Jetson Nano-Based Mask Detection System with TensorFlow Deep Learning Framework

Muhammad Luqman Bukhori[1], Erwan Eko Prasetiyo[2]

[1] Teknik Dirgantara, Sekolah Tinggi Teknologi Kedirgantaraan, Jl. Parangtritis KM 4.5 Druwo, Bangunharjo, Sewon, Bantul, 55143 (tel.: (0274) 418 248, 439 6163; email: m.luqman@sttkd.ac.id)
[2] Aeronautika, Sekolah Tinggi Teknologi Kedirgantaraan, Jl. Parangtritis KM 4.5 Druwo, Bangunharjo, Sewon, Bantul, 55143 (tel.: (0274) 418 248, 439 6163; email: erwan.eko@sttkd.ac.id)

[Received: 8 September 2022, Revised: 20 October 2022]
Corresponding Author: Muhammad Luqman Bukhori

ABSTRACT — Indonesia is one of the countries experiencing COVID-19 impacts. Various measures have been conducted to prevent the spread of this virus. One of the efficient measures to prevent this impact is by implementing a strict health protocol and proper mask-wearing. Mask-wearing monitoring continues to be carried out in office buildings, supermarkets, and other public spaces. The supervisor's role is indispensable in supervising proper mask-wearing. However, a supervisor has limitations in conducting supervision, creating a gap for people not to comply with mask-wearing rules properly. Therefore, it is necessary to have a system that works automatically to assist supervisors in monitoring proper mask-wearing. This paper aims to design a computer vision capable of detecting whether or not a person wears a mask using the TensorFlow deep learning framework. TensorFlow is used for its efficiency in processing digital image data. The classification of digital image data in TensorFlow uses a Keras deep learning structure. As a result, it is lightweight and can be used on embedded devices such as Jetson Nano to detect mask-wearing in real time. The stages of a mask detection system consisted of image dataset collection, feature extraction, data separation, modeling, model training, and model implementation. TensorFlow deep learning framework processed image data directly through a webcam. When the camera captured the object of the person not wearing the mask properly, the monitor screen displayed a red box on the face. The sign can help the supervisor when conducting supervision. The test results show that the system successfully correctly detects unmasked people, with an accuracy of 99.48%. In addition, the system also managed to detect people wearing masks properly and got an average accuracy of 99.12%. The monitor displays a green box on the face when the detected person properly wears a mask.

KEYWORDS — Computer Vision, Jetson Nano, Mask Detection, Deep Learning, Keras, Framework, TensorFlow.

I. INTRODUCTION

COVID-19 is an infectious disease caused by SARS-CoV-2. People who contract COVID-19 will experience mild to moderate symptoms and can recover without special treatment. However, some people can become seriously ill and require medical assistance, and some cannot survive [1]. On the other hand, the number of COVID-19 infection cases is increasing. Data from the end of July 2022 showed that COVID-19 cases in Indonesia had reached more than 6 million cases, while globally, there have been more than 550 million cases [2]. COVID-19 can spread from an infected person's mouth or nose through small fluid particles when the person coughs, sneezes, speaks, sings, or breathes.

The spread of COVID-19 can be controlled by implementing health protocols, including maintaining distance, wearing masks, washing hands frequently, avoiding crowds, vaccinating, and maintaining hygiene. Medical masks are very effective in preventing the transmission of COVID-19 because they can filter particles up to 95% [3]. However, a study shows that many people ignore or are reluctant to wear masks in public areas and indoors [4]. Attention must be paid to the behavior of those who disregard the implementation of this health protocol. The situation can endanger others because it increases the COVID-19 transmission risk [5]. Prevention efforts have been carried out by providing hand washing stations, installing thermal cameras to check body temperature, and placing special officers to supervise mask-wearing and health protocol enforcement [6]. As there are not enough supervisors to keep an eye on everyone's mask use, it is still possible for people to disobey the mask-wearing rules in public places, especially indoors.

An artificial intelligence system is needed to detect masked people in real time by considering several things related to controlling health protocols in public areas. Computer vision is the answer to this system because it can be used for tasks such as object recognition, visual tracking, semantic segmentation, to image restoration [7]. In previous research, computer vision has been applied to human object detection systems to build robotic interactions with humans [8]. Data was detected based on body parts, such as the head, shoulders, arms, and legs, with multiple gesture conditions. The experimental results showed that two datasets (INRIA and Caltech) provided good effectiveness, as previously determined. In addition, real-time detection of multiple pedestrian objects has also been carried out [9]. The proposed method decides whether or not the object detected in the current frame is the same as that in the previous frame, as well as updates the coordinates of some objects and the corresponding histogram during the tracking process. The experiment results show that the proposed dual object tracking method outperforms the existing method in challenging situations with partial occlusion.

