Mobile and Forensics
Vol. 7, No. 1, March 2025, pp. 12-23
ISSN: 2714-6685 (online), 2656-6257 (print)
DOI: http://dx.doi.org/10.12928/mf.v1i1.11951

EMBEDDED SYSTEM FOR AUTOMATIC MASK DETECTION USING YOLOv4 DEEP LEARNING AND PyQt5 INTERFACE

[1,*]Arif Fadllullah, [2]Nelson Mandela Rande Langi, [3]Ikhsan Maulana, [4]Laura Nur Meilindy, [5]Muhammad Adhiya Yudhistira Akbar, [6]Mukti Dika Rahman

[1-6]Computer Engineering Department, Faculty of Engineering, Universitas Borneo Tarakan, Tarakan, Indonesia
[1,*]arif.fadl@borneo.ac.id, [2]usernelson90@gmail.com, [3]imsansss44@gmail.com, [4]lmeilindy@gmail.com, [5]adhiyayudhistira@gmail.com, [6]muktidika212@gmail.com
[*]correspondence email

Abstract

The use of masks remains essential, particularly in high-risk areas for disease transmission such as health centers, hospitals, and crowded public spaces. However, many individuals still neglect mask usage in these environments, increasing the risk of disease spread, including COVID-19. Therefore, this study proposes the development of an embedded system based on Raspberry Pi 4 for automatic mask detection using YOLOv4 deep learning and a PyQt5 interface. The system features a simple and compact design with a user-friendly graphical user interface (GUI) to detect mask usage on multiple faces in a single detection. The proposed system was successfully developed and tested on 40 data samples using two testing schemes. The first scheme involved multi-face mask detection with positional and mask color variations at distances of 1 m, 2 m, and 4 m, achieving average precision, recall, F1_score, and sound notification accuracy of 1, 0.872, 0.927, and 0.722, respectively. The second scheme, incorporating positional variations and random unmasked samples, resulted in average values of 0.872, 0.878, 0.874, and 0.878, respectively. These findings demonstrate that the proposed embedded system effectively detects masks on multiple faces with varying colors and angles in a single detection with good accuracy. Further research is recommended to optimize detection sensitivity, accuracy and to evaluate the system's frame per second (FPS) performance in more diverse environments, such as high object density, different camera resolutions, and varied lighting conditions.

Keywords: Multi-face Mask, Embedded System, Raspberry Pi, YOLOv4 Deep Learning, PyQt5

INTRODUCTION

In mid-2020 to 2022, the Indonesian government implemented lockdowns across office, economic, social, and educational activities due to the high transmission rates of COVID-19. As a result, all activities were either suspended or conducted remotely (work from home, or WFH). However, after the government successfully reduced COVID-19 transmission rates through vaccination programs and strict enforcement of COVID-19 protocols, public and private institutions, schools, and universities reopened, allowing in-person work and learning to resume. This transition was accompanied by strict adherence to COVID-19 protocols, including wearing masks, handwashing, physical distancing, avoiding crowds, and limiting mobility [1].

From 2022-2023, COVID-19 protocol regulations have started to loosen. In many areas such as health centers, airports, hospitals, schools, and other crowded places, people can be seen neglecting to wear masks [2]. Some of them, in addition to not wearing masks, also do not limit interactions, do not maintain physical distance, and continue to gather in crowds when entering rooms, thus still posing a risk of transmitting COVID-19. Extra attention should be paid to people when entering public transport, health centers and hospitals to ensure they always comply with COVID-19 protocols, especially in terms of wearing masks [3].

Article History: *Received November 5, 2024; Revised February 27, 2025; Accepted March 5, 2025*

Typically, conventional mask checks are conducted at the entrance through direct verbal warnings. However, this method is subjective as some officers in crowded areas also seem to be negligent in wearing masks, leading people to become accustomed to removing their masks. This condition makes the area unsafe, uncomfortable, and unhealthy for the community as it has the potential to become a breeding ground for rapid COVID-19 transmission. Therefore, the development of real-time mask detection technology is necessary to immediately detect individuals who are not wearing masks when entering crowded areas [4].

Previous research has explored digital techniques for mask detection systems. For instance, a mask detection system based on an Arduino-integrated camera, LCD, LED, and buzzer [5] achieved 100% accuracy for data captured at distances of 50 to 100 cm. However, testing was limited to a single data sample, the system required numerous components which increased costs, and it could only detect one object at a time, making it less effective for multiple-object detection. Another study employed a mask detection method using a combination of the Viola-Jones algorithm and CNN (Convolutional Neural Network) with an accuracy rate of 84.23%. However, the sample images used were static and not real-time [6]. Additional research on mask detection for visually impaired individuals used Raspberry Pi hardware combined with CNN Keras and TensorFlow for real-time video detection, achieving an accuracy of 98%. However, video streaming with Raspberry Pi in this study only achieved an FPS of 0.33, with mask detection limited to distances under 3 meters [7]. Moreover, a real-time mask detection system based on TensorFlow and ssdmobilenet in Python [8] achieved an accuracy of over 97%. However, the dataset used was very limited, and mask detection testing on sample images lacked variety. Another approach involved using the Haar Cascade method (Viola-Jones architecture) [9] [10] and the single-shot multibox detector and mobilenetv2 algorithm [11] for real-time image detection, achieving accuracy rates of over 88% in each study. However, these studies were unable to detect masks on multiple faces at once [11] and were not robust under low light conditions [9].

