Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 22, No. 1, November 2022, pp. 241∼258
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v22i1.2350

Automatic Door Access Model Based on Face Recognition using Convolutional Neural Network

Tjut Awaliyah Zuraiyah[1], Sufiatul Maryana[2], Asep Kohar[3]
Universitas Pakuan, Bogor, Indonesia

Article Info

Article history:

Received August 30, 2022
Revised September 28, 2022
Accepte November 25, 2022

Keywords:

Automatic door access
Amazon face recognition
Convolutional Neural Network
Face Recognition
Raspberry

ABSTRACT

Automatic door access technology by utilizing biometrics such as fingerprints, retinas and face structures is constantly evolving. Face recognition and the use of masks are widely used to access doors automatically, so there is difficulty recognizing someone who is wearing a mask. The study aimed to create an automated door access model using convolutional Neural Network (CNN) algorithms and Amazon Rekognition as cloud-based software. The CNN algorithm is applied to classify faces wearing masks or not wearing masks. The CNN architecture model uses sequential, convolution2D, max polling 2D, flatten dan dense. The hardware includes the Raspberry Pi, USB Webcam, Relay, and Magnetic Doorlock. The test results were obtained from the results of the accuracy plot on the Convolutional Neural Network model with an accuracy rate of 99% at an epoch value of 8 with a learning time of 67 seconds

Corresponding Author:

Tjut Awaliyah Zuraiyah,
Departemen of Computer Science,
Universitas Pakuan, Bogor, Indonesia,
Email: tjut.awaliyah@unpak.ac.id

How to Cite: T. Zuraiyah, S. Maryana, and A. Kohar, Facial Recognition-Based Automatic Door Access Model using Convolutional Neural Network, MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 22, no. 1, pp. 241-252, Dec. 2022.

1. INTRODUCTION

A few years ago, the door known to people was just a manual door, but now automatic doors have been widely used. Traditional security systems require a key, password, RFID card or ID card to be able to access. The downside is that it is difficult to remember and can be duplicated or stolen by others. Therefore, a way is needed to increase security in certain rooms that concern authority or privacy. A form of protection that is difficult for others to imitate, open or modify is to use the automatic characterization of the biological characteristics that every human being has always had and characterized. Such characteristics are known as biometrics. These characteristics can be seen in physical characteristics, such as fingerprints, facial expressions, retinas of the eyes, and voice. One of the most commonly used ways to create security systems from these automated doors is to use facial recognition systems [1]. Face recognition is one of the most popular biometric methods [2]. Faces are harder to imitate, modify, or steal when compared to keys or passwords on non-biometric security. The face detection system is expected to provide a warning in the form of markers on faces that do not use masks properly. In addition, this system must be able to recognize all types of masks of various colors and shades. The use of these various motifs requires the application of Artificial Intelligence for data on the training of various mask motifs. Several similar studies have been carried out on a server-based basis, but their success depends largely on the capacity of servers available to be able to process in large quantities. Computer vision is a technology that supports the decision to recognize an image using sensors [3]. One of the techniques that process image data to classify faces that wear masks or not is to use deep learning. Amazon Face Recognition is a Face recognition system created by Amazon Web Services, owned by the Amazon company, launched in 2016 as cloud-based software as a computer vision platform that can perform Face recognition, commonly known as face recognition [2]. Another method in processing Face images and computer vision that is often used is the Convolutional Neural Network algorithm. Algorithm consists of several layers, including Convolutional Layer, Subsampling Layer, and Fully Connected Layer. Some researchers have developed mask-masked or mask-wearing face detection systems using deep learning [4, 5]. The microcontroller module was used to build this system using the Raspberry Pi 4, with the advantages of the main chip of the Cortex-Z72 type microcontroller IC with Video core 6 GPU. There is also a USB 3.0 port that functions as a USB communication line that can be directly connected to the PC so that the program flashing process can take place faster, and it is the right choice in building this system. A single board computer (SBC) the size of a credit card. This component has been equipped with all the functions of a complete computer, using the arm's System-on-a-Chip (SOC), which is packaged and integrated on top of a circuit board [6–8].

2. RESEARCH METHOD

The research method used in this study includes several stages including: the stages of project planning, research, part testing, mechanical design, electrical design, software design, functional test, integration in Figure 1.

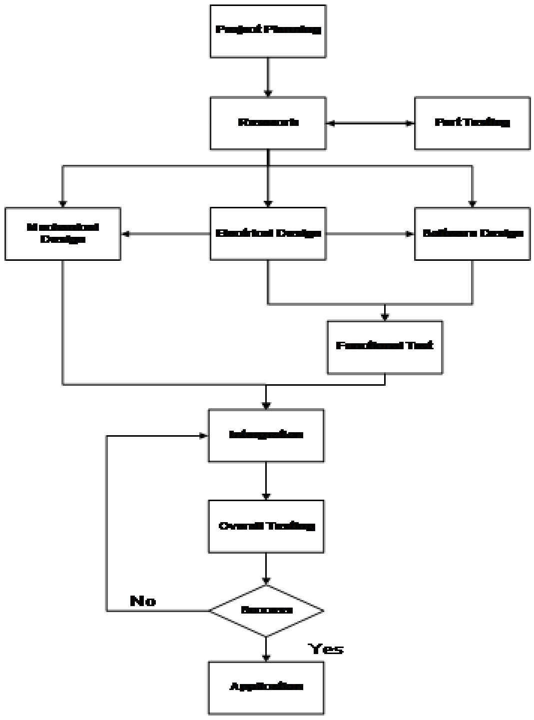


Figure 1. Flowchart of research stages

2.1. Project Planning

The planning stage of the research project is the activity of the system creation process. The components needed in designing the system are the Raspberry Pi 4 4GB, Magnetic Door Lock, USB Webcam, and Relay.