In the COVID-19 pandemic era, there have been several studies that have developed object detection systems to detect people wearing face masks [10]. The face mask detection was built using deep learning techniques. This technique has been proven to gain an accuracy of up to 98%. Reference [11] states that in identifying masked persons, it is necessary to collect up to thousands of image data, and it is necessary to separate the class of facial image data. Another system was developed using a simple convolutional neural network (CNN) with Raspberry Pi 4 module [12]. The experiment's results showed that using CNN, masked facial recognition can be carried out with an accuracy of 97.67%. CNN is a type of deep neural network architecture because of the high network depth and is extensively applied to image data [13]. Some other types of deep neural network architectures are artificial neural networks (ANN) [14], recurrent neural networks (RNN) [15], long short-term memory networks (LSTM), as well as self-organizing maps (SOM). These architectures have several challenges, including neural network obscurity, data quality assurance, data security, and artificial intelligence production classes [16]. Each has its disadvantages and advantages, but the architecture suitable for computer vision in processing images (images or videos) is CNN [17] because it can classify the smallest parts of the interconnected nodes [18]. Moreover, it has been used by several deep learning frameworks, one of which is TensorFlow [19]. TensorFlow has been packaged into a Python library that can be used in machine learning [20] because it has complex resources and requires a relatively short time to identify masked or unmasked people. Reference [21] concluded that Tensorflow-Keras could detect masked faces with an accuracy of 93% by using the Raspberry Pi 4 module device. Based on the conducted analysis, there is a lack of performance in the employed device. The benchmark on the Raspberry Pi can only overcome 8.1 FPS, while another device, namely the NVIDIA Jetson Nano, can overcome 11.4 FPS [22].

There are various methods in studies on mask detection systems. Mask detection testing on the face was successfully carried out using the YOLO algorithm [23]. YOLO is an algorithm that uses the single-stage detection (SSD) method. It is different from the multi-stage detection (MSD) owned by CNN [24], R-CNN, Faster R-CNN [25], and Mask R-CNN. It was applied in machine learning using mini PCs such as the Raspberry Pi module [26]. However, the results were less effective, and the detection speed was very slow because it relied solely on the central processing unit (CPU) when processing. This study aims to design a mask detection system using Jetson Nano, an alternative to mini PCs based on internal graphical processor units (GPUs).

This paper presents computer vision learning techniques assisted by OpenCV, Keras, and TensorFlow libraries. The developed computer vision used the NVIDIA Jetson Nano Developer Kit. This device has a GPU embedded in it to handle and process all digital image data. Its small size enables this device to be placed in various locations to record real-time videos. Subsequently, the Logitech C920 camera assisted this implementation as a source of high-quality image data. The image data captured by the camera would be processed using Phyton libraries, i.e., TensorFlow and Keras. Those two libraries would process all incoming videos and image data and identify everyone who was not wearing a mask properly.

The main objectives of this study are: 1) developing a computer vision that can detect people with or without masks; 2) applying an identification method that produces visual information in the form of a red box for people who are not wearing a mask and a green box for those who use the mask properly; 3) being able to identify the number of people detected wearing masks or not; and 4) conducting mask detection testing using different mask colors.

This study presents four main sections as follows. The introduction section describes the background of the problem, research gaps, and objectives. The methodology demonstrates the solution to research problems related to detection systems on embedded devices and deep learning framework stages. The results and analysis section describes the model detection testing and detection through a live webcam. The conclusions convey the findings obtained based on the performed analysis. Finally, acknowledgment is addressed to several parties who participated in and supported the implementation of this research.

