

## Real-Time Occluded Face Identification Using Deep Learning

Muhammad Fachrurrozi<sup>1</sup>, Anggina Primanita<sup>2\*</sup>, Rafly Pakomgan<sup>3</sup>, Abdiansah<sup>4</sup>

<sup>1,2,3,4</sup>Informatics, Faculty of Computer Science, Sriwijaya University  
<sup>1,2,3,4</sup> Jl. Srijaya Negara, Bukit Besar, Kec. Ilir Bar. I, Kota Palembang, Sumatera Selatan 30128, Indonesia.

### ABSTRACT

#### Article:

Accepted: March 28, 2023

Revised: April 14, 2023

Issued: April 30, 2023

© 2023 The Author(s).



This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

#### \*Correspondence Address:

anggina.primanita@ikom.unsri.a  
c.id

One of the most difficult aspects of face identification is face occlusion. Face occlusion is when anything is placed over the face, for example, a mask. Masks occlude multiple important facial features, like the chin, lips, nose, and facial edges. Face identification becomes challenging when important facial features are occluded. Using one of the deep learning algorithms, YOLOv5, this work tries to identify the face of someone whose face is occluded by a mask in real-time. A special program is being created to test the effectiveness of the YOLOv5 algorithm. 14 people's data were registered, and each person had 150 images used for training, validation, and testing. The images used are regular faces and mask-occluded faces. Nine distinct configurations of epoch and batch sizes were used to train the model. Then, during the testing phase, the best-performing configuration was chosen. Images and real-time input were used for testing. The highest possible accuracy of image identification is 100%, whereas the maximum accuracy of real-time identification is 64%. It was found during the testing that the brightness of the room has an influence on the performance of YOLOv5. Identifying individuals becomes more challenging when there are significant changes in brightness.

**Keywords:** *YOLOv5, deep learning, occluded face identification, real time identification*

## 1. INTRODUCTION

The human face is a distinct characteristic of the human body. Each individual on the planet possesses a distinctive combination of facial features [1]. Numerous details about a face can be shown, such as skin tone, facial bone structure, gender, and facial expressions. Faces may be used to portray a range of scenarios and can be used as research material for digital images [2].

Face identification, a discipline within computer science, involves the capacity to ascertain an individual's identity by examining the distinctive facial features and patterns present in human faces [3]. The initial step of this process involves detecting the face and its landmarks in an image. Following pre-processing, the main facial area is inputted into the back-end network, which performs facial feature extraction and face matching. Face identification has significantly advanced and improved in light of current discoveries in deep learning and wide object detection [4].

Face occlusion refers to the act of covering the face with various objects or items, including glasses, masks, scarves, hands, hair, or hats. This occlusion significantly reduces the effectiveness of face detection [5]. Partial occlusion poses a significant challenge for face identification, as it becomes difficult to recognize a face when certain parts are obscured. For example, the eyes can be hidden by glasses, the ears can be hidden by a hijab, the forehead and ears can be hidden by hair, half of the face can be hidden by a mask, and the other half can be hidden by facial hair like a mustache or a beard. These factors can lead to a decline in the system's performance [6]. The focus of this study is on the identification of occluded faces specifically through the presence of masks.

Deep learning is a popular method employed for face recognition tasks. It involves the utilization of artificial neural networks with multiple layers, allowing for the analysis of data at intricate levels of abstraction. The primary advantage of deep learning models lies in their capability to automatically extract important features from input data using general-purpose learning techniques. These models find applications in various domains, including image reconstruction, speech recognition, video analysis, and audio processing [7].

In conventional face identification, deep learning relies on significant facial features,

including the chin, lips, nose, eyes, forehead, and facial edges. However, when a face is covered by a mask, deep learning struggles to identify the face due to the obstruction of crucial facial features [8]. The challenge with occluded face identification lies in the system's ability to identify faces solely based on the eyes and forehead. One deep learning method used in object recognition, specifically in this context, is You Only Look Once (YOLO).

YOLO, a widely used system for detecting objects, demonstrates exceptional performance in real-time object detection. YOLOv5, in particular, stands out due to its low RAM requirements, making it more convenient for deployment and integration with Internet of Things (IoT) devices [9].

