Face Mask Detection using Convolutional Neural Network

Ronald Mustika Putra
Computer Science Department
BINUS Online Learning
Bina Nusantara University
Jakarta, Indonesia
Ronald.putra@binus.ac.id

Emny Harna Yossy
Computer Science Department
BINUS Online Learning
Bina Nusantara University
Jakarta, Indonesia
emny.yossy@binus.ac.id

Suharjito
Computer Science Department
BINUS Online Learning
Bina Nusantara University
Jakarta, Indonesia
suharjito@binus.edu

Immanuela Puspasari Saputro
Computer Science Department
BINUS Online Learning
Bina Nusantara University
Jakarta, Indonesia
immanuela.saputro@binus.ac.id

Devriady Pratama
Computer Science Department
BINUS Online Learning
Bina Nusantara University
Jakarta, Indonesia
devriady.pratama@binus.ac.id

Teguh Prasandy
Information Systems Department
School of Information Systems
Bina Nusantara University
Jakarta, Indonesia
teguh.prasandy@binus.edu

2023 8th International Conference on Business and Industrial Research (ICBIR) | 979-8-3503-9964-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICBIR57571.2023.10147730

***Abstract*—The current development of artificial intelligence (AI) technology makes it possible to detect and classify objects, such as whether a person is wearing a mask or not, using deep learning technology. The system was built using the Python language for modeling and using the Flutter language. Classification of face masks using the Convolutional Neural Network (CNN) method, and the MobileNetV2 architecture. This model use datasets in image format with Mask and Non-Mask classifications. From the test results, the mask detection model gives good results. Testing objects carried out at the furthest distance of one meter gave good results, while system notifications on mobile application also worked well by sending notifications to mobile application when an object was detected by the CCTV tool and displaying it on mobile application.**

***Keywords*—CCTV, Object Detection, Convolutional Neural Networks, Masker, Raspberry Pi.**

I. INTRODUCTION

Currently, Indonesia and other countries around the world are facing the Coronavirus Disease 2019 pandemic or known as COVID-19. The China National Health Commission reported several cases of pneumonia of unknown cause in the city of Wuhan, China. Then, after that, there was a rapid spread of the virus until January 31, 2020, the World Health Organization (WHO) stated the global health emergency [1]. WHO gave the name of the disease as Novel Corona Virus (Cov-19) or commonly referred to as COVID-19, because research shows a close relationship with the Coronavirus that causes Severe Acute Respiratory Syndrome (SARS), which was later referred to as Severe Acute Respiratory Syndrome Corona Virus. 2 or SARS-CoV-2 [2]. COVID-19 is an infectious disease whose spread has yet to be contained. Therefore, the government urges public awareness in minimizing the risk of spreading, namely by complying with and implementing the 3M movement, namely wearing masks, maintaining distance, and washing hands.

The use of masks can reduce aerosols or virus particles into the surrounding air exhaled by the wearer, thereby reducing the possibility of the spread of the virus. For a long time the use of masks has been recommended, especially for people with influenza or respiratory diseases to prevent the spread of saliva (droplets) [3]. One way to control the spread of COVID-19 is to build an application to detect people using masks.

Mask detection applications can not only be used to control the implementation of health protocols. Before the COVID-19 pandemic, many similar applications had been developed to monitor visitors within a company for security purposes. The system not only uses deep learning for mask detection but can also send notifications in the form of messages on telegrams to related security officers [4]. This study using a Raspberry Pi 3 B+ with limited memory to run video streaming, which is only capable of running 20-32 fps, of course, if we want to use it as CCTV/streaming this is not enough. For the notification itself, because it uses Telegram, the system cannot do history recording for quarterly reporting, because Telegram only sends text message data and images linearly.

In this study, a monitoring application will be developed that can real-time monitor and detect whether someone is wearing a mask or not. This application also sends a notification in the form of mobile application that can be accessed by the security officer concerned. So it is hoped that security officers can monitor in real-time if someone violates Health.

