Pattern recognition for facial expression detection using convolution neural networks

Mohammad Yazdi Pusadan [a,1,*], James Rio Sasuwuk [a,2], Septiano Anggun Pratama [a,3], Rahma Laila [a,4]

[a] Department of Informatics, Faculty of Engineering, University of Tadulako, Indonesia
[1] yazdi.diyanara@gmail.com; [2] jamesriosasuwuk99@gmail.com; [3] septiano93@gmail.com; [4] lailarahmah.ella@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received May 28, 2024
Revised July 10, 2024
Accepted December 30, 2024
Available online November 30, 2025

Keywords
Emotion recognition
Viola-Jones
CNN
VGG-16
Classification

ABSTRACT

The COVID-19 pandemic was a devastating disaster for humanity worldwide. All aspects of life were disrupted, including daily activities and education. The education sector faced significant challenges at all levels, from kindergarten to elementary, junior high, and high school, as well as in higher education, where learning had to be online. Human emotions are primarily conveyed through facial expressions resulting from facial muscle movements. Facial expressions serve as a form of nonverbal communication, reflecting a person's thoughts and emotions. This research aims to classify emotions based on facial expressions using the Convolutional Neural Network (CNN) and detect faces using the Viola-Jones method in video recordings of online meetings. We utilize the VGG-16 architecture, which consists of 16 layers, including convolutional layers with the ReLU activation function and pooling layers, specifically max pooling. The fully connected layer also employs the ReLU activation function, while the output layer uses the Softmax. The Viola-Jones method is used for facial detection in images, achieving an accuracy of 87.6% in locating faces. Meanwhile, the CNN method is applied for facial expression recognition, with an accuracy of 59.8% in classifying emotions.

1. Introduction

The COVID-19 pandemic is a heartbreaking disaster for all humans on Earth. All activities and life are certainly disrupted due to this outbreak. The education field is also hampered, from kindergarten through elementary, junior high, and high school levels, and even in universities, the teaching and learning process must be carried out online. This prevents lecturers and teachers from monitoring their students' faces and emotions. Naturally and intuitively, humans utilize powerful facial expressions to communicate and express their feelings in social interactions. In these interactions, interpreting emotional states is important for good communication. Emotional states are reflected in words, gestures, and especially facial expressions.

Rapid facial expression recognition is becoming an important part of computer systems, as it is the most expressive way for humans to show emotions [1]–[4]. A person's emotions can be observed through facial expressions, including those of students, which reflect their intentions, social relationships, and personalities. At this time, technological development led to research on facial emotion classification using various methods. Facial emotion classification in technology is a part of computer vision and artificial intelligence (AI) [5]. For example, in the current development of Artificial Intelligence, Facebook can recognize people in an uploaded photo and provide suggestions for tagging the person involved. This capability is a branch of Artificial Intelligence, namely Computer vision [6]–[8]. Notwithstanding considerable progress in AI and computer vision, facial expressions' precise and efficient recognition remains difficult. Facial expression classification is essential in numerous applications, such as education, human-computer interface, and social behavior study. Nevertheless, current methodologies face constraints in precision, real-time processing, and adaptability to diverse facial expressions. Consequently, additional research is required to formulate more resilient and effective methods for facial expression identification [9], especially with student behavior analysis and emotion detection in dynamic settings.

The Convolutional Neural Network (CNN) method enables and has been reported to have promising performance in classifying emotions from video recordings [10]–[12]. Classification models using CNN have been reported to produce better performance in accuracy, precision, and recall compared to other baselines [13]–[15]. One of the main factors behind the better performance of CNN is an equal distribution of training data. Creating an emotion recognition system for human faces from video recordings will enable the detection of multiple faces in a single video recording. Therefore, this research aims to develop a classification technique based on CNN that can recognize changes in facial expressions based on four types (happy, sad, angry, and neutral) of expressions used to determine a person's emotions on video conference recordings.

2. Method

2.1. Dataset

The data set used video conference recordings, such as Zoom or Microsoft Teams, on a 1-page video containing 2 people, 5 people, and 10 people. The training data is from Facial Expression Recognition (FER). The FER2013 dataset from https://www.kaggle.com/datasets/msambare/fer2013 contains 35,887 facial images across seven emotion categories. Four basic human emotions, happy, sad, angry, and neutral, will be used to design the emotion recognition system on facial expressions from video recordings.

2.2. Training Model CNN

In this stage, the author conducts training on the CNN model [11], [12], [16] in classifying facial images. The data for training the CNN model is taken from the Kaggle FER2013 dataset, which selects four labels: happy, sad, angry, and neutral. The initial stage was resizing the image to 48 x 48, and the grayscale stage was used to change the image's color to grey [10], [17]. Then, in the data augmentation stage, the processes include randomly zooming and rotating the training image with a maximum of 90^0 to get variation images. The process of training the CNN model can be seen in Fig. 1.

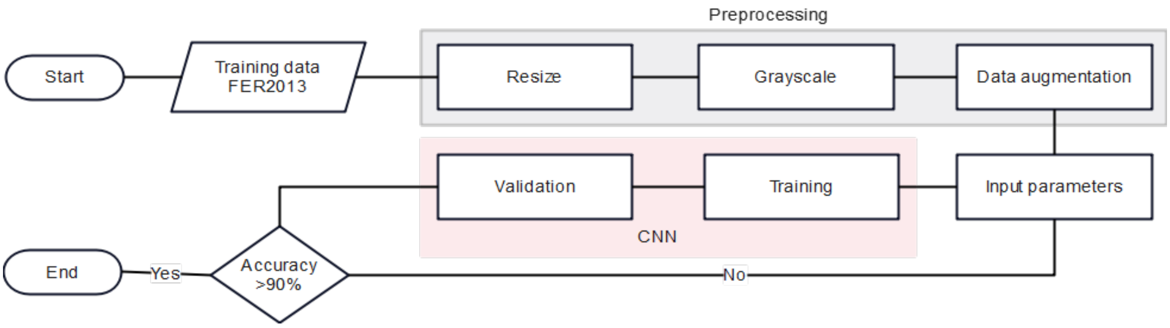


Fig. 1. CNN model training process

2.3. Testing

Testing aims to measure the system's performance to recognize emotions on human faces properly and correctly. The process of testing the system that has been designed can be seen in Fig. 2.

