

Detection of Participants Facial Expressions in Video Conference Using Convolutional Neural Network Algorithm

Karimuddin Hakim Hasibuan¹, Hermawan Syahputra²

^{1,2}Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Medan, Indonesia

Abstract.

Purpose: The purpose of this research is to develop an architecture based on the Convolutional Neural Network (CNN) algorithm to detect facial expressions during video conferences. The goal is to address the problem of understanding participants' emotions and expressions during online video conferencing sessions. The aim is to create a system that can analyze facial expressions in images and determine the corresponding emotions.

Methods/Study design/approach: Data was collected by capturing facial expression images from 10 students using a webcam. Preprocessing techniques, such as cropping, converting images to grayscale, and data augmentation, were applied to ensure data variation. The CNN model was trained using the processed data and evaluated using test data (a subset of the dataset), new data (external data) and video conference recording.

Result/Findings: The CNN model achieved a high training accuracy of 97.5% using an image size of 128x128 and 2000 epochs. The model architecture consists of 2 Conv2D layers, 3 BatchNormalization layers, 2 MaxPooling layers, 2 dropout layers, 1 flat layer, 1 dense layer, and 1 output layer. When tested on facial expression data, the model achieved with 97,5% accuracy on the training data and 93,33% accuracy on the test data. The model was also able to detect the facial expressions of participants in the video conference.

Novelty/Originality/Value: The novelty of this research lies in developing a CNN-based system to detect facial expressions in video conferences by analyzing facial images. This approach addresses the challenge of understanding participants' emotions and expressions during online video conferencing sessions, which can contribute to better communication and interaction among participants.

Keywords: Facial Expressions, Video Conference, Convolutional Neural Network.

Received Month 20xx / **Revised** Month 20xx / **Accepted** Month 20xx

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



1. INTRODUCTION

The COVID-19 pandemic has drastically changed interpersonal and group communication, leading to limitations in individual activities. To overcome these limitations, humans have increasingly relied on online technologies for remote communication. Video conferencing has emerged as a prominent technology enabling audio and visual interactions, bridging geographical gaps. Platforms like Zoom, Microsoft Teams, and Google Meet have provided users with the convenience of face-to-face interactions in virtual settings [1].

Video conferencing facilitates real-time long-distance communication, enabling participants to interact with each other through full-screen mode or screen sharing [2]. Its widespread adoption across various sectors, including education, has become instrumental in addressing the challenges of remote learning. By allowing students to engage with teachers visually and audibly in real-time, video conferencing has contributed significantly to overcoming global learning challenges.

However, during video conferences, it becomes challenging for educators to gauge the participants' expressions and attitudes directly. Lack of responsiveness from participants can result in minimal

¹*Corresponding author.

Email addresses: 1st author email (last name), 2nd author email (last name), 3rd author email (last name)

DOI: [10.24114/j-ids.xxxxx](https://doi.org/10.24114/j-ids.xxxxx)

interaction, hindered discussions, and even display inappropriate facial expressions, such as appearing rigid, tense, or gloomy [3]. In the context of online education, maintaining positive character traits becomes crucial, with politeness being one of the essential moral values that individuals should possess [4].

Understanding facial expressions plays an important role in understanding and recognizing a person's character. Facial expressions have been found to be a powerful way to convey emotions and intentions, with research conducted by Mehrabian in 1971 showing that facial expressions contribute significantly (55%) to message communication [5]. The ability to detect and analyze facial expressions can provide insight into the attitudes of participants, thereby improving online communication and learning experiences.

In this era of digital advancement, image processing and recognition technologies, including facial expression recognition, have been extensively researched and implemented using artificial intelligence (AI) techniques [6]. One of the most powerful AI techniques is deep learning, which has revolutionized pattern recognition tasks and outperformed traditional machine learning methods [7]. Convolutional Neural Networks (CNNs), a class of deep learning algorithms, have shown exceptional capabilities in image recognition tasks and have become widely adopted in various fields, including facial expression recognition [8].

While deep learning and CNNs have demonstrated remarkable performance, the need for real-time facial expression analysis during video conferences remains a challenge. The ability to detect politeness or other expressions through facial analysis in real-time video conferencing settings is yet to be fully explored. Therefore, this research aims to implement the CNN algorithm for facial expression recognition during video conferences, with a focus on identifying expressions related to politeness and other emotions in a real-time context. By developing such a system, educators and participants can gain valuable insights into the effectiveness of their online interactions and create a more engaging and respectful learning atmosphere.

2. METHODS

2.1. Research location and time

The research location was conducted at Medan State University which is located at Williem Iskandar Street, Pasar V Medan Estate, Percut Sei Tuan, Deli Serdang, North Sumatra. The specific location selection was conducted in the Computer Science Study Program Room 77.01.07. The research time was from March to April 2023.