II. LITERATURE REVIEW

A. MobileNet v2

Deep learning itself is a machine learning implementation that aims to imitate the workings of the human brain using an Artificial Neural Network / artificial neural network algorithm that uses metadata as input and processes it using several hidden layers to calculate the output value [5]. One method of deep learning is convolutional neural networks. CNN is a type of neural network that is commonly used in image data. The use of this CNN is to detect and recognize objects in an image [6]. The CNN architecture used in this research is MobileNet v2. MobileNetV2 is the second version of MobileNets (development of MobileNetV1) with the addition of new features, namely linear bottlenecks, and shortcut connections between bottlenecks. The results of accuracy and speed from MobileNetV2 are better than the previous version. This is evidenced in the image classification experiment using ImageNet. In this experiment, MobileNetV2 showed better accuracy results than MobileNetV1 with fewer parameters [7].

B. Rapsberry Pi4

The Raspberry Pi is a credit-card-sized single-board computer or SBC. The Raspberry Pi has been equipped with all the functions as a complete computer, using an ARM SoC (System-on-a-chip) which is packaged and integrated on a PCB (Printed Circuit Board). This device uses an SD card for booting and long-term storage [8]. Raspberry Pi device is a minicomputer that is the size of an ID Card and has the ability to process logical operations like a PC computer. In this system, the raspberry device is used as the main controller of the system whose job is to capture images, send images to the server, manage camera logic operations, and store image meta-data as a database.

C. Previous Researchs

Research conducted uses deep learning, namely Convolutional Neural Network (CNN) to detect whether someone is wearing a mask or not. in this study using a sample image/image of an object where the sample contains sample data of people wearing masks and not wearing masks[9]. The drawback in this study is the distance when detected using a video stream which is only up to a maximum distance of 2.3 meters.

Similar research which was to create a real-time facemask recognition system using deep learning techniques through the Convolutional Neural Network (CNN). The dataset collected contains 25,000 images using a resolution of 224x224 pixels and achieves a 96% accuracy rate for the performance of the trained model [10]. This system creates a notification in the form of an alarm in the form of sound when an object is detected not wearing a mask. Multi-Task Convolutional Neural Network (MTCNN) is used in research for facial recognition [11]. MTCNN is used as a feature extractor and Support Vector Machine (SVM) as a classifier. The system built focuses on facial recognition using masks. The image is given a face image that uses a mask and does not use a mask to recognize someone's face.

With current technological developments, there are many ways to do face recognition, one of which is by using several machine learning technologies. Research conducted created a face mask recognition system using several machine learning technologies such as TensorFlow, Keras, OpenCV, and Scikit-Learn [12]. The method used is to detect the face from the image correctly and then identify whether there is a mask or not. This system is also applied to images of moving objects. This method obtains an accuracy of up to 95.77% and 94.58% respectively on two different datasets in each machine learning.

III. Research Methods

The hardware design in this research consists of designing a CCTV system tool based on raspberry pi4. The type of Rasberry used is the Rasberry pi 4 Model B with 8 GB of RAM and 32 GB of memory. The camera used is PiCamera version 1.3 with 5 MP. The operating system used is Raspberry pi OS (32-bit) latest release. The CCTV system on the device is made using the Python programming language. Mask Detection using OpenCV, Keras/tensorflow and Deep Learning. The notification system is made in the form of mobile apps with the Flutter programming language and utilizes the Firebase Cloud Messaging service for push notifications. For data storage media using the MariaDB Database. Rest API on the backend is built using the PHP programming language. Testing the mask detection system is carried out by testing the accuracy of the mask detection model at 50cm to 1 meter with the position of the object in front view, side view, top view and the number of objects is more than 1 object. Testing the durability of the tool is done by measuring the average temperature of the hardware device when it is turned on for 24 hours.

The research method used to detect masks that will be applied in this study is the Convolutional Neural Network (CNN) method with the MobileNet V2 algorithm. Convolutional Neural Network (CNN) is a type of neural network commonly used in image data. CNN can be used to detect and recognize objects in an image. MobileNets itself is a convolutional neural network (CNN) architecture that can be used for devices that have limited resources such as mobile phones. The advantage of mobilenet is that it has high speed and accuracy, The research method used to detect masks that will be applied in this study is the Convolutional Neural Network (CNN) method with the MobileNet V2 algorithm. Convolutional Neural Network (CNN) is a type of neural network commonly used in image data. CNN can be used to detect and recognize objects in an image. MobileNets itself is a convolutional neural network (CNN) architecture that can be used for devices that have limited resources such as mobile phones. The advantage of MobileNet is that it has high speed and accuracy but still runs on devices with low specifications. MobileNet released its second version in April 2017. Just like MobilenetV1, MobileNetV2 still uses depth wise and pointwise convolution. MobileNetV2 adds two new features: 1) linear bottlenecks, and 2) shortcut connections between bottlenecks. MobileNetV2 was chosen to assist system modeling in detecting objects. In the next development will use the latest version.