Numerous studies have been conducted to identify faces that are occluded by masks. These studies have utilized different methods such as ResNet-50 architecture, which produced an accuracy rate of 47.91% [10]; Convolutional Neural Network (CNN), which achieved an accuracy rate of 87% [11]; a combination of Principal Component Analysis (PCA) and deep learning, achieving an accuracy rate ranging from 85% to 95% [12]. The accuracy of these methods varies depending on several factors, including the resolution of the image, lighting conditions, and the position of the head.

Research by Al Tamimi and Ali [13] utilized YOLOv5s to detect whether individuals were wearing face masks. Real-time images were used in the experiments, and the results demonstrated that video and real-time video achieved a high level of accuracy. The study suggested that increasing the epoch value during the training process is crucial for improving accuracy. However, it emphasized the need for careful monitoring to prevent system overload when adjusting the epoch value.

A different research study examined the use of face recognition for a library attendance system [14]. The study specifically focused on using the YOLOv5 approach to identify masked faces as part of the attendance system. To detect several objects, the system needed 0.14267 seconds for image input and an average of 0.013 seconds for video input. The recognition system's average frame rate was 76.92 frames per second (fps). The tests were run using 1000 epochs.

Research conducted in the past has demonstrated that YOLOv5, a deep learning

system, is highly proficient and efficient in identifying individuals' faces and discerning whether they are wearing a mask or not. The main contribution of this study is the use of YOLOv5 in real-time to identify people who are currently wearing masks.

This manuscript represents the original findings of our research.

## 2. METHODS

The process is divided into five stages: data collection, acquisition of occluded face data, training, testing with images as input, and real-time testing.

### 2.1 YOLOv5

The You Only Look Once (YOLO) algorithm was designed to detect objects in real-time. For detection, the detection system used a localizer or repurposed classifier. The YOLO method detects objects using Convolutional Neural Networks. YOLO employs a Deep Learning approach to detect objects in images. YOLO excels at image classification and prediction [14]. You Only Look Once (YOLO) algorithms adopt a unique approach where the entire image serves as input to the network. They predict both the position and category of the bounding boxes that encompass objects, leveraging the features extracted from the entire image. This holistic approach allows YOLO to efficiently and accurately detect objects by considering the contextual information present in the entire image [15]. Since its inception in 2016, the YOLO algorithm has undergone continuous refinement and development. It has evolved into five primary versions [16]: YOLOv1, YOLOv2, YOLOv3, YOLOv4, and YOLOv5 [14].

YOLOv5, the fifth version of the YOLO algorithm, is implemented using Python [17] and built upon the PyTorch framework [18]. YOLOv5 is recognized for its exceptional detection speed and accuracy, largely attributed to its utilization of an open-source network architecture [19]. In 2020, Glenn Jocher introduced YOLOv5, a one-stage target recognition algorithm [20]. YOLOv5 offers four different models for data training: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These models vary in their network designs, number of layers, and number of parameters used. Each model is tailored to suit

different requirements and offers varying levels of performance and computational complexity. The network architecture of the four YOLOv5 models is depicted in Figure 1. The YOLOv5 architecture is comprised of three elements: the backbone model, the neck model, and the head model [14].

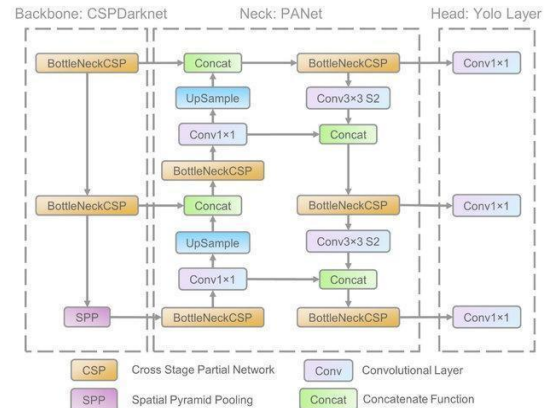


Figure 1. Network architectures of YOLOv5s model [21]

Deep learning is a subfield of machine learning that takes inspiration from the intricate structure of the human brain. Similar to how humans employ structured and logical thinking to make decisions, deep learning aims to replicate this process using an algorithm known as a neural network [22]. Deep learning leverages nonlinear information processing techniques to excel in tasks such as feature extraction, pattern recognition, and classification [23].