1. Analyze Hardware Needs

1. Microcontroller Module Selection
   The microcontroller module used to build this system only uses the Raspberry Pi 4, with the advantages of the main chip of the Cortex-Z72 type microcontroller IC with a Video core 6 GPU. There is also a USB 3.0 port that functions as a USB communication line that can be directly connected to a PC so that the program flashing process can take place faster, and it is the right choice in building this system.
2. In making this automatic door lock model, the hard equipment used includes raspberry pi, USB Webcam, Relay, and Magnetic Doorlock. AUSB Webcam camera is because of its small shape but has a fairly good resolution of 6 Megapixels, so it is very suitable for building this system.
3. The key used is Magnetic Doorlock, the advantage lies in this lock mechanism. This key utilizes electric current to create a magnetic force that is the main restraint so that this key will not be dangerous if something undesirable happens, such as fire, power outage, or other disasters, because this key will automatically open when there is no electric current [9–12].

2. Analyze Software Needs

The software used in this study is as follows: Visual Studio Code is a software used to facilitate the development of systems to be built, starting from writing program sources, debugging, and terminal trials, Python language is one of the languages used in microcontroller programs in addition to Basic and Assembly languages.

2.2. Research

After system planning, proceed with the initial research of the system to be created. At the research stage, the initial design of the mechanical circuit and components of this door lock system model was carried out to ensure that all components could run optimally. The system uses one Raspberry Pi 4 with a face image captured using a USB Webcam connected to the Raspberry Pi. The face image will be sent by the Raspberry Pi 4 to the Amazon Web Services server where there is a database containing the registered face data, so there will be a process of matching face data by Amazon Face Rekognition, which then the result of the matching will be returned to the Raspberry Pi 4, if the face is registered, then the system will ask to use a mask, after which the Webcam will recapture the image of the recognized face to be processed using the Convolutional Neural Network algorithm to detect the use of a mask on the face, after the mask is detected, the Raspberry Pi 4 will trigger a Relay to disconnect the electric current on the Magnetic Doorlock which will make the lock open.

2.3. Component Testing (Part Testing)

Testing the components to be used using a multimeter is carried out at this stage. Testing using Raspberry Pi monitoring is carried out by looking at the output of each component connected to Raspberry via USB co-ops [13]. Testing using a multimeter includes testing each component's input and output voltages.

2.4. Mechanical Design

Mechanical design D is an important thing to consider, as can be seen in Figure 2. In general, the application needs for mechanical design include:

1. Shape
2. Resilience and flexibility to the environment
3. Placement of electronic modules
4. Testing of designed mechanical systems
5. Form of system interface size design

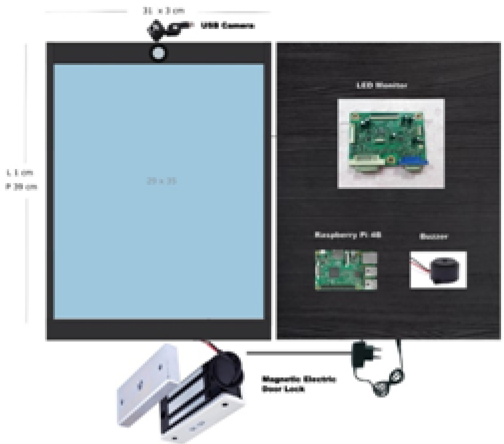


Figure 2. Mechanical System Design

Figure 2 The design of the mechanical system explains that there is a screen with a USB Webcam connected to the Magnetic Doorlock, and there is an electronic circuit to process commands and transmit data from the camera. The electronic circuit design uses the Raspberry Pi 4 microcontroller as the main processor and the Monitor LED as the interface between humans and tools.

2.5. Electronic Design

Schematic design of the circuit using draw.io software based on the block diagram in figure 2. Architectural Planning includes the Raspberry Pi 4 as the main data processor that will receive input in the form of a face image from a USB Webcam which will later be used for the facial recognition process and detection of the use of masks on the face, to open the Magnetic Doorlock.

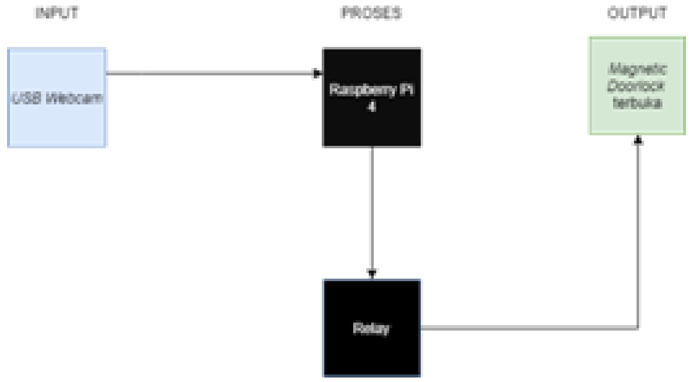


Figure 3. Block Diagram

The block diagram, as shown in figure 3, was then developed in the form of a hardware design. The architectural planning above is explained that there is a Raspberry Pi 4 as the main data processor that will receive input in the form of a face image from a USB Webcam which will later be used for the Face recognition process and detection of the use of masks on the face, to open the Magnetic Doorlock.