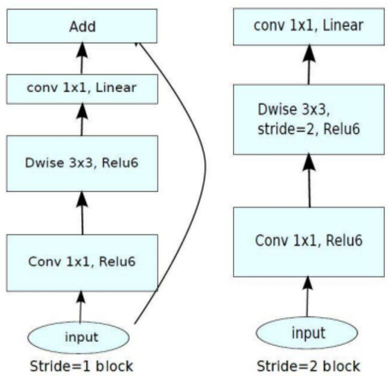


Fig. 1. Residual block on MobileNetV2 [13].

The proposed system is made to design the required system based on the existing problems; the proposal is how to make an application that can not only detect masks but can also send warning notifications to the user. And the application is expected to have a storage database so that in the future if there is a need to issue data in the form of reports or statistical data, the system can display the data because the system will be built in this study uses a storage database. The figure 2 is a flow chart of the proposed mask detection device system.

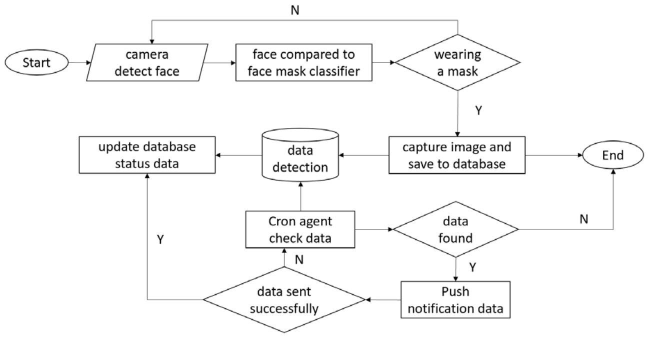


Fig. 2. Flow chart for mask detection and notification system.

The system will detect faces, then compare them with the face mask classifier. The face will be captured and stored in the database. Data detection will update the database data status. Cron agent will check data on data detection and send notification. The figure 3 is a flow chart of the notification application on the proposed mobile apps.

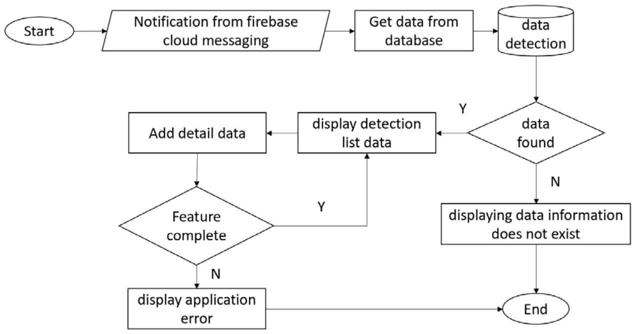


Fig. 3. Flow chart for the system on mobile apps.

The notification comes from firebase cloud messaging to collect data by means of data detection. If the data is not found, the system displays information that data is not available. If data is found, the system will display a list of detection data. Next, the system will display detailed data.

IV. Result and Discussion

A. Software Design

The mask detection system is an IoT-based system that utilizes an artificial intelligence technology where this system can run on raspberry pi devices. This system can classify people who wear masks and those who do not. This system also has a notification designed using mobile apps. Where the results of the data are taken from a database that has been stored by the monitoring tool and for the results of the photo object taken from a hard drive storage. From the results of this study, it is hoped that the creation of a CCTV monitoring system that can detect masks on objects of people, and can send alarm notifications to mobile apps application used by the relevant task force, making it easier to monitor a room/area, because there is no need to monitor for 24 hours. non-stop hours and also help the government's efforts to prevent and control the spread of COVID-19 can be realized and implemented properly.

The software function consists of a list of findings, notifications, and details of findings. The list of findings contains data captured by the detection mask system which is displayed in the form of a list view. This data is in the form of information on the date and time of the violation and image capture of the violation that occurred. Notifications are used as information if there are violations that are caught by the mask detection tool. Notification in the form of a popup on the device and contains information on the number of the latest findings. The details of the findings are used as information if there are violations that are caught by the mask detection device. This data contains information on the time and date of the violation, image capture of the violation that occurred along with the location of the device that captured the object.