Based on issues identified in similar research on mask detection systems, such as the inability to detect multiple people simultaneously, detection errors when several individuals enter a room together, misidentifications due to mask variety, and sensitivity to low or changing light. This study proposes the development of an embedded real-time automatic mask detection tool using a deep-learning multi-face detector. This system integrates Raspberry Pi 4 as the embedded system, a camera sensor, YOLOv4 as the deep-learning detector, and PyQt5 as the application interface. YOLOv4 is a single-stage, real-time object detection algorithm, an advancement of YOLO [12]. In single-stage object detection, the neural network predicts multiple bounding boxes and classifies objects (mask/no mask) in a single evaluation, making YOLO and its successors fast for real-time/live video detection. YOLOv4 offers advantages such as fast processing times, the ability to analyze multiple objects simultaneously, and stable performance under low-light conditions and diverse object angles [12][13]. It uses the CSPDarknet53 backbone, which is more memory-efficient compared to the previous backbone (Darknet-53 in YOLOv3), making it more suitable for embedded devices [14]. When hardware resources are limited (e.g., Raspberry Pi or Jetson Nano), it can run on a single GPU with 8GB of VRAM, such as the GTX 1080 Ti [15]. The model complexity also plays a role, as YOLOv5 and later versions use the PyTorch framework, which, while offering greater flexibility, requires more dependencies and memory compared to the Darknet framework used by YOLOv4 [16]. Additionally, it offers a clearer open-source license and is not commercially bound, unlike YOLOv5 and later versions, which is developed by Ultralytics, potentially leading to challenges in commercialization or research publication [17].

By integrating the YOLOv4 algorithm into Raspberry Pi 4, camera footage from CCTV or similar devices can automatically detect whether people are wearing masks. If an individual is detected without a mask, the system will automatically save that frame to Raspberry Pi 4, display a notification via the PyQt5 interface with a message and sound/alarm, and highlight the detected object without a mask in a bounding box on a mini touchscreen LCD. PyQt5, a Python library for creating graphical user interfaces (GUI) using the Qt toolkit, developed by the Qt company, provides an interactive and visually appealing platform for presenting detection processes and outputs [18]. With the PyQt5 interface, the proposed system can deliver user-friendly, cross-platform (Windows, macOS, RaspberryOS, and Linux) desktop applications that make mask detection processes intuitive and visually engaging.

METHODS

Fig. 1. The proposed hardware design

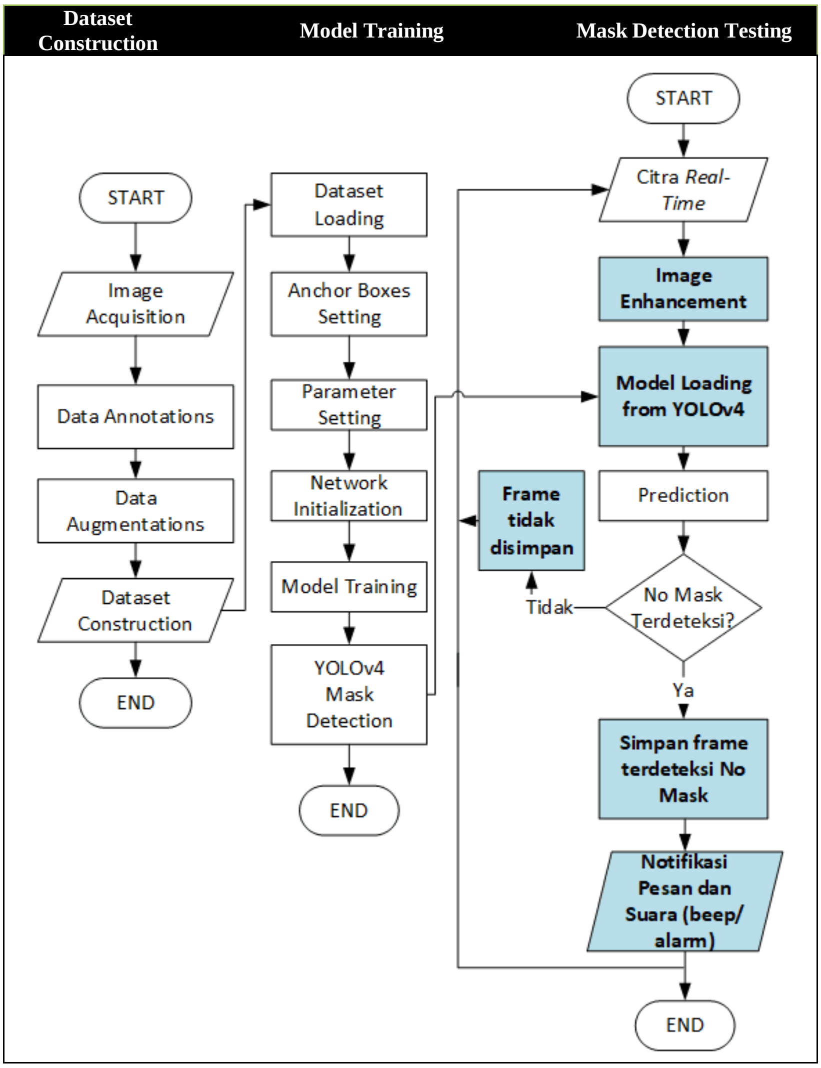

Fig. 2. The proposed software design

Fig. 1 shows the proposed hardware design of the system with three main hardware components: the camera sensor, the panel box (Raspberry Pi 4 and LCD), and the camera sensor stand. The

