Journal of Soft Computing Exploration

Homepage: htttps://shmpublisher.com/index.php/joscex

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Adaptive deep learning based on FaceNet convolutional neural network for facial expression recognition

Maulana Malik Ibrahim Al-Ghiffary[1], Nur Ryan Dwi Cahyo[2], Eko Hari Rachmawanto[3], Candra Irawan[4], Novi Hendriyanto[5]

[1,2,3,5]Study Program in Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
[4]Study Program in Information System, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia

Article Info

Article history:

Received Aug 15, 2024
Revised Sep 16, 2024
Accepted Sep 19, 2024

Keywords:

Facial expression
Adaptive deep learning
FaceNet convolutional neural network
Image recognition

ABSTRACT

Facial recognition technology has become increasingly crucial in various applications, from personal identification, security, and human-care. Facial recognition has numerous practical applications, ranging from assessing mental health and well-being through facial expressions to evaluating customer satisfaction in service quality ratings. This study aims to develop a facial recognition model using a Convolutional Neural Network (CNN) with FaceNet architecture. The proposed method utilizes an advanced deep learning approach to generate high-quality facial embeddings, enhancing the model's ability to accurately identify and verify individuals. Our methodology includes training the CNN with FaceNet architecture, achieving an impressive average accuracy of 99.93%, with precision, recall, and F1-score all reaching 100%. The model demonstrated both high accuracy and efficiency, with an average training time of 13 minutes and 51 seconds. Future research should explore incorporating data augmentation, K-fold cross-validation, and additional transfer learning techniques to further enhance model performance and generalization. These advancements could lead to more resilient and flexible facial recognition systems capable of functioning effectively in diverse and challenging real-world conditions.

Corresponding Author:

Nur Ryan Dwi Cahyo,
Faculty of Computer Science,
Universitas Dian Nuswantoro,
Semarang, Indonesia
Email: ryanwaver23@gmail.com
https://doi.org/10.52465/joscex.v5i3.450

1. INTRODUCTION

In communications or interactions between people, facial expression plays a huge role in understanding the communication, facial expression itself is a form of non-verbal communication that is created by one or more facial muscle movements, facial expression can be used to convey feelings of the communications participant since 93% of common communication relied on a person's feeling, facial expression also contains 55% of information while sound and language only contains 7% and 38% respectively, therefore facial expression can be used to interpret the information substantially and maintain a good flow in a discussion or communications between people [1], [2], [3] . On the other hand, facial expression recognition can be used in several fields for example it can be used as an indicator for satisfactory level of a service or a

product, this opens up opportunity for a business to increase their customer's loyalty by understanding their satisfactory level through facial expression [4].

Facial expression recognition can also be used in the education field where it can be used to recognize how students behave and act while following the lesson taught in class, by recognizing the student's facial expression future teaching method can be perfected by analyzing whether the student is attracted or distracted with current teaching method. Not only in the educational fields, it can also be used in medical fields where facial expression recognition can be used to analyze the condition of a patient and help medical workers determine suitable treatment efficiently [5]. In other words, facial recognition expression has many utilizations in various fields and it has become one of the most popular research topics.

Facial expression classification can be categorized as facial expression recognition, recently computer vision technology has been utilizing deep learning methodology as it shows great performance in terms of classification, identification, and target detection [3]. Convolution Neural Network (CNN) [6] as one of the methodologies used in deep learning, is able to extract image features accurately and since then has been used in different areas of computer vision [3]. CNN is able to learn and neatly combine image features automatically without any manual selection, therefore CNN is considered to perform greatly in feature extraction of facial expression recognition especially for the expressions of sadness, fear, and contempt [7]. Several previous studies and research have been conducted regarding the use of Convolution Neural Network for facial expression recognition and or classification, the following paragraph has presented related works.

Research [2] is performing facial expression recognition using the Convolution Neural Network algorithm with a dataset containing 28,821 of image data divided into seven classes of expression being angry, disgust, fear, happy, neutral, sad, and surprise class, the result obtained shows that the proposed method is able to achieve validation score of accuracy, precision, and recall of 65%, while it also achieves training score of 90%. Research [3] is performing facial expression recognition using a Convolution Neural Network with a modified activation layer called LS-ReLu, the dataset used is the FER2013 publicly available dataset, the analysis reveals that the proposed technique reaches an average accuracy of 90.16%.

Research [4] is performing facial expression recognition using MobileNet-V2 architecture, the dataset used contains 22,619 of image data divided into six classes of expression being angry, fear, happy, neutral, sad, and surprise class, the result obtained shows that MobileNet-V2 architecture is able to achieve training accuracy of 100% and validation accuracy of 40%. Research [5] is performing facial expression recognition using modified VGG-19 architecture where the architecture's network depth is increased, the result obtained shows that the proposed approach is capable of attaining an accuracy score of 96%.

Research [7] is performing facial expression recognition using the Convolution Neural Network algorithm while performing image alignment and cropping in the pre-processing stage, the dataset used contains 10,708 image data and 327 video sequences that are divided into seven classes of expression anger, contempt, disgust, fear, happiness, sadness, and surprise, the analysis shows that the proposed technique is capable of reaching an accuracy score of 97.38%.

