JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 10.30811/jtrik.v8i1.7422
Vol. 8 No. 1 2025 pp. 56-69

Mask Detection and Facial Recognition System Using a Convolutional Neural Network Algorithm

M. Fariz Maulana[1], Hendrawaty[2], Muhammad Rizka[3*]

[1,2,3] Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

*Corresponding Author: riska@pnl.ac.id

Article info: Received 04/02/2025, Revised 17/02/2025, Accepted 04/03/2025

Abstract

During the current COVID-19 pandemic, wearing a mask is mandatory for everyone when engaging in various activities to prevent the spread of the COVID-19 virus. So far, the detection of mask use has been done manually through observation by officers. This method has several limitations, namely that it cannot be performed every time and in every place. Based on these problems, the researchers designed a Mask Detection System and Face Recognition Using Convolutional Neural Network. This system is designed for the Department of Information and Computer Technology at the Lhokseumawe State Polytechnic. The system input is in the form of a real-time video camera that can detect students both wearing and not wearing masks. Every student who does not wear a mask will be recognized by their face. The system will send a notification to the officer if any students are not wearing masks. The system utilizes a mask dataset sourced from the Kaggle website, and for facial recognition, it employs facial data from 4C IT class students. The results obtained in the making of the CNN model that were tested resulted in 99% accuracy, and real-time testing resulted in an overall accuracy rate of 75%, mask detection accuracy of 83%, face detection accuracy of 71%.

Keywords: mask, detection, pandemic, covid-19, facial recognition

1. Introduction

The coronavirus infection known as COVID-19 (Coronavirus Disease, 2019) was first discovered in Wuhan, China, in late December 2019. This virus is highly contagious and has spread to almost every country, including Indonesia, within just a few months. This has led several countries to implement lockdown policies to prevent the virus from spreading further. This situation has impacted all aspects of people's lives, including the economy, social life, culture, defense, security, and politics[1]. Wearing a mask is crucial during the current pandemic. Masks or face coverings function as respiratory protection by protecting air entering and exiting the body (through the mouth and nose) from exposure to objects that transmit diseases through the respiratory tract, including the COVID-19 virus. Because COVID-19 is primarily transmitted through droplets, masks can act as a barrier against droplets, both from within and from others.[2] However, this only applies if the majority, if not all, of the population follows the guidelines. If many people do not, mask use as a preventative measure becomes less effective. The government has implemented the New Normal, an initiative aimed at maintaining public productivity during the COVID-19 pandemic. Under this new normal, people must adopt healthier lifestyles and adhere to health protocols, such as maintaining physical distance and always wearing masks in public spaces, to prevent the spread of COVID-19 while engaging in outdoor activities. Therefore, discipline is required from every member of the public to follow existing protocols [1].

Lhokseumawe State Polytechnic (PNL) is an educational institution located in Lhokseumawe. During the implementation of the PPKM (Restrictions on Public Activities) policy, the Lhokseumawe State Polytechnic campus implemented strict health protocols, limited on-campus activities, and conducted lectures online. Once the COVID-19 situation stabilizes, face-to-face lectures will resume for a trial period, initially for two cohorts of students.

Lectures will be conducted by health protocols, including vaccination, regular swab tests, physical distancing, hand washing, and mask-wearing. One of the most important aspects of implementing health protocols is wearing a mask during lectures. Students are checked for mask-wearing by lecturers or security.
The following are the problems with manually detecting masks at the entrance of the Lhokseumawe State Polytechnic ICT Building:

- A dedicated officer must be assigned to detect people who are not wearing masks.
- Officers must be on standby at the entrance.
- Officers' detection/checking time is limited to a few hours.
- If no officers are present, there will be no one to detect people who are not wearing masks.