In developing a monitoring system, several hardware and software are needed. Hardware consists of a Raspberry Pi 4 with a Pi camera, 32 GB microSD Memory for storage media from Raspberry devices, and a Wifi Router for networking. Software includes Python 3.8, OpenCV, Tensor Flow & Keras, Apache 2.4, PHP 8.0, MariaDB 10.6, Flutter 2.0, Centos 8, and Raspberry OS Buster.

In designing a mask detection system using raspberries with notifications based on mobile apps, there are 2 system design approaches, namely a procedural approach to designing hardware that will read and enter data into the database, and hardware that will be designed using Python language and an OOAD (Object-oriented analysis) approach. and design) on system design on mobile apps that will receive and display data from hardware. Mobile apps will be made using Flutter language and for backend will be made using PHP language. So that in the development of this application, several UML diagrams are used such as use case diagrams, activity diagrams, class diagrams and sequence diagrams. Database design using entity relationship diagrams.

The architectural design for both the CCTV system with mask detection and monitoring applications is as follows:

1) The architectural design of the CCTV system is described as follows:

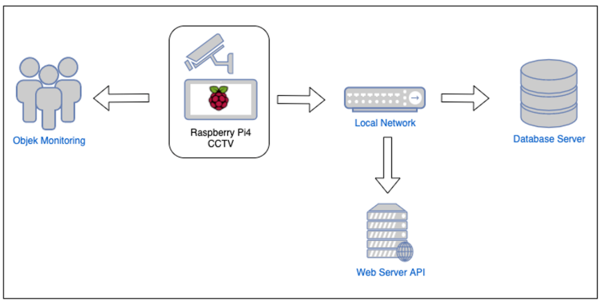


Fig. 4. Monitoring System Architecture.

In the CCTV system architecture there is a camera as a tool to get object data, then the Raspberry Pi will detect the object and perform data processing to find out whether the object is wearing a mask or not. then the object data is stored in file storage, and if it is detected that the object is not wearing a mask, the system will input the data into a database where all are connected to the local network.

2) The design of the monitoring application architecture is as follows:

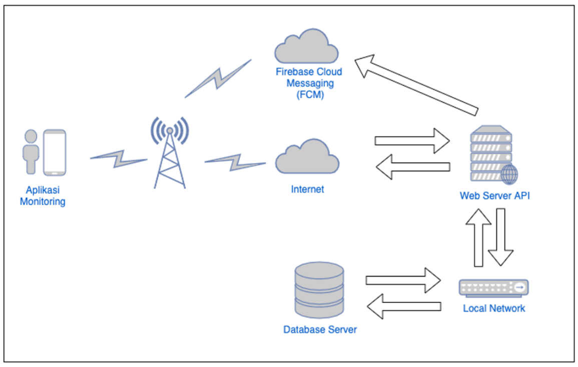


Fig. 5. Monitoring Application Architecture.

In the monitoring application architecture, the client and server are connected via the internet. The client will make a request to the server to get data information. Periodically through the background process the application will check the data to the server to get the latest notification data. When there is data that must be notified, the web api system will push notifications using services from Firebase Cloud Messaging (FCM). And on the server side, data requests from the network are handled by a web server api which will retrieve data from the database and file storage via the local network.

3) CCTV device design

In the CCTV system, the tool used is a raspberry pi4 which is equipped with a camera / pi camera. The tool is placed in an acrylic case. The raspberry pi components used are raspberry pi4 model B, raspberry pi camera rev. 1.3. Fan and heatsink, case acryliv for raspberry and camera, and adapter. CCTV tools are described as follows:

Fig. 6. CCTV System with Raspberry Pi4.

4) Application server design

Application server design Application server consists of API web server, database and support component. The specifications of the api web server used in this study are Intel i5 2.5 GHZ Processor, 2 GB RAM, 50 GB HDD, Linux Operating System, Apache with PHP module. For the specifications of the database server used in this study is Intel i5 2.5 GHZ Processor, 4 GB RAM, 50 GB HDD, Linux Operating System, MariaDB Version 10.6.0. The router is used as a liaison between the local network and the internet network.

B. Dataset

The dataset used is taken from the internet. The dataset is needed because this research uses deep learning methods, so it requires a lot of data to be studied. The dataset is an image taken directly from the Kaggle site because this research is the first step to get a face detection mask model before entering the next stage of system development. There are 1000 datasets separated in the Train, Test and Validation folders. And the train folder is separated into 2 sub folders Mask and Non-Mask with each containing 300 images. The following is an example of a dataset from the Mask and Non-Mask sub folders.