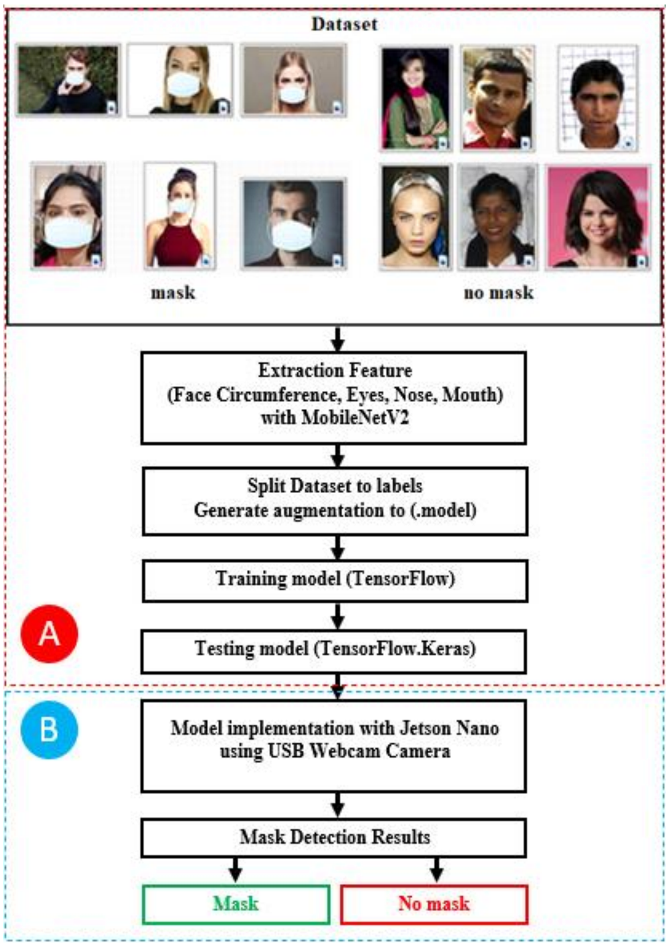


Figure 1. Developed flowchart.

II. METHODOLOGY

CNN is part of an artificial neural network commonly used in image data recognition and processing [27]. The CNN algorithm has neurons designed to work like the frontal lobe, particularly the visual cortex areas of human and animal brains [28]. In previous studies, computer vision was used to extract characteristics to manually achieve better classification. The use of CNN can achieve classification goals by producing convolution layers, dimensionality reduction, to facial feature extraction.

A. THE DEVELOPED MODEL

The applied learning algorithm uses the MobileNetV2 model. MobileNetV2 is a development of CNN's previous architecture model called MobileNetV1. MobileNet is one of CNN's architectures designed to address the need for high computing [29]. The fundamental difference between the MobileNet architecture and other CNN architectures is the use of convolution layers with a filter whose thickness is almost the same as the input image. MobileNet divides convolutions into depthwise convolution and pointwise convolution.

Depthwise convolution and pointwise convolution are notions of separable depth and spatial dimensions of filters. For example, Sobel filters are used in image processing to detect edges. The filter used has nine parameters with a convolution

Figure 2. Training loss chart appearance and model accuracy.

TABLE I
MODEL NETWORK EVALUATION

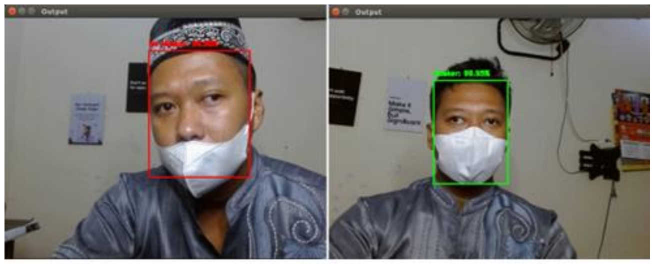
	Precision	Recall	F1-score	Support
with_mask	93%	100%	99%	413
without_mask	100%	93%	97%	498
Average	97%	97%	98%	
Total				911

of 1 × 1 in terms of the channel. MobileNetV2 has two additional features: 1) linear bottlenecks and 2) shortcut connections between bottlenecks. Reference [30] mentions that, in the bottleneck section, there are inputs and outputs between models. At the same time, the inner layer encapsulates the model’s ability to convert inputs from lower-level concepts to higher-level descriptors. As with residual connections in traditional CNNs, shortcut bottlenecks allow for faster training and greater accuracy. The developed model began by loading a dataset for mask detection. Python deep learning libraries used for data preparation are OpenCV, Keras, and TensorFlow. This library functions to train classifications with MobileNetV2, as shown in Figure 1.

There were two stages in the model introduction: training with the TensorFlow library and testing with the TensorFlow.Keras library [31]. Before the training began, all datasets needed to be extracted first based on facial features, namely the face, eyes, nose, and mouth circumferences. The pre-trained datasets were directly saved as model data with a *.model* extension. At the testing stage, the previously saved model was called using TensorFlow.Keras library. The model implementation was applied to Jetson Nano embedded devices via a webcam to read video images in real-time. The results showed as masked if someone was caught on camera wearing a mask. Conversely, if not, the results showed as no-mask.

B. IMAGE DATASETS

The developed detection system has several stages. The initial stage of the system was to collect datasets in the form of images of masked people and non-masked people. The collected dataset consisted of some data already on the internet and additional image data from a model. There was no standard number specified in the determination of this dataset. The most important thing about the dataset was to categorize the type of image data used, wearing a mask covering the nose, not covering the nose, or not wearing a mask at all. This study used a total of 1,360 image data consisting of 680 image data of objects wearing masks, 120 image data of objects wearing



(a) (b)

Figure 3. Results of mask detection testing on models wearing masks, (a) without covering the nose, (b) covering the nose.

masks without covering the nose, and 560 image data of objects without wearing masks. After all the data was collected, the data were divided into two groups: with and without masks. The with-mask group category was image data in which the objects properly wore a mask by covering the mouth and nose. The without-mask category was image data in which the objects were without a mask and an image in which the object wore a mask but did not cover the nose. The system could recognize all collected image data.

