ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240914001990
Original scientific paper

# Comparative Analysis of CNN Architectures for Eight-Class Facial Expression Recognition: A Performance and Error Pattern Study

Kyoungjong PARK

**Abstract:** This paper presents a systematic evaluation of deep learning architectures for facial expression recognition, focusing on improving recognition accuracy through advanced CNN models. This paper investigates three different architectures: Conv2D with Max Pooling (M1), Conv2D with Max Pooling & Dropout (M2), and EfficientNet-B0 (M3), and examines their effectiveness in recognizing eight different facial expressions (Anger, Content, Disgust, Fear, Happiness, Neutral, Sadness, and Surprise). The experimental framework uses the Tsinghua facial expression database, which has a baseline recognition rate of 79.08% by human evaluators. The study yields several significant findings through rigorous comparative analysis using standardized metrics, such as accuracy measurements and confusion matrices. The EfficientNet-B0 model achieves superior performance with an average accuracy of 86.47%, while Conv2D with Max Pooling demonstrates robust performance at 81.68%, both exceeding the accuracy of human evaluators. Notably, the Conv2D with Max Pooling & Dropout model shows reduced effectiveness at 73.25%. Heat map analysis reveals specific recognition patterns: happiness achieves the highest recognition rate (96%), while sadness shows the lowest (63%). The study provides three main contributions: (1) empirical evidence for the superiority of EfficientNet-B0 for facial expression recognition, (2) comprehensive error pattern analysis through heat map visualization, and (3) practical insights into the limitations of dropout layers in expression recognition tasks. These findings advance the technical understanding of CNN architectures in emotion recognition systems and provide practical guidelines for implementing efficient facial expression recognition systems in real-world applications.

**Keywords:** Deep Learning; CNN; Conv2D with Max Pooling; Conv2D with Max Pooling&Dropout; EfficientNet-B0; Facial Expressions Recognition

## 1 INTRODUCTION

The ability to understand what others are thinking without verbal communication can significantly improve relationships. This is especially crucial in service industries, where success or failure often hinges on customer emotions. Knowing and responding to the customers' feelings can greatly enhance customer response effectiveness. However, direct verbal communication of emotions can already influence the outcome, making it difficult to alter a customer's decision. Therefore, it is essential to understand and respond to the customer's emotional state before a decision is made, and one of the most effective ways to gauge a customer's emotional state is by observing facial expressions. Facial expressions are among the most significant biological and social visual stimuli [1, 2]. They serve as vital indicators of emotional states and are extensively studied across various fields such as clinical, developmental, personality, and physiological research [2, 3].

Research in facial analysis has evolved from mere face recognition aimed at identifying individuals [4, 5] to facial expression recognition, which focuses on understanding the underlying emotional states [1, 2, 6-11]. Facial expression recognition is widely utilized in fields such as human-robot interaction and infotainment systems [8]. The traditional machine learning approaches for detecting facial expressions from images or videos typically rely on feature extraction techniques. However, these methods face challenges in accurately identifying relevant features. To address this, recent deep learning techniques, particularly those using convolutional neural networks (CNN), have demonstrated superior performance by automatically extracting and classifying features without manual intervention [6, 12-15].

Recent developments in facial expression recognition have proposed innovative solutions to address technical challenges:

- Transformer-based architectures: Transformers have demonstrated exceptional performance in feature extraction for facial expression recognition. For example, the Swin Transformer model has been applied to facial expression recognition tasks, effectively integrating audio-visual data to enhance recognition accuracy through multi-modal fusion [16]. Additionally, FaceXformer unifies various facial analysis tasks, including expression recognition, by employing advanced transformer-based techniques [17].
- Self-supervised learning methods: Self-supervised learning has reduced the reliance on annotated datasets while maintaining robust performance. Notable methods include SSL-FER, which generates positive pairs for training through temporal shifts and face-swapping techniques [18]. Another approach is Multi-Task Multi-Modal Self-Supervised Learning, which employs contrastive and clustering losses to enhance feature learning [17].
- Multi-modal data integration: Combining modalities such as voice and physiological signals has enhanced real-world recognition accuracy. The FSRT (Facial Scene Representation Transformer) integrates appearance, head pose, and facial expression features for more comprehensive recognition tasks [19].