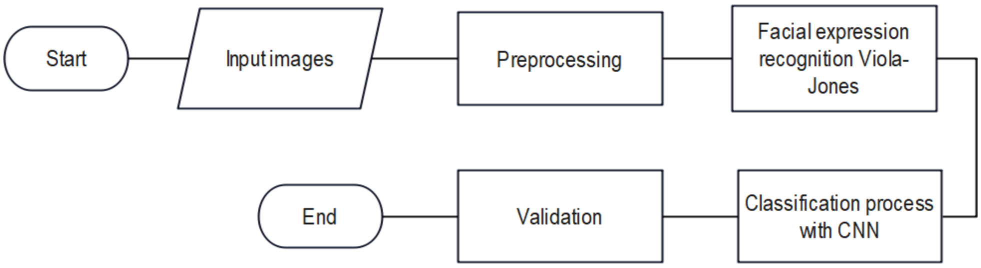

Start
Input images
Preprocessing
Facial expression recognition Viola-Jones
Classification process with CNN
Validation
End

Fig. 2. Model testing process

For testing purposes, the input image was taken from the recorded videos to measure the model's accuracy. The process begins with the input of images, where facial images are provided for analysis. Next, the preprocessing stage prepares the images by enhancing the quality and normalizing the data for better recognition accuracy. The Viola-Jones method is then applied for facial expression recognition, ensuring accurate face detection within the image. Following this, the classification process using CNN is carried out, where the model analyzes facial features to categorize emotions. Finally, the validation phase assesses the classification's accuracy and reliability. A confusion matrix was used to evaluate the performance of a classification method, measuring accuracy, precision, recall, and F1 score.

2.3. Emotion

Humans can intentionally experience certain facial expressions, but they usually occur unintentionally due to human feelings or emotions. Human emotions are mainly expressed by facial expressions, which are the result of facial muscle movements [18]–[20]. Facial expressions are a form of nonverbal communication expressed through people's thoughts. For instance, smiling means warmth, raising eyebrows with an expression of surprise, and frowning with fear and anxiety. Human emotion is a feeling or mental turmoil arising in a person due to stimuli, both from within oneself and outside.

It is often difficult to hide certain feelings or emotions from the face. Basic emotions are generally divided into two categories: primary and secondary emotions. Primary emotions [9], [3] include surprise, pleasure, anger, sadness, fear, and disgust, while secondary emotions include empathy, jealousy, and confusion. The best way to understand emotions is through facial expressions rather than body movements, since emotions are primarily displayed on the face, not the body. The body shows how a person deals with emotions. No specific body movement pattern always expresses the emotion a person is feeling, but each emotion [4], [1] has a specific facial expression. Fig. 3 shows some basic human emotions. In this research, we select four emotions: Angry, Happy, Sad, and Neutral.

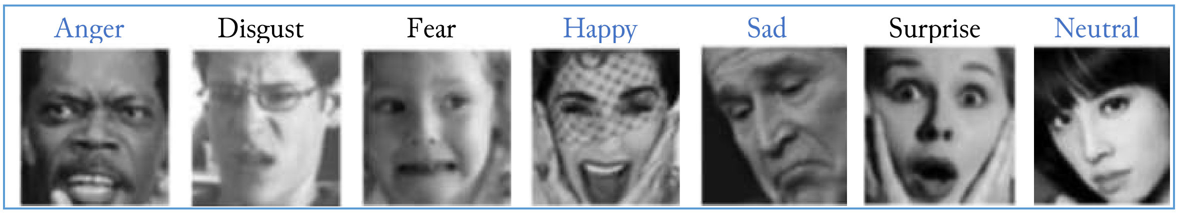


Fig. 3. Basic Human Emotion

2.4. Viola-Jones

Among many face detection methods, Viola-Jones is the most commonly used method [21], [22]. Through a classifier formed based on training data, images are classified based on simple feature values to perform face detection. The Viola-Jones method combines four main keys: Haar-like feature, Integral Image, AdaBoost learning, and Cascade Classifier [23]. Compared to pixel-by-pixel calculation, using Integral Image to extract Haar features increases the calculation time. The main advantages of the Viola-Jones algorithm are a high detection rate and the ability to track faces in images with a low error rate [24]. An outline diagram of the face detection process using the Viola-Jones algorithm [23] can be seen in Fig. 4.

There are several stages of the Viola-Jones method for the face detection process [25], namely: i) the first process is to read a sample of facial images from the input of an image in which there is a human face; ii) from the image that has been entered, the Haar feature reading process [22], [26] is then conducted by processing the image into boxes to get the difference in value from the dark area and the light area. If the difference in value between light and dark areas is above the threshold value, it can be concluded that the feature exists.

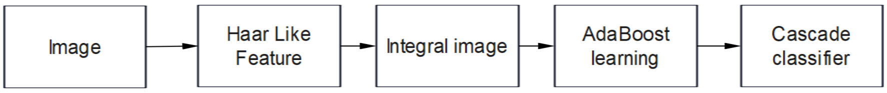


Fig. 4. Viola-Jones method

After the Haar process, the Integral Image method efficiently determines the presence or absence of hundreds of Haar features in an input image at different scales. The definition of integral is adding small units together. In this case, the small units are the values of the image pixels. The integral value of each pixel is the sum of all the pixels from the top and left to the bottom, and the whole image is summed up with integer operations per pixel; (iii) To select good and specific Haar features that will be used later and to set the threshold value, a method from machine learning, namely AdaBoost, is used. This method combines many weak classifiers into a strong classifier by treating AdaBoost as a filter stage, producing a classifier good enough to classify images. During the filter process, if any filters fail to pass a region of the image, that region is immediately classified as non-face. Still, when a filter passes a region of the image and until it passes the entire filter process contained in the filter sequence, then that region of the image is classified as face; and (iv) the next stage is the cascade classifier. The weights assigned by AdaBoost influence the order of filters in the cascade. The filter with the largest weight is placed first in the process, aiming to remove non-face-image regions as quickly as possible.