The proposed solution is to create a system capable of assisting officers in mask detection by applying a Convolutional Neural Network (CNN) method in real time through a camera. The system is divided into two main stages: facial location detection and facial classification. Input consisting of multiple faces is detected using the Haar Cascade Classifier method. Then, each detected face is classified using the CNN method. If a mask is detected, the system recognizes the face and notifies the officers/security personnel.
This system is expected to assist officers in mask-checking and increase student awareness of mask-wearing.
CNNs, like most neural network methods, are composed of layers of neurons with adjustable weights and biases. Each neuron receives input from the input layer, performs a dot product on the subsequent layers, and produces an output at the output layer.[3] Generally, the more layers a network has, the higher its accuracy and complexity. However, at a point, diminishing returns occur, where increasing the number of layers does not improve network performance.[4] The network is trained to take raw data and correlate it with the final score. The higher the final score, the more biased the network is toward the specific configuration that produced that score [5]. CNNs are deep neural networks, a neural network architecture consisting of two or more hidden layers. A distinctive characteristic of CNNs is that the input data is always two-dimensional and in the form of images, allowing for specific properties that can be encoded into the network architecture [6].
CNN generally consists of three types of neuron layers: Convolutional Layer, Pooling Layer, and Fully Connected Layer. These layers are then stacked to form a complete CNN architecture. In the first layer, the Convolution Layer, a small-dimensional filter is applied to the incoming image (represented in matrix form). When it exceeds a certain pixel value, the filter produces an output in the form of an image matrix. The second layer, the Pooling Layer, downsamples the data to reduce the computational cost of processing the matrix. In the third layer, the Fully-Connected Layer, is a layer that has a full connection to the previous layers, this layer is the same as in a normal neural network [7]. In CNN, there are five types of layers used, namely:

a. Convolutional Layers
   Constituting the backbone of a CNN, the convolution layer takes image data and applies a sliding kernel/filter to the image. The kernel then extracts features using a dot product. The resulting dot product is then passed on to the next layer. Each kernel in the convolution layer has a weight equal to the kernel size, which is then modified using backpropagation. A single convolution layer can have more than one kernel type, each with a different weight and extracting different features. All these kernels then form a new image (the dot product image) with a channel depth corresponding to the number of kernels. The convolution layer accepts input with a 2-D spatial dimension (matrix), while the fully connected layer only accepts input with a 1-D spatial dimension (array).
b. Pooling Layers
   Pooling layers are used to downsample the image matrix, reducing the cost of the architecture. This layer has no weights and only acts as a filter.

c. Fully Connected Layers
   The FC layer in CNNs is the same as the FC layer found in other neural networks. The FC and Convolution layers have similar functions in training and storing model weights. However, because it only supports data with one spatial dimension, FC should only be implemented at the end of the architecture. This prevents data loss by converting 2D spatial image data to 1D spatial data.
d. Padding Layer
   When convolution is performed, the image size will decrease.
   To maintain the image size, the image is padded using a Padding Layer. The padding layer does not store the neural network weights.
e. Dropout Layer
   A dropout layer is inserted to randomly disable neurons in the previous layer (usually placed after the fully connected layer). The use of a dropout layer is generally employed to prevent overfitting, as neurons become too dependent on each other. As a result, a more independent and flexible model is produced, with improved training performance [5].

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

Python, founded in 1991 by developer Guido Van Rossum, is a high-level programming language with an interpreted, general-purpose, and object-oriented programming style. Python allows programmers to code more simply and concisely than Java or C++[8]. Some of the advantages of Python chosen for this research include its numerous available libraries, which are still being developed. Another advantage is its ability to integrate with other programming languages, such as C, C++, or Java [2]. This research utilized several libraries, including Keras, TensorFlow, Scikit-Learn, NumPy, OpenCV, and Matplotlib. Keras is a library designed for building machine learning systems. It is widely used in machine learning development because it is easy to implement and saves script code. Numpy is a library that simplifies calculations for multidimensional arrays. In this research, the input data is an image with many pixel values and will be entered into a multidimensional array. OpenCV is a library for image processing and computer vision. Matplotlib is a library used to display analytical data in chart form and can also display images neatly.

