Detection of Certain Objects Wearing Masks in Real Time To Prevent the Spread of the Virus (Yolov3)

RusdiantoRoestam[1], Kusdarnowo Hantoro[2*], Amir Dahlan[3]
[1]Computer engineering Faculty of Computing President University, Indonesia
[2]Informatics, Faculty of ComputerSciences,UniversitasBahayangkara Jaya, Indonesia
[3]Data and Information Center, National Research and Innovation Agency, Indonesia
rusdianto@president.ac.id, kusdarnowo@dsn.ubharajaya.ac.id(*Corresponding Author), amir005@brin.go.id

Abstract. A significant increase in the spread of the corona virus (COVID-19) in the community is currently happening due to people not following the health protocol rules set by the ministry of health. One of the rules is to require people to wear masks while they are outside the home. Measures need to be implemented in anticipation of situations where people do not wear masks in public spaces. Therefore, the establishment of a mask detection system is chosen as a solution in order to solve the mentioned problem above. A real-time identification system for people wearing masks is proposed to be developed in this paper. The system utilizes Yolov3 with Darknet -53 as a deep learning mask detector and OpenCV as a real-time computer vision library, so that people doing activities in a public space captured by a video can be recognized and detected when they do not wear masks. In implementing deep learning, a data set of 4000 images is divided into two classes, i.e.,2000 images with masks for data testing purposes and another 2000 images without masks for training custom objects. The Extreme Programming (XP) method as part of the Agile Process Model is adopted for system development. Computer language support and the latest system development tools have made it possible to utilize this method in an effort to develop this system rapidly. Requirement Analysis is conducted to obtain required processes before designing system. Writing code and testing the system will be the next step before the system is declared ready to be implemented in the public space. By adopting the XP development method, all of the above steps can be implemented repeatedly until the system delivers the expected results. According to test findings on this image, the individual whose face was taken by the camera is recognized as "wearing a mask" with an accuracy of 0.95 or 95%.

.**Keywords**: Arduino Uno, Raspberry Pi, Yolov3, Mask Detection, COVID 19

Received May 2022 / **Revised** November 2022 / **Accepted** November 2022

INTRODUCTION

The corona virus pandemic disease (COVID-19) causes a health and economic crisis in every sector which causes limited mobility of the community. The spread of this virus is still continuing, so the government recommends that public health is the main thing that must be resolved immediately. Enforcement to the community such as social distancing, washing hands, reducing mobility, staying away from crowds and wearing masks are very important things in the government's recommendations in reducing the number of virus transmission in this pandemic.

Indonesia is one of the largest countries infected with the corona virus with a total of 2.18M case, 1.88M recovered, and a total of 54.481 deaths at the end of June 2021, with an average of 5,000 – 20,683 people infected per week. Transmission occurs when an exposed person is in closed contacts (within ±1 meter) that have respiratory symptoms (for example, coughing or sneezing) or often referred as droplets have the potential to infect, because of the risk of being exposed to the mucosa (mouth and nose). The airborne transmission with particle $<5\mu m$ in diameter can last for a longer period and a distance of more than 1 meter.

Wearing a mask for people having any activity in the public area is a very effective measure because this way will cover the source of the virus that comes from the human body (i.e. nose and mouth).[1] Technical Guidance for activities that have been established by WHO also requires people to always use masks during interactions and activities in public areas.

DOI: 10.24014/coreit.v8i2.17184

Although it has been given the obligation for everyone to wear a mask during activities in public spaces, there are still many people who use the mask imperfectly, for example, the nose is still visible. The picture of a situation like this, of course, remains risky and dangerous because the potential for virus transmission will still occur. More intelligent countermeasures need to be put in place to solve problems like this.

This paper proposes a system that is able to detect people who are not wearing masks or the masks used are not used properly. The system built will use cameras to scan people in public spaces. A warning will be given if it is identified that there are people who are not wearing masks or the masks used are not perfect. This system implements Yolov3 and Darknet-53 for the identification of captured objects and utilizes Arduino Uno, Raspberry Pi, LCD, camera, and monitor as components that complete prototype development.

Prototype will work when someone is in the range and scope of the object detecting camera. The data obtained on the camera will be processed by the system as information that will be used as output [2]. The results of the process obtained will be displayed on the monitor screen and LCD.