Despite these advancements, accurately interpreting emotions based on facial expressions remains challenging.Variations due to individual, cultural, gender, and social differences can affect the clarity of the expressed emotions, leading to potential misinterpretations. When individuals clearly express their emotions through facial expressions, it is easier to understand their inner thoughts. However, emotions expressed through the face may not always be conveyed due to individual, cultural, gender, and social differences, making it difficult to accurately interpret the emotional state. Thus, while the classification of facial expressions from everyday images or videos is not always perfectly accurate, it continues to be an effective method for emotional state prediction.

The uniqueness of this study lies in its approach to addressing these challenges through the following contributions:

1. Development and evaluation of three CNN-based models (Conv2D with Max Pooling, Conv2D with Max Pooling & Dropout, and EfficientNet-B0) to enhance recognition accuracy across diverse conditions.
2. Heat map-based error analysis to identify misclassification patterns and provide deeper insights into the correlations between similar expressions.
3. Investigation of computationally efficient architectures, such as EfficientNet-B0, to reduce the resource demands of facial expression recognition systems.

To address these challenges and improve the accuracy of facial expression recognition**,** the following process is implemented. First, individuals are asked to display specific emotions through their facial expressions. Multiple evaluators assess these expressions, and if their evaluations concur, the expression is selected. The person making the expression must also agree that it accurately represents the intended emotion. If there is disagreement, the process is repeated until a consensus is reached. The finalized facial expression data are then employed to train and validate deep learning models, ensuring robustness and reliability in recognition tasks [20, 21].

This study uses images of eight facial expressions (Anger, Content, Disgust, Fear, Happiness, Neutral, Sadness, and Surprise) evaluated by human judges, achieving an accuracy of 79.08% (range: 70.19% - 88.87%) as reported by Yang et al. [2]. These images are used as training and validation data.

Based on this dataset, the objectives of this study are twofold:

1. Verification of CNN-Based Models: To verify whether the proposed CNN-based deep learning models achieve higher accuracy than the empirical accuracy of 79.08% reported by Yang et al. [2].
2. Error and Correlation Analysis: To analyze the recognition errors of the eight facial expressions using the proposed CNN-based models and examine error correlations through heatmap analysis.

This paper is structured as follows: Section 1 explains the necessity and purpose of the study. Section 2 reviews previous research on classical and deep learning-based methods for facial expression recognition and discusses methods for classifying facial expressions. Section 3 details the research model and methodology used to achieve the study's objectives. Section 4 introduces the experimental models and methods used and analyzes the results focusing on expression accuracy and facial expression error evaluation. Finally, Section 5 summarizes the findings, discusses the limitations, and proposes directions for future research.

## 2 LITERATURE REVIEW

Facial expressions representing human emotional states have currently reached the research stage where eight types of expressions are classified using deep learning, with ongoing efforts to incorporate more diverse facial expression data into research. Facial expression recognition technology can be divided into classical methods and deep learning-based methods.

### 2.1 Classical Methods

Classical methods involve extracting unique features of facial expressions from input images or videos and classifying them using machine learning. Studies on feature extraction for facial expressions include techniques such as Active Appearance Models (AAM), Gabor wavelets, Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), histograms of Local Phase Quantization (LPQ), Haar, Histogram of Oriented Gradient (HOG), Ferns, and Modified Census Transform (MCT). Methods used to classify the extracted features include AdaBoost learning and Support Vector Machines (SVM) [4, 6, 8].

Heo & Kang [22] summarized the techniques for extracting features and classifying facial expressions used in facial expression recognition.

Classical feature extraction methods can be explained as follows:

AAM [23] is a technique that matches a statistical model of the face to an image by learning the confusion of model parameters and errors in the derived image to perform efficient iterative matching.

Gabor wavelets [24] use features extracted by Gabor wavelets from facial images or videos for facial expression recognition.

LBP [25] is widely used in image recognition due to its effectiveness in recognizing objects, insensitive to lighting changes, and ease of computation. Recent techniques have been developed to reflect facial radii and various facial angles to improve facial expression recognition accuracy [26, 27].

SIFT [28, 29] identifies easily distinguishable feature points and extracts feature vectors for local patches centered on those points. This method divides patches around feature points into blocks and uses a vector obtained by concatenating histograms of the gradient directions and magnitudes of pixels in each block.