Haar-like features, also known as Haar Cascade Classifiers, are rectangular features that provide specific indications of an image. The Haar Cascade Classifier originated from the ideas of Paul Viola and Michael John. The idea behind Haar-like features is to recognize objects based on simple feature values, not the pixel values of the object's image [9]. The Haar Cascade Classifier algorithm is one algorithm used to detect faces. This algorithm is capable of quickly and in real time detecting objects, including human faces. The Haar Cascade Classifier algorithm has the advantage of fast computation because it relies only on the number of pixels in a rectangular image. OpenCV has a model based on this Cascade Classifier concept, called the FrontFace HaarCascade Classifier, which can be used to detect faces [10].

2. Methods

A. General Architecture

Figure 1 explains the general architecture of the system design. The process begins by collecting datasets, which consist of mask datasets and face datasets. Each dataset is then preprocessed. The datasets are then separated into training and test data. The training data is used to train the algorithm, while the test data is used to test, evaluate, and measure the performance of the machine learning model. The machine learning model is then designed and implemented on the website.

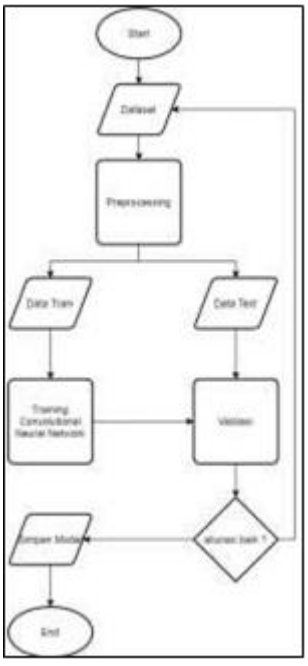


Figure 1. General Architecture

B. Dataset

The dataset used in this study consisted of 12,000 images. The dataset is divided into two categories: mask data and face data. The mask data consists of 1,000 images taken from www.kaggle.com/ using the keyword "face

mask." The face data consists of 11 student faces, each containing 100 images.

C. Preprocessing

The preprocessing process aims to reduce (resize) each image to a specific size to facilitate the process of designing a machine learning model. The steps involved include:

- Grayscale

  Grayscale converts all RGB images to grayscale. This stage is used to simplify the image model.

- Resize

  The resize process is carried out to equalize the size of all images to be processed. This process facilitates and speeds up the analysis process because the image size is compatible with the device's capabilities.

- Augmentation

  This stage addresses the problem of small datasets across several classes. Data can be added to increase the sample size. The augmentation performed in this study included image rotation and either adding or subtracting light. After this augmentation, the number of images in each dataset increased 20-fold, from 100 to 2,000 for each student.

- Labeling

  This stage involves labeling the faces and masks of students for each augmentation. Machine learning algorithms cannot directly process categorical data, particularly for classification problems that utilize deep learning methods. In the label encoding process, categorical data is converted into numerical values.

D. Machine Learning Model Design

The machine learning model design stage begins with defining training data (x) and test data (y). Training data (x) is independent and is used to train the algorithm, while test data (y) is dependent and is used to test, evaluate, and measure the performance of the machine learning model. The training data proportion is 85%, and the test data proportion is 15%. The stages of the convolutional neural network algorithm are:

- Convolution & ReLU
  The image processed in the previous stage will undergo a convolution process. In this stage, the image will be represented as a matrix consisting of numbers 0 to 255. Then, the image will be convolved with several filters. The results will be forwarded to the ReLU process, which is performed by converting each convolution image element with a value below 0 (minus/negative) to 0.
- Pooling Layer
  The output value of the convolution layer is divided into several small boxes, and then the max pooling function is used to extract the maximum value from each box to construct a predetermined image matrix, thereby reducing the dimensionality of the feature matrix.
- Flatten
  The input values are converted into an array of pooled values. Each result of the Pooling Layer process is converted into a one-dimensional array.
- Fully Connected Layer
  The process by which a flattened matrix is fed through the neural network to predict the output probability. The flattened value from the process will be used to train the neural network.