The system used implements Yolo and Darknet. Yolov3 is a state of art object detection algorithm that divides the input images into the SxS grid. The size depending on the input size is 13x13, 26x26, 52x52 [3][4]. Since the bounding box is mapped based on the resulting confidence value, Yolo will predict the class of objects contained in the bounding box and their probabilities, so that a class probability map is formed. In Yolov3, the feature extractor that previously used Darknet-19 became Darknet-53, and also the objcet detection process which now uses 3 scales where at each scale it uses 3 anchor boxes so that it has an impact on increasing the ability to detect smaller objects. Darknet is an open-source neural network framework written in C and CUDA. Darknet has fast speed, easy installation and supported by CPU and GPU. Darknet features are under development for machine learning processes, such as object detection and classification [5][6]. The feature extractor of the Yolo-v3 contains 53 convolutional layers. This is a hybrid of the previous generation v2 and v1.

The development of this system uses a customized dataset with predetermined details. The use of this system uses 4000 sample images which are divided into 2 parts, the first part is for users who do not wear masks as many as 2000 images and the second part is 2000 images for users who wear masks[1][7]. This data collection is taken through internet sources with the help of Firefox Extension "DownThemAll" which can download images simultaneously. The data that has been collected will be labeled using application labeling to get an annotation object file on the image.[8][9][10]

Image labelling is a process of identifying and marking object in the image. Image labelling that can optimize a process to generate meta data or make recommendations to user based on the details in the specified image [11][12]. Meta data or called Annotation is a process of capturing the desired object in the image, so that it can be recognized and understood by the machine through computer vision. Use this this to train AI models by training deep learning algorithms, studying and understanding patterns, which are represented through annotations.[13][14]

This paper uses several microcontroller-based devices and others as components in the mask detection prototype. The explanation of the components used in the prototype is discussed along with the literature theory behind it.

Arduino Uno Arduino UNO is a microcontroller board based on the 8-bit Atmega32P microcontroller. The components contained in the ATmega328P consist of several other components such as a crystal oscillator, voltage regulator and many components supporting the microcontroller. Arduino has 14 digital input and output pins (6 of which are PWM output), 6 pin input analog, connection USB, jack power barrel, header ICSP, and reset button. The Arduino UNO can be used to communicate with computers, other Arduino board or other microcontrollers.

The Atmega328P microcontroller provides UART TTL (5V) serial communication which can be done using digital pin 0 (Rx) and digital pin (Tx) [4]. The ATmega16U2 on board transmits this serial communication via USB and appears as a virtual communication port to software on the computer. Firmware using standard USB, and no external driver required. The Arduino software includes a serial monitor for sample text data sent to the Arduino board. On the board the Rx and Tx LEDs will flash when the data is being transmitted via the USB to serial chip and USB connection to the computer. The ATmega328P L2C and SPI communications.

Raspberry Pi The Raspberry Pi is an SBC (single board computer) with the size of a credit card that has been equipped with functions like a complete computer, using a System on Chip ARM integrated on a circuit board (PCB). The ability to run the Linux operating system and several applications such as applications LibreOffice, multimedia (audio and video) or programming. The Raspberry Pi model B has 521mb of RAM and is also equipped with an Ethernet port. Data storage does not use a hard disk or solid-state drive, but uses an SD memory car for booting and storage. The Raspberry Pi 3 Model B features an Open GL ES 2.0 GPU, hardware-accelerated Open VG, and 1080p30H.264 high-profile decoding and capable of 1 Gpixels/sec. Several changes were made by the raspberry pi such as an increase in the processor, an increase in Bluetooth low energy (BLE) connectivity and BCM43143 Wi-Fi. The Raspberry Pi also improves resource management by up to 2.5 Amps to support more powerful external USB devices.

Webcam (web camera) is a real-time camera whose images can be viewed via the World Wide Web, instant messaging programs, or video calling applications. A webcam is a small camera that is connected to a computer via a USB port, COM port or via Ethernet or Wi-Fi.

I2C Module is a display system that uses a 16x2 character LCD dot matrix based on the Hitachi 44780 IC. High speed I2C serial bus produced by DFRobot. The dot matrix LCD display system can be connected to the Arduino board only by using 2 (two) A4 and A5 in addition +5 Volt DC voltage source. Analog A4 and A5 from Arduino UNO are connected to SDA, SCL from serial board. Library files are required so that Arduino UNO can be used to run the LCD. I2C is a two- way serial using two dedicated channels for sending and receiving data. I2C consists of SCL (Serial Clock) and SDA (Serial Data) which carry data between I2C and its controller.