C. Mask Detection System Model

The machine learning model of mask detection in this study uses tools / IDE from Anaconda, namely jupyter-notebook. Where in anaconda we can create a virtual environment first. The following are the steps taken to create a model for mask detection.

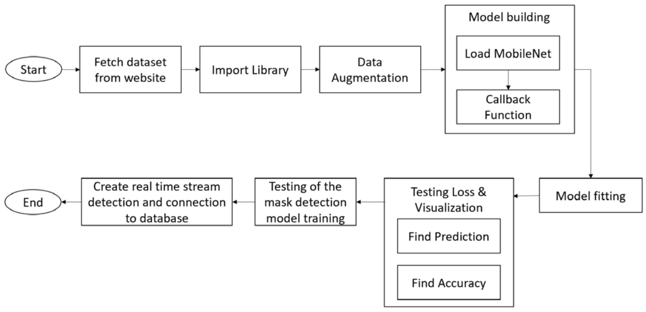
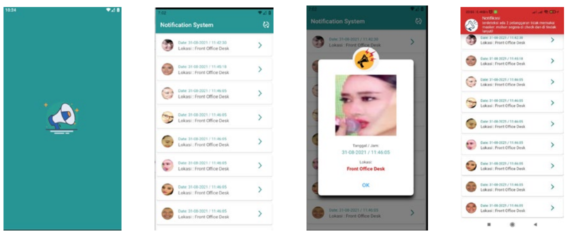


Fig.7. Mask Detection System Model.

D. User Interface

The display of the mask detection notification application consists of a) home page, b) findings list page, c) findings detail page, and d) notification page.



(a) (b) (c) (d).

Fig 8. Flow of notification application user interface from a-d.

E. Evaluation

Evaluation is carried out on hardware devices and mask detectors and evaluation on mobile apps systems. Evaluation on hardware devices checks the temperature of the heat generated by the tool. When the appliance is turned on for 24 hours there is a difference in value when the appliance uses the heatsink and fan as a cooler with the appliance when not using the heatsink and fan. Here's a comparison table.

TABLE I. COMPARISON OF TEMPERATURE VALUES ON DEVICES THAT ARE TURNED ON 24 HOURS

Using the Fan	Using the Heatsink	Temperature
Yes	Yes	35^0C
Yes	No	60^0C
No	No	89^0C

From the value above, it can be seen that the device shows a low temperature by using a cooling fan and heatsink.

In the evaluation of mask detection, testing is carried out on how the system can recognize objects that are wearing masks or not wearing masks when recorded on camera. From the results of making a mask detection model, the accuracy rate reached 99% (0.99) with lost data of 3.8% (0.038). Accuracy reaches 99% because what is being tested is still a division of the training data.

TABLE II. MODEL EVALUATION

Epoch	Loss	Accuracy	Val_loss	Val_acc
1/50	0.5265	0.7711	0.3535	0.8268
2/50	0.1610	0.9613	0.3608	0.8431
3/50	0.0572	0.9912	0.2433	0.9085
4/50	0.0391	0.9894	0.2220	0.9346
5/50	0.0209	0.9965	0.1955	0.9510
		…		
45/50	0.0012	1.0000	0.0407	0.9902
46/50	0.0015	1.0000	0.0404	0.9902
47/50	0.0016	1.0000	0.0402	0.9902
48/50	0.0015	1.0000	0.0398	0.9902
49/50	0.0017	1.0000	0.0393	0.9902
50/50	0.0022	1.0000	0.0389	0.9902

From table above, we know that the accuracy increases starting from the second epoch, and the loss decreases after that. The final evaluation of the model can be seen in the table below.

TABLE III. MODEL EVALUATION

	Precision	**Recall**	**F1-Score**	**Support**
With Mask	0,99	0,99	0,99	306
Without Mask	0,99	0,99	0,99	306
Accuracy			0,99	612
Macro avg	0,99	0,99	0,99	612
Weighted avg	0,99	0,99	0,99	612

The graph of training loss and accuracy can be seen from the following figure where each line is depicted in a different color and for the X axis it is an iteration of the epoch while the Y axis is Loss/Accuracy.