2.4.1. Haar-like Feature

Image classification is based on a feature value. Haar features [27]–[29] are determined by subtracting the average value of pixels in dark regions from those in light regions. All images with lower values than others and little information are discarded. There are two types of features [30], which are based on the number of rectangles [25] (light and dark) contained in it, namely 2,3,4 rectangles, as can be seen in Fig. 5.

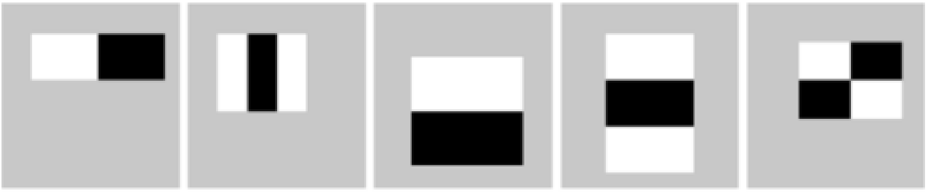


Fig. 5. Haar Like Features

2.4.2. Integral Image

Integral images are data structures and algorithms that add values to parts of the image matrix. The integral value for each pixel is the sum of all pixels from top to bottom. The whole image can be summed with per-pixel integer operations from the top left to the bottom right. Then, to select the specific Haar feature [28], which is specified to be utilized and to set its threshold value, a machine learning method called AdaBoost [31] is used. For an example of integral image calculation, see Fig. 6.

1	2	3
4	5	6
7	8	9

Input Image

1	3	6
5	12	21
12	27	45

Integral Image

Fig. 6. Integral image calculation

2.4.3. AdaBoost Learning

AdaBoost [32], [33] combines many weak classifiers to create a strong classifier by combining multiple AdaBoost classifiers [34] as a series of filters that are efficient enough to classify image regions. Each filter is a separate AdaBoost classifier consisting of a weak classifier or a Haar filter. During the filtering process, if any filter fails to pass an image, that region is immediately classified as non-face. However, when a filter passes an image region and gets through all the filter processes in the filter chain, the image area is classified as a face.

2.4.4. Cascade Classifier

A characteristic of the Viola-Jones method is the existence of a cascade classifier [35]. The Cascade Classifier [36] is a method that is tasked with rejecting image areas that are not detected by using a classifier trained by the AdaBoost algorithm at each classification level. At the first classifier level, the inputs are all sub-windows images. For sub-windows, images that are able to pass the first classifier will become input for the next classifier, and so on. If there is a sub-windows image that is able to pass all levels of the classifier, then the sub-windows image is declared as a face, while sub-windows that fail to pass the classifier will be eliminated. An illustration of the cascade classifier [36] can be seen in Fig. 7.

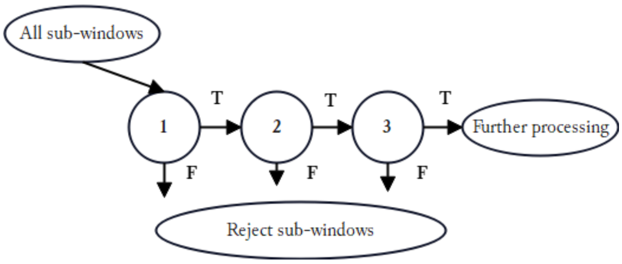


Fig. 7. Ilustrasi cascade classifier [36]

2.5. Convolutional Neural Network

Convolutional Neural Network (CNN) [11], [17] is one type of neural network that is commonly used to process image data. CNN can be used to recognize objects or classify an image. CNN is included in the deep learning group [12] which uses a convolution layer in it which aims to convolve an input image with a filter. CNN consists of two main stages, namely feature learning and classification. The feature learning stage consists of convolutional layer, ReLU (activation function) and pooling layer, while the classification stage consists of flatten, fully connected layer, and prediction. There are two methods that CNN [12], has, namely classification using feedforward and the learning stage using backpropagation. An illustration of the whole CNN process [12] can be seen in Fig. 8.

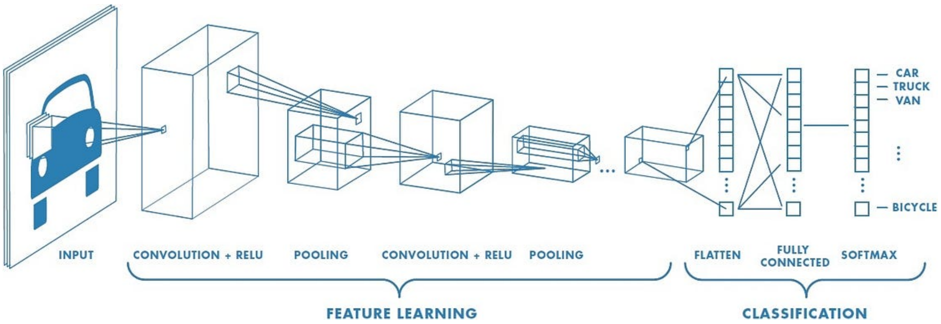


Fig. 8. Full CNN process illustration [12]

2.6. Convolutional Layer

The convolutional layer is a layer that computes the output of neurons connected to local areas of the input image [37]. Each neuron uses a filter to perform dot multiplication between a small area connected to the input image with a shifted filter. The filter in this layer is a 2-dimensional array that can be 7x7, 5x5, 3x3, 1x1. In CNN, a convolutional layer performs feature extraction on an image. The equation used in the convolutional layer calculation process is equation (1).

$$(i,j) = \sum_{m}\sum_{n} w_{m,n}^{l} \cdot o_{i+m,j+n^{+b}}^{l-1} \tag{1}$$

where the convolution calculation results at the position (x,y), represented as (i,j), is determined within a convolutional layer where l denotes the layer, $o(i,j)$ represents the input, $w(m,n)$ is the filter, and b is the bias. The dimensions of the image are given by l for the pixel row and j for the pixel column.

[38] can be seen in Fig. 9.