LCD (Liquid Crystal Display) is an electronic component that functions to display data in the form of characters, symbols, letters and graphics, with the small size LCDs are paired with microcontrollers. The LCD provides a shape module that has power pins, contrast settings, and light supply controls.

The utilizing this component, development can form into a series of prototypes that can be used as a tool with the function of detecting someone in the use of a mask and not using a mask..

METHODS

The mask detection system automatically reminds the user to stay in the correct condition of using the mask. The use of a webcam camera is one of the real-time visual captures to take sample data from someone who is wearing a mask or not wearing a mask, when someone is in the range and scope of the camera's view the system will know if the mask is wearing the right condition or not in the right condition.

YOLOv3 and DarkNet-53 deep learning model explains that the application can detect certain objects. Utilize of formulas that take different co-ordinate, such as pw, ph, tx, ty, tw, th, cx , and cy. The variables used for bounding box dimensions. The values of the obtained boundary boxes (x-axis, y-axis, height and width) will adjust. This study uses 216 images as a dataset for classification into different sets. Yolo annotation is used as data labeling to give value for the DarkNet-53 model, where the training is adjusted according to the configuration and obtained accuracy 97.68% during the training.

An object that is obtained after the system is running will trigger the LCD to display a warning message to the user, while the monitor displays the results on the screen such as a bounding box, image classification and accuracy obtained. The used of LCD as a warning tool can make it easier for users to more easily realize, when someone has been identified on the mask detection machine.

The system will show a presentation of the user's accuracy in the use of masks and the system will show the high and low accuracy of anyone using the use. The use of masks in conditions of wearing masks correctly, then the presentation gain will be high and when the use is under conditions that are not correct then the presentation.

Yolo is a state-of-the-art object detection algorithm that divides the input image into a grid SxS. The size of the grid cells depending on the input size is 13x13, 26x26, and 52x52. Confidence which is the probability value of the existence of anobject in the bounding box. Since the bounding box is mapped based on the resulting confidence value, Yolo will predict the class of objects contained in the bounding box and their probabilities, so that a class probability map is formed (Figure 1).

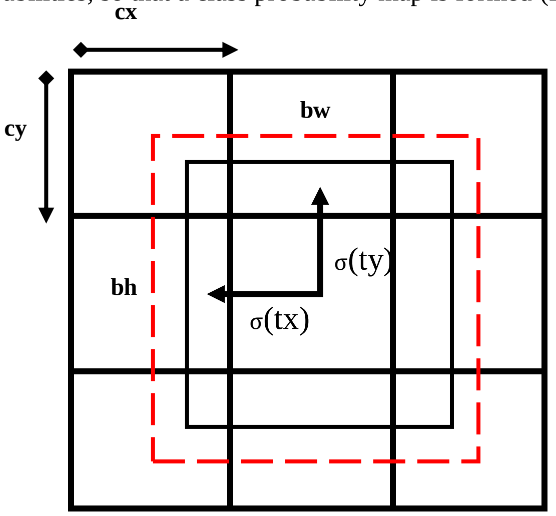

Figure 1 Grid of Bounding Box

$$b_x = \alpha(t_x) + cx \qquad (1)$$

$$b_y = \alpha(t_y) + cy \qquad (2)$$

$$b_w = p_w e1w \qquad (3)$$

$$b_h = p_h e1h \qquad (4)$$

This research uses the Extreme Programming (XP) method, a software development model that simplifies the stages in the development process so that it is more adaptive and flexible. Extreme programming does not focus on coding but covers several aspects of the system under development, with XP being able to adapt to rapidly changing requirements. The Extreme Programming model is interpreted as a light method in emphasizing intense communication, until the work is interactive and incremental.

This research was conducted on the basis of the written theory of the XP methodology used with changes to the use of datasets in object detection systems with appropriate conditions during a pandemic. Changes to the components in the prototype are also changed with much needed importance in order to achieve better goals

This research begins with the identification of system requirements. There are several more activities that need to be done to complete the overall system development. This includes system design, system implementation, and real-time object identification testing. All of the above will be discussed as follows.