Raspberry Pi 4 in the panel box is used for processing the proposed system's software, while the LCD displays detection results, and the number of objects (students) detected without masks. Fig. 2 shows the proposed system hardware positioned directly next to the door, with an estimated system height of 1.6 meters. The minimum specifications for the mini PC used are as follows: 1) Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC with speeds up to 1.5GHz; 2) RAM: 4 GB; 3) Operating System: Raspberry Pi OS; 4) Programming Language: Python with OpenCV 4.2 library; 5) Framework: YOLOv4; 6) Audio: 3.5mm audio jack; 7) Storage: microSD card for OS and data storage; 8) GPIO: 40-pin GPIO header to connect various external devices such as sensors, actuators, and other modules; and 9) Camera: Raspberry Pi Camera Module V2 with 8 MP or similar, with a minimum resolution of 240p.

The proposed software design is shown in Fig. 2, consisting of three main sections: dataset construction, model training, and mask detection testing. Since this study focuses more on the development of an embedded system for real-time mask detection, the software development emphasis is on the mask detection testing section, highlighted with a light green box. Meanwhile, the dataset construction and the weight file generated from the mask detection model training using YOLOv4 used in this study were obtained from previous research [19]. In this study [19], the dataset was obtained from Kaggle: https://www.kaggle.com/alexandralorenzo/maskdetection, with a total of 768 images used for training (after augmentation processes such as rotation, resizing, and flipping). This dataset includes various types of images representing real-world scenarios, such as diverse lighting conditions, different types of masks (e.g., surgical masks, cloth masks), and varied facial angles (frontal, side, and partially obscured). This diversity enhances the model's robustness when deployed in real-world environments, ensuring higher accuracy in detecting individuals wearing and not wearing masks in different situations.

The dataset was then trained using YOLOv4 with several parameters, including *batch* = 64, *subdivisions* = 16, *width* = 416, *height* = 416, *channels* = 3, *learning_rate* = 0.001, *burn_in* = 1000, and *max_batches* = 6000. The training results were saved in the yolov4-hql_last.weights file (the file can be accessed at https://bit.ly/yolov4-hql_last). This pre-trained weight file was then utilized in our proposed study, allowing us to focus more on integrating the mask detection algorithm with the Raspberry Pi 4 hardware, thereby maximizing the efficiency and performance of the developed embedded system. This weight file will be used as a tool for analysing mask detection testing from real-time images that have undergone an image enhancement process (contrast and brightness adjustments) to generate prediction results with a class value of either 0 (object with a mask) or 1 (object without a mask). If the prediction results show class = 0, the image frame will not be saved on the Raspberry Pi 4. However, if the prediction results show class = 1, the system will save the image frame on the Raspberry Pi 4 and activate a message-sound notification for objects not wearing a mask through the PyQt5 interface.

After the proposed system was successfully developed, testing was conducted to determine whether the system's results were valid. This research is quantitative and empirical, based on real-world observations comparing the outcomes generated by the proposed system with actual observed results (direct observation). The study lasted for three months, with system testing conducted on a laboratory scale using one of the classrooms in the Computer Engineering Department at UBT for three days under constant indoor lighting conditions. The testing involved 40 data samples, with the testing scheme considering variations in mask colour, mask position, and image capture distances of 1 meter, 2 meters, and 4 meters. The testing schemes are detailed as follows:

- Testing Scheme 1: Data collection involved capturing 4 samples per real-time image generated by the system. All individuals in the test wore masks with colour variations of black, white, and two other randomly selected colours. Image capture was performed in three positions: facing forward (FD = 0°), facing the right side (RS = 45°), and facing the left side (LS = -45°).
- Testing Scheme 2: Data collection was conducted by capturing 4 samples per real-time image, with random sample selection allowing some individuals not to wear masks. As in the first scheme, images were taken in the positions of facing forward (FD), right side (RS), and left side (LS).

The test results are analyzed by comparing the proposed system's predicted performance with actual conditions [20] using *precision*, *recall*, and *F1_score*, along with the accuracy of the sound notification (*SN_accuracy*) ON status to verify True Negative (TN) cases as an interpretation of the system's prediction accuracy. These values are based on the following classifications: a) TP (True Positive), representing the number of masked objects correctly detected as masked by the system; b) TN (True Negative), representing the number of unmasked objects correctly detected as unmasked by the system; c) FP (False Positive), representing the number of noise objects (unmasked) incorrectly detected as masked by the system; and d) FN (False Negative), representing the number of masked objects incorrectly detected as unmasked by the system.