2.2. Population and sample

The research population and sample comprised 10 students from the Computer Science Study Program at Medan State University.

2.3. The research flow

The flow of the research is represented in the figure below.

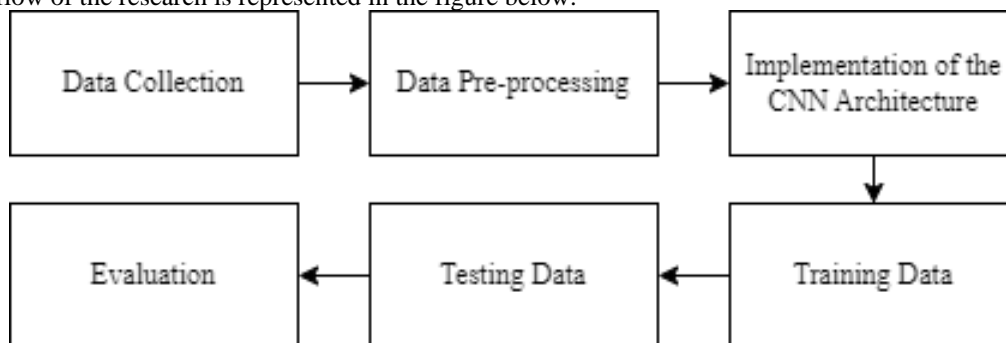


Figure 1. Research flow.

Figure 1 presents the research process consists of several stages. The first stage involves data collection, where the collected data is divided into training and testing sets. The collected facial expression data is then categorized based on predefined and validated facial expression indicators. The next stage is data preprocessing, which involves improving the quality of the collected data through image processing techniques. Afterward, the Convolutional Neural Network (CNN) algorithm is implemented to recognize the input objects in the system. The implemented CNN model is trained and tested, where the training phase introduces and trains the model to recognize the target objects, while the testing phase evaluates the model's accuracy in object recognition. Finally, the model is evaluated based on accuracy and confusion matrix to

assess its performance in detecting facial expressions in the test data samples. A detailed walkthrough of the research is shown below:

a) Data Collection

In this research, for the purpose of identifying facial expressions, an object in the form of a photo of facial expressions is needed to be identified. In this research, the case study that will be used is a photo of facial expressions of Medan State University students. Then, the object of this research will be photographed manually using a webcam camera. Meanwhile, the expressions that will be identified in this study are divided into 6 types of facial expressions, namely normal, happy, sad, afraid, angry, and surprised. So students are asked to express as naturally as possible to show these indicators. To validate that the expression displayed is correct and appropriate, the researcher determines several characteristics that match the expression.

In a research conducted by Nendya et al [9] which quoted from the research of The Duy Bui [10], a textual description of several facial conditions when expressing. Textual descriptions can help to understand the description of a person's behavior, feelings, or emotions. The following indicators used in detecting facial expressions participants can be seen in Table 1:

Table 1. Facial expression indicators.

No	Expression	Textual Description
1	Normal	- Relaxed position of the eyes, eyebrows, cheeks, and lips. - Closed lips.
2	Happy	- Open mouth. - Corners of the mouth pulled towards the ears and normal eyebrow position.
3	Sad	- Inner eyebrows bent upwards. - Slightly closed eyes and relaxed mouth position.
4	Fear	- Raised and pulled eyebrows. - Inner eyebrows bent upwards and tense eye shape.
5	Surprised	- Wide open upper eyelids and relaxed lower eyelids. - Opened jaw and Raised eyebrows.
6	Angry	- Inner eyebrows pulled downwards. - Wide open eyes and lips pulled together, showing teeth.

Table 1 provides information on the textual descriptions for the facial expressions used as indicators in collecting the data. Based on these facial expressions indicators and textual descriptions, participants are requested to express themselves according to the respective expression categories and are then photographed from 10 different angles, starting from the front, right/left side, and shifting in the opposite direction to capture the specific expression indicators. The goal is to obtain image variations for each expression. With the predetermined indicators and sample size, there are 100 images per expression, resulting in a total dataset of 600 facial expression images. The dataset is divided into training and testing data, with a split ratio of 90%:10%. This distribution is chosen due to the limited available data and the need for a substantial amount of data for CNN algorithms to learn effectively.

b) Data Pre-processing

In the research conducted by Lopes et al [11] on expression identification, the images used for training were subjected to preprocessing techniques to enhance their suitability for the CNN algorithm. This research project acknowledges and adopts the same preprocessing techniques due to the limited quantity of available data.