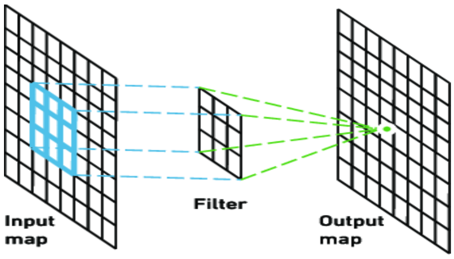


Fig. 9. Ilustrasi proses convolutional layer [38]

In the convolutional layer process, there are two variables that can determine the size of the output matrix, namely Stride and Padding. [39], [40].

2.7. Pooling Layer

One technique that increases the efficiency of CNN is the pooling layer [41]; this layer functions as a filter that optimizes an image and reduces computation in neural networks. The process in the pooling and convolutional layers shifts a window in the input image. The number of strides will determine each shift shifted across the feature map or activation map area.

In application, the commonly used pooling layers [42] are max pooling and average pooling. A kernel of size n*n (2x2) is moved across the matrix, and for each position, the maximal value is taken and inserted into the corresponding position of the output matrix, this is called max pooling, and in the case of average pooling, a kernel of size n*n (2x2) is moved across the matrix and for each position the average is taken of all the values and inserted into the corresponding position of the output matrix. Using a pooling layer reduces the size of the feature map dimensions generated from the convolutional layer process, thus speeding up the computational process due to fewer updated parameters and overfitting. Max pooling [42] is the most commonly used method in CNNs. An example of the max pooling process can be seen in Fig. 10.

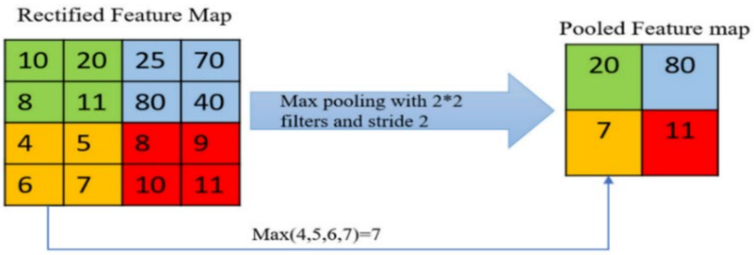


Fig. 10. Proses max-pooling [42]

2.8. ReLU (Rectified Linear Units)

ReLU [43] is an activation function that is responsible for normalizing the value generated by the convolutional layer, for example, the rectifier activation function normalizes the value so that there is no value below 0 by using the $max(0,x)$ function [44]. ReLU will display the value directly if the value is positive, while for negative values, it will be given a value of 0. If there is an input matrix x then the ReLU value is obtained in the equation (2) [43] as follows

$$g(x) = \max\{0, x\} = \left\{\begin{matrix} 0 & for\ x \leq 0 \\ x & for\ x > 0 \end{matrix}\right\} \tag{2}$$

2.9. Flattening

Flattening [45] is an operation that will convert a matrix into a vector with one dimension. The flattening process [46] will change the feature map [47] on the previous layer that has been processed and become a one-dimensional vector so that the feature map [48] that has been flattened can be classified using a fully connected layer [49] and Softmax [50].

2.10. Data Augmentation

Data augmentation is the process of changing and modifying images so that the computer recognizes the changed image as different from the previous image [51], [52]. This data augmentation technique can improve the performance of the trained CNN because by performing augmentation techniques, the model created gets additional data that can greatly help the training process using CNN [53]. Augmentation is done by rescaling, rotating, zooming, and flipping.

2.11. VGG 16

VGG-16 is an architecture developed by Simonyan and Zisserman in 2014 [54]. The VGG architecture was developed in Simonyan and Zisserman's research, which examined the effect of layer depth on CNN on error values. The research was also included in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) competition, and the VGG architecture managed to place 2nd in the competition. The VGG-16 architecture [54] can be seen in Fig. 11.

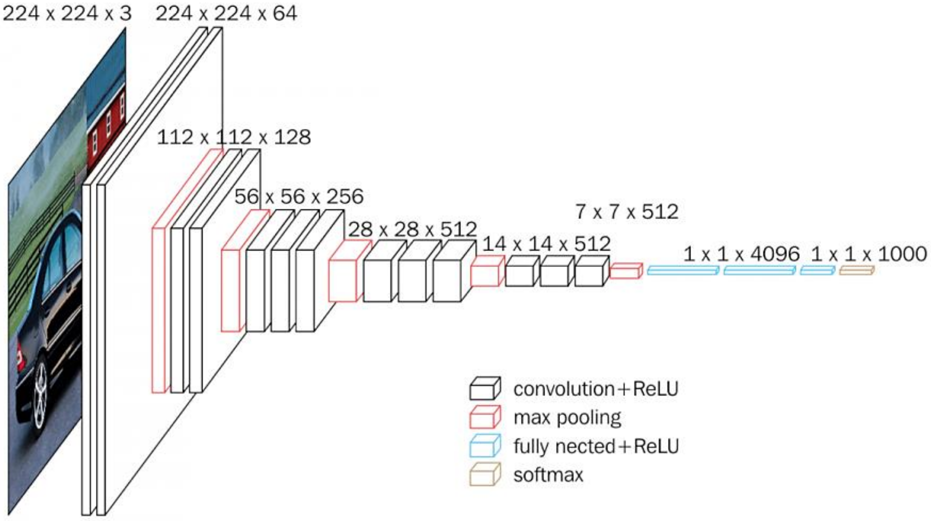


Fig. 11. VGG-16 Architecture [55]

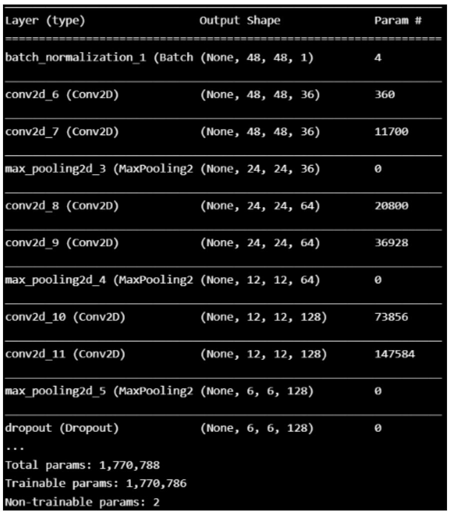
VGG-16 is a VGG architecture version with a layer depth of 16 [55]. In this system design, we will use the VGG-16 architecture with a layer depth of 16, which consists of a convolutional layer using the ReLU activation function, a pooling layer which is more precisely max pooling, then the last one using a fully connected layer using the ReLU activation function and for the output layer a Softmax activation function will be used [55].