RESULT AND DISCUSSIONS

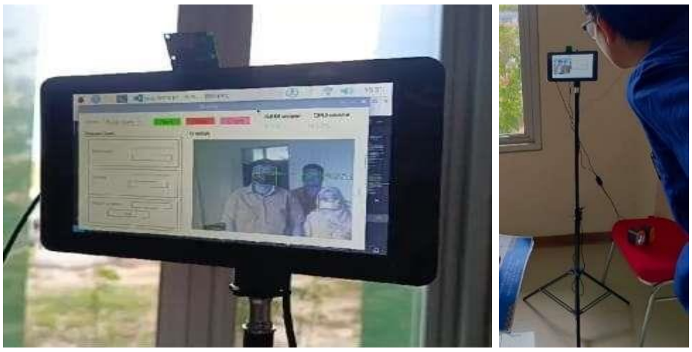

Fig. 3. The hardware of the proposed system

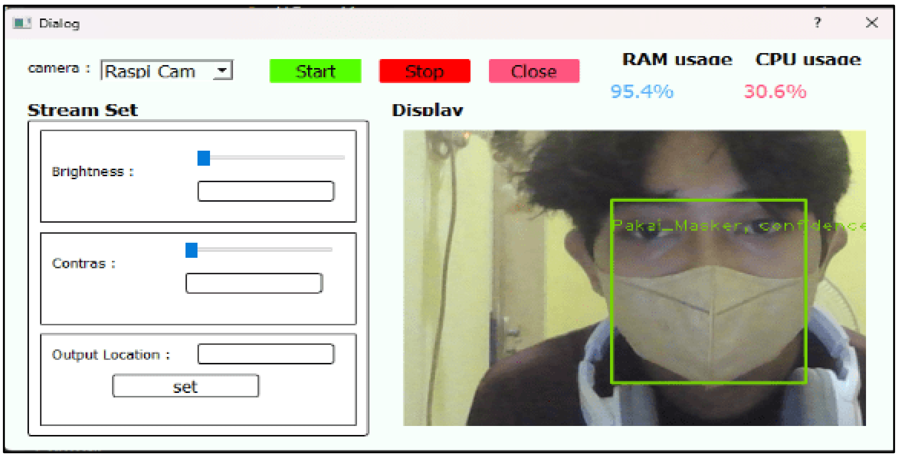

Fig. 4. The software of the proposed system

Fig. 3 illustrates that this research successfully developed a device system architecture using components such as the Raspberry Pi 4 Model B as an embedded system to process YOLOv4 as the deep-learning detector and PyQt5 as the system interface. The Raspberry Pi camera module rev 1.3 functions as the camera sensor, positioned on top of the device, with a tripod to adjust the sensor’s position as needed, an adapter as the device's power supply, and a touchscreen LCD and sound system used to display detection notifications through messages and audio alerts.

The research has successfully developed the software architecture of the system using a simple and minimalist PyQt5 interface, as shown in Fig. 4. This software includes several features that users can utilize, including: a) Users can select one of several cameras connected to the Raspberry Pi 4 mini PC; b) Users can adjust the brightness and contrast as part of the image enhancement process for real-time object images directly through this GUI display; c) In the output location menu, users can specify the folder for storing output images in the form of videos detected by the system; and d) The software provides information on the percentage of RAM and CPU usage of the Raspberry Pi 4 Model B mini PC used to process this proposed system. To obtain test results on whether the proposed system is valid, the system needs to be tested first. The first testing scheme involves capturing 4 samples simultaneously for each image acquisition, where all individuals wear masks with color variations of black, white, and two random colors, positioned facing forward (FD = 0°), right side (RS = 45°), and left side (LS = -45°). Fig. 5 shows the first testing scheme captured 1 meter. The second testing scheme involves capturing 4 samples simultaneously for each image acquisition, with some samples randomly not wearing masks, positioned facing forward (FD), right side (RS), and left side (LS). Fig. 6 shows the second testing scheme captured 1 meter.

Fig. 5. Testing multi-face mask detection with positional and color variations at 1 meter

Fig. 6. Testing multi-face mask data with positional variations and random samples without masks at 1 meter.

The results of the first and second testing schemes at 1 meter for 40 samples can be seen in Table 1 and Table 2.

Table 1. The testing results of sample data with positional and color variations at 1meter

<table>
<tr><th rowspan="3">Sample</th><th colspan="15">Position</th><th colspan="5" rowspan="2">Total</th></tr>
<tr><th colspan="5">Forward (FD)</th><th colspan="5">Right Side (RS)</th><th colspan="5">Left Side (RS)</th></tr>
<tr><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>SN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>SN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>SN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>SN
(0 = OFF)</th></tr>
<tr><td rowspan="2">1-4</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="2">12</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">3</td></tr>
<tr><td colspan="5"></td><td colspan="5"></td><td colspan="5"></td></tr>
<tr><td rowspan="2">5-8</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="2">12</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">3</td></tr>
<tr><td colspan="5"></td><td colspan="5"></td><td colspan="5"></td></tr>
<tr><td rowspan="2">9-12</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="2">12</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">3</td></tr>
<tr><td colspan="5"></td><td colspan="5"></td><td colspan="5"></td></tr>
</table>

Sample	Position															Total				
	Forward (FD)					Right Side (RS)					Left Side (RS)									
	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN (0 = OFF)
13-16	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0	12	0	0	0	3
																				
17-20	4	0	0	0	0	4	0	0	0	0	4	0	0	0	0	12	0	0	0	3
																				
21-24	4	0	0	0	0	4	0	0	0	1	4	0	0	0	0	12	0	0	0	3
25-28	4	0	0	0	0	4	0	0	0	1	4	0	0	0	0	12	0	0	0	3
29-32	4	0	0	0	0	4	0	0	0	1	4	0	0	0	0	12	0	0	0	3
33-36	4	0	0	0	0	4	0	0	0	1	4	0	0	0	0	12	0	0	0	3
37-40	4	0	0	0	0	4	0	0	0	1	4	0	0	0	0	12	0	0	0	3
															Total	120	0	0	0	30
															Precision (*p*)	(120/ (120+0)) = 1				
															Recall (*r*)	(120/ (120+0)) = 1				
															F1_score	2*(1*1)/ (1+1) = 1				
															SN_accuracy (SN = 0)	(30/ 30) = 1				

Table 2. The testing results of samples with positional variations and random samples without masks at 1-meter.