Identifying System Requirements

Identification of system requirements is an important stage to determine all requirements of the system, especially related to its functions and features. The system developed is expected to have the ability to detect the use of masks when people are active in public spaces. This effort is one of the measures to minimize the potential spread of the Covid-19 virus in public spaces. The easiness in implementing the system is also underlined as an expected feature.

From the brief explanation about the expected features of the system, some system requirements can be identified and listed in details as follows.

- □ The system is able to be installed in any public space as well as controlled wirelessly from a remote location. To realize this ability, an internet (Wi-Fi) connection should be provided. The system itself also needs the ability to connect to Wi-Fi (internet connection).
- □ This system is able to capture scenes of community activities in public spaces, especially people passing by. For this purpose, the system should be equipped with the camera as an additional device to obtain the input video.
- □ The system can recognize anyone who is not wearing a mask when passing and caught on camera. An intelligent process is needed by the system so that the process can recognize passers-by who are not wearing masks.

Design system

This design consists of hardware and software. The Software part begins with the use case diagram of the Mask detection process as shown in Figure.2. The image data is taken from video material captured by the video camera which is represented in the “Acquire from camera stream” use case. Recognizing image is processed in “Image Data Process” Use Case before being classified in “Image Classification” use case. Data that is identified as person wearing the mask is forwarded to “Output Image 0” use case and the processing result will be an action and notification that the person is allowed to enter the gate (it is represented by “Please Enter” data). Whereas, the data that is identified as person not wearing the mask is directed to “Output Image 1” use case in which the result of the process is providing notification that the indicated person is not having a permission to enter the gate (“No Entry” data represents this result).

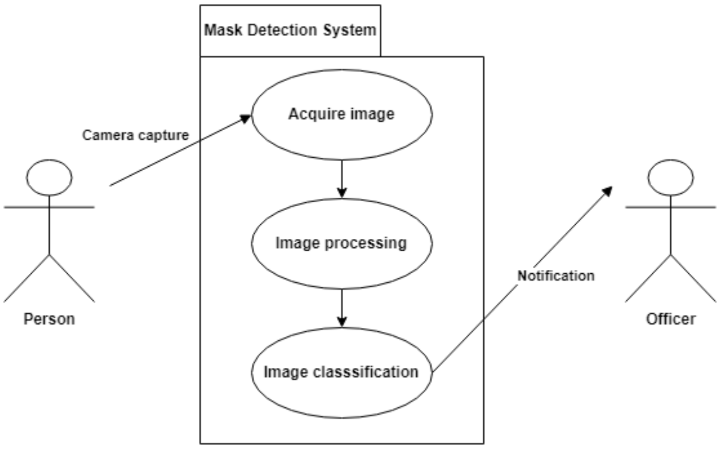

Figure 2 Use Case Mask Detection System

To make it clear the software design, a flowchart diagram is required. Figure 2 and 3 shows the UML diagram of the system. As stated in the Use Case diagram above and Activity diagram below, acquiring image data from the video camera is the first process in this system. All image data captured by the video camera will be gathered here (which is represented with the “Acquire new image from stream camera” process) for further data recognizing process. The next process is to detect and recognize image data of people wearing masks or not. A dataset as a result of the learning process is needed in the process. This dataset is obtained from the process of training the machine to be able to recognize images of people wearing masks using training data and testing data. The process of mask detection is represented in the actions of “Detect mask” and checking whether the mask is detected or not.

The recognition process itself is represented in the following process. Whether the image data of people wearing masks or not must be normalized. This is conducted in the “Normalize Image” process. The normalized image data is then framed through the use of the bounding box function and the confidence level is determined before the prediction is set. The color of the image data needs to be changed to Gray scale for convenience in the next process. The converted Gray-scale image will be processed for recognition which in the end the identified information is extracted and displayed in the resulting image.