Based on the presented literature review, Convolutional Neural Network or CNN is considered to work well with image data classification, therefore it can be used to perform facial expression recognition with image data. This research's primary objective is to develop a facial expression recognition system using CNN algorithm with FaceNet architecture, as facial recognition has many possible implementation in the real world scenario, for example it can be used in the medical field especially mental-health and well-being treatment where facial expressions is able to display the patient's current condition, another example is it can be used in customer service field where facial expressions can indicate customer satisfaction and frustation upon receiving services. It is believed that this research can be used as a proof of concept in various fields that include facial expression recognition, therefore this research can contribute in several fields as previously mentioned. The classification is categorized into seven distinct classes: angry, disgusted, fear, happy, neutral, sad, and surprised. The dataset employed in this study is a public dataset featuring 28,821 images organized into seven different classes of facial expressions.

2. METHOD

This section begins with the Data Collection phase, where the dataset has undergone data augmentation for training and evaluation. Following data acquisition, the dataset is processed through a Batch Layer, which organizes the data into manageable subsets to facilitate efficient training and reduce computational overhead. Subsequently, the data is subjected to L2 Normalization within a deep learning architecture. This step ensures that the features are standardized, contributing to the stability and convergence of the model during training. Once normalized, the data proceeds to the Embedding Face Representation stage, where a deep neural network generates compact and discriminative feature vectors that encapsulate the essential characteristics of each facial expression. The Triplet Loss function is then employed to refine the

embeddings further by minimizing the distance between similar facial expressions and maximizing the distance between dissimilar ones, thus enhancing the model's capability to distinguish between different expressions effectively. Last phase, the model's performance is assessed by calculating the Percentage Accuracy (%), providing a quantitative measure of its effectiveness in recognizing facial expressions. This metric serves as a critical indicator of the success of the proposed method in achieving accurate and reliable facial expression recognition. The approach to research methodology flow is illustrated in Figure 1.

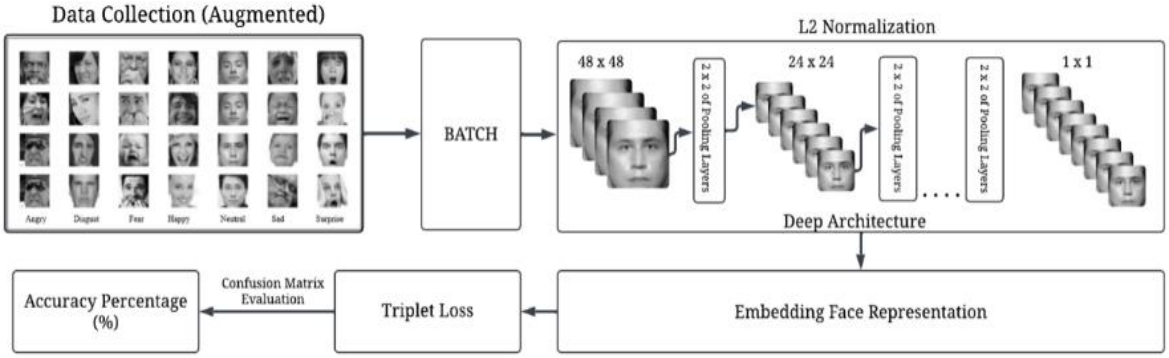

Figure 1. Research Methodology

Data Collection

The dataset utilized in this research is a public dataset available at the Kaggle site and can be accessed through this link https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset. This dataset contains a total of 30,487 image data that is divided into seven expression classes, each class has its own portioning for training and validation at 80% and 20% for each classes respectively. The seven classes and their proportion are shown in the following table, the training and validation data amount is already portioned as mentioned previously.

Table 1. Dataset Distribution and Portioning

No.	Class	Training Data Amount	Validation Data Amount
1.	Angry	3,993	960
2.	Disgust	436	111
3.	Fear	4,103	1,018
4.	Happy	7,164	1,825
5.	Neutral	4,982	1,216
6.	Sad	4,983	1,139
7.	Surprise	3,205	797

The image data in this dataset are uncolored or appears in grayscale color, and the resolution for each image data is 48 pixel * 48 pixel. The sample images taken from the dataset are displayed in Figure 2.