Sample	Position															Total				
	Forward (FD)					Right Side (RS)					Left Side (RS)									
	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN (1 = ON)
1-4	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3
																				
5-8	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3
																				
9-12	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3
																				
13-16	1	3	0	0	1	1	3	0	0	1	1	3	0	0	1	3	9	0	0	3
																				
17-20	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3

Sample	Position															Total				
	Forward (FD)					Right Side (RS)					Left Side (RS)									
	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN	TP	TN	FP	FN	SN (1 = ON)
																				
21-24	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3
25-28	3	1	0	0	1	3	1	0	0	1	3	1	0	0	1	9	3	0	0	3
29-32	3	1	0	0	1	3	1	0	0	1	3	1	0	0	1	9	3	0	0	3
33-36	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3
37-40	2	2	0	0	1	2	2	0	0	1	2	2	0	0	1	6	6	0	0	3
															Total	63	57	0	0	30
															Precision (*p*)	(63/ (63+0)) = 1				
															Recall (*r*)	(63/ (63+0)) = 1				
															F1_score	2*(1*1)/ (1+1) = 1				
															SN_accuracy (SN = 1)	(30/ 30) = 1				

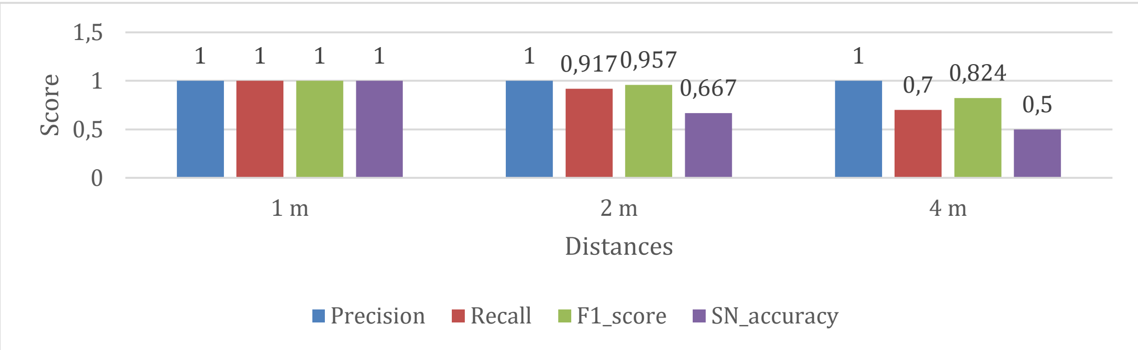

Fig. 7. The results of multi-face mask detection testing with positional and color variations at 1 meter, 2 meters, and 4 meters (Testing Scheme 1)

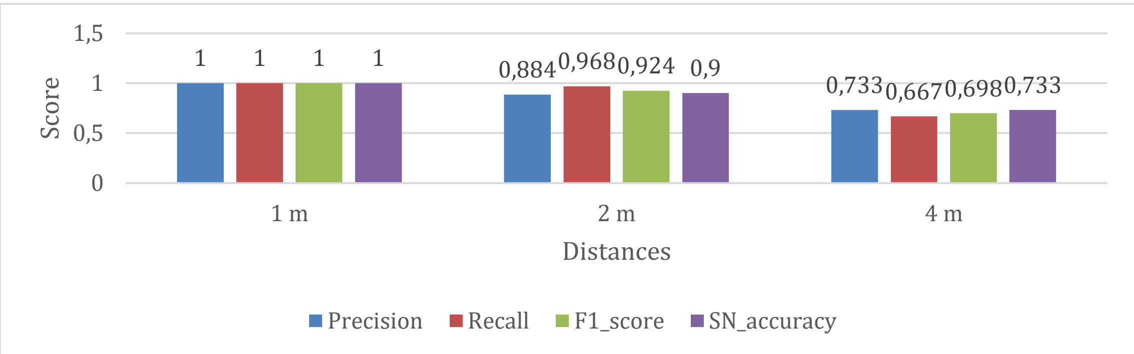

Fig. 8. The results of multi-face mask detection testing with positional variations and random samples without masks at 1 meter, 2 meters, and 4 meters (Testing Scheme 2)

The testing results in Table 1 and Table 2 demonstrate that the proposed system can achieve an average *precision*, *recall*, and *F1_score* of 100% in detecting multi-face mask usage (4 face samples in a single real-time image capture) for all samples with mask position and color variations at 1-meter distance. The system can also identify unmasked objects among masked objects for all sample data at 1-meter distance, maintaining an average *F1_score* of 100%. Additionally, the average accuracy of notification/sound messages (SN) in detecting unmasked noise at 1-meter distance reached 100%. This means that the sound alert is triggered accurately, image frames are stored on the microSD, and bounding boxes appear through the PyQt5 interface whenever unmasked sample data is detected. Then, the average accuracy of sound notification (SN) in detecting unmasked noise at 1-meter distance also achieved 100%. This indicates that the sound alert is triggered precisely, image frames are stored on the microSD, and bounding boxes appear on the PyQt5 interface if there is unmasked sample data.