C. FEATURE EXTRACTION

Feature extraction was the second stage of the developed model. The image data were resized to 224 × 224 pixels, then converted to array data using MobileNetV2 encoding. The result obtained was the coordinate point of the region of interest (ROI) of the face in the image data obtained by calculating the circumference of the face, mouth, nose, and eyes. The data known as the ROI coordinates were then saved into an array to be used in the loop function. The data was saved into labels according to predefined categories. The following is a snippet of the script for creating data labels.

```
# perform encoding on the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)
```

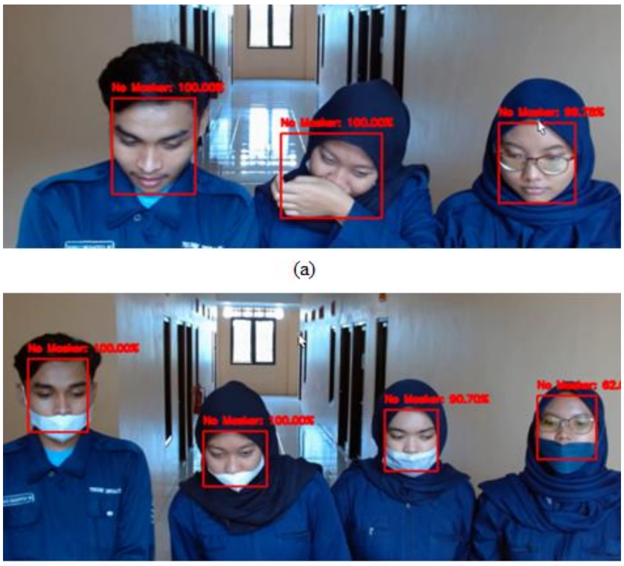
Since array data can only read numeric data, predefined categories were converted into numerical form. The number 0 was a category of labels without masks, while the number 1 was a label category for masked image data.

D. DATA SEPARATION AND MODELING

This stage was a unity of the second stage. The total data tested at this stage was 20%, while the remaining 80% was used for data training. Model creation has several stages: training the image generator, adding model parameters, compiling, training the model, and saving the model for further prediction with the Jetson Nano device. The following is a snippet of a label creation program from a dataset to train an image generator.

```
# partition the data into splits using 80%
 # training and 20% for testing
(trainX, testX, trainY, testY) =
   train_test_split(data, labels,
   test_size=0.20, stratify=labels,
   random_state=42)

# construct the training image generator
 # for augmentation
aug = ImageDataGenerator(rotation_range=20,
   zoom_range=0.15, width_shift_range=0.2,
   shear_range=0.15, height_shift_range=0.2,
   horizontal_flip=True,
   fill_mode="nearest")
```



(a)

(b)

Figure 4. Mask detection testing result with the model, (a) without a mask and (b) with mask without covering the nose.

E. MODEL TRAINING

Model training was performed to obtain a single file as a *filename.model*. This file served as a face detection model to determine whether or not the image taken from the camera was a masked image. The training was conducted directly with some settings through the Jetson Nano Developer Kit 4GB device. The number of iterations was determined by 50 iterations and a batch sample (BS) of 32 data. The results of the model evaluation are presented in Table I and also shown by the graphic display in Figure 2. Equation (1)-(3) can be used to calculate the evaluation of model training results [32].

$$Precision = \frac{[TP]}{[TP + FP]} \quad (1)$$

where TP is a true positive, TN is a true negative, and FP is a false positive.

Precision is a metric showing the number of expected positive values or values that are actually true. For example, if TP is worth 1 and FP is worth 1, the result obtained for the actual value is 0.5. Recall statistics are used as a quantity of an algorithm’s ability to classify all positive cases to make it easier to remember the proportion of correctly identified positives. Mathematically, recall is written in (2).

$$Recall = \frac{[TP]}{[TP + FN]} \quad (2)$$

where FN is the false negative.

$$F1\ score = 2 \times \frac{[Precision \times Recall]}{[Precision + Recall]}. \quad (3)$$

The $F1\ score$ test accuracy was quantified. This evaluation step provided the most accurate findings against a balanced dataset. This stage was the verification stage of accurate predictions. A positive value in the equation indicated that the image inputted on the label truly corresponds to a predetermined label’s prediction. Meanwhile, the negative image was actually the image corresponding with the correct category but got the wrong prediction.

Figure 2 shows the results of the model training evaluation against the accuracy with unreadable data. The graph shows that the accuracy data displays values between 0.8 and 1. The accuracy obtained is outstanding because the value is more than 0.8. Data loss is data that should not be visible above the number 0.2. Figure 2 shows that the data loss is almost toward the value of 0. These results prove that the training can correctly identify the model image.

