SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

Implementation of Face Recognition for Lecturer Attendance Using Deep Learning CNN Algorithm

Fajhar Muhammad Ramdhan[1], Agung Triayudi[2*], Eri Mardiani[3]

Informatics Study Program, Faculty of Communication and Information Technology, Universitas Nasional, Jakarta, Indonesia

Author Email: Fajarmuhammad1087@gmail.com[1], agungtriayudi@civitas.unas.ac.id[2], erimardiani@civitas.unas.ac.id[3]

Abstract. Using the Convolutional Neural Network (CNN) algorithm, this research aims to create a better lecturer attendance application that improves the attendance system and creates peace of mind when lecturers arrive at national universities. The author analyses the results of applying deep learning algorithms to an experimental face recognition system that uses convolutional neural networks. The purpose of this study is to show that deep learning algorithms can improve the accuracy and efficiency of recording presence. In addition, the goal of this research is to create a timekeeping application using face recognition technology that is expected to have a high level of accuracy. In addition, this research includes a modification of the CNN model. This modification resulted in an epoch value of 75 for training of 100% and test of 95%. Analysis of results, drawing conclusions, and suggestions for additional development are the final stages of this research. Evaluation of the integrated system is done by collecting actual attendance data and comparing it with the attendance records created by the system. This validation will help explain the performance of the system and find problems or vulnerabilities that may need to be fixed.

Keywords: Attendance, CNN Algorithm, Deep Learning, Face Recognition

1 Introduction

With the rapid development of technology, there are many advantages, especially in the field of information technology. [1]. There are many benefits of the human face recognition research field [2]. One of the biometric identification methods is face recognition. This method is usually divided into two types, the first is appearance-based methods and the second is feature-based methods [3]. Face recognition algorithms are generally better than before, but thanks to the help of machine teaching algorithms, large amounts of available data, and new technologies as input, face recognition technology has evolved. [4].

Every educational institution must maintain valid attendance records as a measure of their academic performance. This advancement in technology can be applied to university attendance systems, which will collect attendance data to determine the percentage of attendance to teaching and learning activities. [5].

A lot of data on human biological traits are used for various purposes thanks to the advancement of science and technology today. The unique biological traits of each person can provide information about a person's identity, such as information on the human body such as fingerprints, retina, voice patterns, and facial patterns (facial recognition). [6]. Face recognition system is one of the most developed and rapidly growing systems. It is an artificial intelligence system capable of recognizing or identifying human faces from digital images, both images and videos. This is done by identifying, recognizing, and comparing images of previously unknown faces with a database of faces stored in the database. [7].

The facial recognition method records attendance by detecting a person's face. It records a photo of each employee's face and automatically matches it with the data in the database indicating their participation. The selection of the facial recognition method as the attendance method is based on the fact that the face image is the most important data content and is difficult to change. [8]. Therefore, this tool can be used to determine attendance or absenteeism. Of course, this is very helpful for the HR division in assessing performance by attendance. [9].

The authors examine the results of using deep learning algorithms on an experimental face recognition system that uses convolutional neural networks. Their expectation is that the recording of presence will be more accurate and efficient. In addition, this research is expected to help in the development of timekeeping

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

applications that use facial recognition technology that is expected to have high accuracy and assess the results of deep learning in attendance applications. [10].

2 Methodology

In this study the data used is sourced from the National University campus. after the data is obtained, then the data testing of photo samples will be implemented with the Convolutional Neural Network algorithm. After the implementation is continued with the Confusion Matrix testing method. Then the algorithm is carried out by getting accuracy, precision, recall and f1-score to find out the results of the algorithm.

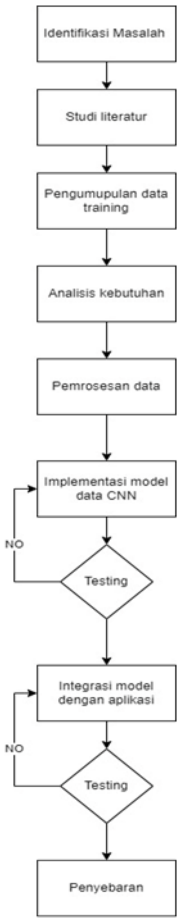

Figure 1. Process Design Flow

2.1 Data Collection Methods

Data collection in this research is based on a collection of images/photos that researchers take as data sets and training data. This data collection uses a device.

Table 1. Data Collection

No	Class	Sum of Data
1	Face 1	20
2	Face 2	20
3	Face 3	20
4	Face 4	20
5	Face 5	20
6	Face 6	20
7	Face 7	20

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

8	Face 8	20
9	Face 9	20
10	Face 10	20
11	Face 11	20
12	Face 12	20
13	Face 13	20
14	Face 14	20
15	Face 15	20
16	Face 16	20
17	Face 17	20
18	Face 18	20
19	Face 19	20
20	Face 20	20

2.2 Preprocessing

In this study, a convolutional neural network (CNN) algorithm is used by transforming the CNN model by using an input of the form 128 x 128. The CNN model is divided into two stages: future learning and classification. The future learning stage consists of three convolution layers, three 3 x 3 filter sizes, three pooling layers, and three 2 x 2 filter sizes. The classification stage consists of flatten, fullconnected, and sigmoid activation. A total of 700 training data and 300 testing data. In addition, in this study, the CNN model used an input image of 80 x 80 pixels with 3 x 3 filters, 20 epochs, and a learning rate of 0.001. This study also simulated CNN algorithms that work in three layers or layers, namely convolutional, pooling, and fullconnected. These steps are part of the development of a CNN model that uses the Deep Learning CNN algorithm to identify lecturer faces in the attendance application. The data set used is divided into training data and testing data with a division of 80% training data and 20% testing data.

2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of Deep Learning algorithm model that is capable of performing classification and recognition tasks through images.

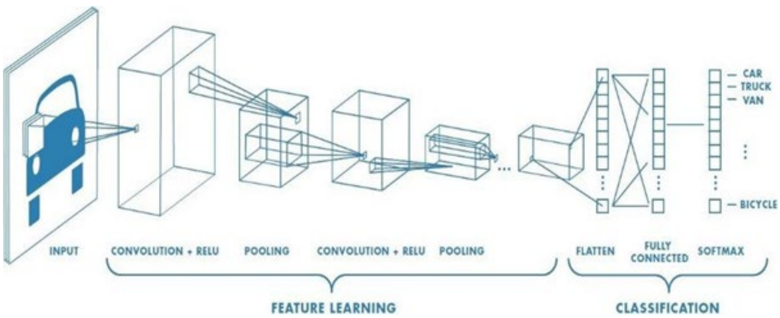

Figure 2. Convolutional Neural Network

Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm that requires 4 main types of layers to build its architecture namely: Convolutional Layer; Activation Layer (ReLU); Pooling Layer; Dense Layer.

CNNs are used for face recognition on social media such as Facebook. Other related studies show that compared to legacy machines, CNNs are highly successful in face recognition, facial expression recognition, facial number recognition, and facial object segmentation learning algorithms. The accuracy of face recognition also reached an average of above 90%. Below is an example of a CNN network designed to develop facial expression recognition applications.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

3 Results and Discussion

The CNN algorithm, works in 3 layers. The first layer is convolutional layer, the second layer is pooling layer, and the third layer is fully connected. In the first layer, the input image is defined as a matrix. The following below is the convolutional layer calculation.

112	69	51	59
79	48	47	63
56	43	46	60
47	42	46	59

1	0
1	0

$$\begin{bmatrix}112 & 69\\ 79 & 48\end{bmatrix} * \begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix} = \begin{matrix}191\end{matrix}$$

112	69	51	59
79	48	47	63
56	43	46	60
47	42	46	59

1	0
1	0

$$\begin{bmatrix}69 & 51\\ 48 & 47\end{bmatrix} * \begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix} = \begin{matrix}191 & 117\end{matrix}$$

112	69	51	59
79	48	47	63
56	43	46	60
47	42	46	59

1	0
1	0

$$\begin{bmatrix}51 & 59\\ 47 & 63\end{bmatrix} * \begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix} = \begin{matrix}191 & 117 & 98\end{matrix}$$

112	69	51	59
79	48	47	63
56	43	46	60
47	42	46	59

1	0
1	0

$$\begin{bmatrix}79 & 48\\ 56 & 43\end{bmatrix} * \begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix} = \begin{matrix}191 & 117 & 98\\ 135 & & \end{matrix}$$

In this calculation, the matrix is 4 x 4 with a 2x2 kernel. At this stage, each row and column of the matrix marked with color is multiplied by the kernel until the final row and column stage. And have the final result as follows.

112	69	51	59
79	48	47	63
56	43	46	60
47	42	46	59

1	0
1	0

191	117	98
135	91	93
103	95	92

$$\begin{bmatrix}46 & 60\\ 46 & 59\end{bmatrix} * \begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix} = \begin{matrix}191 & 117 & 98\\ 135 & 91 & 93\\ 103 & 95 & 92\end{matrix}$$

After performing *convolutional* calculation, the result is determined by *max pooling*.

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

191	117	98
135	91	93
103	95	92

$$\begin{bmatrix} 191 & 117 \\ 135 & 91 \end{bmatrix} = \begin{bmatrix} 191 & \\ & \end{bmatrix}$$

191	117	98
135	91	93
103	95	92

$$\begin{bmatrix} 117 & 98 \\ 91 & 93 \end{bmatrix} = \begin{bmatrix} 191 & 117 \\ & \end{bmatrix}$$

191	117	98
135	91	93
103	95	92

$$\begin{bmatrix} 135 & 91 \\ 103 & 95 \end{bmatrix} = \begin{bmatrix} 191 & 117 \\ 135 & \end{bmatrix}$$

191	117	98
135	91	93
103	95	92

$$\begin{bmatrix} 91 & 93 \\ 95 & 92 \end{bmatrix} = \begin{bmatrix} 191 & 117 \\ 135 & 95 \end{bmatrix}$$

Max pooling is determined from the convolutional results by comparing the highest number. This process is done until the last row and column, and get the max polling result as follows.

$$\begin{bmatrix} 191 & 117 \\ 135 & 95 \end{bmatrix}$$

3.1 Results from Training Data

From the explanation above, the CNN algorithm is applied to the phyton programming language with the following google collabs results:

Layer (type)	Output Shape	Param
rescaling_1 (Rescaling))	(None, 180, 180, 3	0
conv2d (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d(MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_1 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_1(MaxPooling2D)	(None, 45, 45, 32	0
conv2d_2 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 22, 22, 64)	0
flatten (Flatten)	(None, 30976)	0
dense (Dense)	(None, 128)	3965056
dense_1 (Dense)	(None, 15)	1935

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

In this result, layer rescaling_1 has an outshape (180,180,3) with parameter 0. Then the activation size is 180*180*3 = 97200. Conv2d, this layer has a parameter value of 448. Conv2d_1, has a parameter of 4640. Conv2d_2 has a parameter of 18496.

Parameters are generally the weights learned during training. Technically, it is the content of the matrix that contributes to the predictive power of the model, which is changed during the backward propagation process. From the calculations using google collab and phyton, it can be concluded that the results are as follows.

Total params: 3990575 (15.22 MB)
Trainable params: 3990575 (15.22 MB)
Non-trainable params: 0 (0.00 Byte)

In this section the datasets are trained until the end, ranging from 1 to 20 datasets. Epoch refers to 1 model training cycle where all datasets are trained once, in 1 epoch the model parameters are updated.

Epoch 1/10
4/4 [==========================] - 25s 8s/step - loss: 2.8360 - accuracy: 0.0833 - val_loss: 2.4405 - val_accuracy: 0.2333
Epoch 2/10
4/4 [==========================] - 3s 927ms/step - loss: 2.1876 - accuracy: 0.6083 - val_loss: 1.8592 - val_accuracy: 0.6667
Epoch 3/10
4/4 [==========================] - 4s 977ms/step - loss: 1.2299 - accuracy: 0.8750 - val_loss: 0.6157 - val_accuracy: 1.0000
Epoch 4/10
4/4 [==========================] - 3s 850ms/step - loss: 0.3190 - accuracy: 0.9583 - val_loss: 0.0813 - val_accuracy: 1.0000
Epoch 5/10
4/4 [==========================] - 5s 1s/step - loss: 0.0525 - accuracy: 1.0000 - val_loss: 0.0176 - val_accuracy: 1.0000
Epoch 6/10
4/4 [==========================] - 3s 823ms/step - loss: 0.0101 - accuracy: 1.0000 - val_loss: 0.0253 - val_accuracy: 1.0000
Epoch 7/10
4/4 [==========================] - 3s 910ms/step - loss: 0.0039 - accuracy: 1.0000 - val_loss: 0.0201 - val_accuracy: 1.0000
Epoch 8/10
4/4 [==========================] - 5s 1s/step - loss: 0.0025 - accuracy: 1.0000 - val_loss: 0.0034 - val_accuracy: 1.0000
Epoch 9/10
4/4 [==========================] - 3s 915ms/step - loss: 9.1407e-05 - accuracy: 1.0000 - val_loss: 0.0031 - val_accuracy: 1.0000
Epoch 10/10
4/4 [==========================] - 3s 933ms/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 2.5975e-04 - val_accuracy: 1.0000

3.2 Grouping Data Graph

The following is a view of the coding script to generate a graph image of the grouping data.

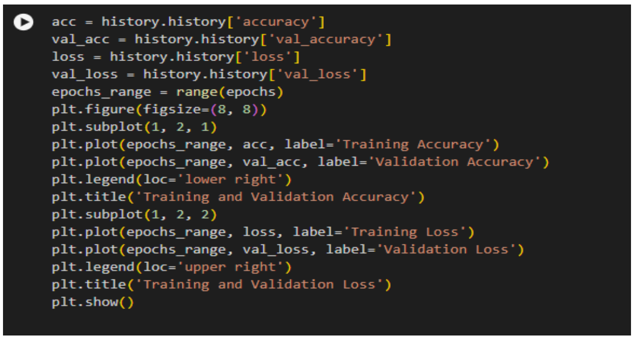

Figure 3. Script Code Graph Clustering Data

SaNa: Journal of Blockchain, NFTs and Metaverse Technology
Vol 2, Issue 1, February 2024, Pages 72-79
ISSN: 3030-9832 (Media Online)
DOI: https://doi.org/10.58905/sana.v2i1.270

The following is a graphical display of accuracy validation training and loss validation training where the grouping data graph describes a data or result that has been trained.

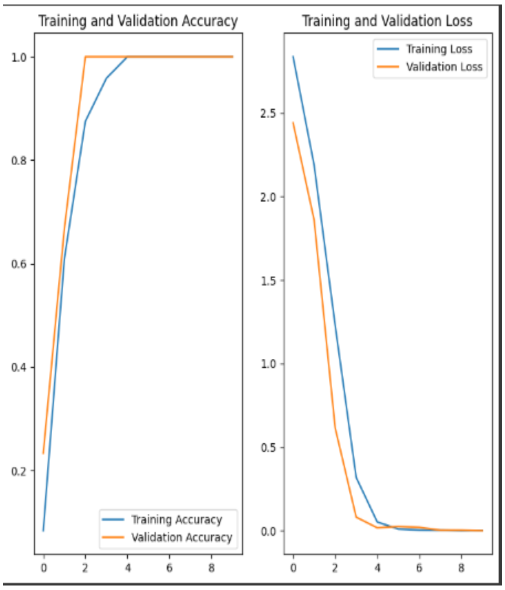

Figure 4. Graph of Clustering Results

3.3 Individual Data Graph

Here is a look at the script coding to generate individual data graphic images

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs_range = range(epochs)
plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

Figure 5. Script Code Graph Individual Data

The following is a graphical display of accuracy validation training and loss validation training where the individual data graph illustrates a data or result that has been trained.

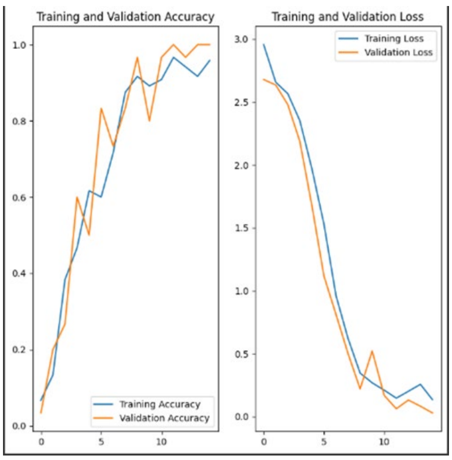

Figure 6. Individual Result Chart

4 Conclusion

Based on the research that has been done by the author, it can be concluded that face recognition can detect faces for the needs of lecturer attendance. Face Recognition does not succeed in detecting faces if there are no images or photos in the image testing data. Face recognition using the convolutional Neural Network algorithm, based on the results of experiments that have been carried out, that the application of the convolutional neural network algorithm can be used to detect and recognize faces for lecturer attendance needs, with sufficient lighting and faces that match the testing data created.

References

1. W. G. Alfeus Adi Saputra, "Cover page of face recognition implementation on android-based attendance system using convolutional neural network (cnn) method and lock gps," 2022.
2. W. M. Saputra, H. A. Wibawa, and P. Wajah, "Face recognition using algorithms," vol. 2, no. 1, pp. 102-110, 2013.
3. Supriana and Y. Pratama, "Face recognition new approach based on gradation contour of face color," Int. J. Electr. Eng. Informatics, vol. 9, no. 1, pp. 125-138, 2017, doi: 10.15676/ijeei.2017.9.1.8.
4. S. Suwarno and K. Kevin, "Analysis of Face Recognition Algorithm: Dlib and OpenCV," J. Informatics Telecommun. Eng., vol. 4, no. 1, pp. 173-184, 2020, doi: 10.31289/jite.v4i1.3865.
5. M. Azamy, A. B. Ariwibowo, and I. Mardianto, "Face Recognition Implementation with MTCNN on Attendance System Prototype at Trisakti University," vol. 1, no. 1, pp. 73-88, 2023.
6. [Anugrah Fahrul Ramadhan, "FACE RECOGNITION APPLICATION FOR STUDENT ABSENCE BY USING THE 128D EMBEDDING METHOD OF STUDENTS BY USING THE 128D EMBEDDING METHOD," 2022.
7. [J. Jtik, J. Teknologi, B. T. Utomo, I. Fitri, and E. Mardiani, "Application of Face Recognition in Online Academic Applications," vol. 5, no. 4, pp. 1-5, 2021.
8. [R. Jannah and A. I. Nurhidayat, "WEB PRESENCE IMPLEMENTATION," pp. 1-12, 2023.
9. [H. Jusuf, A. Azimah, and Winarsih, "Development of Lecturer Attendance System Application Using Fingerprint (Digital Fingerprint) at National University," Rekayasa Teknol. J., vol. 5, no. 1, pp. 3-8, 2013.
10. Y. Hartiwi, E. Rasywir, Y. Pratama, and P. A. Jusia, "Face Recognition Experiment with Indoor Positioning System features using CNN Algorithm," vol. 22, no. 2, 2020.