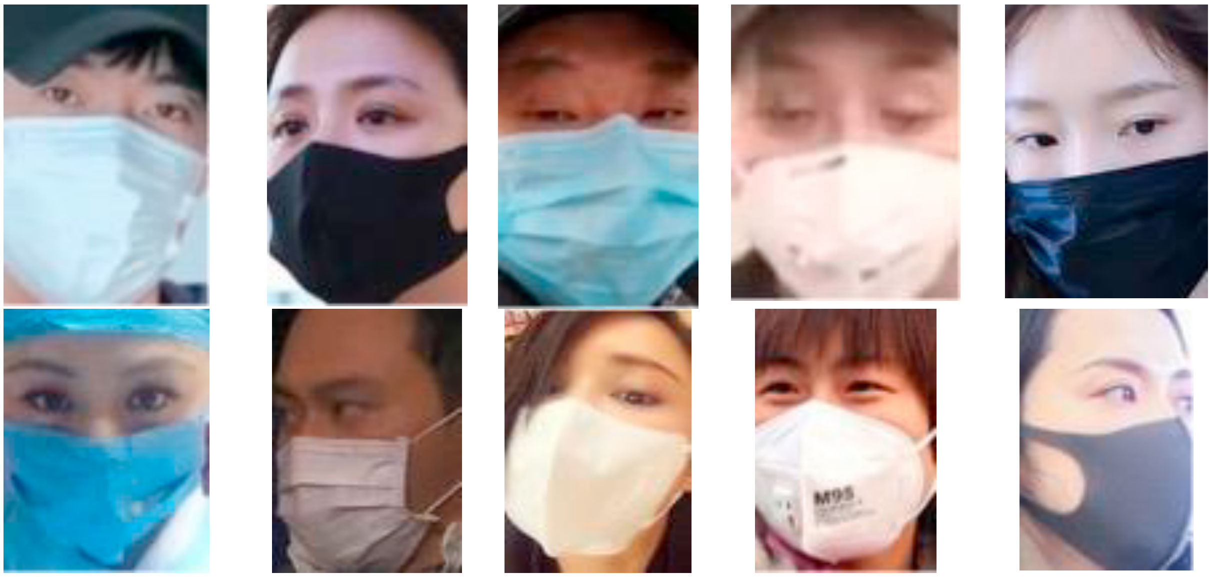


Fig. 4. Examples of images of humans with the mask used to train the network [11]

and then Haar cascades were used to find the face. This face detection method works only with greyscale images, and hence the conversion was a prerequisite. Then if there is only one face present in the frame, the program decides whether there is a facemask or not using the ANN developed. If it is present, the program delays for two seconds and then checks the presence of the mask once again. This step is done because the noise from other frame attributes induces some errors. This noise is filtered using this double-check, and the program starts working much more reliably. The actuator is initiated if the positive response is obtained from both checks. Otherwise, a message to put on the facemask is displayed.

Fig. 5. Examples of images of humans without the mask used to train the network [11]

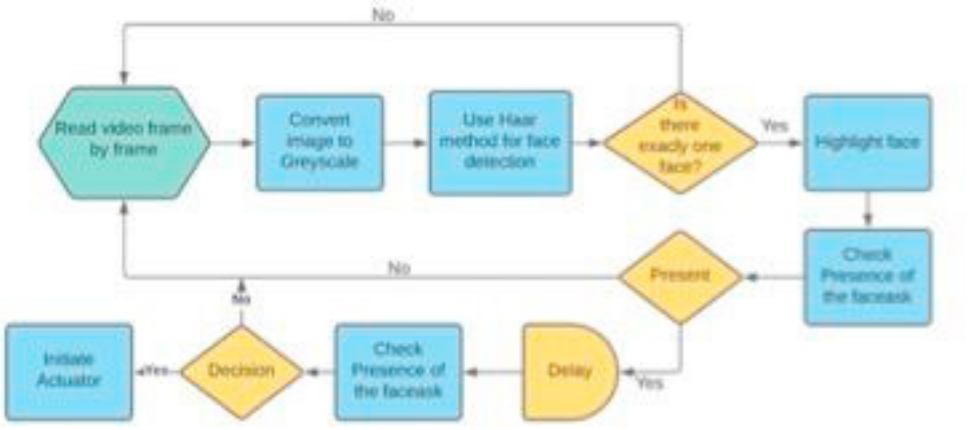

No
Read video frame by frame
Convert image to Greyscale
Use Haar method for face detection
Is there exactly one face?
Yes
Highlight face
Check Presence of the facemask
Present
No
No
Initiate Actuator
Yes
Decision
Check Presence of the facemask
Delay
Yes

Fig. 6. Algorithm to check the presence of the facemask

1.1. Communication between microcontroller and Python program

If the output of two-stage checkups is true, a corresponding signal is generated, sent to the serial port, and then captured by the microcontroller. This signal triggers the microcontroller to start the actuator, which opens the door. Then 6-second delay is imposed to let a person pass through. Afterward, the door is closed, and the program returns to the initial idle stage to check the next person.