TABLE II
UNMASKED DETECTION PREDICTION TESTING

No	Number of Models	Live Camera Detection Testing		
		Number of Faces	*Detection Prediction*	*Accuracy*
1	1 people	1 red box	No Mask	100.00%
2	2 people	2 red boxes	No Mask	100.00%
3	3 people	3 red boxes	No Mask	99.93%
4	4 people	4 red boxes	No Mask	99.90%
5	5 people	5 red boxes	No Mask	100.00%
	Average Accuracy			99.97%

F. MODEL IMPLEMENTATION

The model implementation was applied to Jetson Nano devices. This device is a portable minicomputer with a relatively small size. NVIDIA created Jetson Nano to develop a small and resilient embedded system running multiple neural networks applications, such as image classification, object detection, segmentation, and speech processing [33].

The specification of the employed Jetson Nano was the Jetson Nano Developer Kit 4GB. Jetson Nano is an affordable kit. Despite its affordability, it has a quality that is capable of image processing, armed with a Maxwell 128-core GPU, ARM A57 quad-core CPU @1.43GHz, with maximum FHD video encode and decode capability (1080p).

The model implementation on the mask detection system device would work after an active webcam detects a person. If a face was detected, reprocessing was done by resizing the pixel size of the image, converting it to an array matrix, then processing it using MobileNetV2. After that, the input model was saved to predict the comparability data with the model that had been subjected to a previous training process. If the prediction obtained was more dominant wearing a mask, the result displayed is a green box that says “Mask”. If the prediction obtained was more dominant without wearing a mask, the displayed result was a red box that said “No Mask”.

III. RESULTS AND DISCUSSION

A. MASK DETECTION TESTING ON MODELS

Model testing from the dataset needed to be performed to prove that the processed prediction results actually produced valid test data. Based on Figure 3, the test results on the previous model data show that models wearing masks without covering the nose are recognized by the system as non-masked. It is in accordance with the applied method. Models wearing masks by covering the nose are also correctly recognized, that is, properly masked.

Figure 3(a) shows that the system has successfully detected an image of a model wearing a mask without covering the nose with the prediction that the model is not wearing a mask, marked with a red box labeled “No Mask”. The test was performed five times and obtained an accuracy and precision of 99.18%. The trained algorithm successfully indicated that any model who did not wear a mask or wear a mask but did not cover their nose was considered not wearing a mask.

Figure 3(b) shows that the image of the model wearing a mask with a nose covered is successfully detected by the

TABLE III
DETECTION PREDICTION TESTING OF WEARING A MASK WITHOUT COVERING THE NOSE

No	Facial Conditions	Number of Models	Number of Mask Colors	Live Camera Detection Testing		
				Number of Faces	*Detection Prediction*	*Accuracy*
1	Wearing a mask without covering the nose	1 people	1 color	1 red box	No Mask	100.00%
2		2 people	2 colors	2 red boxes	No Mask	100.00%
3		3 people	3 colors	3 red boxes	No Mask	98.80%
4		4 people	3 colors	4 red boxes	No Mask	98.18%
5		5 people	4 colors	5 red boxes	No Mask	98.00%
	Average accuracy					98.99%

TABLE IV
DETECTION PREDICTION TESTING OF WEARING A NOSE COVERING MASK

No	Facial Conditions	Number of Models	Number of Mask Colors	Live Camera Detection Testing		
				Number of Faces	*Detection Prediction*	*Accuracy*
1	Wearing a mask and covering the nose	1 people	1 color	1 green box	Mask	99.80%
2		2 people	2 colors	2 green boxes	Mask	99.84%
3		3 people	3 colors	3 green boxes	Mask	98.87%
4		4 people	3 colors	4 green boxes	Mask	98.55%
5		5 people	4 colors	5 green boxes	Mask	98.54%
	Average accuracy					99.12%

system with the model prediction wearing the mask correctly, which is marked with a green box labeled “Mask”. The test was carried out five times and obtained an accuracy and precision of 99.85%.

B. MASK DETECTION TESTING THROUGH A LIVE CAMERA

In this test, the tested models were those not previously used as a training dataset. In other words, the model used was purely someone else. Two criteria categorized mask detection testing on a live camera. The first criterion was without wearing a mask, and the second criterion was wearing a mask. In the first criterion, two conditions stated that people were not masked, namely when people were completely unmasked, as in Figure 4(a), and people wearing masks without covering their noses, as Figure 4(b) shows.

Figure 4(a) shows the results of detection testing without wearing a mask with three models, namely one male and two females. The test results were obtained in the form of two models declared non-masked with 100% accuracy and one other model was declared non-masked with an accuracy of 99.70%. Under these conditions, all models tested were declared unmasked, according to the actual circumstances. The testing data are presented in Table II.