The preprocessing consists of three steps. Firstly, cropping is applied to extract the facial region from the images, ensuring that only the relevant area is considered. Secondly, the images are converted to grayscale, simplifying the computation process and aiding feature extraction. Lastly, image augmentation techniques are implemented to increase the diversity of the data. Various transformations such as rescaling, shifting, rotating, zooming, shearing, and flipping are applied with specific parameter values to enrich the dataset.

These preprocessing steps pave the way for the subsequent CNN architecture. The CNN model is designed to process the preprocessed images and construct a suitable model to classify and identify expressions. Ultimately, this CNN architecture will be utilized in the testing phase to evaluate and obtain the desired outputs.

c) Implementation of the CNN Architecture

In this step, a CNN architecture is designed to detect facial expressions in video conference. The architecture used in this study is an extension of the previous CNN model developed by Liu et al [12] and further enhanced by Lopes et al [11] to create a powerful CNN model with a limited amount of data. The CNN architecture can be visualized through the block diagram shown in Figure 2.

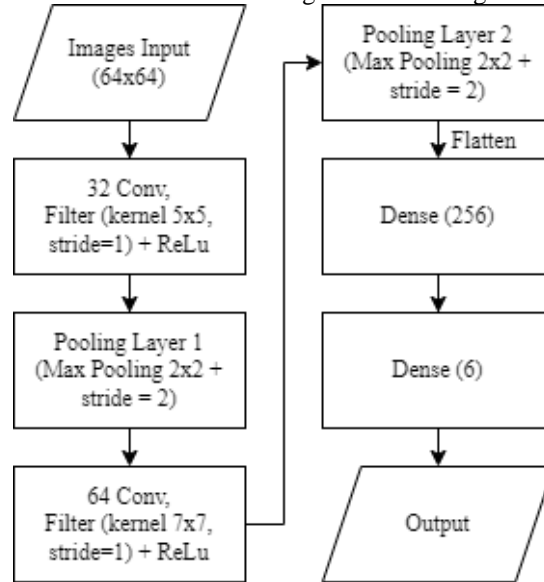


Figure 2. CNN Architecture.

Figure 2 shows the architecture of the convolutional neural network (CNN) algorithm, the sequence of the convolution process can be explained as follows:

1. The images are augmented and stored as variables.
2. These variables are then used as input in the CNN model, which is set to a size of 32x32x1. Different image sizes, such as 64 and 128 pixels, are used for variation. However, for the purpose of this explanation, only the 64x64 size will be discussed. The number 1 in the image size denotes the grayscale channel.
3. The first convolution process takes place using a 5x5 kernel and 32 filters, with a stride of 1, resulting in a 60x60x32 feature map that is activated using ReLu.
4. Subsequently, max-pooling is performed with a kernel size of 2x2 and a stride of 2 pixels.
5. The pooled feature map is then subjected to the second convolution process, utilizing a 7x7 kernel and 64 filters, producing an 24x24x64 feature map that is activated using ReLu. Increasing the number of filters during the pooling process allows for greater information variation from the available data.
6. The second max-pooling is carried out with a kernel size of 2x2 and a stride of 2 pixels, resulting in a 12x12x64 feature map.
7. Subsequently, the flatten process is performed to transform the feature map into a 1-D list vector with a length of 4608 pixels. This result will serve as input for the fully connected layer.
8. Following the flatten stage, the fully connected layer consists of a multilayer perceptron (MLP) with 3 layers: an input layer with 1 neuron, a hidden layer with 256 neurons, and an output layer with 6 neurons for image classification.

The CNN architecture formation process is achieved by defining the necessary layers to construct the CNN architecture using the available functions in TensorFlow's Keras.

d) Training and Testing Data

The purpose of training the architecture model is to recognize features in images and mark the activated neurons when classifying the images. The preprocessed training data and several learning parameters need to be initialized before the training process. These parameters include the learning rate, which is set using the Adam optimizer to update the weights during each iteration of training data, the batch size, which determines the number of data samples passed to the neural network in each epoch, and the number of epochs, which specifies the number of complete passes through the dataset during training. After initializing the necessary parameters, the CNN algorithm is applied to learn the features and classify the images. During the model training process, the established CNN architecture directly

processes the prepared data. The training process is performed over several pre-determined epochs. In each iteration, the system automatically displays the accuracy and loss values from the training data. Once the training process is completed, the formed model is saved and then used for the testing process.

When the system is executed, it automatically records the screen display of a Zoom meeting that includes 10 students whose facial expression images have been used for training in the data collection phase as participants in the testing, as well as one participant who does not have sample data to evaluate the system's detection success. As a result, the system can display output label boxes that contain the expressions and their respective categories. Examples of these output labels include: 1) Normal; 2) Happy; 3) Sad; 4) Fear; 5) Surprised; 6) Angry.

e) Evaluation

The evaluation of the trained model involves assessing its performance using various metrics, including the confusion matrix, precision, recall, and F1-Score. These metrics provide insights into the model's accuracy and effectiveness in classifying expressions during a video conference.