3. Results and Discussion

The results achieved in this study are the accuracy of detecting facial images from various video recording inputs. The first test dataset is from video conference recordings, and the second test is from live cameras or live detection using the Viola-Jones method. Furthermore, the accuracy of emotion classification (angry, happy, sad, and neutral) on human faces was assessed using video recordings and live camera-based CNN methods.

3.1. CNN model

The CNN experiment model using the VGG-16 architecture with 200 epochs is shown in Fig. 13. The CNN model gets high accuracy for emotion classification on each training data and image validation data taken from the Kaggle FER2013 dataset with the accuracy of the training was 88.39%, and the accuracy of the validation data is 72% that shown in Fig. 14.



```
Layer (type)                 Output Shape              Param #
=================================================================
batch_normalization_1 (Batch (None, 48, 48, 1)         4
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 48, 48, 36)        360
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 48, 48, 36)        11700
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 24, 24, 36)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 24, 24, 64)        20800
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 24, 24, 64)        36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 12, 12, 128)       73856
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 12, 12, 128)       147584
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 6, 6, 128)         0
_________________________________________________________________
dropout (Dropout)            (None, 6, 6, 128)         0
...
Total params: 1,770,788
Trainable params: 1,770,786
Non-trainable params: 2
```

Fig. 12. VGG 16 Architecture Layer Details

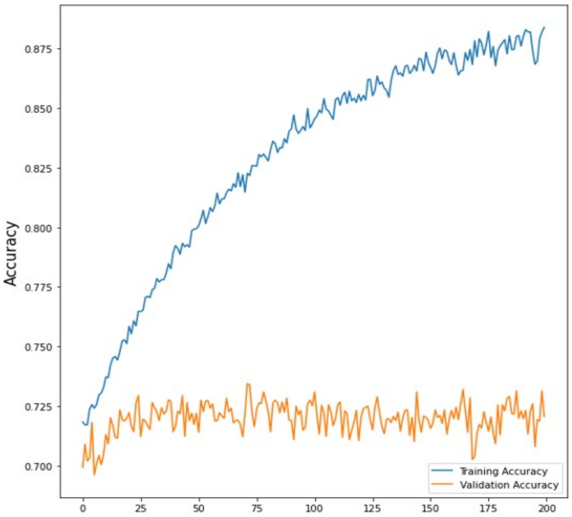


Fig. 13. Graphic of accuracy (200 epochs)

3.2. Experiment-based application output

Data from video conference recordings were used for the first test. The dataset comprised 584 images, split into 408 for training and 176 for testing. The Second test data from the live camera has 52 images. For each well-detected result in the image, whether from video recordings or camera detection, the emotion is directly categorized into one of four categories: angry, happy, neutral, or sad. The results are analyzed manually and using the multi-class confusion matrix method. Examples of test results using a live camera are shown in Fig. 15, and those using video recordings are shown in Fig. 16.

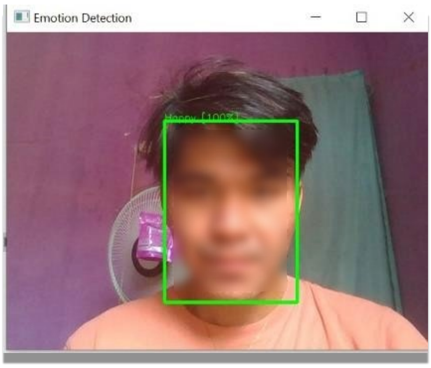


Fig. 14. Live of the camera

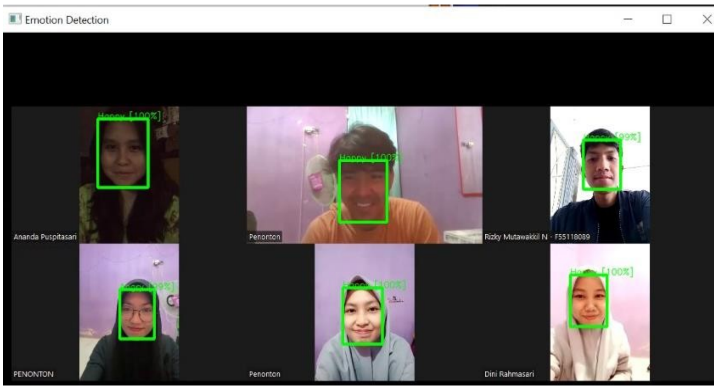


Fig. 15. Video recording

The face detection process using the Viola-Jones method runs well. It achieves high accuracy on the first test data from video conference recordings and the second test data from live cameras. The accuracy obtained from the test data in the first test was 78.9%, and in the second test, the accuracy was 96.2%.

3.3. Accuracy of training data from video recording

Table 1 shows the training accuracy results calculated from video conference recordings totaling 408 images, divided into 12 video conference recordings per emotion type: happy, angry, sad, and neutral. There are 102 instances of each emotion category: angry, happy, neutral, and sad. The accuracy of emotion classification is calculated as follows: 9 out of 102 images are correctly classified, yielding an accuracy of 8.8%. Happy emotions are correctly classified in 63 out of 102 images, with an accuracy of 61.7%. Neutral emotions are correctly classified in 25 out of 102 images, achieving an accuracy of 24.5%. Sad emotions are correctly classified in 35 out of 102 images, with an accuracy of 34.3%.

Table 1. Accuracy results of training data through video recordings

Type of emotions	Total	Detected true	Accuracy
angry	102	9	8.8%
happy	102	63	61.7%
neutral	102	25	24.5%
sad	102	35	34.3%

3.4. Accuracy of test data through video recording

The Accuracy of test data from Table 2 is calculated from video conference recordings containing up to 176 images, divided into 5 recordings for each emotion type: happy, sad, angry, and neutral.