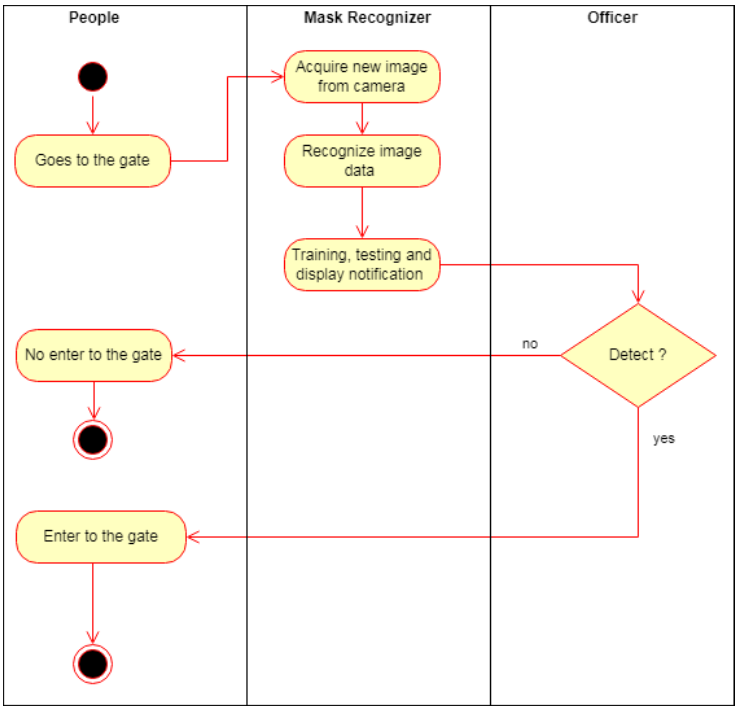

Figure 3 Activity Diagram Mask Detection System

Furthermore, explanation of the face image component processing is provided in order to make it clear the recognizing process stated above. The face component images are processed by the system as illustrated in flowchart (Figure.4). This includes faces framed using the bounding box function and accuracy calculation results. The framed face is taken as input data for the machine learning process and the accuracy data is displayed on the monitor screen as the result of the accuracy calculation in machine learning.

The software section is equipped with the OpenCV platform, for the development of real-time object detection systems. The intelligent process implemented in the software will allow the system to recognize person not wearing the mask. The results of the recognition process are displayed on the monitor screen along with a bonding box that provides the boundaries of the person's face, information on people who are wearing or not wearing masks and accuracy in the form of numbers. The video displayed on the monitor screen indicates that the shooting process has been going well. All information including video is displayed on the monitor screen in real time.

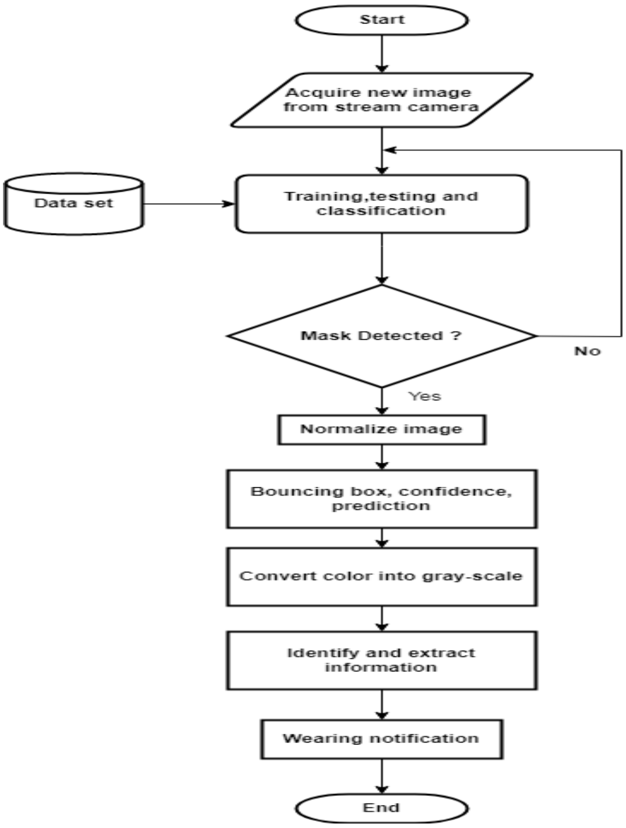
Start
Acquire new image from stream camera
Data set
Training,testing and classification
Mask Detected ?
No
Yes
Normalize image
Bouncing box, confidence, prediction
Convert color into gray-scale
Identify and extract information
Wearing notification
End

Figure 4 Flowchart Mask Detection System

The LCD module is only used to display information that provides an indicator that there are people who are not wearing masks that are captured by the system. Appropriate action related to this disciplinary violation can be taken as a form of preventive measure against the spread of the virus.