3. RESULT AND DISCUSSION

3.1. Data collection result

In this research, a sample of 10 student participants from Medan State University was used. Various facial expression photos were taken from these participants to create the dataset for this study. The sample consists of 6 male and 4 female participants, with a total of 6 individuals having a round face shape, 3 individuals with an oval face shape, and 1 individual with a heptagon face shape. Before collecting the data, the researcher defined facial expression images as indicators which were validated and used as reference samples to display appropriate facial expressions. These example images serve as reference samples to demonstrate the variations in facial expressions.

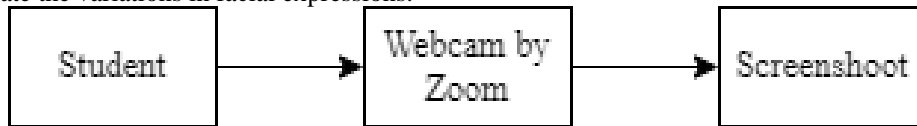


Figure 3. Data collection process on the sample.

Figure 3 presents the process of collecting data in the form of images of facial expressions from the participants is depicted. The participants were instructed to sit facing a webcam that was connected to the Zoom Meetings application. Then, the facial expressions that appear will be screenshotted as facial expression data. Subsequently, the participants were asked to display facial expressions corresponding to the validated indicator images. The Zoom Meetings screen displaying the participants' facial expressions was captured through screenshots, which were then saved as image datasets. An example of the captured screenshots can be seen in Figure 4.

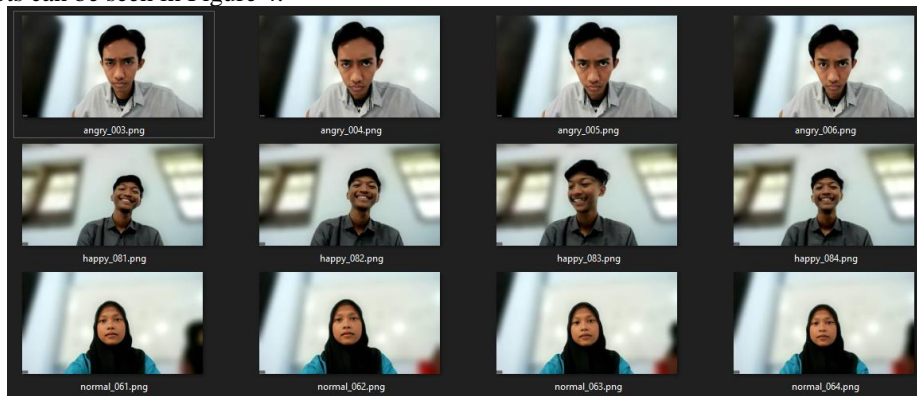


Figure 4. Sample image of data capture results.

Figure 4 shows an example of the results of data collection on the sample. There are 6 classes of facial expressions: Normal, Happy, Sad, Fear, Surprised, and Angry. Each class consists of 100 images, resulting in a total of 600 images as shown in Table 2.

Table 2 represents different facial expressions along with the corresponding number of images available for each expression. There are six types of expressions in total, each consisting of 100 images. The expressions include Normal, Happy, Sad, Fear, Surprised, and Angry. These images will be used for

training and evaluating the Convolutional Neural Network (CNN) algorithm for facial expression recognition during video conferences

Table 2. Distribution of facial expression image data.

No	Expression	Number of pictures
1	Normal	100 images
2	Happy	100 images
3	Sad	100 images
4	Fear	100 images
5	Surprised	100 images
6	Angry	100 images

3.2. Data pre-processing result

In this step, the collected data undergoes several steps to prepare it for the training phase. The steps are cropping, grayscale and image augmentation which can be seen in Figure 5.

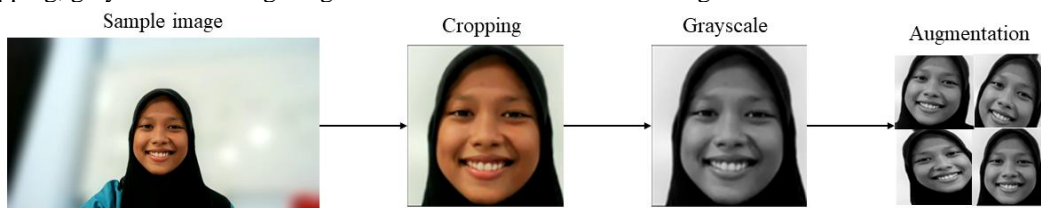


Figure 5. Example of the results of the image preprocessing step on the data.