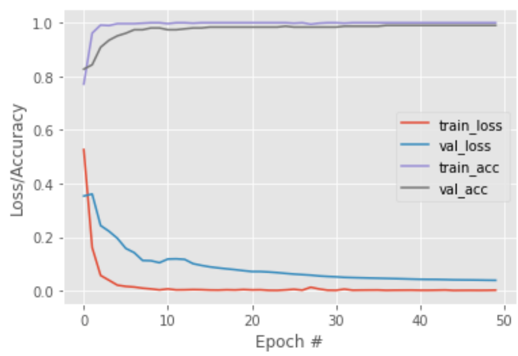


Fig.9. Training loss/accuracy chart.

After the mask detection model is created, the next step is to test it on real-time streaming. Testing of the mask detection tool is carried out at the position of the object as follows:

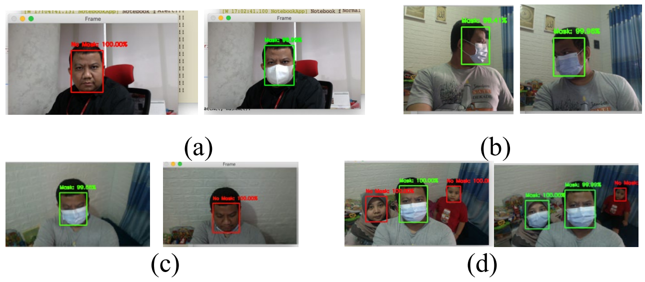


Fig.10. (a)The position of the object in front view, (b) The position looks right and left, (c) View the camera position from above, (d) The camera position from above with the objects in white face mask, a different position and the farthest distance of the object is about 1 meter.

From the results above, it is found that the mask detection model can recognize objects that are wearing masks using a mask of one color variation and not wearing masks in various positions and the number of objects. And the tool can still detect objects well when the object is 1 meter away. When the object is more than 1 meter away, the tool cannot detect the object anymore and tends to be inaccurate, this is because the camera used in this study is a low-resolution camera with a low fps as well.

Tests carried out on mobile apps are using the blackbox method and testing the user interface on the menus contained in the application. Testing for the functions and menus contained in mobile apps application consists of a notification function when the application is not accessed, a notification function when the application is accessed, a splash display when the application is first accessed, a function displays detection data, a function displays details of detection data, a button function refresh, Paging function with lazy load, error handling function when data is not found, error handling function when there is no internet connection. The results obtained by the function run well and can display the desired data.

Evaluation is also carried out on the interface design that uses the Eight Golden Rules [15]. The results obtained include:

1. Strive for consistency The appearance of the page on the application is very consistent with the green color from the start of the application opening to the menu. The layout and letters are also the same on each page,
2. Carter to universal usability: This application is designed to be very easy to use, both for new users and those who are already used to using the application. Buttons are represented by attractive icons that are easy to understand for their functions. so that it can be used by the user quickly to find out and view the contents of the intended page, among others.
3. Offer informative feedback: Based on this rule the application is made using very informative dialogs. from notifications to when the user presses the back button, a dialog will appear that brings up the choice of whether to exit the application or not.
4. Design dialogs to yield closure: Based on this rule, every application design dialog that is displayed leads to a sequence of actions from the system, for example on notifications when the application is closed and if there is notification information and the notification is clicked it will open the application

5. Prevent errors: Based on this rule, the application for the back button has a function where when the user performs the back button, the system will ask first.
6. Permit easy reversal of actions: Every page in the application has been designed for errors to occur to the user, such as when there is a detailed popup from the detection page, an OK button is provided which will make the popup application close. Likewise, if the user clicks on the background, the application popup will close.
7. Support internal locus of control: In this application the user has control over all actions taken either when navigating or pressing buttons on the page.
8. Reduce short-term memory load: The application uses a cache system on the device, this is used in the page paging feature on pages that display a list of notifications.

Other tests carried out on the service cron job are to ensure that the service runs well and smoothly and performs the specified command. In this case the command given to the service is to access the URL address http://127.0.0.1/pushnotif/pushnotif.php every 5 seconds every 1 minute.