Next, we conducted the same testing as in Table 1 and Table 2, but with objects captured at 2-meter and 3-meter distances. The testing results can be seen on Fig. 7 and Fig. 8. The test results of multi-face mask detection with positional and colour variations at distances of 2 meters and 4 meters (see Fig. 7) show a decline in system performance. At 2 meters, *recall* dropped to 0.917, and *F1_score* decreased to 0.957, while *precision* remained at 1. The decrease in *recall* indicates an increase in *false negative (FN)*, where the system began missing some masks that should have been detected. The audio notification accuracy (*SN_accuracy*) also significantly declined to 0.667, indicating that only 66.7% of audio notifications matched the actual conditions. This decline may have been caused by reduced image quality or facial resolution at greater distances, particularly affecting dark-coloured masks or those with skin-tone-like colours. At 4 meters, the performance drop became more evident, with *recall* plummeting to 0.7 and *F1_score* falling to 0.824, while *precision* remained perfect at 1. *SN_accuracy* reached its lowest point at 0.5, showing that only half of the audio notifications were accurate. These findings indicate sensitivity and accuracy issues in the system at longer distances. The average metric values from 1, 2, and 4 meters show *precision* of 1, *recall* of 0.872, *F1_score* of 0.927, and *SN_accuracy* of 0.722. Although *precision* remained consistent without any *false positives (FP)*, the drop in *recall* and *SN_accuracy* indicates system limitations in mask detection and accurate audio notifications as the observation distance increases.

The test results with positional variations and random samples without masks at 2 and 4 meters (see Fig. 8) also revealed a decline in system performance. At 2 meters, the system showed *precision* of 0.884, *recall* of 0.968, *F1_score* of 0.924, and *SN_accuracy* of 0.9. The decrease in *precision* indicates the occurrence of *false positives (FP)*, where the system might incorrectly identify masked individuals as not wearing masks or wearing them improperly. At 4 meters, the performance drop became more significant, with *precision* falling to 0.733, *recall* to 0.667, *F1_score* to 0.698, and *SN_accuracy* to 0.733. The drop in *recall* indicates an increase in *false negative (FN)*, where the system failed to detect individuals who should have been identified as not wearing masks or wearing them incorrectly. The decline in *SN_accuracy* shows that only about 73.3% of audio notifications matched the actual conditions. The main factor contributing to this decline might be reduced image quality and facial resolution at greater distances, particularly under specific lighting conditions or when mask colours closely match skin tones. Overall, the average metric values from 1, 2, and 4 meters show *precision* of 0.872, *recall* of 0.878, *F1_score* of 0.874, and *SN_accuracy* of 0.878. While the system generally has good capabilities in mask detection and providing accurate audio notifications, the performance decline at longer distances highlights its limitations in maintaining sensitivity and detection accuracy, as well as accurate audio notifications under various conditions.

These research findings indicate that the proposed system has a high level of validity for implementation as an automatic mask detection tool, particularly demonstrating excellence in maintaining perfect precision (1.0) during tests with positional and color variations of masks or random samples without masks, especially at closer distances (1 and 2 meters). The advantages of this system include the ability to detect multi-face masks in a single detection, reducing the need for direct contact between mask users and mask inspectors. Moreover, the system is robust against diverse backgrounds/noise and various mask colors, while the software and hardware prototypes are simple and compact, allowing for application in any area with sufficient lighting. However, some limitations were identified, particularly the decline in *recall* and *SN_accuracy* as the observation distance increases. At 4 meters, performance significantly drops, both in detecting masked individuals and in the accuracy of audio notifications. Factors such as image quality, facial resolution, mask colours resembling skin tones, and specific lighting conditions affect the system’s performance. Moreover, attention to frame per second (FPS) performance is necessary to ensure the system remains responsive in various environmental conditions and when detecting a high number of objects.

Further research is recommended to test this system in scenarios with high object density, different camera resolutions, and varied lighting conditions to enhance sensitivity and detection accuracy in more diverse environments. Evaluating FPS performance is also crucial to ensure the system operates optimally in real-world scenarios with high workloads, such as crowded public areas. With continued development and testing, this system has the potential to become a reliable solution for supporting health protocols by automatically and accurately detecting mask usage in various situations. Finally, this research offers a cost-effective alternative for automatic mask detection, promoting more objective health protocol enforcement in public spaces. Security personnel at airport entrances, health centres, and hospitals would no longer need to constantly monitor mask compliance, as the proposed system can automatically detect violations and trigger notifications, alarms, or sounds if unmasked individuals attempt to enter. Additional advantages include easy installation, a compact design, and a user-friendly interface.