Figure 5 shows an example of the results of each image preprocessing step on the data consisting of cropping, grayscale and augmentation. Firstly, the cropping process is performed using Python programming language to automatically crop the images, and manual adjustments are made if necessary to ensure the entire face is captured. The cropped images are then resized to a standardized size, such as 500x500 pixels or square. Next, the images are converted to grayscale to simplify the color dimension.

After the cropping and grayscale conversion, the image augmentation step is applied using the ImageDataGenerator from Keras in Python. This step involves various transformations such as rescaling, shifting, rotating, zooming, shearing, and flipping, with specific parameter values for each transformation. Image augmentation increases the diversity of the dataset and helps prevent overfitting during training.

Once the data has completed the preprocessing stage, it is ready to be fed into the CNN architecture for training. These preprocessing steps ensure that the images are in a suitable format and contain enough variations for the CNN model to learn effectively and accurately classify facial expressions during the training phase.

3.3. Implementation of the CNN Architecture result

The Convolutional Neural Network (CNN) architecture used to perform facial expression detection can be seen in Figure 6 below.

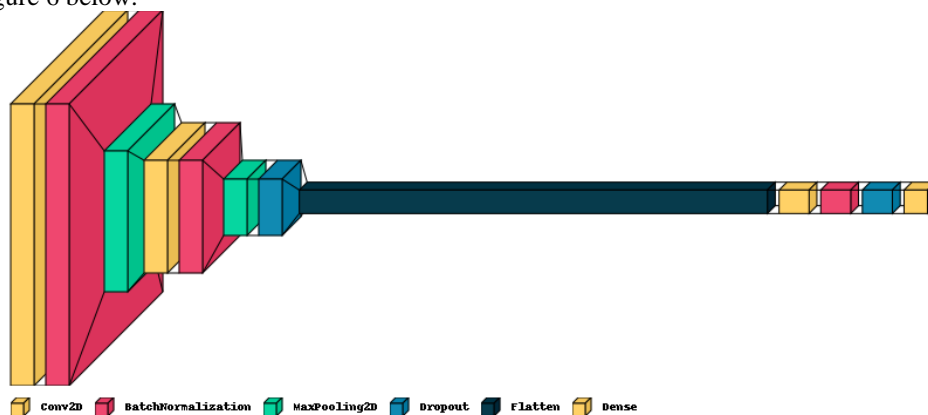


Figure 6. CNN architecture visualization.

Figure 6 shows the visualization of the CNN architecture used. There are 6 parts applied to the architecture namely Conv2D, Batch Normalization, MaxPooling2D, Dropout, Flatten, and Dense. The CNN

architecture applied to the sample image dataset with a size of 64x64 and 128x128 consists of multiple layers, each serving a specific purpose in the facial expression recognition process. The first layers, including Conv2D, Batch Normalization, and MaxPooling2D, are responsible for extracting features from the input images and reducing the feature map's dimensionality. These layers help in identifying important patterns and characteristics in the facial expressions.

Following that, additional Conv2D, Batch Normalization, MaxPooling2D, Dropout, and Flatten layers are applied to further process the data and create a more compact representation of the features. The Dropout layer is particularly helpful in preventing overfitting by randomly dropping some neurons during training, enhancing the model's generalization capabilities.

The process continues with a Dense layer, Batch Normalization, Dropout, and the final Dense output layer. The Dense layers are fully connected and play a critical role in combining the features extracted from the previous layers to make predictions about the facial expressions.

3.4. Training process results

The training process was conducted on a Google Collaboratory platform with 12GB RAM and 12GB GPU specifications. During the training process, various learning parameters were applied. The number of epochs used were 1000, 1500, and 2000 epochs, and the image sizes used were 64x64, and 128x128. The batch size used was 64. The collected data, which has gone through the preprocessing stage, is used to train the CNN model to learn the features and patterns in the data.

The output of this process is a trained CNN model that is ready to be tested. The training process iterates according to a specified number of epochs, where each epoch represents one iteration through the entire training data to extract the required feature representation. In the training process, an optimizer is used to update the weights during the backward-pass phase. The optimizer used is the Adam optimizer with a learning rate of 0.0001.

After the training process is complete, the accuracy and loss values obtained based on various epochs and image size parameters are recorded in Table 3.

Table 3. Training data results.