One of the main advantages of Deep Learning models is their ability to extract valuable features from input data using general-purpose learning techniques. As a result, these models find application in various domains, such as image, speech, video, and audio reconstruction [7].

### 2.2 Data Collection

In this research, two image data compilations were utilized. The first compilation consists of 80 regular face images and 70 mask-occluded face images captured from 10 people using a Samsung S21 Ultra Smartphone Camera. The second compilation includes 50 regular face images and 50 mask-occluded face images of 10 people from the Labeled Faces in the Wild (LFW) Dataset. Both compilations were stored in the jpg format.

### 2.3 Data Acquisition

Once the image compilation is completed, the images undergo a labeling process using the Roboflow website [24]. This process involves annotating the images to identify and label the object classes present in each image frame. The result of this labeling process is a text file in a specific format that contains information about the list of object classes found in each image frame. This labeled file serves as a reference for subsequent analysis and training of the deep learning model. Figure 2 depicts the acquisition method for masked face image data.

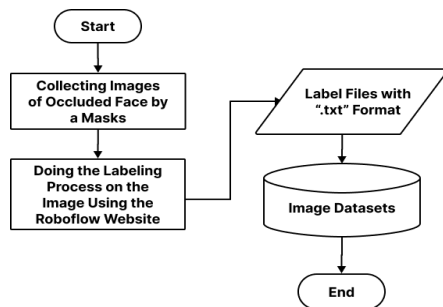


Figure 2. Workflow of occluded face data acquisition

The following phase in the process is data pre-processing, which involves resizing the images to 416x416 pixels. This pre-processing step guarantees that the input size is consistent for future analysis and model training. The images are converted and saved as ".jpg" files with a 416x416 pixel size. Figure 3 depicts the workflow of data pre-processing. Furthermore, augmentation techniques are used to improve the dataset. Each image is subjected to a random set of modifications, such as brightness, saturation, and grayscale conversion.

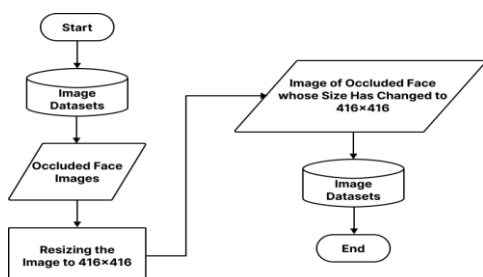


Figure 3. Workflow of data pre-processing

### 2.4 Training

During the training phase, the occluded face images and corresponding labels in the dataset are used to train the YOLOv5s model.

The output of this training process is the weight model, which is crucial for the success of the program in identifying occluded faces. The comprehensive flow of the training process can be observed in Figure 4, while Figure 5 depicts the results obtained from the training process captured through the specialized software developed for this research.

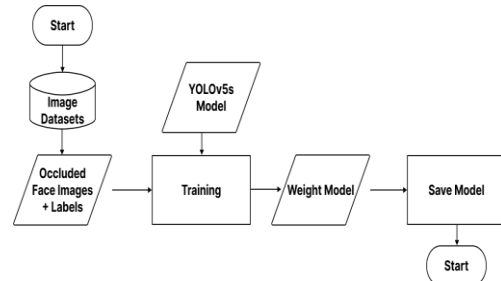


Figure 4. Workflow of the training process



Figure 5. Collage of training process results.

### 2.5 Testing

There are two testing phases in this experiment. The first is testing using image input, while the second is testing using real-time video input.

The aim of the image input testing phase is to assess the YOLOv5 model's accuracy in identifying occluded faces. Figure 6 illustrates the sequence of steps involved in the training process using image input.