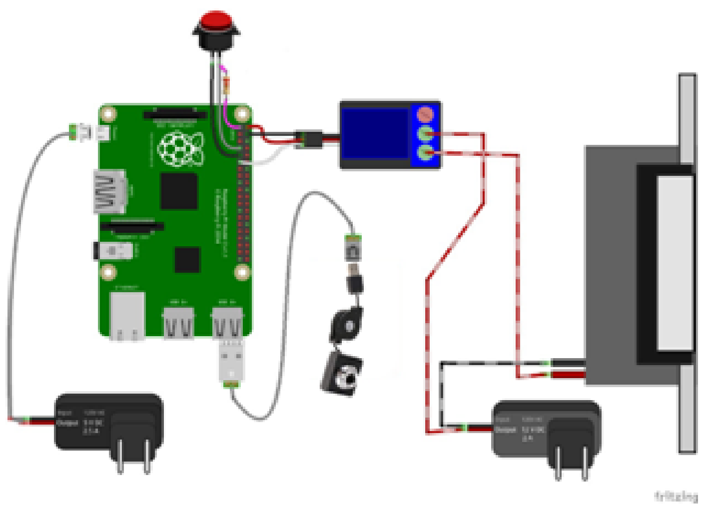


Figure 4. Circuit Schematic

The voltage source uses electricity that will supply current to each component. The voltage entering each component of the Relay, USB Webcam, and Raspberry Pi 4 is 5V, while the Magnetic Doorlock is 12V. Circuit schematic can be seen in Figure 5.

2.6. Software Design

The system software design was created with the Python Programming Language on the Raspberry Pi based on the flowchart in figure 5 and figure 6.

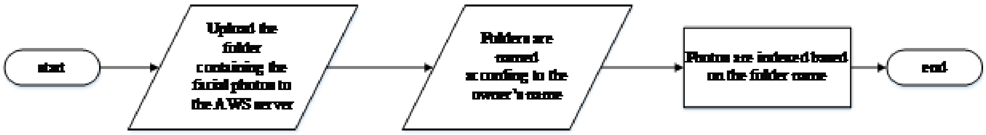


Figure 5. Flowchart Training System

The flowchart can be divided into two parts, namely, the flowchart when the system conducts the employee training or registration process and when the system carries out the Face recognition and mask detection process, which is the main process in this system. The training process is conducted on Amazon S3's cloud servers using Amazon Rekognition. This process is executed using a Personal Computer.

Initially, the training begins with uploading photos of employees registered as recognized persons to Amazon S3 or S3 Buckets. The photos are stored in their respective employee folders with folder names based on the employee's name. The next process in this training process is that Amazon Rekognition will index or label each uploaded photo based on the name of the folder where the photos are located the photo is saved. With this, every photo uploaded during this training process will have a label based on the employee's name, which is the folder's name, which can then be recognized by the system when carrying out the Face recognition process.

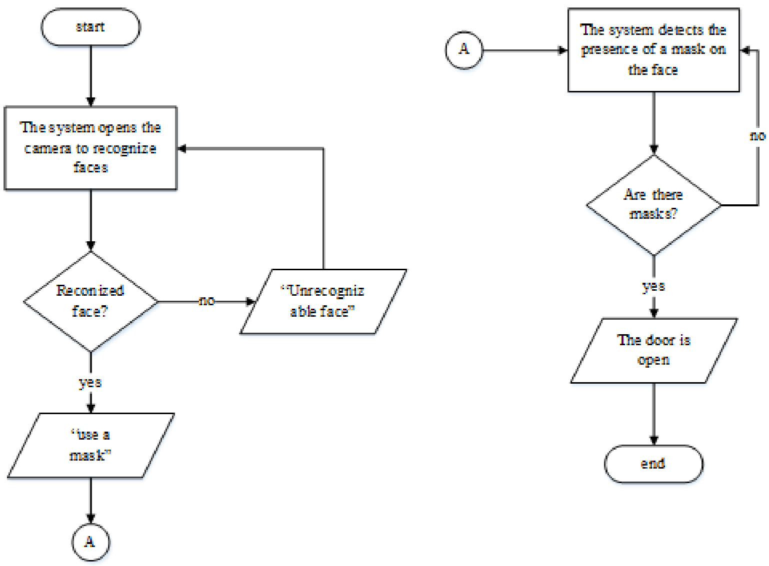


Figure 6. Overall System Flowchart

This flowchart describes the main process of this built system, namely the Face recognition and detection of masks on faces. This process begins with a Webcam that captures an image of the employee's face when the face is detected on a Webcam that is already connected to the Raspberry Pi. After that, Raspberry Pi will send the employee's Face image captured by the Webcam to Amazon Web Service to match the photo of the employee registered with Amazon S3 using Amazon Rekognition. If the photo of the face captured by the Webcam is recognized, then Amazon Web Service will return the result to Raspberry Pi that the face is recognized [14]. After that, the camera will detect the presence of a mask on the face of the recognized employee. If a mask is detected, the Raspberry Pi will trigger the Relay to cut off the electricity to the Magnetic Doorlock so that the lock will open.

2.7. Functional Test

Functional tests are performed on pre-designed software. This testing process is carried out to improve the performance of the software in controlling the electrical design and eliminate and anticipate errors from the software created. When the software system has been tested, it enters the assembly process.

2.8. Integration

In this process, the assembly process is carried out based on the design process, both mechanical, electronic, and software design.

1. Model Building Process

The model creation process includes the process of training a model with training data, as well as a model testing process to find out how accurately the model has been created. The model creation process starts with the training step, carried out by training with machine learning models used in the Face recognition process and face mask detection process. In the Face recognition process, the model used is Amazon Face Rekognition. Meanwhile, in the detection process of wearing a mask on the face, the model used is the Convolutional Neural Network algorithm.

The process of creating a model for Face recognition is carried out on a cloud server or Amazon Web Services using Amazon Rekognition [15]. In contrast, the process of creating a mask detection model is carried out on a Personal Computer. The Raspberry Pi here is not used for the model creation process but only as an executor of the model that has been built.

2. Amazone Face Recognition

In the Amazon Face Rekognition model training process, AWS-owned services will be used, namely Amazon S3, Amazon Rekognition, and Amazon Cloudwatch [16].

Amazon S3 or Simple Storage Service is an object storage service offering industry-leading scalability, data availability, security, and performance. In the model training process, this service acts as a database used to store photos of employees to be registered. There, photos of employees to be registered are stored in a folder with each employee's name.