No	Epochs	Image Size	Accuracy	Loss	Training Duration	GPU Usage	RAM Usage
1	1000	64	0,9333	0,1845	2 jam 9 menit 26 detik	1.3GB	2.1GB
2	1000	128	0,9528	0,1259	2 jam 17 menit 48 detik	3.8GB	4.2GB
3	1500	64	0,9583	0,1330	3 jam 27 menit 12 detik	1.3GB	3.9GB
4	1500	128	0,9708	0,0799	3 jam 45 menit 7 detik	3.8GB	4.2GB
5	2000	64	0,9653	0,0903	4 jam 24 menit 4 detik	1.3GB	3.8GB
6	2000	128	0,9750	0,0736	4 jam 7 menit 5 detik	3.8GB	4.2GB

Table 3 shows the results of the training process on the data. These results show the movement of accuracy and loss across the number of epochs and image sizes used as parameters. There is also the GPU and RAM usage which are the computational resources used as well as the training duration. The training results with 2000 epochs and image size of 128x128 achieved the highest accuracy, with a value of 0.9750 or 97.5%, and the training duration was 4 hours, 7 minutes, and 5 seconds. On the other hand, the lowest training result was obtained with 1000 epochs and image size of 64x64, with a training accuracy of 0.9333 or 93%, and a training duration of 2 hours, 9 minutes, and 26 seconds.

Figure 7 shows the visualization of accuracy and loss values during the training process based on the number of epochs. The training result shown that the accuracy achieved with the 128x128 image size is the highest compared to the other image sizes. The increase in accuracy from 1500 epochs to 2000 epochs is not very significant, ranging from 97% to 97.5%. However, the training time between 1500 epochs and 2000 epochs is longer. The same trend can be observed in the slight decrease in loss between 1500 epochs and 2000 epochs. In addition, it is clear that the model with an image size of 128x128 consistently has the lowest line among the other two lines in the loss section of the graph. This indicates that the loss rate for this image size is the lowest.

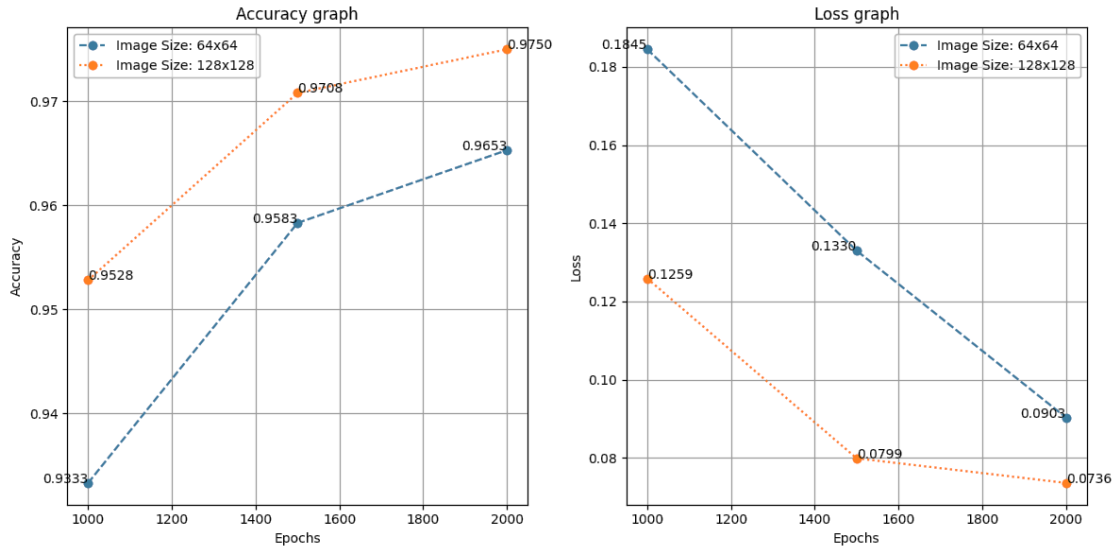


Figure 7. Training result accuracy and loss graph.

Based on the training results with different epoch parameters, the CNN model trained with 2000 epochs and image size of 128x128 emerged as the top-performing model for the training phase. Throughout the training process, the model underwent numerous iterations, updating its weights and biases with the training dataset. By training it for 2000 epochs, the model had sufficient exposure to the data, enabling it to potentially capture intricate patterns and representations from the images.

The choice of a 128x128 image size during training is essential as it affects the model's ability to recognize features in the data. Larger image sizes can allow the model to grasp more detailed information, which is beneficial in recognizing complex patterns. However, a balance must be struck between image size and computational resources, as larger images require more memory and processing time. By selecting the model with the highest accuracy, we prioritize its ability to generalize well on unseen data, demonstrating its effectiveness in making accurate predictions in real-world scenarios during the testing phase.

3.5. Testing process result

The testing process involves evaluating the model's performance on test data and in video conference recording. The purpose of testing is to assess the performance of the model after it has undergone training.

a) Testing results on test data.

In the testing process on the test data, 60 images were randomly selected, representing 10% of the total dataset, with each class of facial expression consisting of 10 images. Each image in the test data is individually predicted using the pre-trained CNN model. Examples of the test results based on the facial expression classes can be seen in Table 4.