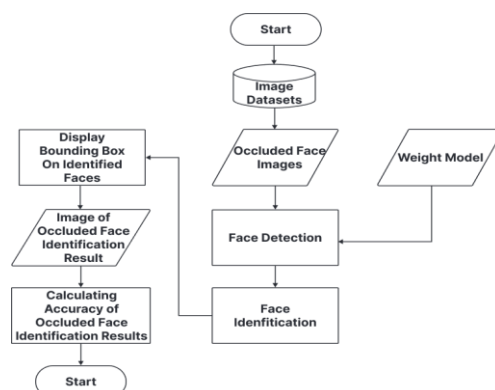


Figure 6. Model testing process utilizing images as input

During testing with image input, the first step involves providing images of faces that have been occluded by a mask. The program then detects the faces in these images. Upon successful detection, the program proceeds to identify the face using the weight model obtained from the training process. A bounding box is added to the identified face, and the resulting image is displayed by the program. The final step involves calculating the accuracy of the occluded face identification results.

Likewise, real-time video input from a camera was used throughout the testing phase to evaluate the response time of YOLOv5 in conducting real-time identification of occluded faces. The workflow of real-time training can be seen in Figure 7.

In real-time testing, the initial step is to grant the program access to the camera. The user then positions their face in front of the camera. Subsequently, the program detects the face and proceeds to identify it using the weight model obtained from the training process. If an occluded face is identified, the program captures a screenshot. Once the program is stopped, the screenshot results are displayed for a duration of 2 seconds. The final step involves calculating the response time based on the results of real-time occluded face identification.

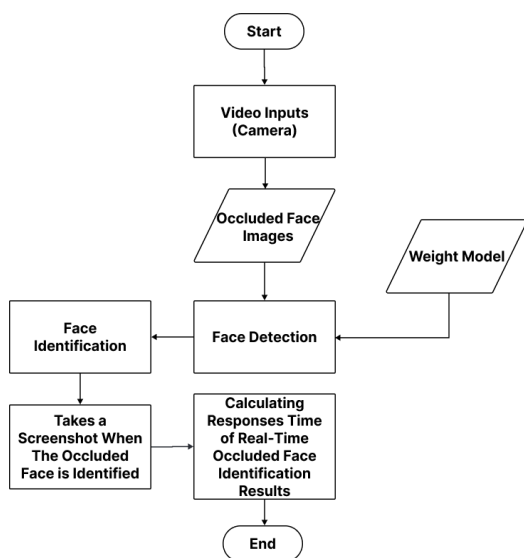


Figure 7. Workflow of real-time testing

## 2.6 Evaluation

To evaluate the recall, F1-score, precision, and accuracy of the model employed in this study, a confusion matrix is utilized. The details of the confusion matrix is shown in Table 1.

Table 1. Confusion Matrix [2]

		Predicted Value	
		True	False
Actual Value	True	TP (True Positive)	FP (False Positive)
	False	FN (False Negative)	TN (False Negative)

The value of precision, recall, accuracy and F1-score can be calculated based on Table 1, as follows:

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

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

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$F1 - Score = \frac{2 \times recall \times precision}{recall+precision} \quad (4)$$

## 3. RESULTS AND DISCUSSION

This section presents and discusses the results obtained from data acquisition, training, and testing phases.

### 3.1 Occluded face data acquisition results

The collected data underwent preprocessing phase, which involved resizing and augmentation by applying random changes to the saturation, brightness, and grayscale. As a result of this process, a primary dataset called Compilation I was generated, consisting of 450 masked face images and 450 corresponding labels. Furthermore, this process also led to the creation of a secondary dataset called Compilation II, which includes 300 masked face images and 300 corresponding labels. The distribution of images and labels for training, validation, and testing purposes can be observed in Table 2.

### 3.2 Training phase results

The training process on both the primary and secondary datasets involved nine combinations, hereafter is called as configurations, of epochs and batch sizes. The specific configurations of epochs and batch sizes are provided in Table 3.

Table 2. The quantities of data acquired for each compilations

Dataset Name	Training	Validation	Testing
Compilation I	300	90	60
Compilation II	210	60	30

Table 3. Training configurations

Configuration	Epoch	Batch Size
1	50	16
2	250	16
3	500	16
4	50	24
5	250	24
6	500	24
7	50	32
8	250	32
9	500	32

The selection of the best model is based on monitoring the accuracy value at each epoch, and the model with the highest accuracy is chosen. Stochastic Gradient Descent (SGD) is utilized as the optimizer. The training process incorporates a technique where the training is halted at a certain epoch if the accuracy value does not improve for 100 consecutive epochs following the highest accuracy value. The output of this training process is the best weights model obtained after running multiple epochs. The training was performed on Google Collab media, utilizing the Nvidia Tesla T4 GPU. The training results from the best configuration, configuration 9, are depicted in Figure 8(a) for Compilation I and Figure 8(b) for Compilation II.