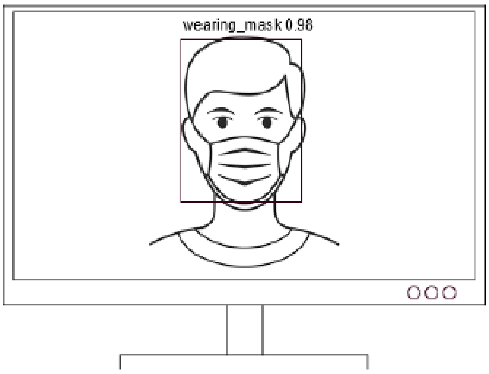

Figure 5 LCD Monitor displayed wearing mask

Information not only comes from the LCD module, but also comes from the Monitor screen that is connected to the server. The bounding box, image class, accuracy, and image are captured by the camera and then all will be displayed on the monitor screen (see Figure 5).

RESULT AND DISCUSSION

The testing effort will be focused on the function of the system in recognizing people faces. In this case, people who walk and move towards the camera will be recognized by their faces, whether wearing a mask or not wearing a mask. A Scenario to carry out tests need to be prepared with the aim of ensuring that the required functions work properly as specified at the beginning of the system design.

The first test aims to ensure that the system is able to recognize the face of walking person approaching the camera. The system needs to frame the person face and process the captured image data in the server running the machine learning process. This process has been described in detail in the design system above (see Figure 4.). The framed face is indicated by a bounding box displayed on the monitor screen.

This face image data is processed to be recognized by machine learning and the result is data that mentions whether the person is wearing a mask. The calculation process to obtain facial image accuracy is also carried out in the machine learning process. The results of facial image recognition and accuracy calculations are sent and displayed on the monitor screen. Figure 9 shows all the results of testing the facial image recognition process. The test results on this image show that the Face of the person captured by the camera is identified as "wearing a mask" and the accuracy is 0.95 or 95%.

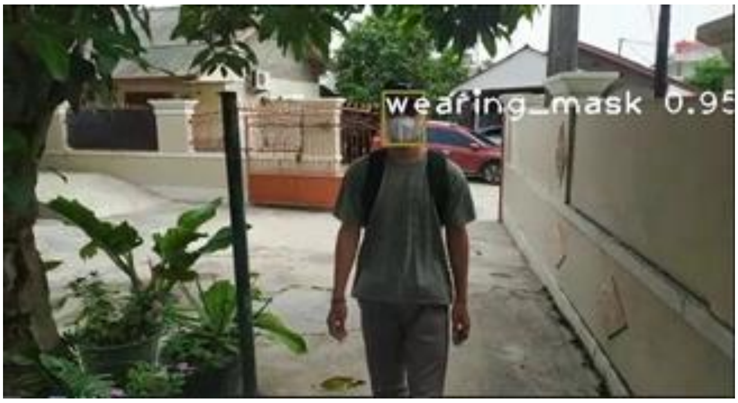

Figure 6 Detected person wears a mask

The recognition process not only provides information regarding whether the identified person wears a mask (Figure 6) and the accuracy level, but also information regarding whether the identified person is allowed to enter the gate (to enter certain public spaces) is provided as a result of the classification process. This process was detail discussed in the system design above (see also Figure 1. and Figure 2). The results of this process are sent and displayed on the LCD screen so that the security authorities can take appropriate action based on this information. The identified person is allowed to enter the gate when the LCD Screen displays the information "Please Enter!" meanwhile the one who are not allowed to enter the gate when the information "No entry!" displayed on the LCD screen. Figure 7 shows the result of this classification process.

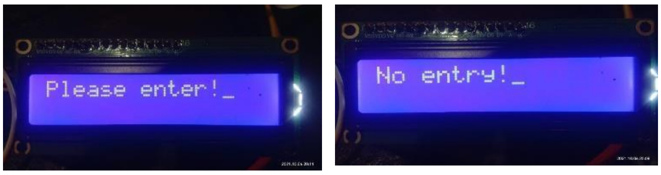

Figure 7 Displayed entry statuses

In summary, the scenarios and results of the system testing can be seen in Table 1. Scenarios and Tests for the face recognition process are listed and represented at the beginning of this table, ranging from taking pictures of people walking towards the camera to recognizing whether someone's face is wearing a mask. This includes a process for calculating the facial recognition accuracy of an identified person. The last scenario and Test listed in the table is for the result of classification process which is displayed on the LCD screen.