Figure 4(b) shows the detection testing results of people wearing masks without covering their noses. There are four models detected and declared non-masked with a box marked red. The two models on the left (male and female) show that the accuracy result is 100% unmasked, while the other two models show different accuracy results, i.e., the third model of 90.70% and the fourth model of 62%. The accuracy results in this fourth model indicate that the model and the object detected to have a similarity below the average when the system calculates the process. Both the conditions in Figure 4(a) and Figure 4(b), which have been tested, successfully show that all models are not wearing masks, even though in the actual condition, they are wearing masks but without covering the nose, as shown in Table III.

A total average accuracy of 99.48% was obtained based on the average prediction accuracy of detections of models that do not wear masks and models that wear masks without covering the nose. All tests showed that models detected directly using cameras and models that had gone through previous training both showed the correct prediction, i.e., the result of the detection of “No Mask.”

Further testing is shown in Table IV, where all models wear masks correctly until the mask covers their noses. This test was carried out five times on each number of models. The first test was carried out by one model wearing a white mask. The experimental results showed that the system could detect people wearing masks correctly and the average detection accuracy obtained was 99.80%.

The second test was carried out by two models wearing white and gray masks. The test results showed that the system could detect people wearing masks, marked with green boxes, with an average accuracy of 99.84%.

The third test was conducted by three models with three different mask colors. The results showed that the system could detect masks correctly, with an average accuracy of 98.87%. Four models conducted the fourth test, but two wore masks of the same color, which were white. The performed tests successfully performed a detection with an average accuracy of 98.87%.

Five models performed the fifth test by applying four different mask colors. The color of the worn mask was white, worn by two people, gray by one person, black by one person, and light blue by one person. The test results showed that the system successfully detected the mask, marked by the appearance of five green boxes, with an average detection accuracy of 98.54%. This success percentage was obtained from models consisting of two male models and three female models. Each model wore a mask of a different color. The first model was declared masked with an accuracy of 68.56%, while in another model, the obtained accuracy was more than 90%. Based on the results of the performed tests, all models could be detected wearing masks, even though the worn masks have different colors. Based on the calculations of the five tests carried out, the average accuracy obtained in the system was 99.12%.

The results of detection tests from all predefined categories were successfully performed. The next test is to detect masked and unmasked models in one condition, i.e., simultaneously in front of the camera. Masked and unmasked models can be correctly identified. This test had five models, i.e., two male and three female models. Testing successfully detected one of the female models not wearing a mask, with an accuracy of 100%. Meanwhile, four other models were detected that be wearing masks correctly.

The system developed using Jetson Nano could process live image data with a processing time of up to 0.114 seconds. This data was obtained from the average of each FPS of an image from a web camera. This processing speed was much higher than the process on other minicomputer devices. For example, a previous study using the Raspberry Pi 4 minicomputer 4 version 4GB took a processing time of approximately 7 seconds. Therefore, it can be stated that this Jetson Nano device performs better for processing and identifying objects directly.

IV. CONCLUSION

A device using Jetson Nano to detect masked and unmasked objects or people has been successfully designed and manufactured. Testing showed that the tested model, consisting of men and women, was successfully detected with the TensorFlow deep learning framework. The testing of the non-masked model was successfully performed with an average detection accuracy of 99.48%, with the number of models tested by as many as five people. Masked model testing was also successfully conducted with an average detection accuracy of 99.12% for the five people detected. The masks worn by the model consisted of black, gray, white, and light blue masks. TensorFlow can correctly detect all masks.

The study has only conducted testing to detect people with or without masks. There needs to be further research related to the accuracy of the range that can be detected by the system based on outdoor and indoor conditions. In the following research, it is necessary to develop tests with the range of the device to the detected model in order to get optimal predictive results.

CONFLICT OF INTEREST

During conducting the research and the writing of this paper entitled "Jetson Nano-Based Mask Detection System with TensorFlow Deep Learning Framework", the writing team had no conflict of interest with any party.

AUTHOR CONTRIBUTION

The following is a division of each author's contributions to conducting this study. Conceptualization and implementation of the hardware program, Muhammad Luqman Bukhori; device designer, Erwan Eko Prasetiyo; writing −drafting of the original drafts, Muhammad Luqman Bukhori and Erwan Eko Prasetiyo; data collection, Muhammad Luqman Bukhori.

ACKNOWLEDGMENT

A word of gratitude was conveyed to the Education Fund Processing Institute of the Ministry of Finance and the Directorate General of Higher Education, Ministry of Education, Culture, Research, and Technology Number: PRJ-79/LPDP/2021 and 6/E1/HK.02.06/2021 for financial assistance and support in the implementation of the Scientific Research Program. The researcher also expressed his gratitude to Sekolah Tinggi Teknologi Kedirgantaraan (STTKD) Yogyakarta for providing research funding support in accordance with contract number Skep/63/XII/2021/STTKD. Gratitude is also extended to fellow cadets and employees involved as a testing model and all parties who participated in preparing and supporting the availability of research testing sites.