Table 2. The Accuracy Results of Test Data Through Video Recordings

Type of emotions	Total	Detected true	Accuracy
angry	44	8	18.1%
happy	44	28	63.6%
neutral	44	13	29.5%
sad	44	22	50%

The results of the accuracy calculation obtained on angry emotions that are correctly classified in the image with the number of images of angry emotions are 8 out of 44 images, with an accuracy of 18.1%, on happy emotions that are correctly classified in the image with the number of images of happy emotions are 28 out of 44, with an accuracy 63.6%, on neutral emotions that are correctly classified in the image with the number of images of neutral emotions are 13 out of 44 images, with an accuracy 29.5%, and Sad emotions classified correctly in the image with the number of images of happy emotions are 22/44 by 50%. The total accuracy of correctly classified data obtained based on test data is 40.3%, and the total accuracy of correctly classified data obtained based on the number of detected faces is 51.1%.

3.5. Test data accuracy with live cameras

The result of the accuracy of the test data through the live camera from Table 3 is calculated from the data that has been collected from as many as 52 camera capture images, and the types of emotions on the face used are the same as the first test, namely, happy, angry, sad and neutral. In the overall dataset of 52 images, there are 13 angry, 13 happy, 13 neutral, and 13 sad emotions. The accuracy for correctly detected angry emotions is 5/13 (38.4%), for correctly detected happy emotions is 12/13 (92.3%), for correctly detected neutral emotions is 10/13 (76.9%), and for correctly detected sad emotions is 10/13 (76.9%). The total accuracy of correctly classified data obtained based on test data is 71.1%, and the total accuracy of correctly classified data obtained based on the number of detected faces is 68.5%.

Table 3. Test Data Accuracy Results Via Live Camera

Type of emotions	Total	Detected true	Accuracy
angry	13	5	38.4%
happy	13	12	92.3%
neutral	13	10	76.9%
sad	13	10	76.9%

3.6. Confusion matrix multi-class

The multi-class confusion matrix is one method for measuring the performance of a classification method. The Multi-class confusion matrix provides information on how the system's classifications compare with the expected classifications. The Confusion Matrix Multi Class will be implemented on the first test data using video recordings and on the second test data using live cameras. The classification results are shown in Table 4 for the first test data and in Table 5 for the second test.

Table 4. First data testing of the classification confusion matrix for multi-class

	angry	happy	neutral	sad
angry	8	18	2	4
happy	0	28	3	4
neutral	1	18	13	3
sad	1	12	1	22

Table 5. Second test data multi-class confusion matrix classification

	angry	happy	neutral	sad
angry	5	3	1	4
happy	0	12	1	0
neutral	0	1	10	2
sad	0	3	0	10

Based on Tables 4 and 5, the collected and classified data are used to compute Accuracy, Precision, Recall, and F1-Score. The results for Accuracy, Precision, Recall, and F1-Score are shown in Table 6.

Table 6. Accuracy, precision, recall, and F1-score values

	accuracy	precision	recall	F1-*score*
Data testing 1	51.07%	62.98%	50.81%	56.24%
Data testing 2	68.51%	77.24%	71.15%	74.07%

Based on Table 6, the Multi Class Confusion Matrix percentages obtained on the first test data are: 51.07% for accuracy, 62.98% for precision, 50.81% for recall, and 56.24% for F1-score. In the second test, the accuracy was 68.51%, the precision was 77.24%, the recall was 71.15%, and the F1-score was 74.07%.

4. Conclusion

The conclusions obtained by the author from the results of research and testing conducted on the human facial emotion recognition program through video recordings using the Convolutional Neural Network method are as follows: i) the results of testing the keys on the human facial emotion recognition program using the black box testing method and confusion matrix can run well. The functions and features in the program have run according to what was designed; ii) the results on the first test data, using video recordings, the average accuracy obtained in the face detection process is 78.9%, and for the recognition of emotions from detected faces is 51.1%; iii) the results on the second test data, that is through a live camera or live detection, the average accuracy obtained in the face detection process is 96.2%, and for the recognition of emotions from detected faces is 68.5%; and iv) the overall system average accuracy of the first test data through video recording and the second test data through live cameras obtained an accuracy of 87.6% in the face detection test and 59.8% for emotion recognition of detected human faces. For further development research, it is determined based on: i) for the accuracy of human facial emotion recognition, it still gets low accuracy, so additional methods are needed such as adding transfer learning techniques or also looking for other architectures that are more suitable for emotion recognition; ii) can detect directly from the Zoom application or other video conference applications used, so that users don't need to do the recording process manually and input the recording results into the program; and iii) the human facial emotion recognition program not only detects 4 human facial emotions but can add other types of emotions to classify human facial emotions such as emotions of shock, fear, disgust and others.

Acknowledgment

The authors would like to thank the Faculty of Engineering, University of Tadulako, which has provided financial assistance in this research. The authors also thank several parties who supported the availability of data during the research.

Declarations

Author contribution. All authors contributed equally to the paper's main contribution. All authors read and approved the final article.

Funding statement. This research was funded through the DIPA Research Fund, Faculty of Engineering, Fiscal Year 2024.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] Z. Abidin and Alamsyah, "Wavelet based approach for facial expression recognition," *Int. J. Adv. Intell. Informatics*, vol. 1, no. 1, pp. 7–14, 2015, doi: 10.26555/ijain.v1i1.7.

[2] M. Sajjad *et al.*, "A comprehensive survey on deep facial expression recognition: challenges, applications, and future guidelines," *Alexandria Eng. J.*, vol. 68, pp. 817–840, 2023, doi: 10.1016/j.aej.2023.01.017.

[3] D. Y. Liliana, "Emotion recognition from facial expression using deep convolutional neural network," *J. Phys. Conf. Ser.*, vol. 1193, no. 1, 2019, doi: 10.1088/1742-6596/1193/1/012004.

[4] J. H. Kim, B. G. Kim, P. P. Roy, and D. M. Jeong, "Efficient facial expression recognition algorithm based on hierarchical deep neural network structure," *IEEE Access*, vol. 7, pp. 41273–41285, 2019, doi: 10.1109/ACCESS.2019.2907327.