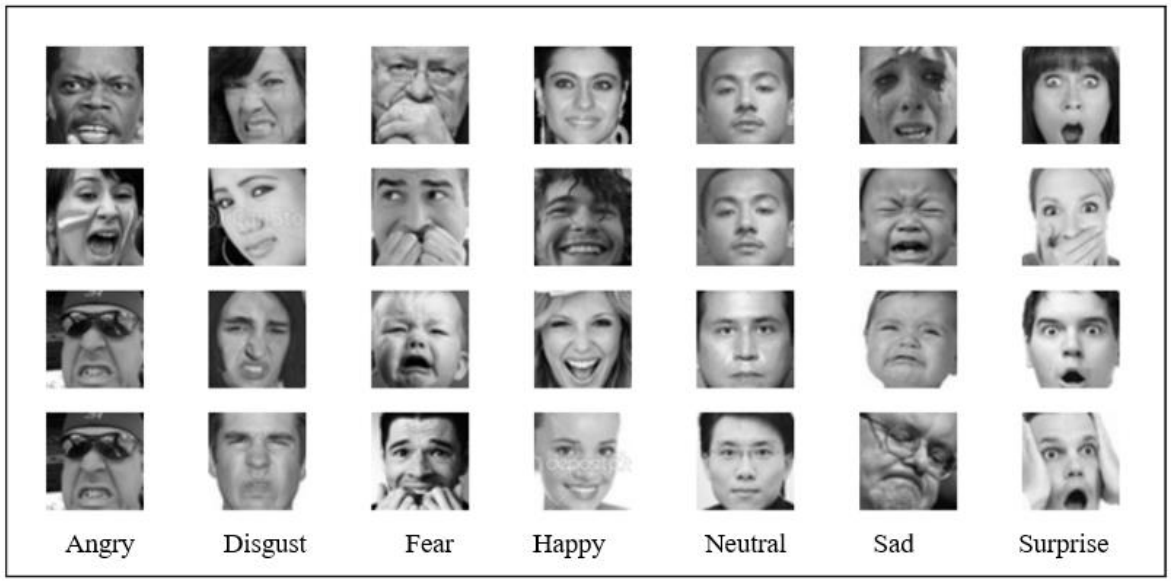

Figure 2. Sample Datasets Each Class

Data Augmentation

In this study, data augmentation [8], [9] methods such as rotation, cropping, and X and Y transformations were employed to effectively double the size of the training dataset. The original dataset comprised 3,993 images for the "Angry" class, 436 for "Disgust," 4,103 for "Fear," 7,164 for "Happy," 4,982 for "Neutral," 4,983 for "Sad," and 3,205 for "Surprise." After applying data augmentation techniques such as rotation, cropping, and X and Y transformations, the dataset for each class was effectively doubled. This resulted in 7,986 images for "Angry," 872 for "Disgust," 8,206 for "Fear," 14,328 for "Happy," 9,964 for "Neutral," 9,966 for "Sad," and 6,410 for "Surprise." This expansion of the dataset enhances the model's ability to learn from a more diverse set of examples, improving its performance and robustness in facial expression recognition tasks [10], [11]. Based on data augmentation algorithm can be seen in Table 2.

Table 2. Data Augmentation Algorithm

1st Algorithm: Augmentation of Datasets

```
Input = path_of_datasets
function augment_dataset (original_dataset):
    augmented_dataset = []
    for each image in original_dataset:
        augmented_dataset.add (image)
        # 1. Rotation Augmentation
        rotated_image = rotate_image (image, angle = random_angle())
        augmented_dataset.add(rotated_image)
        # 2. Cropping Augmentation
        cropped_image = crop_image(image, crop_size = random_crop_size())
        augmented_dataset.add (cropped_image)
        # 3. X and Y Transformations
        transformed_image_X = transform_image_X (image, shift = random_shift ())
        augmented_dataset.add(transformed_image_X)
        transformed_image_Y = transform_image_Y (image, shift = random_shift ())
        augmented_dataset.add (transformed_image_Y)
        # Optionally combine augmentations (e.g., rotated and cropped)
        combined_image = rotate_image (crop_image (image, crop_size = random_crop_size ()), angle = random_angle ())
        augmented_dataset.add(combined_image)
    end
    # Ensure the dataset is doubled
    while len (augmented_dataset) < 2 * len(original_dataset):
        augmented_dataset.add(random_augmentation(image))
    return augmented_dataset
end
```

The functions $augment\_dataset$ perform specific augmentations on the input of path image, such as rotating, cropping, and shifting the image in the X or Y direction. Helper functions like $random\_angle$, $random\_crop\_size$, and $random\_shift$ generate random values for these augmentation parameters to introduce variability. The augment_dataset function serves as the main process, applying these augmentations to the original dataset and ensuring that the dataset size is effectively doubled.

Convolutional Neural Networks

CNN [12] is based on the development of Artificial Neural Network, CNN is a neural network that comprises convolutional layers, pooling layers, and fully connected layers [13], [14], [15]. CNN utilizes its specific kernel in order to extract features from an image, later a process named activation process is performed in order to preserve positive values while negative values are mapped into 0 value [16], [17], [18], then in the pooling layer a process named map reduction process is performed in order to reduce parameters that needs to be learned by the network, and finally in the fully connected layer K dimension is obtained where K is the predictable classes [14], [19].

FaceNet Architecture

FaceNet [20], [21], [22] is an architecture based on the Deep Convolution Neural Network (DCNN) developed by Google in order to integrate facial detection and facial recognition into one framework. FaceNet works by mapping a person's face into and Ecludian spaces, where Ecludian spaces itself is a collection of geometrical points. This mapping process is also called "the embedding process" in FaceNet, it works by measuring the length of spatial distance between different faces that directly corresponds to the similarity between faces, this spatial distance will get closer if one face is similar to others and vice versa [12], [13]. Further explanation regarding FaceNet architecture is explained in the following Figure 3 and 4.