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

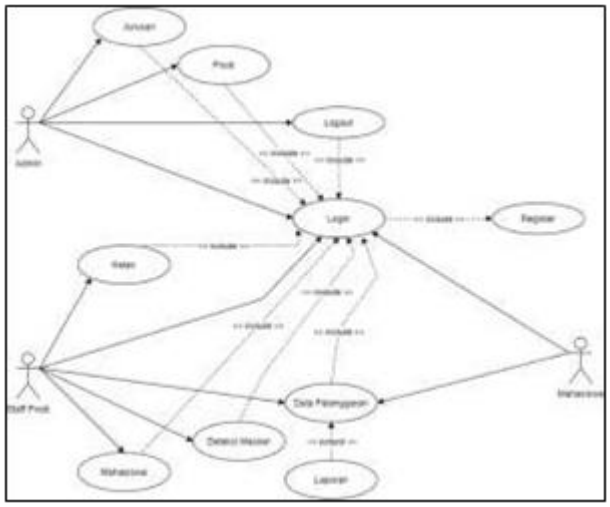


Figure 2. Use Case Diagram

E. Use Case Diagram

A use case diagram is one of the various types of UML (Unified Modeling Language) diagrams that illustrate the interaction between a system and its actors. Use cases can describe the type of interaction between system users and the system. This study encompasses three distinct use cases: admin, study program staff, and student use cases.

The explanation of Figure 2 is as follows:

- Students

Students are one of the users of this system. Students can only view violation data.

- Study Program Staff: The Study Program Staff is one of the users of this system. Study Program Staff can perform more processes than students, such as inputting class and student data, viewing violation data, and viewing the mask detection monitor display.
- Admin: Admins are the users who manage this system. Admins can perform processes such as inputting department and study program data, as well as creating accounts for study program staff.
- Department: Departments contain department data that will relate to study programs.
- Study Programs: The study program has data that will relate to the class.
- Classes : Classes contain class data that will relate to study programs.
- Students : Students contain student data that will relate to violations.
- Violation Data : The Violation Data page stores data on students who commit violations, such as not wearing a mask.
- Reports : The Reports page allows printing reports on mask violation data.
- Mask Detection : Mask Detection is a menu for viewing the mask detection camera display.
- Login : Users must log in to access various features in the application.
- Register : Register is the first step users take to register their data in the application to access it.
- Logout : Explains the activities that users can perform. Selecting this menu will log them out of the application.

F. Entity Relationship Diagram

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

The Entity Relationship Diagram (ERD) used in this system includes the table entities users, department, study program, class, student, and violation. The relationships between the tables in this mask detection and facial recognition application are illustrated in Figure 3.

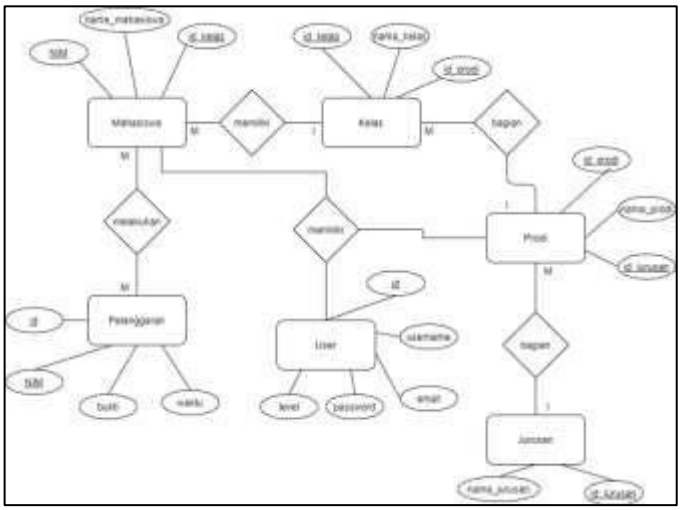


Figure 3. Entity Relationship Diagram

Figure 3 illustrates the ERD for the Mask Detection and Facial Recognition System, utilizing the Convolutional Neural Network Algorithm, which comprises six primary entities. The following explains them:

- User Table : The user table has five attributes: id, username, email, password, and level. ID serves as the primary key.
- Department Table : The Department table has two attributes: department_id and department_name. id_jurusan serves as the primary key.
- Study Program Table : The Study Program table has three attributes: study program_id, student_name, and department_id. id_prodi serves as the primary key, and id_jurusan serves as the foreign key.
- Class Table : The Class table has three attributes: class_id, class_name, and study program_id. id_kelas serves as the primary key, and id_prodi serves as the foreign key.
- Student Table : The Student table has three attributes: student ID, student name, and class ID. NIM serves as the primary key, and id_kelas serves as the foreign key.
- Violation Table : The Violation table has four attributes: id, student ID, proof, and time. ID as the primary key and NIM as a foreign key.

3. Results and Discussion

A. Process Implementation

The process implementation outlines the procedures and steps involved based on the machine learning model design described above.

- Grayscaling

  The grayscaling process converts all RGB images into grayscale images. The results of the grayscaling process are shown in Figure 4.

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN K
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

Gambar 4. Grayscaled Image

- Image Resizing

  The image resizing process reduces all images to a 50x50 resolution. The results of the image resizing process are shown in Figure 6.

Figure 5. Resized Image

- Augmentation

  The augmentation process is performed on each image in the dataset. Transformations include image rotation, image shifting, and random grayscale color adjustments, both light and dark, within a predetermined parameter range. After this augmentation, the number of images in each dataset increases 20-fold, from 100 to 2,000 for each student. The results of the augmentation process are shown in Figure 6.

Figure 6. Image Augmentation Results

- Labelling

  The dataset labeling process is shown in Figure 7. The dataset labeling process is carried out after the dataset has been augmented. Each augmented image is labeled based on the folder name within the dataset. Each labeled dataset contains 2,000 images, each with a numerical value. The labels also serve to

translate the output values from numerical to categorical values, allowing the detection results to be understood.

```
le = LabelEncoder()

le.fit(names)

labels = le.classes_

name_vec = le.transform(names)

categorical_name_vec = to_categorical(name_vec)
print("number of class :", len(labels))
print(labels)
```

✓ 0.5s

```
number of class : 12
['1857301003' '1857301019' '1857301021' '1857301023' '1857301033'
 '1857301035' '1857301041' '1857301047' '1857301055' '1857301059'
 '1857301061' 'mask']
```

Figure 7. Labelling Process

- Model Training

The machine learning model design stage begins with defining the training data (x) and test data (y). The training data (x) is independent and is used to train the algorithm, while the test data (y) is dependent and is used to test, evaluate, and measure the performance of the machine learning model. The training data comprises 85% of the total data, consisting of 20,400 images, and the test data comprises 15% of the total data, consisting of 3,600 images. There are several processes involved in using the convolutional neural network algorithm, including convolution layers, pooling layers, and fully connected layers, which consist of flattened and dense layers. A summary of the layer structure is shown in Figure 8.

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

Layer (type)	Output Shape	Param #
conv2d_36 (Conv2D)	(None, 48, 48, 64)	640
conv2d_37 (Conv2D)	(None, 46, 46, 64)	36928
max_pooling2d_18 (MaxPooling2D)	(None, 23, 23, 64)	0
conv2d_38 (Conv2D)	(None, 21, 21, 128)	73856
conv2d_39 (Conv2D)	(None, 19, 19, 128)	147584
max_pooling2d_19 (MaxPooling2D)	(None, 9, 9, 128)	0
flatten_9 (Flatten)	(None, 10368)	0
dense_27 (Dense)	(None, 128)	1327232
dense_28 (Dense)	(None, 64)	8256
dense_29 (Dense)	(None, 12)	780
activation_9 (Activation)	(None, 12)	0

Total params: 1,595,276
Trainable params: 1,595,276
Non-trainable params: 0

Figure 8. Model Layers Stucture

- Testing and Evaluation

  In this test, the authors tested 10 epochs. Figure 9 shows the accuracy results for 10 epochs. The test results indicate that the model effectively adapts the training data during the learning phase. This is illustrated by the increasing accuracy graph as the number of epochs increases. The model achieved an accuracy of 99.90% and a val_accuracy of 99.74%. The val_accuracy value shows the comparison between the validation data results and the model's predictions for each training period.