Table 4. Several examples of model testing results on test data.






No	Image	True label	Predict	Status
1		normal	normal	True
2		sad	angry	False
3		surprised	surprised	True
4		happy	happy	True
5		surprised	surprised	True

Table 4 presents the example of model testing results on test data. Each row represents an instance where the model made predictions on the given images and compared them to the true labels. In the first row, the model predicted "normal" for an image labeled as "normal," and the prediction was correct

(True). Each image in the test data will be evaluated so that the model can show its performance against the given test data. The table allows us to evaluate the performance of the model in classifying facial expressions. It shows both correct and incorrect predictions, enabling us to calculate metrics such as accuracy, which indicates the overall performance of the model on this particular task. The misclassification in row two suggests that the model might need further improvements to accurately distinguish between sad and angry expressions.

The accuracy of the model in making predictions based on facial expressions can be determined through the confusion matrix displayed in Figure 8. The confusion matrix provides a detailed breakdown of the model's performance, showing the number of true positive, true negative, false positive, and false negative predictions for each class. From this matrix, that can calculate various evaluation metrics, such as accuracy, precision, recall, and F1-score, which help assess the overall effectiveness of the model in classifying different facial expressions.

Test Data Confusion Matrix Result							
TARGET \ OUTPUT	happy	angry	normal	sad	fear	surprised	SUM
happy	10 16.67%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	10 100.00% 0.00%
angry	0 0.00%	10 16.67%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	10 100.00% 0.00%
normal	0 0.00%	0 0.00%	10 16.67%	0 0.00%	0 0.00%	0 0.00%	10 100.00% 0.00%
sad	0 0.00%	2 3.33%	0 0.00%	8 13.33%	0 0.00%	0 0.00%	10 80.00% 20.00%
fear	0 0.00%	1 1.67%	0 0.00%	0 0.00%	8 13.33%	1 1.67%	10 80.00% 20.00%
surprised	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	10 16.67%	10 100.00% 0.00%
SUM	10 100.00% 0.00%	13 76.92% 23.08%	10 100.00% 0.00%	8 100.00% 0.00%	8 100.00% 0.00%	11 90.91% 9.09%	56 / 60 93.33% 6.67%

Figure 8. Testing results on test data based on facial expressions.

Figure 8 displays the confusion matrix which provides a comprehensive visual representation of the model's performance on facial expression recognition using the test data. By comparing the model's predictions with the actual labels for each class, this matrix shows the ratio of the number of images with actual labels to their predicted labels. In the resulting confusion matrix, it can be seen that the model was able to detect 56 images correctly, while 4 images were misclassified. From these results, the accuracy can be calculated as 93.33%. The precision, recall and F1-score values for each expression class can be seen in Table 5.

Table 5. Precision, recall and f1-score values of test data on facial expressions.

Expressions	Precision	Recall	F1-score
Happy	1	1	1
Angry	1	0,76	0,86
Normal	1	1	1
Sad	0,80	1	0,89
Fear	0,80	1	0,89

Surprised	1	0,91	0,95
-----------	---	------	------

Table 5 displays evaluation metrics, including precision, recall, and F1-score, for various facial expressions classified by the model. Precision represents the proportion of correctly predicted positive instances out of all predicted positive instances, while recall measures the proportion of true positive instances correctly identified by the model out of all actual positive instances. The F1-score balances both precision and recall.

The results show that the model achieved perfect precision for "Happy," "Normal," and "Surprised" facial expressions, indicating accurate positive predictions and high confidence in these classifications. Additionally, these classes demonstrated high recall values, implying that the model effectively captured most of the actual positive instances. As a result, the F1-scores were excellent, indicating the model's strong performance in accurately classifying these expressions.

However, for "Angry," "Sad," and "Fear" facial expressions, there were slight discrepancies between precision and recall, resulting in F1-scores slightly below 1.00. While precision was high for these classes, recall values suggested that the model missed some actual positive instances. Consequently, improvements in recall on these expressions could enhance the model's overall performance in emotion recognition.

b) Result of facial expression detection in video conferencing.

The testing process on video conferencing involved 10 samples whose facial expression data had been trained on the model for recognition. The system records and detects the participants' facial expressions through video conferencing, and categorizes them based on 6 types of expressions that have been previously defined.

The detection process in the system is done by recording the user's monitor screen which is then extracted into a collection of image frames. Each frame is then detected by the stored CNN model, resulting in detected frames. The detected frames, which indicate facial expressions, are then merged back into the video and stored in the local storage of the system device.

To make a final decision regarding the facial expressions displayed by the participants successfully captured by the system, five random frames were selected from the recorded video conference. Each frame was then analyzed to determine the expression of each participant. The decision results can be seen in Table 6.