Amazon Rekognition is a service that AWS or Amazon Web Services provides to analyze images and videos using deep learning. This service will be used as a model for the Face recognition process. Photos that have been uploaded to Amazon S3 will be indexed by Amazon Rekognition based on the name of the folder where the employee's photos are stored. With this, each photo stored in Amazon S3 will have a name that matches the name of the folder where the photo is stored.

The next AWS service to use is Amazon Cloudwatch. These services are used to monitor traffic usage of Amazon Web Services.

First, data in the form of photos of employees who want to be registered as face photos that the system will recognize is uploaded to an Amazon S3 or S3 Bucket. The total photo data uploaded to Amazon S3 is 38 photos. The employee's photo is uploaded and saved into a folder with the employee's name masing respectively, as in figure 7.

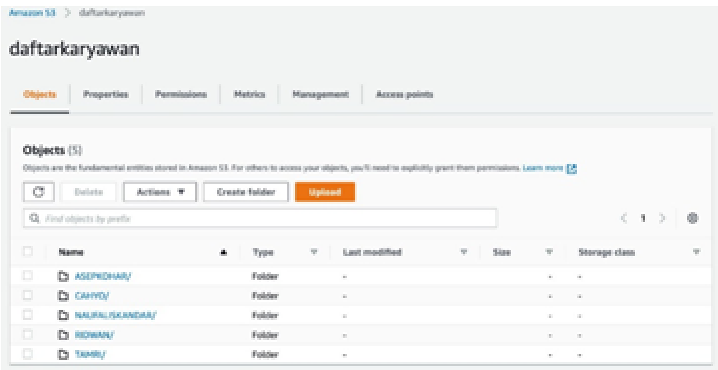


Figure 7. Uploading data to Amazon S3

It then trains the Amazon Rekognition model using the uploaded data to Amazon S3, in figure 8.

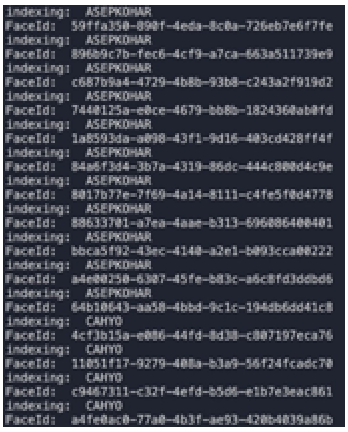


Figure 8. Amazon Rekognition model training process

In this Amazon Rekognition training process , the last step is to test the model that has been trained. The testing process is done by trying Face recognition capabilities on employees whose photos have been uploaded to Amazon S3, in figure 9.

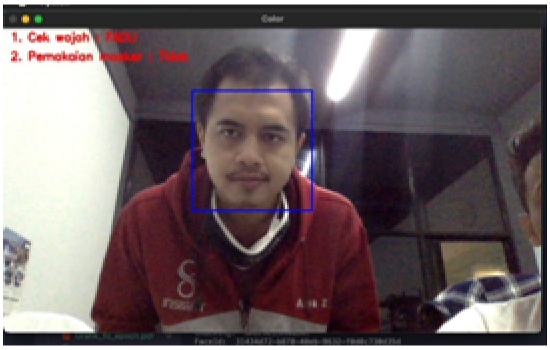


Figure 9. Test results on the faces registered employee

From the testing results, it can be concluded that the Amazon Rekognition model has succeeded in recognizing faces that have been registered with Amazon S3 so that the model is ready to implement in this study.

3. Convolution Neural Network

Convolutional Neural Network or CNN is part of several types that exist in neural networks or artificial neural networks that people commonly use to process image data. CNN is also a development of Multi Layer Perception (MLP) method. Compared to MLP, CNN has a more significant number of dimensions. Convolutional Neural Network is a deep learning algorithm that is included in the feedforward neural network, which means that this algorithm does not form cycles. Figure 10 illustrates a process of classification of vehicle images using a Convolutional Neural Network, which will explain the architecture of this algorithm where an input is entered directly into an artificial neural network, followed by several stages of convolution and Pooling. Next, this process will be handled by a layer called fully connected layer, which will provide an output in the form of a classification of the input [17].

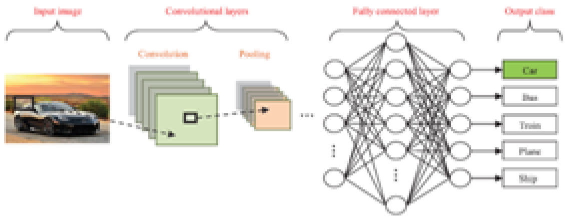


Figure 10. Illustration of image classification using CNN

Convolution or commonly called convolution, is a matrix that serves to filter images. Convolutional Neural Network (CNN) has several layers that are used to filter each process. That process is called the training process. There are three stages in the training process, the first is a Convolutional layer, the second is a Pooling layer, and the third is a Fully connected layer.

a Convolutional Layer
The Convolution layer is the first layer that extracts the features from the inserted image. All data that touches the convolutional layer will undergo a convolution process. In that layer, there will be an "encoding" of an image into a feature map in the form of numbers that represent the image (Feature Extraction). Convolution processes maintain relationships between pixels by studying image characteristics using mathematical operations between image matrices and filters or kernels. A kernel is an operator that is applied to the entire image to get the array value of an image. A kernel is a matrix usually measuring 3*3 or 5*5 with a random value between -1 and 1. Padding functions to add several pixels with a certain amount as needed. To get the final result of the convolution process of the layer, a formula is used as in the following Equation [18]:

$$Output = W - F + 2PS + 1 \quad (1)$$

Where:
W : Image size
F : Filter size/kernel
P : Padding
S : Stride

b Pooling Layer
The Pooling Layer is the next step of the Convolutional Layer. The result of the convolution of the image matrix with filters (kernel) is called the feature map or activation map. Meanwhile, ReLU or Rectified Linear Unit functions to change negative values on the feature map to positive.