In this study, Arduino Uno received commands from the Python code and sent signals to the A4988 module. This module controlled the operation of the 17HS3401 unipolar stepper motor. The motor was powered from the 12 V battery, while its controller received energy from the 5 V pin of the Arduino microcontroller.

4. Test Results

In the following section, the results of the operation of the program are reported. In the first subsection, the checking operation of the Ashyq status is presented with examples of green, red, and blue status outputs. In the second subsection, the facemask presence checking is described.

4.1 Checking the Ashyq Status

The video frame is initially captured and preprocessed as described above. As a result, an image is displayed with only edges between the contrasting colors. This step allows isolating the phone, whose corner edges are found and the resulting region is cropped. Further processing is applied to this image. After the application of the mask, only the edges of the green object are detected. Thus, again the program checks whether the area of the resulting green box is larger than the threshold value. Finally, the detected contour of the green box with status is indicated, and the corresponding message is prompted on the initial image of the phone as shown in the first figure in the sequence.

On the other hand, this program does not allow people with red and blue statues to enter the building, as shown in Figure 7. The program was constrained to recognize the green colour of the specified minimum size on the phone display. In case it cannot be found from the image, and hence the second stage check does not start.

Finally, it should be mentioned that the following program is moderately reactive to the lighting conditions. Under very bright lighting, the boundary of the phone display cannot be found due to insufficient contrast between it and the surrounding environment. However, the program operates more smoothly under standard room lighting as multiple tests showed.

4.2 Checking the facemask presence

The facemask detection algorithm performed satisfactorily in all the tests conducted. Figure 8 shows the operation of this recognizer on the example of a male face in the mall. From Figure 8, it is interesting to note that the program recognizes only proper wearing the mask, which should cover the entire face from chin to eyes. Opening the nose or wearing the facemask in the chin, as shown in Figures 8 b and c, will not allow the person to pass through the door.

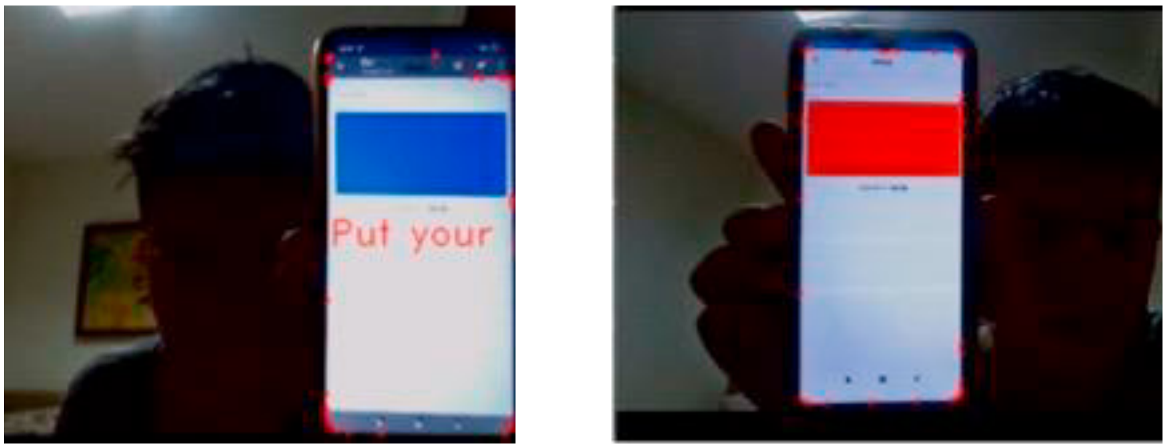


Fig. 7. Examples of how the program operates if: (a) blue and (b) red statuses are captured