REFERENCES

[1] (2021) "Coronavirus Disease (COVID-19): How Is It Transmitted?" [Online], https://www.who.int/news-room/questions-and-answers/item/coronavirus-disease-covid-19-how-is-it-transmitted, access date: 6-Jul-2022.

[2] (2022) "Virus Corona (COVID-19)," [Online], https://news.google.com/covid19/map?hl=id&mid=%2Fm%2F03ryn&gl=ID&ceid=ID%3Aid, access date: 6-Jul-2022).

[3] (2020) "Anjuran mengenai penggunaan masker dalam konteks COVID-19," [Online], https://www.who.int/docs/default-source/searo/indonesia/covid19/anjuran-mengenai-penggunaan-masker-dalam-konteks-covid-19-june-20.pdf, access date: 6-Jul-2020.

[4] H.E. Siahaineinia and T.L. Bakara, "Persepsi Masyarakat tentang Penggunaan Masker dan Cuci Tangan selama Pandemi COVID-19 di Pasar Sukaramai Medan," *Wahana Inov.: J. Penelit. dan Pengabdi. Masy. UISU*, Vol. 9, No. 1, pp. 172–176, Jan.-Jun. 2020.

[5] S.F. Rizqah, Haeruddin, and A.R. Amelia, "Hubungan Perilaku Masyarakat dengan Kepatuhan Penggunaan Masker untuk Memutus Rantai Penularan COVID-19 di Kelurahan Bontoa Maros," *J. Muslim Community Health*, Vol. 2, No. 3, pp. 165–175, Jul.-Sep. 2021, doi: 10.52103/jmch.v2i3.553.

[6] E. Lubis, "Peran Protokoler dalam Menunjang Keberhasilan Kinerja Kepala Bagian Umum Pemerintahan Kabupaten Deli Serdang," *Perspektif*, Vol. 7, No. 2, pp. 362–373, Jul. 2014, doi: 10.31289/perspektif.v4i2.165.

[7] J. Chai, H. Zeng, A. Li, and E.W.T. Ngai, "Deep Learning in Computer Vision: A Critical Review of Emerging Techniques and Application Scenarios," *Mach. Learn. Appl.*, Vol. 6, pp. 1–13, Dec. 2021, doi: 10.1016/j.mlwa.2021.100134.

[8] S.C. Hsu, Y.W. Wang, and C.L. Huang, "Human Object Identification for Human-Robot Interaction by Using Fast R-CNN," *2018 Second IEEE Int. Conf. Robot. Comput. (IRC)*, 2018, pp. 201–204, doi: 10.1109/IRC.2018.00043.

[9] D. Kim *et al.*, "Real-Time Multiple Pedestrian Tracking Based on Object Identification," *2019 IEEE 9th Int. Conf. Consum. Electron. (ICCE-Berlin)*, 2019, pp. 363–365, doi: 10.1109/ICCE-Berlin47944.2019.8966205

[10] S.I. Ali, S.S. Ebrahimi, M. Khurram, and S.I. Qadri, "Real-Time Face Mask Detection in Deep Learning Using Convolution Neural Network," *2021 10th IEEE Int. Conf. Commun. Syst., Netw. Technol. (CSNT)*, 2021, pp. 639–642, doi: 10.1109/CSNT51715.2021.9509704.

[11] M.I. Amin, M.A. Hafeez, R. Touseef, and Q. Awais, "Person Identification with Masked Face and Thumb Images under Pandemic of COVID-19," *2021 7th Int. Conf. Control, Instrum., Automat. (ICCIA)*, 2021, pp. 1-4, doi: 10.1109/ICCIA52082.2021.9403577.

[12] M.R. Alwanda, R.P.K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *Algoritm.*, Vol. 1, No. 1, pp. 45–56, Oct. 2020, doi: 10.35957/algoritme.v1i1.434.

[13] J. Pujoseno, "Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Alat Tulis," Undergraduate thesis, Universitas Islam Indonesia, Sleman, Indonesia, Mar. 2018.

[14] M.N.H. Siregar, "Model Arsitektur Artificial Neural Network pada Pelanggan Listrik Negara (PLN)," *InfoTekJar (J. Nas. Inform., Teknol. Jar.)*, Vol. 3, No. 1, pp. 1–5, Sep. 2018, doi: 10.30743/infotekjar.v3i1.642.

[15] M.B. Herlambang (2019) "Deep Learning: Recurrent Neural Networks homepage on website Epam," [Online], https://www.megabagus.id/deep-learning-recurrent-neural-networks/, access date: 6-Jul-2022.

[16] G. Kaur *et al.*, "Face Mask Recognition System Using CNN Model," *Neurosci. Inform.*, Vol. 2, No. 3, pp. 1-9, Sep. 2022, doi: 10.1016/j.neuri.2021.100035.