[5] H. Sheikh, C. Prins, and E. Schrijvers, "Mission AI: The new system technology, " *Springer*, pp. 1- 385, 2023, doi: 10.1007/978-3-031-21448-6.

[6] E. Kurniawan *et al.*, "Deep neural network-based physical distancing monitoring system with tensorRT optimization," *Int. J. Adv. Intell. Informatics*, vol. 8, no. 2, pp. 185–198, 2022, doi: 10.26555/ijain.v8i2.824.

[7] J. Videira, P. D. Gaspar, V. N. da G. de J. Soares, and J. M. L. P. Caldeira, "Detecting and monitoring the development stages of wild flowers and plants using computer vision: approaches, challenges and opportunities," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 3, p. 347, Oct. 2023, doi: 10.26555/ijain.v9i3.1012.

[8] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, p. 22, Jun. 2018, doi: 10.29099/ijair.v2i1.42.

[9] P. Tarnowski, M. Kołodziej, A. Majkowski, and R. J. Rak, "Emotion recognition using facial expressions," *Procedia Comput. Sci.*, vol. 108, pp. 1175–1184, 2017, doi: 10.1016/j.procs.2017.05.025.

[10] E. S. Nugroho, I. Ardiyanto, and H. A. Nugroho, "Systematic literature review of dermoscopic pigmented skin lesions classification using convolutional neural network (CNN)," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 3, p. 363, Oct. 2023, doi: 10.26555/ijain.v9i3.961.

[11] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Springer Insights Imaging*, vol. 9, pp. 611–629, 2018, doi: 10.1007/978-981-15-7078-0_3.

[12] I. A. Anjani, Y. R. Pratiwi, and S. Norfa Bagas Nurhuda, "Implementation of Deep Learning Using Convolutional Neural Network Algorithm for Classification Rose Flower," *J. Phys. Conf. Ser.*, vol. 1842, no. 1, 2021, doi: 10.1088/1742-6596/1842/1/012002.

[13] A. A. Kurniawan, S. Madenda, S. Wirawan, and R. J. Suhatril, "Multidisciplinary classification for Indonesian scientific articles abstract using pre-trained BERT model," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 2, p. 331, Jul. 2023, doi: 10.26555/ijain.v9i2.1051.

[14] Hanafi, A. Pranolo, and Y. Mao, "Cae-covidx: Automatic covid-19 disease detection based on x-ray images using enhanced deep convolutional and autoencoder," *Int. J. Adv. Intell. Informatics*, vol. 7, no. 1, pp. 49–62, 2021, doi: 10.26555/ijain.v7i1.577.

[15] M. Bramer, *Principles of Data Mining*. London: Springer London, pp. 1-328, 2020, doi: 10.1007/978-1-4471-7493-6.

[16] H. Hanafi, N. Suryana, and A. S. H. Basari, "Dynamic convolutional neural network for eliminating item sparse data on recommender system," *Int. J. Adv. Intell. Informatics*, vol. 4, no. 3, p. 226, Nov. 2018, doi: 10.26555/ijain.v4i3.291.

[17] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, Jan. 2018, doi: 10.1016/j.procs.2018.05.069.

[18] M. T. Vignesh and K. M. Umamaheswari, "Facial expression recognition using Eigen face approach," *Int. J. Health Sci. (Qassim).*, vol. 6, no. March, pp. 1309–1315, 2022, doi: 10.53730/ijhs.v6ns3.5552.

[19] H. Echoukairi, M. El Ghmary, S. Ziani, and A. Ouacha, "Improved Methods for Automatic Facial Expression Recognition," *Int. J. Interact. Mob. Technol.*, vol. 17, no. 6, pp. 33–44, 2023, doi: 10.3991/ijim.v17i06.37031.

[20] U. B. Chavan, "Facial Expression Recognition- Review," in *Conference: IJLTET*, 2013, vol. 3, no. 1, pp. 237–243, [Online]. Available at: https://www.researchgate.net/publication/.

[21] A. A. Elngar, M. Arafa, A. E. R. A. Naeem, A. R. Essa, and Z. A. shaaban, "The Viola-Jones Face Detection Algorithm Analysis: A Survey," *J. Cybersecurity Inf. Manag.*, no. June, pp. 85–95, 2021, doi: 10.54216/jcim.060201.

[22] D. M. Abdulhussien and L. J. Saud, "evaluation study of face detection by Viola-Jones algorithm," *Int. J. Health Sci. (Qassim).*, vol. 6, no. August, pp. 4174–4182, Sep. 2022, doi: 10.53730/ijhs.v6nS8.13127.

[23] R. R. Damanik, D. Sitanggang, H. Pasaribu, H. Siagian, and F. Gulo, "An application of viola jones method for face recognition for absence process efficiency," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, 2018, doi: 10.1088/1742-6596/1007/1/012013.

[24] K. D. Ismael and S. Irina, "Face recognition using Viola-Jones depending on Python," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 20, no. 3, pp. 1513–1521, 2020, doi: 10.11591/ijeecs.v20.i3.pp1513-1521.

[25] M. A. Rajab and K. M. Hashim, "An automatic lip reading for short sentences using deep learning nets," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 1, pp. 15–26, 2023, doi: 10.26555/ijain.v9i1.920.

[26] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," *Image Process. Line*, vol. 4, pp. 128–148, 2014, doi: 10.5201/ipol.2014.104.

[27] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. II, no. November, pp. 1619–1626, 2005, doi: 10.1109/ICCV.2005.129.

[28] I. G. Susrama, M. Diyasa, A. H. Putra, M. Rafka, and M. Ariefwan, "Feature Extraction for Face Recognition Using Haar Cascade Classifier," vol. 2022, pp. 197–206, 2022, doi: 10.11594/nstp.2022.2432.

[29] P. Menezes, J. C. Barreto, and J. Dias, "Face tracking based on haar-like features and eigenfaces," *IFAC Proc. Vol.*, vol. 37, no. 8, pp. 304–309, 2004, doi: 10.1016/s1474-6670(17)31993-6.