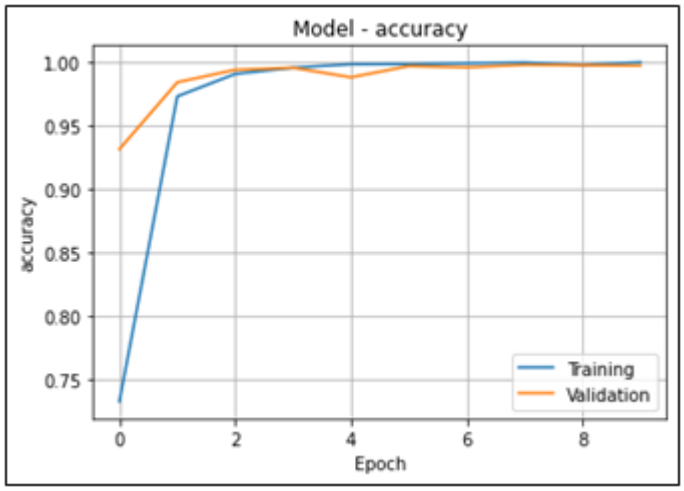


Figure 10. Accuracy Results with 10 epochs

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

The test results indicate that the model effectively adapts the training data during the learning phase. The increasing accuracy graph illustrates this as the number of epochs increases. The model has an accuracy of 99.90% and a val_accuracy of 99.74%. The val_accuracy value shows the comparison between the validation data results and the model's predictions for each training period.

B. System Implementation

The system implementation describes the implementation of the system design.

- Login Page: The login page is the process for logging into an application. The system login page can be seen in Figure 11. Login Page Interface

Figure 11. Login Page

- Dashboard Page

This page is the initial page accessed by the Operator after completing the login process. The Operator Dashboard page displays the system's data, such as: Number of Students, Number of Violations, Number of Majors, Number of Study Programs, and Number of Classes. Figure 12 shows the Operator Dashboard Interface.

Figure 12. Dashboard Page

- Violation Data Page

On this page, study program staff can view detailed student data and delete student violation data from the student violation table.

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

Figure 12. Violation Page

Violation data is automatically entered into the program if it detects a student not wearing a mask. The student violation page displays the following in Figure 12. The Student Violation Data Page Interface.

- Violation Details Page

  The violation details page displays the identity and photo of students who are not wearing masks. The displayed identity includes name, student ID number, class, study program, major, and time of the violation. This data is automatically entered by the system when students are not wearing masks. The Violation Details page interface is shown in Figure 13.

Figure 13. Violation Details Page

- Violation Report Page

  The report page is accessible only to staff members of the study program. This page displays reports of detected violations. To print the report, users can filter by month, department, and study program. Then, click "Print Report," and the report will be automatically downloaded. The report page, as seen in Figures 14, displays the report results from the previously filtered data.

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10

Figure 14. Violation Report Page

- Camera Page

The camera page is a dedicated page for the study program staff to display the mask detection monitor.

Figure 15. Face Acquisition Without Using Mask

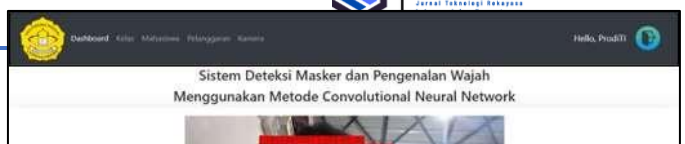
This page displays real-time video from the camera used for mask detection. Faces not wearing masks are marked with a red box, and the displayed face frame and student ID data are sent to the student violation database. Faces detected wearing masks are identified with a green box. Figure 15 shows a face detected not wearing a mask, and Figure 16 shows a face wearing a mask.

Figure 16. Face Acquisition Using Mask

C. System Testing

System testing is the final stage of software development. System analysts conduct comprehensive system testing to ensure proper system performance. This stage involves testing the mask detection system and facial recognition capabilities.