[17] O. Kembuan, G.C. Rorimpandey, and S.M.T. Tengker, "Convolutional Neural Network (CNN) for Image Classification of Indonesia Sign Language Using Tensorflow," *2020 2nd Int. Conf. Cybern., Intell. Syst. (ICORIS)*, 2020, pp. 1-5, doi: 10.1109/ICORIS50180.2020.9320810.

[18] R. Tineges (2021) "Algoritma Deep Learning : Kenalan dengan Bagian-Bagian Deep Learning, Yuk!" [Online], https://www.dqlab.id/algoritma-deep-learning-machine-learning, access date: 6-Jul-2022.

[19] (2021) "Convolutional Neural Network With Tensorflow and Keras," [Online], https://medium.com/geekculture/introduction-to-convolutional-neural-network-with-tensorflow-and-keras-cb52cdc66eaf, access date: 6-Jul-2022.

[20] R.M. Pradistya (2021) "Mengenal Tensorflow, Library untuk Keperluan Machine Learning Python" [Online], https://www.dqlab.id/mengenal-tensorflow-library-untuk-keperluan-machine-learning-python, access date: 8-Jul-2022.

[21] Friendly, Z. Sembiring, and H.R. Safitri, "Deteksi Wajah Bermasker Berbasis Tensorflow-Keras untuk Pengendalian Gerbang Akses Masuk Menggunakan Rasberry Pi4," *JIKSTRA*, Vol. 2, No. 2, pp. 45–55, Oct. 2020.

[22] (2021) "TensorFlow_Lite_Face_Mask_Jetson-Nano," [Online], https://github.com/Qengineering/TensorFlow_Lite_Face_Mask_Jetson-Nano, access date: 6-Jul-2022.

[23] F.A.M. Ali, and M.S.H. Al-Tamimi, "Face Mask Detection Methods and Techniques: A Review," *Int. J. Nonlinear Anal., Appl.*, Vol. 13, No. 1, pp. 3811-3823, Jan. 2022, doi: 10.22075/ijnaa.2022.6166.

[24] V.K. Pandey, V.K. Gupta, and S. Kumar, "Face Mask Detection Using Convolutional Neural Network," *2021 3rd Int. Conf. Adv. Comput., Commun. Control, Netw. (ICAC3N)*, 2021, pp. 951–954, doi: 10.1109/ICAC3N53548.2021.9725689.

[25] S. Singh *et al.*, "Face Mask Detection Using YOLOv3 and Faster R-CNN Models: COVID-19 Environment," *Multimed. Tools, Appl.*, Vol. 80, No. 13, pp. 19753–19768, Mar. 2021, doi: 10.1007/s11042-021-10711-8.

[26] V. Saminathan *et al.*, "Face Mask Detection Using Raspberry Pi," *Ann. Romanian Soc. Cell Biol.*, Vol. 25, No. 4, pp. 9982–9988, Apr. 2021.

[27] E.N. Arrofiqoh and Harintaka, "Implementasi Metode Convolutional Neural Network untuk Klasifikasi Tanaman pada Citra Resolusi Tinggi," *Geomatika*, Vol. 24, No. 2, pp. 61-68, Nov. 2018, doi: 10.24895/jig.2018.24-2.810.

[28] Trivusi (2022) "Pengertian dan Cara Kerja Algoritma Convolutional Neural Network (CNN)," [Online], https://www.trivusi.web.id/2022/04/algoritma-cnn.html, access date: 11-Aug-2022.

[29] J. Feriawan and D. Swanjaya, "Perbandingan Arsitektur Visual Geometry Group dan MobileNet pada Pengenalan Jenis Kayu," *Sem. Nas. Inov. Teknol.*, 2020, pp. 185–190, doi: /10.29407/inotek.v4i3.84.

[30] P. Nyoman and P.K. Negara, "Deteksi Masker Pencegahan Covid19 Menggunakan Convolutional Neural Network Berbasis Android," *J. RESTI (Rekayasa Sist., Teknol. Inf.)*, Vol. 5, No. 3, pp. 576–583, Jun. 2021, doi: 10.29207/resti.v5i3.3103.

[31] A. Rosebrock (2020) "COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning," Online], https://pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/, access date: 6-Jul-2022.

[32] F.R. Lumbanraja *et al.*, "An Evaluation of Deep Neural Network Performance on Limited Protein Phosphorylation Site Prediction Data," *Procedia Comput. Sci.*, Vol. 157, pp. 25–30, 2019, doi: 10.1016/j.procs.2019.08.137.

[33] (2021) "Jetson Nano Developer Kit," [Online], https://developer.nvidia.com/embedded/jetson-nano-developer-kit, access date: 7-Jul-2022.