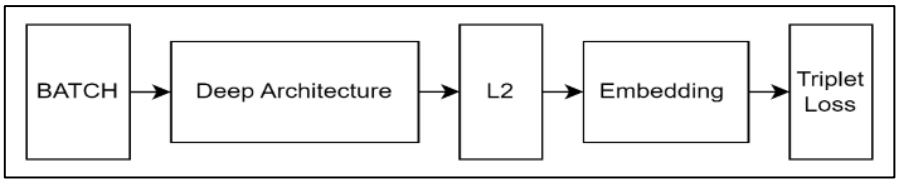

Figure 3. FaceNet Architecture

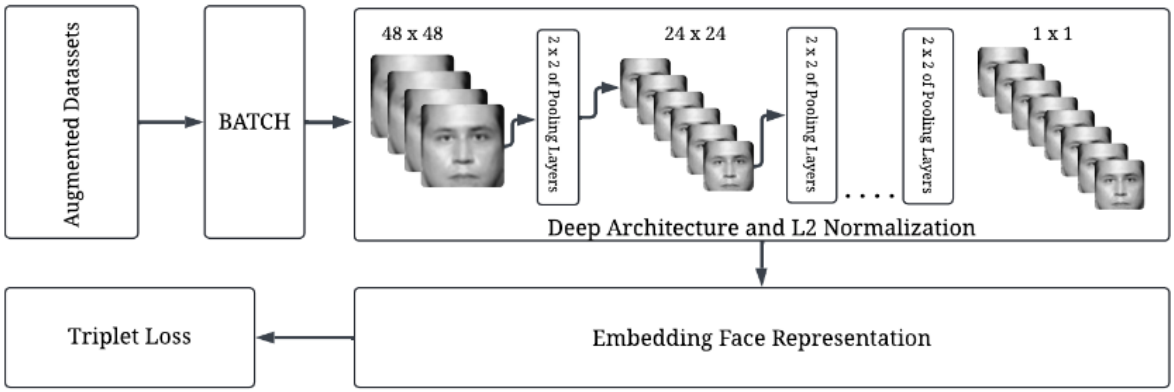

Figure 4. Specific of FaceNet Architecture

FaceNet architecture consist of batch input layer, Deep Convolution Neural Network or DCNN, L2 normalization, the embedding process, and triplet loss calculation. In this architecture, DCNN is used to learn the ecludian spaces of each image and perform training process of the network, later the squared L2 is a normalization process which followed by the embedding process where the embedding space corresponds directly to the similarity between faces, and finally the triplet loss function is used to train the neural network resulting in 128-dimensional vector spaces, this loss function also resulted in making the distance of similar images closer and vice versa [23]. FaceNet layer algorithm can be seen in Table 3.

Table 3. FaceNet Layer Algorithm

2nd Algorithm: FaceNet Layer Algorithm
Input Layer: Input Image: Resize to $48x48\ pixels$. Convolutional Layers: Apply several convolutional layers to extract low-level features. Follow each conv layer with ReLU activation. Apply Batch Normalization after each conv layer. Inception Modules: Use multiple Inception Modules to capture multi-scale features. Each module includes: $1x1, 3x3, and\ 5x5\ convolutions$. Max-pooling layer. Concatenate the outputs of the module. Pooling Layers: Apply Max-Pooling to reduce spatial dimensions. Fully Connected Layers: Flatten the output from the convolutional layers. Pass through fully connected layers. Embedding Layer: Output a low-dimensional embedding vector ($size\ 128\ or\ 512$). L2 Normalization: Normalize the embedding using L2 normalization. Triplet Loss Layer: Apply Triplet Loss during training: Ensure $distance\ (anchor, positive) < distance\ (anchor, negative) + margin.$ Output: Produce the normalized embedding vector. Use embedding for face recognition/verification tasks.

Confusion Matrix

Confusion matrix [24], [25], [26], [27] is employed as a measuring tool for Convolution Neural Network classification model, it includes four primary elements True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP is where the model correctly predicted the positive classes, TN is where the model correctly predicted the negative classes, while FP and FN are where model falsely

predicted the positive class into the negative class and vice versa [15]. Based on these main components, the following equation is used to calculate accuracy, precision, recall, and F1-score as the evaluation metrics of the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

$$F1 - score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (4)$$

3. RESULTS AND DISCUSSIONS

The research commenced with the training phase, utilizing a configuration that included 8 epochs, the Adam, SGD, RMSP optimizer, a learning rate of 0.0001, and a validation frequency of 30 [28], [29], [30]. The training was executed in MATLAB 2021a, which is integrated with Python for enhanced computational capabilities. The resulting training graph, as depicted in Figure 5, illustrates the model's learning curve across the specified epochs. Following the training phase, the model was subjected to testing, with the performance evaluation based on the Confusion Matrix. This matrix provides a detailed view of the classification results, highlighting the accuracy and potential misclassifications. The outcomes of this testing phase are presented in Figure 6, providing a thorough insight into the model's performance in distinguishing between the various classes.

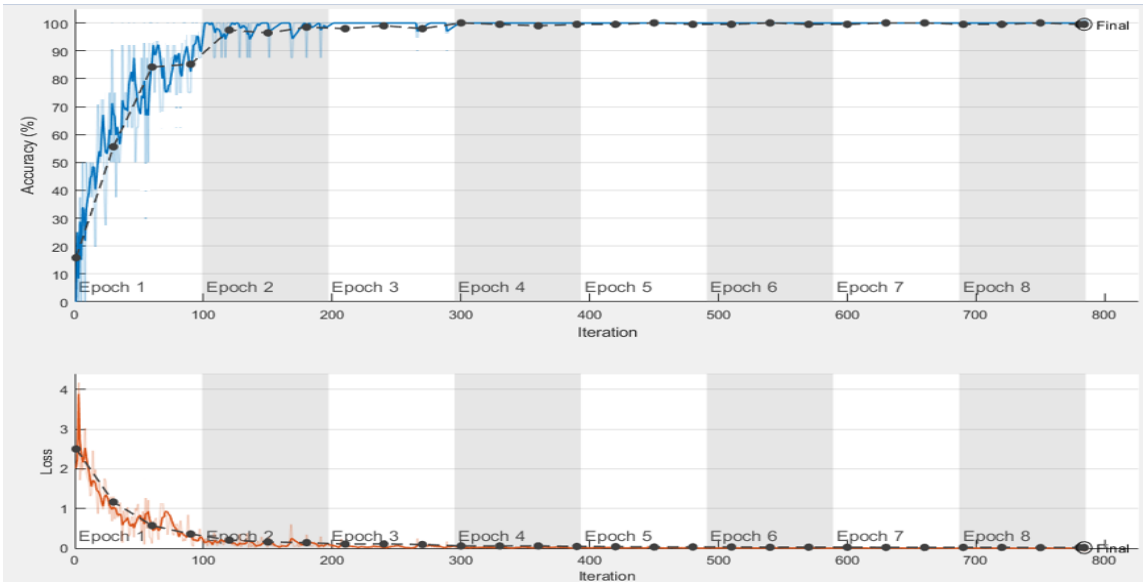

Figure 5. Training Graph

The training graph depicted in Figure 5 demonstrates the model's learning progression throughout the training phase. Based on the analysis of this graph, the evaluation using the confusion matrix was conducted, yielding results that align with the performance metrics detailed in Table 4. These results provide insight into the model's accuracy in classification and the distribution of correctly and incorrectly predicted instances across the various categories. Based on Figure 5, an experiment graph was carried out using the Adam Optimizer which obtained an accuracy result of 99.96%.

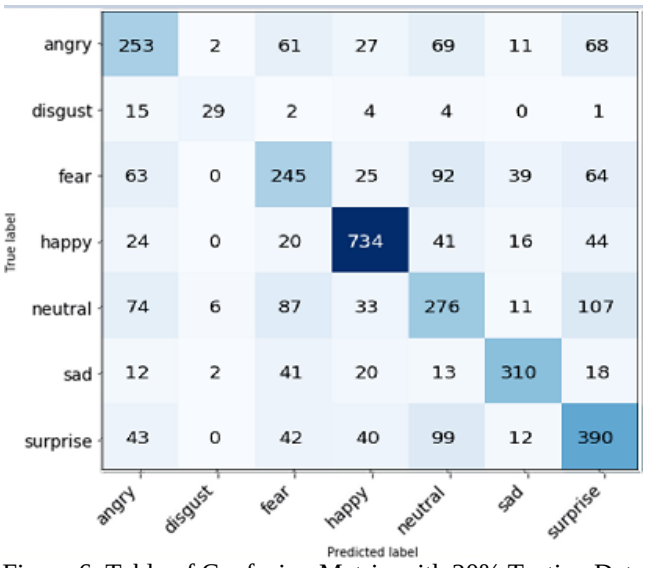

Figure 6. Table of Confusion Matrix with 20% Testing Data

Table 4. Confusion Matrix Evaluation

Training Model	Accuracy	Precision	Recall	F1-Score	Elapsed Time
1st Training Adam (Best)	99.96%	100%	100%	100%	16 Minutes 14 Sec
2nd Training SGD	99.91%	100%	100%	100%	12 Minutes 32 Sec
3rd Training RMSP	99.92%	100%	100%	100%	12 Minutes 47 Sec
Average	**99.93%**	**100%**	**100%**	**100%**	**13 Minutes 51 Sec**

The evaluation results presented in Table 4 demonstrate exceptional performance in terms of accuracy when applying the FaceNet architecture on the CNN layers. The layer parameters used in this experiment is described in the table 5.

Table 5. Layer Parameters

Layer Parameter	Description
Image Input Layer	Image size of 48 x 48 pixels
Convolutional Layer	Filter size of 7 x 7 pixels, with 64 filters in total; Stride value of 2 and padding value of 3
Pooling Layer	Max Pooling Layer over 3 x 3 region; Stride value of 2
Fully Connected Layer	L2 Normalization layer

In addition, the training parameters used in this experiment consist of epoch, batch size, Adam, SGD, and RMSP optimizer, learning rate, and validation frequency. These parameters were obtained through trial and error process of experimenting with several different values parameters, the following are the example of trial and error experiments.

Table 6. Trial and Error Experiments

Trial No.	Epoch	Batch Size	Optimizer	Learning Rate	Val. Frequency	Result (Accuracy)
Trial 1	8	16	Adam	0.0001	30	**99.96% (Best)**
Trial 2	16	32	Adam	0.0001	30	**99.84%**
Trial 3	16	32	SGD	0.0001	30	**99.75%**
Trial 4	16	32	RMSP	0.0001	30	**99.70%**

As shown in the table, the best parameters of this experiment were epoch value of 8 epochs, batch size value of 16, the Adam, SGD, RMSP optimizer, a learning rate of 0.0001, and a validation frequency of 30, therefore the training models achieved nearly perfect accuracy, with individual training sessions yielding accuracies of 99.96%, 99.91%, and 99.92%. The average accuracy across these sessions is an impressive 99.93%, reflecting the robustness of the FaceNet architecture. Additionally, the precision, recall, and F1-score consistently reached 100% across all training sessions, indicating flawless classification performance. The average elapsed time for training was 13 minutes and 51 seconds, showcasing an efficient balance between accuracy and computational efficiency. Although other mentioned studies do not address training time, it is a crucial aspect. A quick training time indicates efficiency in resource management, especially when computational resources are limited and highly complex models may not run effectively. Rapid training combined with high performance suggests the model can perform well even in challenging scenarios, such as those with limited computational resources.

Table 7. Comparison Results based on Related Study

Researcher	Model	Accuracy
Our	**CNN with FaceNet Architecture**	**99.93% (AVR)**
[2]	Classic CNN	90%
[3]	CNN with a modified activation layer ReLu	90.16%
[4]	Mobilenet-v2	100% Accuracy and 40% Validation (High Overfitting)
[5]	CNN with VGG-19	96%
[7]	CNN with Augmentation Pre-processing	97.38%

The outcomes displayed in Table 7 highlight the effectiveness of the proposed CNN model with FaceNet architecture, which achieved an average accuracy of 99.93%. This significantly outperforms other models in the comparison, demonstrating its effectiveness in facial recognition tasks. The superior accuracy of our model reflects the robustness and efficiency of the FaceNet architecture in capturing and representing facial features. In contrast, models such as the Classic CNN and CNN with a modified ReLU activation layer reported accuracies of 90% and 90.16%, respectively, indicating that enhancements like the FaceNet architecture contribute to substantial improvements in performance.

However, it is notable that the Mobilenet-v2 model, despite achieving a 100% accuracy, exhibited a high level of overfitting with only 40% validation accuracy. This suggests that while the model performs flawlessly on the training set, it struggles to generalize to new data, which is a critical issue in practical applications. On the other hand, CNN models with VGG-19 and augmentation pre-processing achieved accuracies of 96% and 97.38%, respectively, showing that these methods also provide strong performance but do not reach the level of accuracy offered by the FaceNet-based approach. Overall, the FaceNet architecture's exceptional accuracy and generalization capability underscore its superiority in facial recognition tasks compared to the other models evaluated. FaceNet integrates facial detection and recognition into a single framework by mapping facial images into Euclidean space. This results in geometrical points that can measure distinct facial features between expressions. FaceNet's advantage in recognizing facial expressions lies in its "Facial Embeddings" feature, making it ideal for facial expression recognition. According to the proposed method, the results of the augmentation technique of each class can be seen in Figure 7.

(a) Original Image

(b) Augmented Image

Figure 7. Results of Augmented Image

This technique involves two primary transformations applied sequentially to an image. First, the image is rotated by 45 degrees around its center. This rotation alters the orientation of the image while keeping its dimensions unchanged. Following the rotation, a translation is applied to the already rotated image. The translation shifts the image 5 pixels to the right and 10 pixels downward. This combined process results in an image that not only changes its orientation but also its position within the coordinate space. The final output, therefore, showcases the effect of these transformations, with the image appearing rotated and offset from its original position.

4. CONCLUSION

In this study, researcher utilized a Convolutional Neural Network (CNN) with the FaceNet architecture, which introduced significant advancements in facial recognition technology. The method achieved an outstanding average accuracy of 99.93%, with precision, recall, and F1-score all reaching a perfect 100% across various training models—Adam, SGD, and RMSProp. Training times were efficiently managed, with durations of 16 minutes and 14 seconds for Adam, 12 minutes and 32 seconds for SGD, and 12 minutes and 47 seconds for RMSProp, resulting in an average training time of 13 minutes and 51 seconds. This high level of performance demonstrates the model's exceptional ability to accurately and swiftly process facial

images. The novelty of the FaceNet-based CNN approach lies in its combination of high accuracy, precision, and computational efficiency. This model offers substantial benefits to society by enhancing security systems, access control, and personalized services through reliable and rapid facial recognition. The advancements achieved with this method promise improved public safety, streamlined user experiences, and broader applicability in various real-world scenarios where accurate identity verification is critical.

From an algorithmic perspective, future research could benefit from incorporating data augmentation techniques to enhance the diversity and robustness of the training dataset. Implementing k-fold cross-validation could further improve the model's reliability and generalizability by ensuring a more comprehensive evaluation that's less dependent on a single validation set. Additionally, integrating other transfer learning methods could leverage pre-trained models from larger datasets to potentially boost performance and reduce training time. From a dataset and methodological standpoint, future research could improve robustness by enhancing datasets to include facial expressions with object occlusions. This would enable the model to recognize expressions even when parts of the face are obscured. Involving other modalities or sensory information—such as body language and gestures—alongside facial recognition could lead to more accurate and context-aware systems for both facial and emotion recognition.

REFERENCES

[1] K. Sarvakar, R. Senkamalavalli, S. Raghavendra, J. Santosh Kumar, R. Manjunath, and S. Jaiswal, "Facial emotion recognition using convolutional neural networks," *Mater Today Proc*, vol. 80, pp. 3560–3564, Jan. 2023, doi: 10.1016/j.matpr.2021.07.297.

[2] P. Adi Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *JURNAL ALGOR*, vol. 2, no. 1, 2020, [Online]. Available: https://jurnal.buddhidharma.ac.id/index.php/algor/index

[3] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences (Switzerland)*, vol. 10, no. 5, Mar. 2020, doi: 10.3390/app10051897.

[4] R. Steven Immanuel Sihombing *et al.*, "Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network (CNN)," *Journal of Creative Student Research (JCSR)*, vol. 1, no. 6, pp. 89–97, 2023, doi: 10.55606/jcsrpolitama.v1i6.3046.

[5] S. Cheng and G. Zhou, "Facial Expression Recognition Method Based on Improved VGG Convolutional Neural Network," *Intern J Pattern Recognit Artif Intell*, vol. 34, no. 7, Jun. 2020, doi: 10.1142/S0218001420560030.

[6] A. Susanto, C. A. Sari, E. H. Rachmawanto, I. U. W. Mulyono, and N. Mohd Yaacob, "A Comparative Study of Javanese Script Classification with GoogleNet, DenseNet, ResNet, VGG16 and VGG19," *Scientific Journal of Informatics*, vol. 11, no. 1, pp. 31–40, Jan. 2024, doi: 10.15294/sji.v11i1.47305.

[7] K. Li, Y. Jin, M. W. Akram, R. Han, and J. Chen, "Facial expression recognition with convolutional neural networks via a new face cropping and rotation strategy," *Vis Comput*, vol. 36, no. 2, pp. 391–404, Feb. 2020, doi: 10.1007/s00371-019-01627-4.

[8] N. P. Sutramiani, N. Suciati, and D. Siahaan, "MAT-AGCA: Multi Augmentation Technique on small dataset for Balinese character recognition using Convolutional Neural Network," *ICT Express*, vol. 7, no. 4, pp. 521–529, Dec. 2021, doi: 10.1016/j.icte.2021.04.005.

[9] Q. A. Putra, C. A. Sari, E. H. Rachmawanto, N. R. D. Cahyo, E. Mulyanto, and M. A. Alkhafaji, "White Bread Mold Detection using K-Means Clustering Based on Grey Level Co-Occurrence Matrix and Region of Interest," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2023, pp. 376–381. doi: 10.1109/iSemantic59612.2023.10295369.

[10] F. J. Moreno-Barea, J. M. Jerez, and L. Franco, "Improving classification accuracy using data augmentation on small data sets," *Expert Syst Appl*, vol. 161, p. 113696, 2020.

[11] X. Jiang and Z. Ge, "Data augmentation classifier for imbalanced fault classification," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1206–1217, 2020.

[12] E. H. Rachmawanto and P. N. Andono, "Deteksi Karakter Hiragana Menggunakan Metode Convolutional Neural Network," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 183–191, Dec. 2022, doi: 10.23887/janapati.v11i3.50144.

[13] N. R. D. Cahyo, C. A. Sari, E. H. Rachmawanto, C. Jatmoko, R. R. A. Al-Jawry, and M. A. Alkhafaji, "A Comparison of Multi Class Support Vector Machine vs Deep Convolutional Neural Network for Brain Tumor Classification," in *2023 International Seminar on Application for Technology of Information and Communication (iSemantic)*, IEEE, Sep. 2023, pp. 358–363. doi: 10.1109/iSemantic59612.2023.10295336.

[14] M. M. I. Al-Ghiffary, C. A. Sari, E. H. Rachmawanto, N. M. Yacoob, N. R. D. Cahyo, and R. R. Ali, "Milkfish Freshness Classification Using Convolutional Neural Networks Based on Resnet50

Architecture," *Advance Sustainable Science Engineering and Technology*, vol. 5, no. 3, p. 0230304, Oct. 2023, doi: 10.26877/asset.v5i3.17017.

[15] I. P. Kamila, C. A. Sari, E. H. Rachmawanto, and N. R. D. Cahyo, "A Good Evaluation Based on Confusion Matrix for Lung Diseases Classification using Convolutional Neural Networks," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, p. 0240102, Dec. 2023, doi: 10.26877/asset.v6i1.17330.

[16] A. Ziaee and E. Çano, "Batch Layer Normalization A new normalization layer for CNNs and RNNs," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Oct. 2022, pp. 40–49. doi: 10.1145/3571560.3571566.

[17] Z. A. Sejuti and M. S. Islam, "An Efficient Method to Classify Brain Tumor using CNN and SVM," in *International Conference on Robotics, Electrical and Signal Processing Techniques*, 2021, pp. 644–648. doi: 10.1109/ICREST51555.2021.9331060.

[18] E. Z. Astuti, C. A. Sari, M. Syabilla, H. Sutrisno, E. H. Rachmawanto, and M. Doheir, "Capital Optical Character Recognition Using Neural Network Based on Gaussian Filter," *Scientific Journal of Informatics*, vol. 10, no. 3, pp. 261–270, Jul. 2023, doi: 10.15294/sji.v10i3.43438.

[19] Y. A. Nisa, C. A. Sari, E. H. Rachmawanto, and N. Mohd Yaacob, "Ambon Banana Maturity Classification Based On Convolutional Neural Network (CNN)," *sinkron*, vol. 8, no. 4, pp. 2568–2578, Oct. 2023, doi: 10.33395/sinkron.v8i4.12961.

[20] C. Wu and Y. Zhang, "MTCNN and FACENET based access control system for face detection and recognition," *Automatic Control and Computer Sciences*, vol. 55, pp. 102–112, 2021.

[21] F. Cahyono, W. Wirawan, and R. F. Rachmadi, "Face recognition system using facenet algorithm for employee presence," in *2020 4th international conference on vocational education and training (ICOVET)*, IEEE, 2020, pp. 57–62.

[22] S. Srinivas and M. P. Selvan, "E-CNN-FFE: An Enhanced Convolutional Neural Network for Facial Feature Extraction and Its Comparative Analysis with FaceNet, DeepID, and LBPH Methods," in *International Conference on Data Management, Analytics & Innovation*, Springer, 2024, pp. 339–354.

[23] X. Xu, M. Du, H. Guo, J. Chang, and X. Zhao, "Lightweight FaceNet Based on MobileNet," *Int J Intell Sci*, vol. 11, no. 01, pp. 1–16, 2021, doi: 10.4236/ijis.2021.111001.

[24] A. Theissler, M. Thomas, M. Burch, and F. Gerschner, "ConfusionVis: Comparative evaluation and selection of multi-class classifiers based on confusion matrices," *Knowl Based Syst*, vol. 247, Jul. 2022, doi: 10.1016/j.knosys.2022.108651.

[25] E. Oktayessofa, C. A. Sari, E. H. Rachmawanto, and N. M. Yaacob, "CLASSIFICATION OF ORGANIC AND NON-ORGANIC WASTE WITH CNN-MOBILENET-V2," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 4, pp. 1173–1180, 2024, doi: 10.52436/1.jutif.2024.5.4.2165.

[26] M. Dolla Meitantya, C. Atika Sari, E. Hari Rachmawanto, and R. Raad Ali, "VGG-16 Architecture on CNN For American Sign Language Classification," vol. 5, no. 4, pp. 1165–1171, 2160, doi: 10.52436/1.jutif.2024.5.4.2160.

[27] M. Zulhusni, C. A. Sari, and E. H. Rachmawanto, "Implementation of DenseNet121 Architecture for Waste Type Classification," *Advance Sustainable Science Engineering and Technology*, vol. 6, no. 3, p. 02403015, Jul. 2024, doi: 10.26877/asset.v6i3.673.

[28] D. Asif, M. Bibi, M. S. Arif, and A. Mukheimer, "Enhancing Heart Disease Prediction through Ensemble Learning Techniques with Hyperparameter Optimization," *Algorithms*, vol. 16, no. 6, Jun. 2023, doi: 10.3390/a16060308.

[29] H. K. Ravikiran, J. Jayanth, M. S. Sathisha, and K. Bindu, "Optimizing Sheep Breed Classification with Bat Algorithm-Tuned CNN Hyperparameters," *SN Comput Sci*, vol. 5, no. 2, Feb. 2024, doi: 10.1007/s42979-023-02544-z.

[30] N. R. D. Cahyo and M. M. I. Al-Ghiffary, "An Image Processing Study: Image Enhancement, Image Segmentation, and Image Classification using Milkfish Freshness Images," *IJECAR) International Journal of Engineering Computing Advanced Research*, vol. 1, no. 1, pp. 11–22, 2024.