CONCLUSIONS

The proposed system was successfully developed and tested on 40 data samples using two testing schemes. In the first scheme, which involved multi-face mask detection with positional and mask color variations at 1 m, 2 m, and 4 m distances, the system achieved average precision, recall, F1_score, and sound notification accuracy of 1, 0.872, 0.927, and 0.722, respectively. In the second scheme, with positional variations and random unmasked samples, the results were 0.872, 0.878, 0.874, and 0.878, respectively. These results indicate that the development of an automatic mask detection system using the Raspberry Pi 4 Model B as an embedded system, with YOLOv4 as the deep learning detector and PyQt5 as the system interface, has proven to be valid and accurate for detecting masks in real-time multi-face images, even with variations in mask colour, position, and unmasked objects. The use of the Raspberry Pi 4 offers advantages in a compact and simple design, requiring minimal space and power. Additionally, the integration of a camera sensor as input, YOLOv4 deep learning as the processing algorithm, and PyQt5 as the system's output interface—equipped with a sound system—enables the system to detect multiple objects wearing or not wearing masks in a single detection while remaining robust against background noise and various mask colours. Further research is recommended to test this system in scenarios with high object density, different camera resolutions, and varied lighting conditions to enhance sensitivity and detection accuracy in more diverse environments. Evaluating FPS performance is also crucial to ensure the system operates optimally in real-world scenarios with high workloads, such as in crowded public areas. With continuous development and testing, this system has the potential to become a reliable solution in supporting health protocols by automatically and accurately detecting mask usage in various situations.

ACKNOWLEDGEMENTS

The author would like to express his/her gratitude to the Universitas Borneo Tarakan (UBT) and Kemdikbudristek (Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi) for supporting the funding for this research.

REFERENCES

[1] I. S. Joyosemito and N. M. Nasir, “Gelombang Kedua Pandemi Menuju Endemi Covid-19: Analisis Kebijakan Vaksinasi Dan Pembatasan Kegiatan Masyarakat Di Indonesia,” 2021. doi: 10.31599/jstpm.v2i1.718.

[2] R. Filip, R. Gheorghita Puscaselu, L. Anchidin-Norocel, M. Dimian, and W. K. Savage, “Global challenges to public health care systems during the COVID-19 pandemic: a review

of pandemic measures and problems," *J. Pers. Med.*, vol. 12, no. 8, p. 1295, 2022.
[3] M. R. Abdul Rashid *et al.*, "COVID-19 pandemic fatigue and its sociodemographic, mental health status, and perceived causes: a cross-sectional study nearing the transition to an endemic phase in Malaysia," *Int. J. Environ. Res. Public Health*, vol. 20, no. 5, p. 4476, 2023.
[4] H. Goyal, K. Sidana, C. Singh, A. Jain, and S. Jindal, "A real time face mask detection system using convolutional neural network," *Multimed. Tools Appl.*, vol. 81, no. 11, pp. 14999–15015, 2022.
[5] A. Nofiar and Muhammad Ridwan, "Alat Pendeteksi Ketepatan Penggunaan Masker Berbasis Arduino Menggunakan Bahasa Pemrograman Python," *SATIN - Sains dan Teknol. Inf.*, vol. 8, no. 1, pp. 69–81, 2022, doi: 10.33372/stn.v8i1.835.
[6] F. A. Hermawati and R. A. Zai, "Sistem Deteksi Pemakaian Masker Menggunakan Metode Viola-Jones dan Convolutional Neural Networks (CNN)," in *Konferensi Nasional Ilmu Komputer (KONIK) 2021*, 2021, pp. 182–187. [Online]. Available: https://www.kaggle.com/prithwirajmitra/covid-face-mask-
[7] M. Aminullah, "Alat Deteksi Masker Dengan Metode Convolutional Neural Network Untuk Tunanetra Pada Era New Normal," in *Seminar Nasional Fortei Regional 7*, 2020, pp. 321–324.
[8] R. F. Muharram and A. Suryadi, "Implementasi artificial intelligence untuk deteksi masker secara realtime dengan tensorflow dan ssdmobilenet Berbasis python," *J. Widya*, vol. 3, no. 2, pp. 281–290, 2022, [Online]. Available: https://jurnal.amikwidyaloka.ac.id/index.php/awl
[9] F. L. Ahmad, A. Nugroho, and A. F. Suni, "Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahaan COVID 19," *Edu Elektr. J.*, vol. 10, no. 1, pp. 13–18, 2021.
[10] G. Aprilian Anarki, K. Auliasari, and M. Orisa, "PENERAPAN METODE HAAR CASCADE PADA APLIKASI DETEKSI MASKER," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 5, no. 1, pp. 179–186, 2021, doi: 10.36040/jati.v5i1.3214.
[11] R. R. Ramdhani, R. I. Adam, and A. A. Ridha, "Implementasi Deep Learning Untuk Deteksi Masker Deep Learning Implementation for Face Mask Detection," *J. Inf. Technol. Comput. Sci.*, vol. 4, no. 2, p. 2021, 2021.
[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv Prepr. arXiv2004.10934*, pp. 1–17, 2020, doi: 10.48550/arXiv.2004.10934.
[13] L. Agustien, T. Rahman, and A. W. Hujairi, "Real-time Deteksi Masker Berbasis Deep Learning menggunakan Algoritma CNN YOLOv3," *J. Teknol. Inf. dan Terap.*, vol. 8, no. 2, pp. 129–137, 2021, doi: https://doi.org/10/25047/jtit.v8i2.246.
[14] Z. Jiang, L. Zhao, S. Li, and Y. Jia, "Real-time object detection method for embedded devices," in *computer vision and pattern recognition*, 2020, pp. 1–11.
[15] N. Klingler, "YOLOv4: A Fast and Efficient Object Detection Model - viso.ai." Accessed: Feb. 25, 2025. [Online]. Available: https://viso.ai/deep-learning/yolov4/
[16] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, 2022.
[17] "YOLOv5 Model License and Pricing." Accessed: Feb. 25, 2025. [Online]. Available: https://roboflow.com/model-licenses/yolov5
[18] B. Gabrial, S. Tangkuman, and J. Rantung, "Rancang Bangun Aplikasi Perancangan Roda Gigi Lurus Menggunakan Bahasa Pemrograman Python," *J. Tekno Mesin*, vol. 10, no. 1, pp. 1–8, 2024, doi: https://doi.org/10.35793/jtm.v10i1.51990.
[19] H. M. Alfarisi, "GitHub - HaiqalLM10/FaceMaskYOLOv4: Face Mask Detection with YOLOv4 dan Darknet (Indonesian)." Accessed: Jun. 14, 2024. [Online]. Available: https://github.com/HaiqalLM10/FaceMaskYOLOv4
[20] S. Singha and B. Aydin, "Automated Drone Detection Using YOLOv4," *Drones*, vol. 5,