Table 1 Result of classification process

Scenario	Expected Result	Evaluation
The Image of walking person approaching the camera is captured.	Captured image of walking person is displayed on the monitor screen.	As expected
The identified person face is framed using a bounding box function.	The person face is framed and displayed on the monitor screen.	As expected
The person face image is taken for recognition process and the result is sent and displayed on the monitor screen. The case is for identified face wearing a mask.	The information “wearing mask” is displayed on the monitor screen when the identified face is wearing a mask.	As expected
The person face image is taken for recognition process and the result is sent and displayed on the monitor screen. The case is for identified face wearing a mask.	The information “not wearing mask” is displayed on the monitor screen when the identified face is not wearing a mask.	As expected
The recognized person face is taken as an input for calculating its accuracy.	The degree of accuracy represented in numbers or fractions is displayed on the monitor screen as a result of this calculation.	As expected
The facial image is taken for the facial image classification process. The information result is displayed on the LCD screen	“Please enter!” or “No entry!” information is displayed on the LCD screen	As expected

CONCLUSION

Calculations to produce accurate facial image data (4000 images) through a machine learning approach. System testing shows that the face of a person wearing a mask is successfully recognized by the system with an accuracy of over 90 percent. The developed system is equipped with the implementation of the Yolov3 algorithm for identification of specific objects. The test results above prove that the algorithm has been implemented and works well and accurately. The information of status whether the identified person is allowed to enter the gate is also successfully displayed on the LCD screen. It is then concluded that the developed system has been working as expected and, in the end, it can be an effort for reducing and avoiding the spread of the covid19 virus in a public space when the system is implemented.

As an early effort to control the spread of the COVID-19 virus, the system built is more than adequate. However, for a perfect system, it is necessary to add other features as an effort to improve system functions so that they are able to handle the problem of the spread of the COVID-19 virus better and more efficiently. Remote control of the system by adopting and implementing mobile and IoT technology can be an effort to improve the system in the future.

ACKNOWLEDGMENT

We thank to Ega Frandika for his help and contribution by preparing prototype hardware and supplying some specific data related to the prototype implementation and testing.

REFERENCES

[1] S. E. Eikenberry *et al.*, “To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic,” *Infect. Dis. Model.*, vol. 5, pp. 293–308, Jan. 2020.

[2] A. Kumar, Z. J. Zhang, and H. Lyu, “Object detection in real time based on improved single shot multi-box detector algorithm,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2020, no. 1, Dec. 2020.

[3] S. Singh, U. Ahuja, M. Kumar, K. Kumar, and M. Sachdeva, “Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment,” vol. 80, pp. 19753–19768, 2021.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 779–788, Dec. 2016.

[5] Z. Liang, J. Shao, D. Zhang, and L. Gao, “Small object detection using deep feature pyramid networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11166 LNCS, pp. 554–564, 2018.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2155–2162, Sep. 2014.

[7] I. D. Apostolopoulos and T. A. Mpesiana, “Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks,” *Phys. Eng. Sci. Med.*, vol. 43, no. 2, pp. 635–640, Jun. 2020.

[8] P. Soviany and R. T. Ionescu, “Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction,” *Proc. - 2018 20th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2018*, pp. 209–214, Sep. 2018.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, Sep. 2014.

[10] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, “Benchmark analysis of representative deep neural network architectures,” *IEEE Access*, vol. 6, pp. 64270–64277, 2018.

[11] N. D. Nguyen, T. Do, T. D. Ngo, and D. D. Le, “An Evaluation of Deep Learning Methods for Small Object Detection,” *J. Electr. Comput. Eng.*, vol. 2020, 2020.

[12] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9908 LNCS, pp. 354–370, 2016.

[13] L. Abraham, A. Urru, N. Normani, M. P. Wilk, M. Walsh, and B. O’flynn, “Hand Tracking and Gesture Recognition Using Lensless Smart Sensors.”

[14] B. Roy, S. Nandy, D. Ghosh, D. Dutta, P. Biswas, and T. Das, “MOXA: A Deep Learning Based Unmanned Approach For Real-Time Monitoring of People Wearing Medical Masks,” *Trans. Indian Natl. Acad. Eng.*, vol. 5, no. 3, pp. 509–518, Sep. 2020.