Table 6. Results of real-time testing of politeness detection.

Person to-	Frame to-					Desc
	1	2	3	4	5	
1	S	S	S	ND	S	H: Happy
2	A	A	A	A	A	A: Angry
3	N	N	N	F	F	N: Normal
4	A	A	A	S	ND	S: Sad
5	S	S	S	N	N	F: Fear
6	F	F	ND	S	S	SU: Surprised
7	F	F	F	F	F	ND: Not Detected
8	A	ND	A	F	F	
9	S	S	S	A	F	
10	A	A	A	F	F	
11	S	F	F	F	F	

Table 6 shows the variation in facial expressions of several individuals in the five observed frames. Some people show consistent expressions in each frame, while others show changes in expression from frame to frame. There are also cases where facial expressions are not detected (labeled "ND"). Some expressions appear more frequently than others, and there are individuals who show different expressions in the same situation. Factors that affect the system's frame limit in detecting participants' facial expressions at once include the number of participants, participants' movements, participants' image quality, and the duration of the video recording.

4. CONCLUSION

Based on the research conducted to detect participants' facial expressions during video conferences using the Convolutional Neural Network (CNN) algorithm, the applied CNN architecture achieved a high training

accuracy of 97.5%. The architecture consists of 2 Conv2D layers, 3 BatchNormalization layers, 2 MaxPooling layers, 2 dropout layers, 1 flatten layer, 1 dense layer, and 1 output layer. The Adam optimizer with a learning rate of 0.0001 was used, and the batch size was 64. The CNN model successfully recognized facial expressions based on pre-trained images, with 97,5% accuracy on the training data and 93,33% accuracy on the test data.

The research contributes to character education in online learning environments by utilizing facial expression recognition to gain insights into participants' engagement during video conferencing. The findings emphasize the importance of character education and offer potential solutions to improve participant response and engagement in online learning platforms. The research opens up opportunities for further advancements in using facial expression analysis as a tool for enhancing online communication and character development.

REFERENCES

- [1] L. Lina, A. A. Marunduh, W. Wasino, and D. Ajiengoro, "Identifikasi emosi pengguna konferensi video Menggunakan Convolutional Neural Network," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 5, p. 1047, 2022. doi:10.25126/jtiik.2022955269.
- [2] N. Ekawardhana, "Efektivitas pembelajaran dengan menggunakan media video conference," *Seminar Nasional Ilmu Terapan*, vol. 4, no. 1, 2020.
- [3] M. H. Dra. Rosnawati, "Pemulihan Karakter siswa pasca pembelajaran daring," *Gurusiana*, <https://www.gurusiana.id/read/drarnosnawatimhum/article/pemulihan-4869238> (accessed Jul. 6, 2023).
- [4] N. Zuriah, *Pendidikan Moral & Budi Pekerti Dalam Perspektif Perubahan: Menggagas Platform Pendidikan Budi Pekerti Secara Kontekstual Dan Futuristik*. Jakarta: Bumi Aksara, 2007.
- [5] S. S. Kulkarni, "Facial image based mood recognition using committee neural networks", thesis, 2006.
- [6] T. T. R. ZHENG, *Artificial Intelligence with Python*. S.l.: SPRINGER VERLAG, SINGAPOR, 2023.
- [7] N. G. Paterakis, E. Mocanu, M. Gibescu, B. Stappers and W. van Alst, "Deep learning versus traditional machine learning methods for aggregated energy demand prediction," 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Turin, Italy, 2017, pp. 1-6, doi: 10.1109/ISGTEurope.2017.8260289.
- [8] Yusuf, A. Wihandika, R. C. Dewi, and Candra, "Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 3, no. 11, 2020.
- [9] M. B. Nendya, L. Husniah, H. Wibowo, and E. M. Yuniarno, "Sintesa Ekspresi Wajah Karakter Virtual 3D Menggunakan action unit berbasis facial action coding system (FACS)," *Journal of Animation and Games Studies*, vol. 7, no. 1, pp. 13–24, 2021. doi:10.24821/jags.v7i1.4239.
- [10] T. D. Bui, *Creating Emotions and Facial Expressions for Embodied Agents*. Enschede, University of Twente: Taaluitgeverij Neslia Paniculata, 2004.
- [11] A. T. Lopes, E. de Aguiar, A. F. De Souza, and T. Oliveira-Santos, "Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order," *Pattern Recognition*, vol. 61, pp. 610–628, 2017. doi:10.1016/j.patcog.2016.07.026.
- [12] K. Liu, M. Zhang, and Z. Pan, "Facial expression recognition with CNN ensemble," 2016 International Conference on Cyberworlds (CW), 2016. doi:10.1109/cw.2016.34.