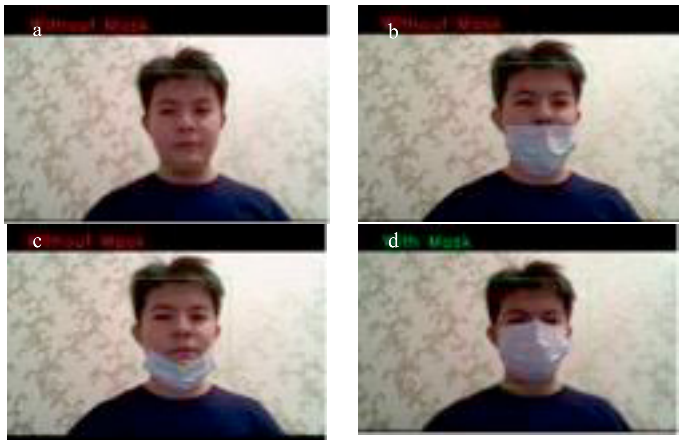


Fig. 8. Example of the operation of the program with the male face

To assess the accuracy of the model, a series of tests on the test dataset was conducted. The test dataset was taken from the MaskedFace-Net image database developed by Cabani et al. [3]. Overall, 1000 images (500 in each image label) were used to assess the model's accuracy. Table 1 shows the results of the tests. As it can be seen, the proposed method can recognize the face without the mask with about 99% accuracy as evaluated on the test set. Meanwhile, the error in recognizing the masked face is higher, and the method's accuracy of drops to 77%. The reason for such discrepancy is that the Haar cascade method is not perfect in the recognition of faces. This method, in several cases, failed to show the face, and when this image is passed to the neural network, it produces an erroneous result.

Table 1. Performance of the proposed Facemask detection method

Without facemask test set	With face mask test set
77.0%	99.1%

5. Conclusion

In conclusion, the program that can identify Ashyk status and mask-wearing factor of a citizen was written. A miniature model of the door assembly was constructed To demonstrate its working principle. For the program, a Python-based open-source OpenCV library was used. The code was saved on the computer and was implemented through an Arduino microcontroller. The motor was activated if two conditions were satisfied. The first condition was the visitor's safe status, and the second condition was the proper mask-wearing. The motor was rotated in one direction to open the door and after 5 seconds in another direction to close the door.

The model shows an accuracy of 77% in identifying masked faces correctly and 99.1% in the identification of non-masked faces. The main limitation is that it fails to recognize the face correctly and passes the wrong result to the face recognizer on several occasions. This failure contributes to the increased error of the Haar Cascade method. Despite this fact, the presented checkup system can be integrated into a tourniquet or sliding doors to substitute the manual checking of the vaccination status and facemask presence.

References

[1] K. Goyal, K. Agarwal, and R. Kumar. (2017) "Face detection AND Tracking: Using opencv," *International conference of Electronics, Communication and Aerospace Technology (ICECA)*.

[2] Q. Chen and L. Sang. (2018) "Facemask recognition for fraud prevention using gaussian mixture model," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 795–801.

[3] Cabani, A., Hammoudi, K., Benhabiles, H. and Melkemi, M., 2021. MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health*, 19, p.100144.

[4] P. Liu, W. Zhang, J. Qiu, Q. Hu, and K. He. (2019) "Reagent color recognition model for Android platform based on opencv and machine learning," *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*.

[5] W. Zhang, C. Zhang, C. Li, and H. Zhang. (2020) "Object color recognition and sorting robot based on OpenCV and machine vision," *IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*.

[6] R. Szabó, A. Gontean and A. Sfiraţ. (2016) "Robotic arm control in space with color recognition using a Raspberry PI," 39th International Conference on Telecommunications and Signal Processing (TSP), 2016, pp. 689-692, doi: 10.1109/TSP.2016.7760972.

[7] Y. Jia, G. Yang, and J. Saniie. (2017) "Real-time color-based sorting robotic arm system," *2017 IEEE International Conference on Electro Information Technology (EIT)*.

[8] Q. R. Zhang, P. Peng, and Y. M. Jin. (2016) "Cherry picking robot vision recognition system based on OpenCV," *MATEC Web of Conferences*, vol. 63, p. 01005.

[9] Z. Q. Cai, W. Luo, Z. N. Ren, and H. Huang. (2011) "Color recognition of video object based on hsv model," *Applied Mechanics and Materials*, vol. 143-144, pp. 721–725.

[10] I. W. Agus Suryawibawa, I. K. Gede Darma Putra, and N. K. Ayu Wirdiani. (2015) "Herbs recognition based on Android using OpenCV," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 2, pp. 1–7.

[11] Balajisrinivas, "Face-mask-detection/dataset at master · Balajisrinivas/face-mask-detection," *GitHub*. [Online]. Available: https://github.com/balajisrinivas/Face-Mask-Detection/tree/master/dataset. [Accessed: 01-Feb-2022].