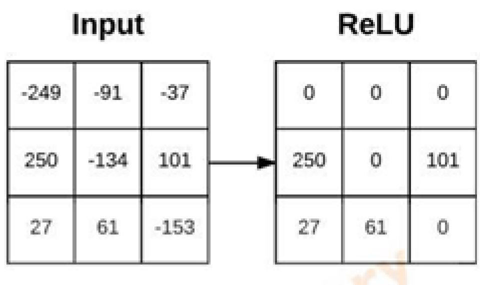


Figure 11. Example of Applying ReLu Activation

The Pooling Layer is composed of a filter with a certain size and stride. For example, the commonly used pooling layers are Maximum Pooling and Average Pooling. The Maximum Pooling process is a layer that reduces the spatial size to reduce the number of parameters and calculations when the image size is too large. For example, if you use Maximum Pooling 2x2 with Stride 2, then at each filter shift, the value taken is the most significant value in the 2x2 area, while Average Pooling will take the average value.

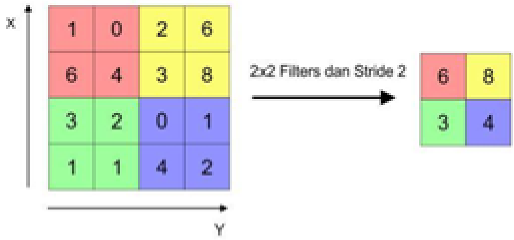


Figure 12. Max Pooling

c Fully Connected Layer
The feature map generated by the previous stage, namely the pooling layer, will be in the form of a multidimensional array. So, before proceeding to the Fully Connected Layer stage, the resulting Feature Map will first go through a "flatten" or reshape process. The flattening process will convert into a vector that can later be used as input from a Fully Connected Layer.

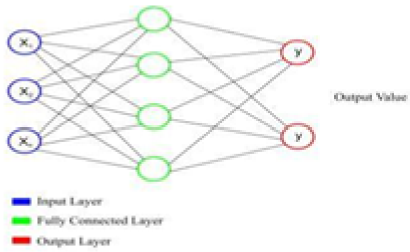


Figure 13. Fully Connected Layer

d Datasets
Training Data is divided into two: training data on Amazon Face Rekognition and training data on Convolutional Neural Network. The data training process on Amazon Face Rekognition is carried out by uploading a photo of the person to be registered on the system as a recognized person into an S3 bucket on the Amazon Web Services server. Each image is entered into a folder named after the ownership of the photo of the face.
Training data on the Convolutional Neural Network for the detection of the use of masks on the face is carried out by training the model using a dataset containing two folders consisting of a folder containing human photos with masks and a folder containing photos of humans without masks. The CNN training process was carried out with a total dataset of 690 photos of humans with masks and 686 photos without masks. The dataset is divided into two, namely 80% as training data and 20% as validation data.

3. RESULT AND ANALYSIS

The way the face detection tool using a mask works is seen in figure 4, including the camera, which will activate and detect the presence of a face. If a face is detected, the camera will capture an image of the face and then send it to the Raspberry Pi to then be sent to Amazon S3 to be compared with the face photo that has been registered in the database, which if the face is recognized, the LED Monitor will display the name of the person whose face was caught on camera. For example, if the face captured by the camera is a registered face, the Monitor LED will display the name of the person. After that, the LED Monitor will display a command to wear a mask, and the Raspberry Pi will trigger webcam to recapture the face image and then process using a Convolutional Neural Network to detect the use of masks on the face. If a mask is detected, the Raspberry Pi will trigger the Relay to cut off the electricity to the Magnetic Doorlock so that the lock will open.

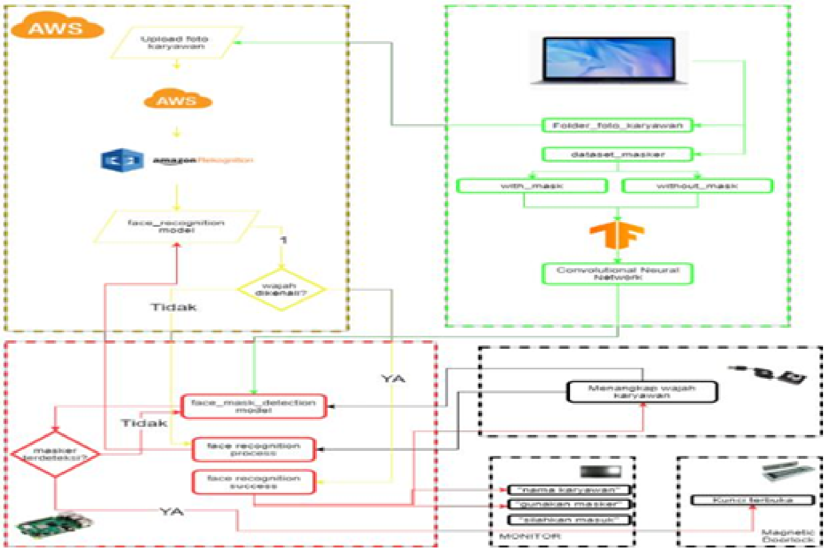


Figure 14. How the system works as a whole

3.1. Modeling Process

The model creation process includes the process of training models with training data and also the process of testing models to find out how accurate the model has been made. The model-making process begins with the training step first, carried out by training with machine learning models used in the facial recognition process and detecting masks on the face. In the facial recognition process, the model used is Amazon Face Rekognition. Meanwhile, in the detection process of using masks on the face, the model used is the Convolutional Neural Network algorithm.

The process of creating a model for facial recognition is carried out on a cloud server or Amazon Web Services using Amazon Rekognition. In contrast, the process of creating a mask detection model is carried out on a Personal Computer. The Raspberry Pi here is not used for the model-making process, but only as an executor of the model that has been built.