LPQ [30] is based on quantizing the phase information in local neighborhoods through the Fourier transform and uses histograms of LPQ labels calculated in local areas as described. It has been noted to perform better than LBP regardless of the image state.

Haar [31] finds feature elements by using the brightness differences between regions in images or videos and combines these features in various sizes and locations to extract characteristics.

HOG [32] divides the entire area into cells of the same size, calculates histograms for pixels with gradients above a certain value in each cell, and concatenates these vectors. HOG shows strong properties against local variations.

Ferns [33, 34] extract feature points from images or videos and calculate local patches around these points, like SIFT, and are known for excellent image-matching performance. Unlike Haar, which uses brightness differences between regions, Ferns uses brightness differences at the pixel level, represented by signs instead of values. Although it has a long training time, Ferns offers faster matching speed and superior performance compared to SIFT and SURF.

MCT [35], a modification of the Census Transform (CT) proposed by Zabih & Woodfill [36], is one of the representative methods in facial recognition, known for its

high recognition rates. The CT result for a specific pixel provides similar results to LBP.

Methods for classifying features extracted by classical methods include:

AdaBoost, developed by Freund and Schapire [37], is a method designed to address the inherent issues in boosting algorithms. SVM (Support Vector Machine) [38, 39] is a supervised learning method that uses a maximum-margin hyperplane in the input space to classify two regions.

SVM performs well with linear hyperplanes and can be adapted for nonlinear classification [22].

### 2.2 Deep Learning-Based Methods

Due to the challenges of finding and deciding appropriate features in classical methods, recent approaches employ CNN-based deep learning techniques that can automatically extract features through machine learning [7, 8, 13, 40, 41].

Notable deep learning-based facial recognition technologies can be found in the works of Kim et al. [42], Hwang [43], Lee et al. [44], and Krizhevsky et al. [45]. These studies demonstrate various deep learning models and techniques that improve facial recognition accuracy in different applications.

Kim et al. [42] explain that facial recognition technology continues to be researched and is applied in areas such as surveillance and biometric systems. They also discuss large facial image datasets.

Krizhevsky et al. [45] trained a large, deep CNN algorithm to classify 1.2 million high-resolution images into 1,000 classes for the ImageNet LSVRC-2010 competition and used dropout regularization to reduce overfitting in the fully connected layers.

Savchenko [11] summarized research results on CNN-based recognition of various facial expressions, demonstrating the efficiency of the proposed deep learning techniques. The results showed excellent performance in facial expression image datasets presented in competitions such as EmotiW 2019 and 2020 challenges: AFEW (Acted Facial Expression In The Wild), VGAF (Video level Group Affect), and EngageWild.

To recognize facial expressions in images or videos, CNN-RNN types and 3D CNN types are commonly used [8].

Lee & Song [8] reported performance of 49.87% for C3D-GRU [9], the highest accuracy among deep learning-based models, using the AFEW (Acted Facial Expression In The Wild) dataset. The low accuracy of the best model highlights the difficulty of recognizing facial expressions.

Lee et al. [10] improved the accuracy to 51.4% based on the research of Lee et al. [9] using AFER data. Savchenko [11] explained that existing methods showed a maximum accuracy of 66.34% for recognizing seven facial expressions from AffectNet dataset images and 62.42% for eight facial expressions. For facial expressions recognized from AFEW video data, a maximum accuracy of 59.27% was reported.

Son & Lee [46] conducted CNN-based research using webcams to detect and recognize faces, presenting a model with an accuracy of about 77%. Kyung [47] demonstrated that simple algorithms like VGG, ResNet, and DenseNet used for recognizing business card images achieved an average accuracy of 97.9%, proving the efficiency of even simple algorithms.

Lee et al. [48] proposed a CNN-based algorithm to correct image rotation, a factor that reduces image recognition accuracy, and obtained a mean absolute error (MAE) of 4.5951 in experiments with 100 business card images.

Ahn & Cho [49] applied CNN-based algorithms such as VGGNET [50], ResNet [51], Densenet [52], and ResNeXt [53] for facial expression recognition. Training without face landmarks showed an accuracy of 34%, and they also conducted research including face landmarks.

Choi et al. [54] proposed a CNN-based facial expression recognition technique, explaining that even with reduced convolutional layers and nodes, facial expression recognition accuracy could be improved for five facial expressions (neutral, happy, sad, angry, surprised).