[30] A. Mohamed, A. Issam, B. Mohamed, and B. Abdellatif, "Real-time Detection of Vehicles Using the Haar-like Features and Artificial Neuron Networks," *Procedia Comput. Sci.*, vol. 73, no. Awict, pp. 24–31, 2015, doi: 10.1016/j.procs.2015.12.044.

[31] Y. Ding, H. Zhu, R. Chen, and R. Li, "An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification," *Appl. Sci.*, vol. 12, no. 12, 2022, doi: 10.3390/app12125872.

[32] R. Wang, "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review," *Phys. Procedia*, vol. 25, pp. 800–807, 2012, doi: 10.1016/j.phpro.2012.03.160.

[33] O. Hornyák and L. B. Iantovics, "AdaBoost Algorithm Could Lead to Weak Results for Data with Certain Characteristics," *Mathematics*, vol. 11, no. 8, 2023, doi: 10.3390/math11081801.

[34] R. Senkamalavalli, M. Balamurugan, R. N. Sundara, and N. Ramshankar, "Improved classification of breast cancer data using hybrid techniques," *Cardiometry*, vol. 6495, no. 26, pp. 487–490, 2023, doi: 10.18137/cardiometry.2023.26.487490.

[35] B. R. Maale and S. Nandyal, "Face detection using Haar cascade classifier," *Int. J. Sci. Res.*, vol. 10, no. 3, pp. 2019–2022, 2021, doi: 10.2139/ssrn.4157631.

[36] R. Padilla, C. C. Filho, and M. Costa, "Evaluation of haar cascade classifiers designed for face detection," *J. WASET*, vol. 6, no. 4, pp. 323–326, 2012. [Online]. Available at: https://www.researchgate.net/publication/303251696_Evaluation_of_Haar_Cascade_Classifiers_for_Face_Detection.

[37] B. Liu *et al.*, "A novel compact design of convolutional layers with spatial transformation towards lower-rank representation for image classification," *Knowledge-Based Syst.*, vol. 255, p. 109723, 2022, doi: 10.1016/j.knosys.2022.109723.

[38] H. Yakura, S. Shinozaki, R. Nishimura, Y. Oyama, and J. Sakuma, "Malware analysis of imaged binary samples by convolutional neural network with attention mechanism," *CODASPY 2018 - Proc. 8th ACM Conf. Data Appl. Secur. Priv.*, vol. 2018-Janua, pp. 127–134, 2018, doi: 10.1145/3176258.3176335.

[39] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[40] F. Alrasheedi, X. Zhong, and P. C. Huang, "Padding Module: Learning the Padding in Deep Neural Networks," *IEEE Access*, vol. 11, no. December 2022, pp. 7348–7357, 2023, doi: 10.1109/ACCESS.2023.3238315.

[41] H. J. Jie and P. Wanda, "Runpool: A dynamic pooling layer for convolution neural network," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 66–76, 2020, doi: 10.2991/ijcis.d.200120.002.

[42] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Appl. Sci.*, vol. 12, no. 17, pp. 1–21, 2022, doi: 10.3390/app12178643.

[43] E. Oostwal, M. Straat, and M. Biehl, "Hidden unit specialization in layered neural networks: ReLU vs. sigmoidal activation," *Phys. A Stat. Mech. its Appl.*, vol. 564, p. 125517, 2021, doi: 10.1016/j.physa.2020.125517.

[44] T. Mao and D. X. Zhou, "Rates of approximation by ReLU shallow neural networks," *J. Complex.*, vol. 79, p. 101784, 2023, doi: 10.1016/j.jco.2023.101784.

[45] X. Liu, S. Li, X. Zheng, and M. Lin, "Development of a flattening system for sheet metal with free-form surface," *Adv. Mech. Eng.*, vol. 8, no. 2, pp. 1–12, 2016, doi: 10.1177/1687814016630517.

[46] A. Gurov, E. Evmenova, and P. Chunaev, "Supervised community detection in multiplex networks based on layers convex flattening and modularity optimization," *Procedia Comput. Sci.*, vol. 212, no. C, pp. 181–190, 2022, doi: 10.1016/j.procs.2022.11.002.

[47] H. Ma, J. Zhang, J. Zhou, X. Zhai, J. Xue, and H. Ji, "Method for constructing multi-dimensional feature map of malicious code," *J. Phys. Conf. Ser.*, vol. 1748, no. 4, 2021, doi: 10.1088/1742-6596/1748/4/042055.

[48] Y. Tang *et al.*, "Beyond dropout: Feature map distortion to regularize deep neural networks," *AAAI 2020 - 34th AAAI Conf. Artif. Intell.*, pp. 5964–5971, 2020, doi: 10.1609/aaai.v34i04.6057.

[49] B. Shah and H. Bhavsar, "Time Complexity in Deep Learning Models," *Procedia Comput. Sci.*, vol. 215, no. 2022, pp. 202–210, 2022, doi: 10.1016/j.procs.2022.12.023.

[50] Q. Jodelet, X. Liu, and T. Murata, "Balanced softmax cross-entropy for incremental learning with and without memory," *Comput. Vis. Image Underst.*, vol. 225, no. January, p. 103582, 2022, doi: 10.1016/j.cviu.2022.103582.

[51] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, no. August, p. 100258, 2022, doi: 10.1016/j.array.2022.100258.

[52] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[53] N. A. M. Roslan, N. M. Diah, Z. Ibrahim, Y. Munarko, and A. E. Minarno, "Automatic plant recognition using convolutional neural network on Malaysian medicinal herbs: the value of data augmentation," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 1, pp. 136–147, 2023, doi: 10.26555/ijain.v9i1.1076.

[54] F. D. Adhinata, N. A. F. Tanjung, W. Widayat, G. R. Pasfica, and F. R. Satura, "Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 2, pp. 230–237, Jul. 2021, doi: 10.26555/JITEKI.V7I2.20758.

[55] P. Gayathri, A. Dhavileswarapu, S. Ibrahim, R. Paul, and R. Gupta, "Exploring the Potential of VGG-16 Architecture for Accurate Brain Tumor Detection Using Deep Learning," *J. Comput. Mech. Manag.*, vol. 2, no. 2, pp. 1–10, 2023, doi: 10.57159/gadl.jcmm.2.2.23056.