no. 3, p. 95, Sep. 2021, doi: 10.3390/drones5030095.

AUTHORS BIBLIOGRAPHY

ARIF FADLLULLAH was born in Tarakan, North Borneo. Indonesia in 1991. He is currently Assistant Professor and Head of the Computer Engineering Department, Faculty of Enginering, Universitas Borneo Tarakan (UBT), Indonesia. He received his S.Pd. (Bachelor of Informatics Education) degree at the Department of Electrical Engineering, Faculty of Engineering, Universitas Negeri Malang, Indonesia, in 2012. His M.Kom. (Master of Informatics) degree in Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, in 2016. Since 2016, he has authored or coauthored in 26 publications: 1 proceedings and 25 journals, with 5 H-index, 1 i10-index, 57 citations, and 8 Intellectual Property Rights (IPR). His research interests include image processing, computer vision, machine learning, and software engineering. He is a member of APTIKOM and PII, Indonesia. He can be contacted at email: arif.fadl@borneo.ac.id / arif.fadl@gmail.com.

NELSON MANDELA RANDE LANGI was born in Tarakan, North Borneo, Indonesia in 2003. He is a current undergraduate student (in his 7th semester) of the Computer Engineering Department, the Faculty of Engineering, Universitas Borneo Tarakan. He is one of the outstanding students who won the national funding for the Student Creativity Program of the Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi in 2023 for his research titled "Automatic Mask Detection System at Class Door Based on Camera with Multi-Face Deep-Learning Detector and Voice Message Notification". His research interests include embedded system and computer vision. He can be contacted at email: usernelson90@gmail.com.

IKHSAN MAULANA was born in Tarakan, North Borneo, Indonesia in 2003. He is a current undergraduate student (in his 7th semester) of the Computer Engineering Department, the Faculty of Engineering, Universitas Borneo Tarakan. He is one of the outstanding students who won the national funding for the Student Creativity Program of the Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi in 2023 for his research titled "Automatic Mask Detection System at Class Door Based on Camera with Multi-Face Deep-Learning Detector and Voice Message Notification". His research interests include computer vision and software engineering. He can be contacted at email: imsansss44@gmail.com.

LAURA NUR MEILINDY was born in Lamongan, East Java, Indonesia in 2004. She is a current undergraduate student (in his 5th semester) of the Computer Engineering Department, the Faculty of Engineering, Universitas Borneo Tarakan. She is one of the outstanding students who won the national funding for the Student Creativity Program of the Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi in 2023 for her research titled "Automatic Mask Detection System at Class Door Based on Camera with Multi-Face Deep-Learning Detector and Voice Message Notification". Her research interests include computer vision and software engineering. She can be contacted at email: lmeilindy@gmail.com.

MUHAMMAD ADHIYA YUDHISTIRA AKBAR was born in Banjarmasin, South Borneo, Indonesia in 2003. He is a current undergraduate student (in his 7th semester) of the Computer Engineering Department, the Faculty of Engineering, Universitas Borneo Tarakan. He is one of the outstanding students who won the national funding for the Student Creativity Program of the Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi in 2023 for his/her research titled "Automatic Mask Detection System at Class Door Based on Camera with Multi-Face Deep-Learning Detector and Voice Message Notification". His research interests include human-computer interaction and computer vision. He can be contacted at email: adhiyayudhistira@gmail.com

MUKTI DIKA RAHMAN was born in Tarakan, North Borneo, Indonesia in 2002. He is a current undergraduate student (in his 7th semester) of the Computer Engineering Department, the Faculty of Engineering, Universitas Borneo Tarakan. He is one of the outstanding students who won the national funding for the Student Creativity Program of the Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi in 2023 for his research titled "Automatic Mask Detection System at Class Door Based on Camera with Multi-Face Deep-Learning Detector and Voice Message Notification". His research interests include image processing and computer vision. He can be contacted at email: muktidika212@gmail.com.