3.2. Analysis of CNN Training Results

The Convolutional Neural Network training process is carried out on a Personal Computer using a dataset consisting of photos of faces wearing masks and photos of faces not wearing masks. The dataset is taken from https://github.com/prajnasb/observations/tree/master/

In the first step, we visualized 1376 total number of images in the dataset in both categories. There are 690 images in the ”wearing a mask” category and 686 images in the ”no mask” category.

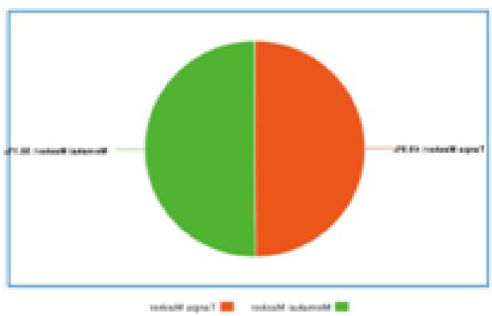


Figure 15. Data Visualization

In the next step, data sharing is carried out into training sets that will be tested using the CNN model. This process uses 80% of the total images for training and the remaining 20% for testing. So there are 552 images for training and 138 images for tests in the "wearing mask" category, and 548 images for training and 138 images for tests in the "no mask" category.

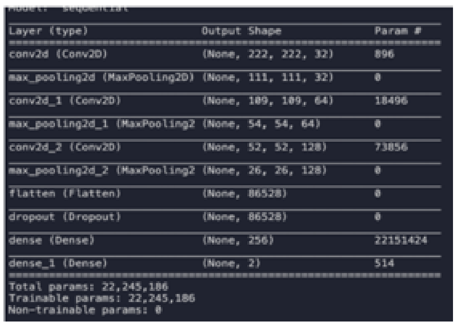
Figure 16. Data Sharing

The CNN model [19] has several layers, including convolution, pooling, and drop out. The model used is a Sequential model, which is a simple model that only needs to add layers to the existing model. Then use several layers, including convolution, pooling, dropout, flatten, and dense layers. In the model shown in Figure 8. For convolution, use the Conv2D sublibrary to start the convolution process specific to 2-dimensional data. The first uses a number of filters 32, kernel size 3x3 and Activation ReLu with input shape according to the resolution used in the dataset, which is 224x224x3. The ReLu parameter contained in Conv2D only creates a delimiter on the number zero, meaning that if x 0, then x = 0 and if x 0, then $x = x$. Then for the second convolution using filter 64, with a pool size of 2x2. The third convolution uses a filter of 128, with a pool size of 2x2. The feature extraction layer is still in the form of a multidimensional array, so it must "flatten" or reshape the feature map into a vector to be used as input from a fully-connected layer. Then a Fully connected (dense) layer is applied to classify the images based on the features extracted in the previous layer and output the probability from each class label.

```
model = tf.keras.models.Sequential([
    Conv2D(32, 3, activation='relu', input_shape=(IMG_WIDTH, IMG_HEIGHT, 3)),
    MaxPooling2D(2,2),

    Conv2D(64, 3, activation='relu'),
    MaxPooling2D(2,2),

    Conv2D(128, 3, padding='same', activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dropout(0.3),
    Dense(256, activation='relu'),
    Dense(2, activation='softmax')
])
```

Figure 17. CNN Layer

The dense layer shows the resulting output, which is two categories. Then a model is obtained, as shown in Figure 18. From the model output in Figure 9, there are several layers with different shape output results, the shape output calculation has differences in each type of layer along with some calculations. This convolution uses a 224x224x3 shape input, then a 32 filter and a 3x3 kernel. With the visualization of the model is visible in figure 18.



Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dropout (Dropout)	(None, 86528)	0
dense (Dense)	(None, 256)	22151424
dense_1 (Dense)	(None, 2)	514

Total params: 22,245,186
Trainable params: 22,245,186
Non-trainable params: 0

Figure 18. CNN model

The graph in Figure 19 shows how accurate the model has been formed. In Figure 19 (left) the x-axis shows the epoch while the y-axis shows accuracy. In Figure 19 (right), the x-axis represents the epoch, while the y-axis represents the loss or error rate of the model. The graph is obtained from the results of plot accuracy in the Convolutional Neural Network model obtained using the matplotlib library, which is commonly used to create plots in the form of graphs or dots. Table 1 shows the results of training using the CNN model. The accuracy value decreases, and the model loss rate increases after passing the 8th epoch because overfitting begins to occur.

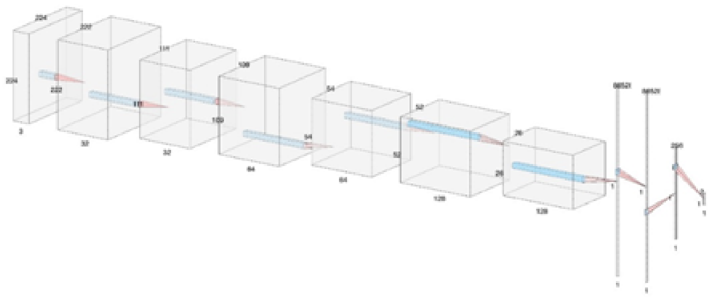


Figure 19. CNN Model Visualization

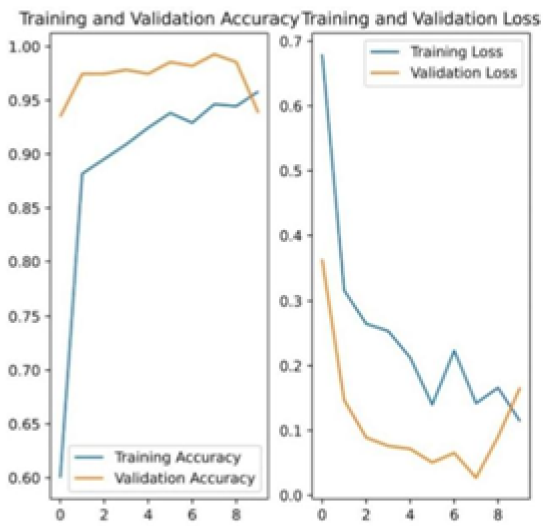


Figure 20. Convolutional Neural Network model training results graph

Table 1. Training results using the CNN model

Epoch to	Learning Time (seconds)	Accuracy %
1	69	93%
2	66	97%
3	71	97%
4	314	97%
5	68	97%
6	61	98%
7	63	98%
8	67	99%
9	167	98%
10	1018	93%

3.3. Modelling Process

The overall system testing process is carried out by implementing a model that has been created, trained, and tested on a series of pre-assembled tools in the form of an automatic door model based on facial recognition for detection of the use of face masks using raspberry pi. After the model is implemented on the tool, the following process is to test whether the tool has worked properly or not. Testing this tool is carried out by making a simulation by installing the tool on an office door. Of course, before starting this simulation, we have already registered a photo of the employee's face that the system will recognize to Amazon S3 so that the employee's face can be recognized.

Before starting the employee photo simulation, the first step is to be registered into the system. Furthermore, the detection process of wearing a mask is seen in figure 12. First of all, the employee shows the employee's face to the Webcam that is connected to this automatic door lock system, if the tool is working properly, the tool will ask the detected face to open the mask, which will later be match the face with the one in the database stored in Amazon S3. After the employee opens the mask, the system will try to recognize whether the face is contained in the database that has previously registered data. If it has been registered, employees can use masks again and will automatically be detected using masks.

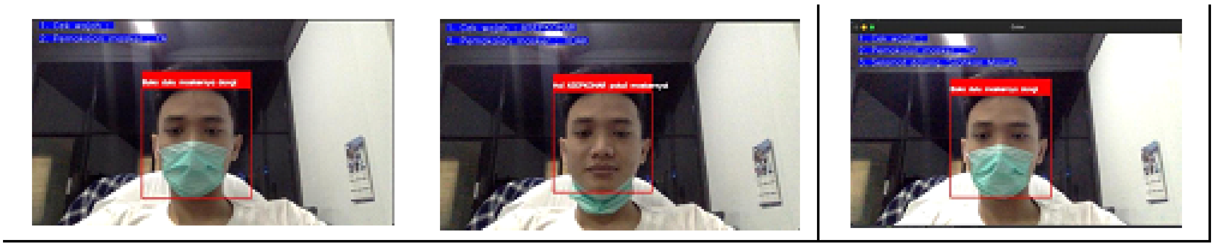


Figure 21. Mask Detection Process

In the last step, if the tool recognizes the employee's face captured by the webcam, then the tool will ask the employee to wear a mask to unlock the magnetic door connected to the system seen in figure 22 and the whole process is completed.

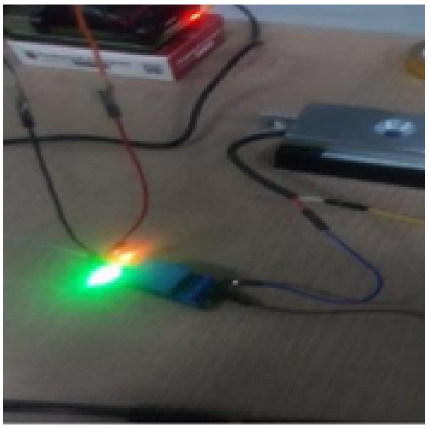


Figure 22. Magnetic door lock open

Table 2 Shows the results of comparing accuracy values between the proposed model and different datasets. It is proved that the proposed model obtained better accuracy values. This proves that using the CNN Sequential model with Dense and Amazon Face Recognition can improve the model's performance in the training and validation process [13–15].

Table 2. The Comparison of Model Performance with Previous Research

Reference	Metode	Database	Akurasi
Hassouneh et al [14]	CNN and LSTM	EEG	87.25%
B. Triwijoyo et al [15]	CNN with Batch Normalization	FER20013	98%
Aslan MF et al [16]	CNN WITHAlexNet, ResNet18, ResNet50, Inceptionv3, Densenet201, Inceptionres-netv2, MobileNetv2, GoogleNet	CXR images collected	96,29%
Our Proposed Method	CNN with Sequential, Dense and Amazon Face Recognition	https://github.com/prajnasb/ observa-tions/tree/master/experiements/data	99%

4. CONCLUSION

This research focuses on creating a facial recognition-based automatic door lock model to detect masks used on the face using raspberry pi, Magnetic Doorlock, Relay and USB Webcam. Input using a USB Webcam that will capture the image of the face. It is then sent to Amazon S3 by Raspberry Pi 4 to be matched with a photo of the face in the database uploaded to Amazon face recognition. If the face is recognized, then the Raspberry Pi 4 will trigger the Webcam to capture the face to detect the use of the mask on the face using 0. When the mask on the face of the recognized employee has been detected, the Raspberry Pi will trigger the Relay to cut off the electricity flowing to the Magnetic Doorlock, which will cause the magnetic field on the magnetic lock to disappear, so that the door can open. The experimental results show the training accuracy level reaches 99%, and the average length of time it takes to perform the facial recognition process, the time it takes to perform the facial recognition process using Amazon Face Rekognition is 0.47 seconds. In comparison, the length of time it takes to carry out the detection process of using a face mask is 0.07 seconds.

5. ACKNOWLEDGEMENTS

We would like to express our gratitude to the Pakuan Siliwangi Foundation for funding research through LPPM Pakuan University as an Internal Grant.