JURNAL TEKNOLOGI REKAYASA INFORMASI DAN KOMPUTER
E-ISSN: 2797-1724, DOI: 00.000/XXXX.YYYY.II.NNN
Vol. 1 No. 1 2025 pp. 01-10



Tabel 1. Testing Results

No	Deteksi	Kondisi	Prediksi	Hasil
1	Masker	Jarak < 1 m dan cahaya cerah	*Mask*	Sesuai
2	Masker	Jarak > 1 m	*Mask*	Sesuai
3	Masker	Jarak > 1 m	*Mask*	Sesuai
4	Masker	Jarak > 1 m	*Mask*	Tidak Sesuai
5	Masker	Jarak < 1 m dan cahaya redup	*Mask*	Sesuai
6	Masker	Jarak < 1 m dan cahaya redup	Tidak dikenal	Sesuai
7	Wajah	Jarak < 1 m	1857301061	Sesuai
8	Wajah	Jarak < 1 m	1857301023	Sesuai
9	Wajah	Jarak < 1 m dan *multidetect*	1857301061 1857301023	Sesuai
10	Wajah	Jarak < 1 m dan *multidetect*	1857301023 Tidak dikenal Tidak dikenal	Sesuai
11	Wajah	Jarak < 1 m dan *multidetect*	1857301023 1857301033 Tidak dikenal Tidak dikenal	Tidak Sesuai
12	Wajah	Jarak > 1 m dan *multidetect*	1857301023 Tidak dikenal Tidak dikenal	Tidak Sesuai
13	Wajah	Jarak > 1 m	1857301003 Tidak dikenal	Sesuai
14	Wajah	Jarak > 1 m	1857301035	Sesuai
15	Wajah	Jarak > 1 m	1857301023	Tidak Sesuai
16	Wajah	Jarak > 1 m	1857301003	Tidak Sesuai
17	Wajah	Jarak > 1 m	1857301003	Sesuai
18	Wajah	Jarak > 1 m	1857301003	Sesuai
19	Wajah	Jarak < 1 m	1857301003	Sesuai
20	Wajah	Jarak > 1 m	Tidak dikenal	Sesuai

4. Conclusion

After testing and implementation, it can be concluded that the system can be used for mask detection and facial recognition. However, factors that influence the system's performance include light intensity, distance, and camera quality. The Mask Detection and Facial Recognition System using the Convolutional Neural Network algorithm achieved an accuracy of 99.90% and a val_accuracy of 99.74% using epochs hyperparameters of 10. In real-time testing, the overall accuracy rate was 75% from 20 trials.

REFERENCES

[1] M. N. Baay, "Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning," ITS, 2021.
[2] A. S. Nainggolan, "Implementasi Convolutional Neural Network Untuk Deteksi Pengguna Masker," UIN Suska Riau, 2021.
[3] H. Abhirawa, "Pengenalan Wajah Menggunakan Convolutional Neural Network," Univ. Telkom, 2017.
[4] Felix, "Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun" STMIK Mikroskil, 2020.
[4] V. D. Win, "Pengenalan Wajah Menggunakan Convolutional Neural Network," Inst. Teknol. Sepuluh Nop. Surabaya, 2018.
[5] I. W. Suartika, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," Inst. Teknol. Sepuluh Nop., 2016.
[6] R. A. Z. Fajar Astuti Hermawati, "Sistem Deteksi Pemakaian Masker Menggunakan Metode Viola-

Jones dan Convolutional Neural Networks (CNN)," 2021.
[7] R. F. Muharram, "Implementasi Artificial Intelligence Untuk Deteksi Masker Secara Realtime Dengan Tensorflow Dan Ssd Mobilenet Berbasis Python," Univ. Indraprasta PGRI, 2021.
[8] B. Budiman, "Pendeteksian Penggunaan Masker Wajah Dengan Metode Convolutional Neural Network," Univ. Tarumanagara, 2020.
[9] G. A. Anarki, "Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker," Inst. Teknol. Nas. Malang, 2021.