In the training process, Compilation I employed 300 training data and 90 validation data, while Compilation II used 210 training data and 60 validation data. The training data is used to train the model to accurately identify faces with occlusions, while the validation data is employed to assess and validate the model generated during the training process.

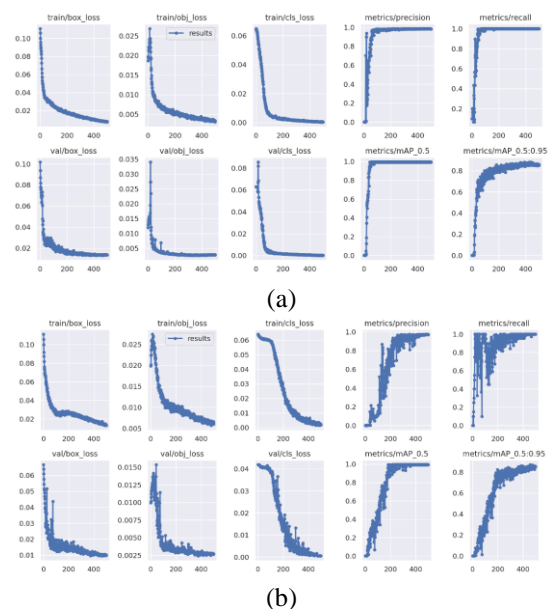


Figure 8. The outcome of the training process using configuration 9 in (a) Compilation I and (b) Compilation II

The performance of a weight model can be considered good if the values of recall, mAP0.5, precision, and mAP0.5:0.95 are close to 1, while the values of box loss, object loss, and class loss are close to 0. In the case of configuration 9, the resulting weight models demonstrate good performance, as indicated by the values of recall, mAP0.5, precision, and mAP0.5:0.95 being close to 1, and the values of box loss, object loss, and class loss being close to 0.

Figure 8(a) illustrates that the precision of Compilation I increases rapidly and steadily, whereas Figure 8(b) depicts that the precision of Compilation II takes a longer time to increase and exhibits instability. This discrepancy is attributed to the disparity in the quantity of training and validation data between the two compilations. Consequently, the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values differ, resulting in a faster increase in precision for Compilation I compared to Compilation II.

The trend of the recall value also differs between the two compilations. Compilation I shows a faster and more stable increase in recall compared to Compilation II. Again, this disparity is influenced by the varying amount of training and validation data, as well as the discrepancies in TP, TN, FP, and FN values discussed earlier.




Furthermore, the difference in image quality between Compilation I and Compilation II contributes to variations in the increase of recall, mAP0.5, precision, and mAP0.5:0.95 values. This difference in image quality can also lead to a weight model with poorer performance.

### 3.3 Determination of Confidence Score

Prior to the testing process, an initial test was conducted to identify the optimal Confidence Score for use in subsequent testing. The purpose of this test was to determine the Confidence Score value that minimizes the occurrence of unexpected bounding boxes displayed by the program. Three proposed Confidence Score values were evaluated: 0.5, 0.6, and 0.75. The results of this Confidence Score determination test are presented in Table 4.

Upon reviewing the test results in Table 4, it is evident that a Confidence Score of 0.75 yields the best outcome compared to the other values. At a Confidence Score of 0.75, only one bounding box is displayed around the face. Conversely, at Confidence Scores of 0.5 and 0.6, the program generates more than one bounding box on the face. Therefore, the selected Confidence Score value for both image-based testing and real-time testing is 0.75.

*Table 4. Values of confidence score obtained from performing real-time testing*

Confidence Score	Real-time Outcome
0.5	
0.6	
0.75	

### 3.4 Image input testing results

During the testing phase, a total of 60 images comprising 10 labels from Compilation I and 30 images with 10 labels from Compilation II were used. The image-based testing was conducted on Google Collab media, utilizing the Nvidia Tesla T4 GPU. In this phase, if the Confidence Score is below 75, the face is considered unidentified. The results of the test, which included occluded face images from both the primary and secondary datasets, are presented in the table for occluded face identification performance evaluation. Four metrics, namely Recall, F1-Score, Precision, and Accuracy were determined to assess the performance of the system.