V. CONCLUSION

The conclusions that can be drawn from the results of this research object detection are the results of the study, the results of the deep learning model with MobileNetV2 used obtained an accuracy of 99% and data loss of 3.8% with 1000 datasets; the results of testing the detection tool from a distance of 50 cm - 1 m with the number of objects more than 1 object, the tool is found to function properly with a good level of accuracy, but because it uses a low-resolution camera so that for a distance of more than 1 meter the tool cannot detect object well; the results of the test on the tool that is turned on for 24 hours, shows the tool using a cooler/fan is better than not using a fan; the test results of the notification agent system are found to be running well where the scheduling agent/cron scheduler that is set runs every 5 seconds to check into the database and send notifications to Firebase Cloud Messaging; blackbox testing of the application is also good where all functions run properly.

Suggestions that can be given for further research are as follows: increasing the number of mask datasets with various use positions and types of masks, this is to train the MobileNet model so that it is expected to achieve high accuracy and low loss values; added face recognition function to detect repeated object images, so that notifications sent with the same object do not occur; using a CCTV camera with a high resolution and fps, so that object detection can be recognized well over long distances; develop a monitoring management system application for user management settings and tool management; develop a dashboard application that can issue reports periodically using the date filter method or the device filter method; using real-time gateway socket technology for push notifications, this is a development of the currently used scheduling/cron technology.

ACKNOWLEDGMENT

Special thanks should be given to Bina Nusantara University BINUS Computer Science Online Learning, our research supervisor for professional guidance and valuable support, useful and constructive recommendations on this project.

REFERENCES

[1] D. Handayani, D. R. Hadi, F. Isbaniah, E. Burhan, and H. Agustin, “Corona Virus Disease 2019,” J. Respirologi Indones., vol. 40, no. 2, pp. 119 – 129, 2020.

[2] P. Zhou, X. Yang, X. Wang, and E. Al., “A pneumonia outbreak associated with a new coronavirus of probable bat origin,” Nature, vol. 579, pp. 270–273, 2020.

[3] C. C. Leung, T. H. Lam, and K. K. Cheng, “Mass masking in the COVID-19 epidemic: people need guidance,” Lancet, 2020, doi: https://doi.org/10.1016/S0140-6736(20)30520-1.

[4] Lambacing, Musakkarul, and Ferdiansyah, “Rancang Bangun New Normal Covid-19 Masker Detektor Dengan Notifikasi Telegram Berbasis Internet Of Things ,” Dinamik, 2020.

[5] P. IEEE et al., “Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask,” 2017 30th SIBGRAPI Conf. Graph. Patterns Images Tutorials, pp. 17–41, 2018, [Online]. Available: https://ieeexplore.ieee.org/document/8250222.

[6] N. Shukla, Machine Learning with TensorFlow, 1st ed. Manning, 2018.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” IEEE Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018.

[8] Bambang Yuwono, S. P. Nugroho, and H. Heriyanto, “Pengembangan Model Public Monitoring System Menggunakan Raspberry Pi,” J. Inform. dan Teknol. Inf., vol. 2, no. 2, 2015, [Online]. Available: http://jurnal.upnyk.ac.id/index.php/telematika/article/view/1409.

[9] [9] [9] T. S. N. P. Putri, M. Al Fikih, and N. Setyawan, “Face Mask Detection Covid-19 Using Convolutional Neural Network (CNN),” Semin. Nas. Teknol. dan Rekayasa 2020., vol. 6, 2020, doi: https://doi.org/10.22219/sentra.v0i6.3801.

[10] Sammy V. Militante and N. V. Dionisio, “Deep Learning Implementation of Facemask and Physical Distancing Detection with Alarm Systems,” 2020 Third Int. Conf. Vocat. Educ. Electr. Eng., 2020, [Online]. Available: https://www.scienceopen.com/document?vid=b12e9114-03c7-413f-8693-1c8282ba0f1b.

[11] M. S. Ejaz and M. R. Islam, “Masked Face Recognition Using Convolutional Neural Network,” 2019 Int. Conf. Sustain. Technol. Ind. 4.0, 2019, doi: 10.1109/STI47673.2019.9068044.

[12] S. M. Almufti, R. B. Marqas, Z. A. Nayef, and T. S. Mohamed, “Real Time Face-mask Detection with Arduino to Prevent COVID-19 Spreading,” Qubahan Acad. J., vol. 1, no. 2, 2021.

[13] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted Residuals and Linear Bottlenecks,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 4510-4520., 2018.

[14] Kaggle, “Covid Face Mask Detection Dataset,” Kaggle.com, 2021. https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset (accessed Oct. 10, 2021).

[15] W. E, “Shneiderman’s Eight Golden Rules Will Help You Design Better Interfaces,” 2020. https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces.