Vol.7 No.1 Tahun 2026
Jurnal Kecerdasan Buatan, Komputasi dan Teknologi Informasi
https://ejournal.unuja.ac.id/index.php/core
E-ISSN : 2774-7875 and P-ISSN : 2775-0124

Human Face Recognition for Attendance System Using Deep Learning Implementation with Convolutional Neural Network Method

Frida Mimi Wahyuni [1], Rizal Adi Saputra [2], Muhammad Ihsan Sarita [3]
[1,2,3] Universitas Halu Oleo, Kendari, Indonesia

Article Info

Article history:

Received : 01 08, 2026
Revised : 03 15, 2026
Accepted : 05 27, 2026

Keywords:

Deep Learning;
CNN;
Face Recognition;
Absence;
Digital Image;

Abstrack

A Convolutional Neural Network-based deep learning approach is utilized to identify and classify various human facial images based on their characteristic differences. The facial recognition system utilizes a convolutional neural network (CNN) model implemented with TensorFlow, complemented by data augmentation techniques to prevent overfitting. The model is trained and validated using a specially prepared dataset, with performance evaluated through loss and accuracy graphs and confusion matrices. The system also supports real-time facial recognition using OpenCV, which identifies faces in videos and automatically records the subject's presence if the identification successfully passes a specified confidence threshold. This demonstrates the potential for integrating Deep Learning into automated facial recognition and attendance recording applications. Once the data training process is complete, the next stage is model testing to assess its performance. Data for testing is randomly sampled using the Python library. The results of this process will indicate the accuracy level of the method used. Confusion Matrix is used to calculate the accuracy value in Deep Learning. The testing process will be carried out through a classification process using the Convolutional Neural Network method. The accuracy of the model based on the Confusion Matrix is approximately 97%.

Corresponding Author:

Frida Mimi Wahyuni,
fridawahyuni09@gmail.com

INTRODUCTION

Nowadays, there are many methods for recording attendance, especially in educational settings. Some places still use manual systems, such as signing attendance registers in a book, which then summarizes the data to create an attendance report. This method has several drawbacks, such as wasting paper, being time-consuming to summarize, being difficult to connect with other digital systems, and being prone to fraud. Therefore, facial detection technology is starting to be used as a modern solution. Before a face can be recognized by the system, it must first be detected through a facial detection process (Miftakhurrokhmat et al., 2021). Security of a room or access to a specific area is a crucial issue in environments that require strict supervision, for example, in the student attendance process. Traditional attendance systems are less effective because they can be misused. Therefore, more sophisticated and accurate technology is needed, such as facial recognition, to ensure safe access to the room (Faiq et al., 1978). Thus, facial detection and recognition technology can be implemented as a modern method in student attendance systems to automate the identification process and improve the accuracy of attendance data (Wahyuni & Sulaeman, 2022).

Facial recognition is known as a fast and efficient verification method, offering greater speed and convenience compared to other biometric methods such as fingerprint or retina scanning (Alwendi & Masriadi, 2021).

According to research conducted by Nugroho et al. (2020), which implemented human expressions into CNN and Deep Learning, this study tested a facial expression recognition model using a large dataset containing 24,029 facial images with seven expressions: anger, happiness, sadness, surprise, neutrality, disgust, and fear. The dataset was further divided into two sets: training (6,106 images) and test (test) data, with representative data selected to maintain image diversity and quality. Testing was conducted with the following main parameters: epochs of 50 and 100, a learning rate of 0.001, and seven output layers, as well as batch size adjustments to improve efficiency and accuracy. The purpose of this testing was to evaluate the model's performance in accurately and efficiently recognizing facial expressions and to provide a basis for further model optimization (Nugroho et al., 2020).

A digital image is an image that has been converted into discrete values by a camera. This image consists of a collection of gray-level values called pixels. The process of converting analog images to digital images is necessary for digital image processing. This conversion produces two types of digital images: still images and moving images (Frenza & Mukhaiyar, 2021). An image can be defined as a visual representation of one or more objects, whether in the form of photographs, medical scans such as X-rays, or images captured by satellites. Generally, images are classified into three categories: binary images, grayscale images, and color images (Nabila, 2024).

Digital image processing is often performed using the Python programming language and PyCharm software. This combination allows developers to view image changes and histograms in real time, simplifying image analysis and processing (Simbolon et al., 2022).

Open Computer Vision (OpenCV) is an open-source library developed specifically for image processing and developing computer visual capabilities that mimic human vision. OpenCV also supports satellite data processing. OpenCV's advantages include its free and open-source capabilities, high flexibility, multi-platform support, compatibility with various programming languages, and a broad developer community (Zulkhaidi et al., 2020).

METHOD

Convolutional Neural Network (CNN)

A Convolutional Neural Network is a deep learning algorithm consisting of 3D neurons arranged in layers. These neurons have three dimensions: width, height, and depth (Chalista et al., 2024). Each layer of a Convolutional Neural Network (CNN) transforms a three-dimensional input volume into a three-dimensional output volume that reflects the activity of the neurons. For example, a color image with width and height as spatial dimensions and three depths represents three RGB color channels: Red, Green, and Blue. This process allows for more accurate and efficient recognition of patterns and features in images in visual data processing (Alwanda et al., 2020).

The CNN model used consists of three convolution layers with 3x3 filters, each followed by a ReLU activation function and a 2x2 Max Pooling method. After the feature extraction process is complete, the feature map is flattened using a Flatten layer, then connected with a Fully Connected Layer of 128 neurons before moving to the output layer using the Softmax activation function, which results in a classification of 20 face classes.

Training Parameters

The training parameters used consist of:

- Optimizer : Adam
- Learning Rate : 0,001
- Batch Size : 32

- Epoch : 20
- Loss Function : Categorical Crossentropy

Deep Learning

This method allows machines to learn to understand and classify objects, such as faces captured in images. One popular architecture in Deep Learning is the Convolutional Neural Network (CNN), which was designed to overcome the limitations of previous methods. With CNNs, the number of parameters that need to be learned can be reduced, and this model is also able to handle various changes in the input image, such as position shifts, rotations, and differences in scale (Dewi & Ismawan, 2021). An example of its application is seen in face recognition systems, where the program is trained to recognize specific faces by breaking down the image structure to find similarities between one image and another (Wijaya & Samodra, n.d.). Deep Learning can be used to develop applications such as facial recognition, language translation, predictive analytics, natural language processing, and pattern recognition (Baay et al., 2021).

Deep learning capabilities emerge from the process of analyzing data sets used to complete specific tasks. In this research, the task is identifying faces by searching for patterns within them. Currently, the development of deep learning technology is very rapid and has been implemented using various artificial intelligence platforms (Guntoro et al., n.d.).

RESULTS AND ANALYSIS

The face recognition process requires three main types of data: training data, test data, and evaluation data. The dataset used in this study consists of several facial images grouped into several classes. Before model training, all images undergo preprocessing and are divided into training data, validation data, and test data. Details of the characteristics of the dataset used are presented in Table 1.

Table 1 Parameters

No.	Parameters	Description
1.	Number of Classes	20 Students
2.	Number of Images	2000
3.	Train	1600
4.	Validation	200
5.	Test	200
6.	Resolusi	150x150 pixel
7.	Aumentasi	Rotation, Zoom, Flip

The dataset used consists of 20 classes, each representing a single student identity. The total dataset consists of 2,000 facial images, divided into 1,600 training images (80%), 200 validation images (10%), and 200 test images (10%). All images were resized to 150 × 150 pixels before processing. To increase data variation and reduce the risk of overfitting, augmentation techniques were applied using ImageDataGenerator, including rotation, zoom, horizontal flip, and other augmentations used in the study.

Table 2 Augmentation Parameters

No.	Augmentation Parameter	Value
1.	Rotation Range	20
2.	Zoom Range	0,2
3.	Width Shift Range	0,2
4.	Height Shift Range	0,2
5.	Horizontal Flip	True
6.	Fill Mode	Nearest

System Implementation

Dataset preparation is the initial step in implementing a facial recognition system. The dataset serves as training and testing data for the model. Data is collected using a camera to obtain facial images of several individuals. The following is an example of a dataset that has been collected.

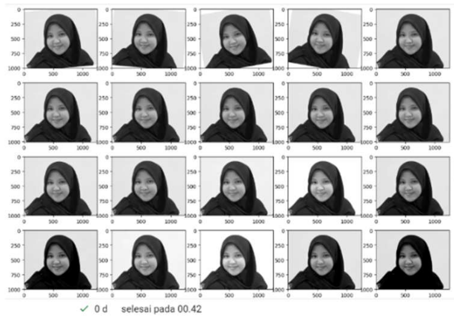

Figure 1 Augmented and Trained Dataset

Data Preprocessing

Next, preprocessing is performed to ensure the data can be used for model training. This stage includes normalization, augmentation, and splitting the dataset into training and testing data.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
import numpy as np

# Membaca dan mengubah Gambar menjadi array
def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder, filename))
        if img is not None:
            images.append(img)
    return np.array(images)

# Mengambil Gambar dari folder
images = load_images_from_folder(full_path)
labels = np.array([0] * len(images))  # Label 0 untuk nama person_name

# Membagi dataset menjadi data latih dan uji
x_train, x_test, y_train, y_test = train_test_split(images, labels,
test_size=0.2, random_state=42)

# Augmentasi data
datagen = ImageDataGenerator(rescale=1./255, rotation_range=20,
zoom_range=0.2, horizontal_flip=True)
datagen.fit(x_train)
```

Figure 2 Data Augmentation Process Code

Model Training

After the pre-processing stage is complete, the data is then used in the model training process. The model used is a Convolutional Neural Network (CNN).

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
Flatten, Dense

# Membangun model CNN
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(250,
250, 3)),
      Conv2D(32, (3, 3), activation='relu',
input_shape=(250, 250, 3)),
      MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')  # Karena hanya ada satu
individu (binary classification)
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# Melatih model
history = model.fit(datagen.flow(x_train, y_train,
batch_size=32), epochs=10, validation_data=(x_test,
y_test))
```

Figure 3 CNN Model Training Code

Model Evaluation

Evaluation data must be separated from training data to ensure unbiased testing results. Some metrics used include accuracy, precision, recall, and F1-score, which together describe the model's ability to recognize and differentiate between classes. Additionally, error analysis is performed to identify conditions that cause incorrect predictions, such as lighting or variations in facial expressions.

After evaluation using static data, the model is tested in real-time conditions to ensure its performance in a real-world environment. A thorough evaluation ensures the model is robust, accurate, and ready to be deployed without the risk of poor performance.

Loss (Training Loss) and Val_loss (Validation Loss)

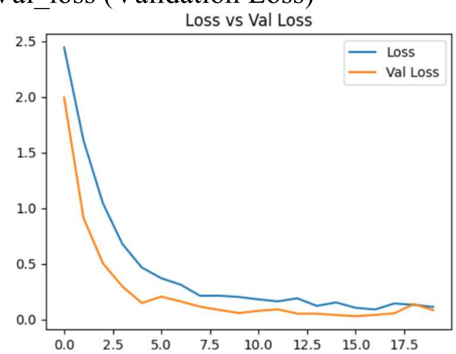

Figure 4 Loss and vall_Loss

Figure 4 This graph depicts the change in loss and val_loss values over 20 epochs in a machine learning model, most likely a neural network model. The loss graph depicts the change in the loss function value during the training process of the machine learning model. This graph helps monitor model performance and adjust hyperparameters. An ideal graph shows a gradual and steady decrease in loss values. If the training loss continues to decrease while the validation loss increases, it indicates overfitting. However, if both decrease in a similar pattern, it indicates the model is learning well. In this study, the graph shows a significant decrease in loss and val_loss early in training and stabilizes after about 50 epochs. The smooth curve and small difference between the two indicate a model with good performance, stability, and no significant overfitting.

Accuracy (Training Accuracy) and Val_accuracy (Validation Accuracy)

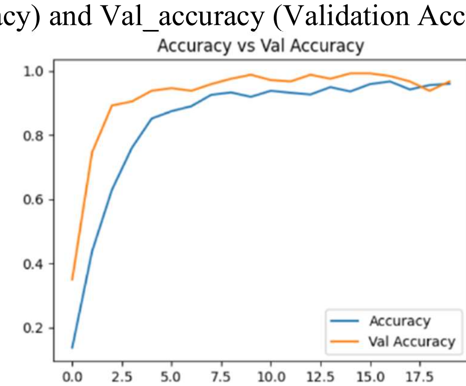

Figure 5 Accuracy and Val_Accuracy

Figure 5 The figure above shows two graphs depicting the change in accuracy and val_accuracy values over 20 epochs in a machine learning model. The blue line represents the model's accuracy on the training data during the training process.

Both graphs exhibit a similar pattern of improvement, with accuracy and val_accuracy values increasing significantly at the beginning of training, then stabilizing as the number of epochs increases. The consistent and stable curves indicate that the model is well-trained and shows no significant signs of overfitting.

In conclusion, this model demonstrates good performance in both training and validation, with a significant increase in accuracy and good stability after approximately 20 epochs. This indicates that the model is capable of predicting both training and validation data with high accuracy.

The purpose of using epochs is to ensure that the model can optimally access and learn from all provided training data samples, thus producing better results.

System Evaluation

System evaluation is conducted to determine whether the system is operating correctly according to its objectives. Through evaluation, it can be determined whether the system is truly meeting its intended objectives. System evaluation in human facial recognition is a crucial part of the development and implementation of facial recognition technology. The Confusion Matrix obtained from the above data is as follows :

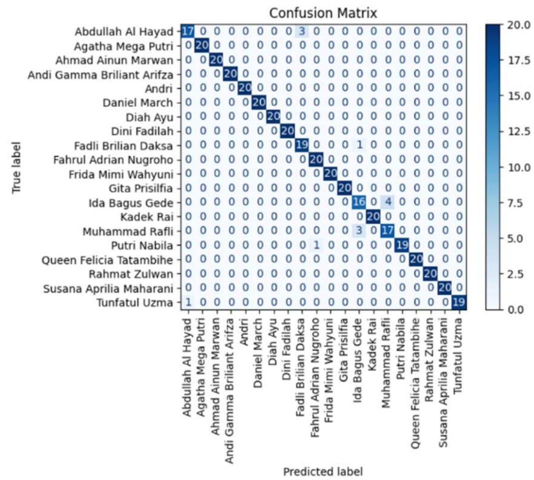

Figure 6 Confusion matrix

In Figure 6, to study the True Positive (TP), False Positive (FP), and False Negative (FN) values as part of the Confusion Matrix analysis, we need to look at the position of each element within the matrix and the context of the model's prediction—whether the result is correct or incorrect.

The following table is used to help identify the position and significance of each TP, FP, and FN value within the Confusion Matrix.

Table 3 Details TP, FP, dan FN

No.	Label	TP	FP	FN	TN
1.	Abdullah Alhayad Arafah	17	1	3	379
2.	Agatha Mega Putri	20	0	0	380
3.	Ahmad Ainun Marwan	20	0	0	380
4.	Andi Gamma Brilliant Arifza	20	0	0	380
5.	Andri	20	0	0	380
6.	Daniel March	20	0	0	380
7.	Diah Ayu	20	0	0	380
8.	Dini Fadilah	20	0	0	380
9.	Fadli Brilian Daksa	19	3	1	377
10.	Fahrul Adrian Nugroho	20	0	1	379
11.	Frida Mimi Wahyuni	20	0	0	380
12.	Gita Prisilfia	20	0	0	380
13.	Ida Bagus Gede	16	4	4	376
14.	Kadek Rai	20	0	0	380
15.	Muhammad Rafli	17	4	3	376
16.	Putri Nabila	19	0	1	380

17.	Queen Felicia Tatambihe	20	0	0	380
18.	Rahmat Zulwan	19	0	1	380
19.	Susana Aprilia Maharani	20	0	0	380
20.	Tunfatul Uzma	19	0	1	380

Information from the Confusion Matrix allows for evaluation of model performance for each class, including a comparison of correct (True Positive) and incorrect (False Positive and False Negative) predictions. This evaluation helps identify classes that are performing well and those that need improvement, for example through additional data or parameter refinement. Based on the Confusion Matrix results, the model achieved an accuracy of approximately 97%, indicating excellent performance and high classification ability, despite a small number of errors. The following are the calculations for the precision, recall, F1-score, and support values for each label.

Table 4 Precission, Recall, F1-Score and Support Value

No.	Label	*Precision*	*Recall*	*F1-Score*	*Support*
1.	Abdullah Alhayad Arafah	0.94	0.85	0.89	20
2.	Agatha Mega Putri	1	1	1	20
3.	Ahmad Ainun Marwan	1	1	1	20
4.	Andi Gamma Brilliant Arifza	1	1	1	20
5.	Andri	1	1	1	20
6.	Daniel March	1	1	1	20
7.	Diah Ayu	1	1	1	20
8.	Dini Fadilah	1	1	1	20
9.	Fadli Brilian Daksa	0.86	0.95	0.90	20
10.	Fahrul Adrian Nugroho	0.95	1	0.98	20
11.	Frida Mimi Wahyuni	1	1	1	20
12.	Gita Prisilfia	1	1	1	20
13.	Ida Bagus Gede	0.80	0.80	0.80	20
14.	Kadek Rai	1	1	1	20
15.	Muhammad Rafli	0.81	0.85	0.83	20
16.	Putri Nabila	1	0.95	0.97	20
17.	Queen Felicia Tatambihe	1	1	1	20
18.	Rahmat Zulwan	1	1	1	20
19.	Susana Aprilia Maharani	1	1	1	20
20.	Tunfatul Uzma	1	0.95	0.97	20

Based on the evaluation results, the model achieved an accuracy of 97%. The average precision value reached 0.97, indicating that most of the model's predictions were correct. The average recall also hovered around 0.97, indicating that almost all faces were successfully recognized. The F1-score of 0.97 indicates a good balance between precision and recall. These results indicate that the CNN model is capable of classifying faces with a high degree of reliability.

The calculation process for Precision, Recall, and F1-score is as follows. I used the name Abdullah Alhayad Arafah as an example :

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% = \frac{17}{17+1} \times 100\% = \frac{17}{18} \times 100\% = 94\% \quad (2.4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100 = \frac{17}{17+3} \times 100\% = \frac{17}{20} \times 100\% = 85\% \quad (2.5)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.94 \times 0.85}{0.94 + 0.85} = 2 \times \frac{7990}{179} = 2 \times 44.63 = 98\% \quad (2.6)$$

Table 4 shows that the model performed very well across all classes, with an overall accuracy of 97%. The model was very effective for prediction in class 4 (F1-Score 0.98) and was quite consistent across the other classes, with precision, recall, and F1-Score values above 0.90 for most classes.

1. System Testing

After the design and coding stages are complete, the next step is program testing. In this study, testing was conducted by running a Python script that activated the camera to capture faces in real time. When a face is detected, a frame appears labeled with the face's name. This process utilizes a Convolutional Neural Network (CNN) and the OpenCV library to access the camera, such as through the cv2.VideoCapture(2) command, which allows for continuous, live image capture. Each captured frame can then be further processed. The facial recognition data is stored in a .csv file, containing information such as name and arrival time, as a test result of the attendance system.

Figure 7 System Test Result

Figure 7 shows the Face Recognition process successfully recognizing a person's face based on the data stored in the dataset. A green box appears to mark the detected face, along with a name taken from the previously stored data..

Figure 8 System Test Data Not Found in Training Data

Upon detection, the system attempts to match facial features against a database of previously registered faces. If the system finds a match in the database, it displays the name or identity associated with that face. Otherwise, as in this case, the system marks the face as "Unknown," as shown in Figure 8, meaning no match was found or the face is not yet registered in the system.

The implementation of a facial recognition system requires attention to the security and privacy aspects of biometric data. All facial images should be stored securely in encrypted storage and should only be accessible to authorized administrators. Furthermore, the use of facial data must obtain user consent and comply with applicable personal data protection regulations. This step is crucial to prevent identity theft and biometric data leaks.

3. Comparison with Other Methods

Compared to the Haar Cascade method, which only functions as a face detector, CNN is capable of automatic feature extraction, resulting in higher accuracy. Compared to the Eigenface and Fisherface methods, CNN is also more robust to changes in lighting, facial expression, and orientation, although it requires a longer training time.

CONCLUSIONS

Metode Convolutional Neural Network terbukti efektif dan efisien dalam proses pendeteksian wajah secara real-time yang dimanfaatkan pada sistem absensi. Sistem ini juga dapat berfungsi sebagai media alternatif untuk validasi kehadiran sehingga dapat meminimalkan terjadinya manipulasi data absensi oleh mahasiswa. Berdasarkan hasil Confusion Matrix, model memperoleh tingkat akurasi sebesar kurang lebih 97%, yang berarti bahwa model mampu mengklasifikasikan dengan benar sekitar 97% dari keseluruhan sampel uji. Nilai akurasi tersebut menunjukkan bahwa kinerja model sangat baik dalam melakukan klasifikasi, meskipun masih terdapat sejumlah kecil kesalahan dalam proses identifikasi. Further research is recommended using a dataset with a larger number of subjects and varying lighting conditions, facial expressions, mask usage, and shooting angles to improve the model's generalizability. Some things to consider include ensuring adequate lighting when registering faces or taking attendance via webcam to ensure clearer images. Furthermore, the dataset size can be expanded and supplemented with a variety of images to help the CNN model learn better and improve accuracy. To prevent data loss, the system should have a regular backup mechanism. Finally, adding a user class to the attendance system can help clarify the identity of each registered user.

REFERENCES

Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, *1*(1), 45–56. https://doi.org/10.35957/algoritme.v1i1.434

Alwendi, A., & Masriadi, M. (2021). Aplikasi Pengenalan Wajah Manusia Pada Citra Menggunakan Metode Fisherface. *Jurnal Digit*, *11*(1), 01. https://doi.org/10.51920/jd.v11i1.174

Baay, M. N., Irfansyah, A. N., & Attamimi, M. (2021). Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning. *Jurnal Teknik ITS*, *10*(1). https://doi.org/10.12962/j23373539.v10i1.59790

Chalista, N., Natun, I., Santhia, M. A., Kaesmetan, Y. R., Studi, P., Teknik, J., & Uyelindo, S. (2024). *Identifikasi Pengenalan Wajah Berdasarkan Jenis Kelamin Menggunakan Metode Convolutional Neural Network (CNN)*. *6*(1). https://doi.org/10.37802/joti.v6i1.694

Dewi, N., & Ismawan, F. (2021). Implementasi Deep Learning Menggunakan Cnn Untuk Sistem Pengenalan Wajah. *Faktor Exacta*, *14*(1), 34. https://doi.org/10.30998/faktorexacta.v14i1.8989

Faiq, H. A., Sabita, H., Lampung, B., Informatika, J. T., & Network, C. N. (1978). *Pengembangan Model Deep Learning Untuk Pengenalan Wajah pada Sistem Keamanan*. *18*(x), 197–209.

Frenza, D., & Mukhaiyar, R. (2021). Aplikasi Pengenalan Wajah Menggunakan Metode Adaptive Resonance Theory (ART). *Multidicsiplinary Research and Development*, *3*(1), 35–42. https://doi.org/10.38035/rrj.v3i3.392

Guntoro, A. L. S., Julianto, E., & Budiyanto, D. (n.d.). *Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network*. 155–160.

Miftakhurrokhmat, Rajagede, R. A., & Rahmadi, R. (2021). Presensi Kelas Berbasis Pola Wajah, Senyum dan Wi-Fi Terdekat dengan Deep Learning. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), 31–38. https://doi.org/10.29207/resti.v5i1.2575

Nabila, A. (2024). *DEEP LEARNING MENGIDENTIFIKASI UMUR MANUSIA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)*. *12*(3), 1836–1843.

https://doi.org/10.23960/jitet.v12i3.4457

Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia. *Algor*, *2*(1), 12–21.

Simbolon, R. W., Siallagan, S., Munte, D., & Barus, B. (2022). Desain Poster Menarik Memanfaatkan Canva. *BERNAS: Jurnal Pengabdian Kepada Masyarakat*, *3*(3), 448–456. https://doi.org/10.31949/jb.v3i3.2904

Wahyuni, S., & Sulaeman, M. (2022). Penerapan Algoritma Deep Learning Untuk Sistem Absensi Kehadiran Deteksi Wajah Di PT Karya Komponen Presisi. *Jurnal Informatika SIMANTIK*, *7*(1), 5–6.

Wijaya, A. M., & Samodra, J. E. (n.d.). *Sistem Presensi Pegawai dengan Face Recognition Menggunakan Deep Learning CNN*. 163–168. https://doi.org/10.24002/jiaj.v4i2.7660

Zulkhaidi, T. C. A.-S., Maria, E., & Yulianto, Y. (2020). Pengenalan Pola Bentuk Wajah dengan OpenCV. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, *3*(2), 181. https://doi.org/10.30872/jurti.v3i2.4033