The sample result of image-based testing on both datasets are presented in Table 5 and Table 6. The custom software developed for this purpose displays the confidence score and identified label above the bounding box surrounding the individual's face. This visual representation provides an indication of the level of clarity with which each individual is identified.

The performance of the occluded face identification system on both datasets is presented in Table 7 and Table 8. These tables provide essential metrics such as precision, recall, mAP\_0.5, accuracy, and F1-score for each compilation after undergoing nine cycles of training and testing. These values collectively demonstrate the effectiveness and performance of the YOLOv5 model utilized.

*Table 5. Some results from testing phase using still image as input on Compilation I*





Image	True Label	Outcome Label
	CIA	CIA
	ICA	ICA
	NIGEL	NIGEL
	RAFLY	RAFLY

Table 6. Some results from testing phase using still image as input on Compilation I.

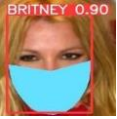



Image	True Label	Outcome Label
	BRITNEY	BRITNEY
	RAY	RAY
	SHERYL	SHERYL
	TOMMY	TOMMY

Table 7. Performance of occluded face identification on Compilation I

C	P	R	m	A	F
1	1.00	0.951	0.99	0.951	0.97
2	1.00	1.00	0.99	1.00	1.00
3	1.00	1.00	0.99	1.00	1.00
4	1.00	0.83	0.99	0.83	0.91
5	1.00	1.00	0.99	1.00	1.00
6	1.00	1.00	0.99	1.00	1.00
7	1.00	0.801	0.99	0.801	0.89
8	1.00	1.00	0.99	1.00	1.00
9	1.00	1.00	0.99	1.00	1.00

C = Configuration, P = Precision, R = Recall, m = mAP<sub>0,5</sub>, A = Accuracy, F = F1-Score

Table 8. Performance of occluded face identification on Compilation II

C	P	R	m	A	F
1	0.00	0.00	0.461	0.00	0.00
2	1.00	1.00	0.99	1.00	1.00
3	1.00	1.00	0.99	1.00	1.00
4	0.00	0.00	0.39	0.00	0.00
5	1.00	0.97	0.99	0.97	0.98
6	1.00	1.00	0.99	1.00	1.00
7	0.00	0.00	0.64	0.00	0.00
8	1.00	1.00	0.99	1.00	1.00
9	1.00	0.901	0.99	0.901	0.95

C = Configuration, P = Precision, R = Recall, m = mAP<sub>0,5</sub>, A = Accuracy, F = F1-Score

According to the outcomes of Compilation I, the configurations with epoch and batch numbers 2, 3, 5, 6, and 9 outperformed the other sets. Similarly, in Compilation II, configurations 2, 3, 6, and 8 outperformed the remaining configurations.

### 3.5 Real-time input testing results

For real-time testing, we utilized the ASUS STRIX SCAR 15 laptop equipped with the external webcam ROG EYE S camera, which has a resolution of 1920x1080 pixels. The testing process was conducted using PyCharm Community Edition 2022.2.1, and the Nvidia RTX 3060 GPU was employed. Similar to the previous experiment, the condition for labeling a face as unrecognized was set to a Confidence Score of less than 75. Table 9 provides detailed information about the accuracy and response time achieved in the experiment, considering nine different combinations of epoch batch sizes.

Table 9. Accuracy and response time for occluded face identification in real time

Configuration	Accuracy	Average Response Time
1	20 %	0.01 seconds
2	40 %	0.02 seconds
3	40 %	0.02 seconds
4	20 %	0.02 seconds
5	50 %	0.02 seconds
6	40 %	0.02 seconds
7	30 %	0.02 seconds
8	<b>50 %</b>	<b>0.02 seconds</b>
9	<b>50 %</b>	<b>0.02 seconds</b>

During the real-time testing, configurations 8 and 9 exhibited the highest accuracy of 50%. Upon closer examination, it was discovered that the images utilized for training the real-time system were captured in an open area. This led to our hypothesis that the system's accuracy was adversely affected by the variability in brightness commonly encountered in such environments.

To prove our hypothesis, four new individual labels were introduced to the system and tested in a room with similar lighting conditions. All four individuals were successfully identified, resulting in a 100% accuracy rate. As a result, the overall program accuracy increased by 14%. However, this improvement came at the expense of longer



average response times. In Table 10, accuracy and response time values are compared before and after the four additional participants were incorporated into the primary dataset.





Table 11 showcases a sample outcome from the real-time testing. Similar to the image input results, the developed software displays the confidence score and the identified label name above the bounding box surrounding the recognized individual. It is important to note that these results are presented in real time, and slight variations may occur when the individual is in motion. In our experiment, we specifically instructed the individuals to maintain a stationary position and face the camera directly, taking into consideration the importance of capturing a frontal view of the face.

*Table 10. Accuracy and response time for occluded face identification in real-time comparison before and after the insertion of additional name labels*

Configuration	Accuracy	ART
9 prior to introducing four new fresh labels	50 %	0.02 seconds
9 following the addition of four fresh name labels	<b>64 %</b>	<b>0.02 seconds</b>

ART = Average Response Time

*Table 11. Real-Time test outcome illustration*

Image	True Label	Outcome Label	Response Time
	CIA	CIA	0.02 seconds
	ICA	ICA	0.02 seconds
	NIGEL	NIGEL	0.01 seconds
	RAFLY	RAFLY	0.02 seconds

## CONCLUSION

The aim of this study was to assess the performance of YOLOv5 in identifying occluded face using images captured from a phone camera and a specific dataset. The results revealed that YOLOv5 achieved exceptional accuracy of 100% in identifying still images. Its real-time implementation shows lower performance with maximum accuracy of 64%. A hypothesis was proposed suggesting that the brightness of the room during data compilation influenced the testing outcomes. In cases where the training images were captured in rooms with drastically different lighting conditions, either excessively bright or dim, the system encountered difficulties in accurately identifying 50% of the individuals. To validate this hypothesis, a new dataset containing images taken in moderately lit rooms was introduced. In this scenario, the system successfully identified individuals without errors, thus confirming the hypothesis. Consequently, the accuracy of the real-time YOLOv5 implementation improved by 14%, but the average response time lengthened compared to the previous setup.

Based on these findings, future research should focus on addressing challenges posed by drastic changes in image brightness. Potential expansions include optimizing image brightness, adding pre-processing and augmentation techniques, and decrease the average response time of the system. These improvements will be crucial in upgrading the system to a larger-scale real-time implementation.

## REFERENCE

- [1] F. Syuhada, I. G. P. Suta Wijaya, and F. Bimantoro, "Pengenal Wajah Untuk Sistem Kehadiran Menggunakan Metode Eigenface dan Euclidean Distance," *J. Comput. Sci. Informatics Eng.*, vol. 2, no. 1, pp. 64–69, 2018, doi: 10.29303/jcosine.v2i1.74.
- [2] L. Novamizanti, H. Gymnovriza, and E. Susatio, "Pengenal Wajah Individu Berbasis 3D Biometrik," *JIKO (Jurnal Inform. dan Komputer)*, vol. 6, no. 1, p. 41, 2022, doi: 10.26798/jiko.v6i1.182.
- [3] R. Bankar, N. Bargat, I. Hanmante, and P. H. Dakore, "Face Recognition Using Facenet Deep Learning Network for

- Attendance System,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3307, pp. 458–463, 2022.
- [4] B. Xu *et al.*, “CattleFaceNet: A cattle face identification approach based on RetinaFace and ArcFace loss,” *Comput. Electron. Agric.*, vol. 193, p. 106675, 2022, doi: 10.1016/j.compag.2021.106675.
- [5] A. Kumar, A. Kaur, and M. Kumar, “Face detection techniques: a review,” *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 927–948, 2019, doi: 10.1007/s10462-018-9650-2.
- [6] S. Anwarul and S. Dahiya, “A Comprehensive Review on Face Recognition Methods and Factors,” *Proc. ICRIC 2019*, pp. 495–514, 2020.
- [7] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review: 2005–2019,” *Appl. Soft Comput. J.*, vol. 90, pp. 2005–2019, 2020, doi: 10.1016/j.asoc.2020.106181.
- [8] A. Anwar and A. Raychowdhury, “Masked Face Recognition for Secure Authentication,” *arXiv Prepr. arXiv2008.11104*, pp. 1–8, 2020, [Online]. Available: <http://arxiv.org/abs/2008.11104>
- [9] N. Hidayat, S. Wahyudi, and A. A. Diaz, “PENGENALAN INDIVIDU MELALUI IDENTIFIKASI WAJAH MENGGUNAKAN METODE YOU ONLY LOOK ONCE (YOLOv5),” *UNEJ e-Proceeding*, pp. 85–98, 2022.
- [10] B. Mandal, A. Okeukwu, and Y. Theis, “Masked Face Recognition using ResNet-50,” *arXiv Prepr. arXiv2104.08997*, 2021, [Online]. Available: <http://arxiv.org/abs/2104.08997>
- [11] R. A. Pratama, S. Achmadi, and K. Auliasari, “Penerapan Metode Convolutional Neural Network Pada Aplikasi Deteksi Wajah Pengunjung Perpustakaan,” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 1, pp. 253–258, 2022, doi: 10.36040/jati.v6i1.4517.
- [12] S. Malakar, W. Chiracharit, K. Chamnongthai, and T. Charoenpong, “Masked face recognition using principal component analysis and deep learning,” *ECTI-CON 2021 - 2021 18th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol. Smart Electr. Syst. Technol. Proc.*, no. May, pp. 785–788, 2021, doi: 10.1109/ECTI-CON51831.2021.9454857.
- [13] M. S. H. Al-tamimi and F. A. Mohammed Ali, “Face mask detection based on algorithm YOLOv5s,” *Int. J. Nonlinear Anal. Appl.*, vol. 14, no. 1, pp. 679–697, 2022, doi: 10.22075/ijnaa.2022.28178.3824.
- [14] Mardiana, M. A. Muhammad, and Y. Mulyani, “Library Attendance System using YOLOv5 Faces Recognition,” *2021 Int. Conf. Converging Technol. Electr. Inf. Eng. Converging Technol. Sustain. Soc.*, no. February 2022, pp. 68–72, 2021, doi: 10.1109/ICCTEIE54047.2021.9650628.
- [15] J. Liu and D. Li, “Research on Moving Object Detection of Animated Characters,” *Procedia Comput. Sci.*, vol. 208, pp. 271–276, 2022, doi: 10.1016/j.procs.2022.10.039.
- [16] Z. Chen *et al.*, “Plant Disease Recognition Model Based on Improved YOLOv5,” *Agronomy*, vol. 12, no. 2, 2022, doi: 10.3390/agronomy12020365.
- [17] C. Jiang *et al.*, “Object detection from UAV thermal infrared images and videos using YOLO models,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 112, no. October 2021, p. 102912, 2022, doi: 10.1016/j.jag.2022.102912.
- [18] F. Jubayer *et al.*, “Detection of mold on the food surface using YOLOv5,” *Curr. Res. Food Sci.*, vol. 4, pp. 724–728, 2021, doi: 10.1016/j.crf.2021.10.003.
- [19] Y. Li, J. Wang, H. Wu, Y. Yu, H. Sun, and H. Zhang, “Detection of powdery mildew on strawberry leaves based on DAC-YOLOv4 model,” *Comput. Electron. Agric.*, vol. 202, no. May, p. 107418, 2022, doi: 10.1016/j.compag.2022.107418.
- [20] G. Jocher, “YOLOv5 in PyTorch,” *GitHub*, 2020. <https://github.com/ultralytics/yolov5> (accessed Oct. 10, 2022).
- [21] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, “A forest fire detection system based on ensemble learning,” *Forests*, vol. 12, no. 2, pp. 1–17, 2021, doi: 10.3390/f12020217.

- [22] M. N. Baay, A. N. Irfansyah, and M. Attamimi, "Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.59790.
- [23] F. M. Qotrunnada and P. H. Utomo, "Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker," *Prisma*, vol. 5, pp. 799–807, 2022.
- [24] Roboflow, "Give your software the sense of sight," *Roboflow*, 2016. <https://roboflow.com/> (accessed Oct. 25, 2022).