6. DECLARATIONS

AUTHOR CONTIBUTION

The first author contribution was to design the overall research outline and build the CNN model. The second author collects datasets, cloud system AWS and performs system integration and the third Author performs part testing and creates mechanical and electrical systems.

FUNDING STATEMENT

This research was funded by the Pakuan Siliwangi Foundation through an Internal Research Grant from Pakuan University

COMPETING INTEREST

The author states that there is no conflig and interest of editors and reviewers regarding the publication of this article.

REFERENCES

[1] H. W. N. Agusti and B. A. Gisela, "Pengenalan Wajah dengan Menggunakan Smartphone : Sistematik Review," *Journal of Indonesian Forensic and Legal Medicine*, vol. 2, no. 2, pp. 156–163, 2020.

[2] A. H. Bachtiar, P. P. Surya, and R. P. Astutik, "Rancang Bangun Dual Keamanan Sistem Pintu Rumah Menggunakan Pengenalan Wajah dan Sidik Jari Berbasis Iot (Internet of Things)," *Jurnal POLEKTRO: Jurnal Power Elektronik*, vol. 1, no. 1, pp. 102–107, 2022.

[3] A. Febriansyah, J. Saputra, and P. Desvirati, "Alat Pendeteksi Suhu Tubuh dan Wajah (Kebutuhan Bukti Kehadiran) Berbasis Data," *Manutech : Jurnal Teknologi Manufaktur*, vol. 14, no. 01, pp. 1–6, 2022.

[4] M. Abdillah, S. Rasyad, and N. Alfarizal, "Implementasi Sistem Pendeteksi Penggunaan Masker Berbasis Raspberry Pi 4 Menggunakan Metode Convolution Neural Network (CNN) pada Proses Screening Protokol Kesehatan COVID-19," *Ijccs*, vol. 6, no. 1, pp. 1–5, 2022.

[5] M. Abdul, R. Irham, and D. A. Prasetya, "Prototipe Pendeteksi Masker Pada Ruangan Wajib Masker untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19," *Prototipe Pendeteksi Masker Pada Ruangan Wajib Masker Untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19*, pp. 47–55, 2020.

[6] M. N. Baay, A. N. Irfansyah, and M. Attamimi, "Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning," *Jurnal Teknik ITS*, vol. 10, no. 1, pp. 64–70, aug 2021. [Online]. Available: http://ejurnal.its.ac.id/index.php/teknik/article/view/59790

[7] B. K. Triwijoyo, "Model Fast Tansfer Learning pada Jaringan Syaraf Tiruan Konvolusional untuk Klasifikasi Gender Berdasarkan Citra Wajah," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 211–221, 2019.

[8] R. R. Ramdhani, R. I. Adam, and A. A. Ridha, "Implementasi Deep Learning Untuk Deteksi Masker Deep Learning Implementation for Face Mask Detection," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 4, no. 2, p. 2021, 2021.

[9] N. S. Salahuddin, N. Iramadhan, S. P. Sari, and T. Saptariani, "Prototipe Sistem Keamanan Pintu Inkubator Bayi melalui Pengenalan Wajah menggunakan Kamera Web dan OpenCV berbasis Raspberry Pi," *Techno.Com*, vol. 21, no. 3, pp. 579–595, 2022.

[10] I. D. Wijaya, U. Nurhasan, and M. A. Barata, "Implementasi Raspberry Pi untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server dengan Pengenalan Wajah Menggunakan Metode Triangle Face," *Jurnal Informatika Polinema*, vol. 4, no. 1, pp. 9–17, nov 2017. [Online]. Available: http://jip.polinema.ac.id/ojs3/index.php/jip/article/view/138

[11] P. Elechi, E. Okowa, and U. Ekwueme, "Facial Recognition Based Smart Door Lock System," *FUPRE Journal of Scientific and Industrial Research*, vol. 6, no. 2, pp. 95–105, 2022.

[12] A. D, "Face Recognition using Machine Learning Algorithms," *Journal of Mechanics of Continua and Mathematical Sciences*, vol. 14, no. 3, jun 2019. [Online]. Available: http://www.journalimcms.org/journal/face-recognition-using-machine-learning-algorithms/

[13] Gaurav Dhiman, Srihari. K, Ramesh. R, and Udayakumar. E, "An Innovative Approach for Face Recognition Using Raspberry Pi," *Artificial Intelligence Evolution*, vol. 10, no. 1, pp. 102–107, aug 2020. [Online]. Available: http://127.0.0.1:8320/index.php/AIE/article/view/62

[14] S. E. Oltean, "Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation," *Procedia Manufacturing*, vol. 32, pp. 572–577, 2019.

[15] M. C. Tosima Manullang, A. Luky Ramdani, and N. Migotuwio, "Design and Architecture of a Public Satisfaction Detection Camera Based on Facial Emotional Analysis," *IOP Conference Series: Earth and Environmental Science*, vol. 537, no. 1, pp. 1–9, jul 2020. [Online]. Available: https://iopscience.iop.org/article/10.1088/1755-1315/537/1/012021

[16] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, "Fawkes: Protecting privacy against unauthorized deep learning models," *Proceedings of the 29th USENIX Security Symposium*, pp. 1589–1604, 2020.

[17] A. Hassouneh, A. Mutawa, and M. Murugappan, "Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG based on machine learning and deep neural network methods," *Informatics in Medicine Unlocked*, vol. 20, no. 1, pp. 2–9, 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S235291482030201X

[18] B. K. Triwijoyo, A. Adil, and A. Anggrawan, "Convolutional Neural Network With Batch Normalization for Classification of Emotional Expressions Based on Facial Images," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 21, no. 1, pp. 197–204, 2021.

[19] M. F. Aslan, K. Sabanci, A. Durdu, and M. F. Unlersen, “COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization,” *Computers in Biology and Medicine*, 2020.