Choi et al. [55] constructed a dataset for six facial expressions (neutral, happy, sad, angry, surprised, disgusted) and conducted research to identify a CNN-based structure that detects facial expression features. The proposed model showed a classification performance of 96.88% while having a shorter execution time compared to other CNN-based structures [51, 56-59].

Recent advancements such as transformer architectures have further improved feature extraction and classification.

Swin Transformers, for instance, use hierarchical attention mechanisms to enhance global and local feature integration, improving recognition accuracy [16]. Similarly, FaceXformer unifies various facial analysis tasks, offering a robust framework for emotion recognition through advanced attention mechanisms [17].

Multi-modal approaches represent another significant development in facial expression recognition. By integrating visual data with other modalities such as audio and physiological signals, these methods provide a more comprehensive understanding of emotions. For instance, the FSRT (Facial Scene Representation Transformer) combines head pose, body language, and facial expressions to enhance recognition accuracy in real-world scenarios [19].

Despite these advancements, challenges remain, particularly in handling overlapping expressions, such as distinguishing "fear" from "surprise," or achieving real-time performance for complex architectures.

### 2.3 Limitations of Existing Methods

Although substantial progress has been made in facial expression recognition, both classical and deep learning-based approaches still encounter various challenges. Classical methods rely heavily on handcrafted features, such as Local Phase Quantization (LPQ) and Modified Census Transform (MCT), which are highly effective under controlled conditions but struggle with variations in lighting, occlusion, and pose [30, 35, 36]. These methods often depend on classifiers like AdaBoost and Support Vector Machines (SVM) to process extracted features, achieving competitive performance on specific datasets [33, 34]. However, their reliance on pre-defined

features limits their adaptability to more complex datasets or subtle emotional nuances.

On the other hand, deep learning methods, particularly those using Convolutional Neural Networks (CNNs), have overcome many of these limitations by enabling automatic feature extraction [45]. However, they require extensive labeled datasets, which are expensive and time-consuming to create. Additionally, while recent transformer-based architectures such as Swin Transformers [16] and FaceXformer [17] offer improved accuracy through attention mechanisms, their computational complexity poses challenges for real-time applications. Furthermore, deep learning models often struggle with overlapping expressions, such as distinguishing "fear" from "surprise," due to subtle similarities in facial features.

Recent multi-modal approaches aim to address these challenges by integrating additional data modalities, such as voice and physiological signals, to provide more robust emotion recognition systems [19]. However, these methods introduce new complexities, including the synchronization of multiple data streams and the lack of standardized benchmarks for evaluation. As a result, while multi-modal approaches show promise, their practical implementation in real-world scenarios remains limited.

## 3 METHODS

CNN, also known as ConvNet, is a deep learning architecture that trains and learns directly from data, utilizing the learned results. CNNs are particularly useful for finding specific patterns in images or videos and are also effective for analyzing audio and time-series data. A CNN can have multiple layers, each tasked with detecting different features.

As shown in Fig. 1, an NN consists of an input layer, hidden layers, and an output layer. The input layer is where the collected data is fed, and the output layer is where the results of the NN technique are observed. The hidden layers, which lie in between, consist of multiple layers and perform learning using the data from the input layer. Because these layers are deep, the term "deep learning" originated.

CNN is a deep learning neural network model designed to mimic the structure of the visual cortex in animals. It is built using a multi-layer perceptron structure that consists of repeated convolutional layers for feature extraction and Pooling layers for information compression (Fig. 2).

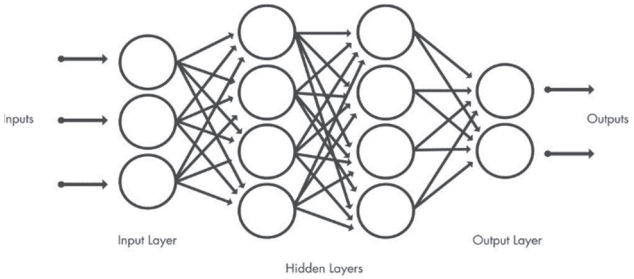


**Figure 1** Basic structure of a NN

Pooling is an operation that converts blocks of a certain size into a representative value, compressing the feature maps derived from the convolutional layers to emphasize specific data. Pooling techniques are widely used in CNN models because they reduce noise and increase learning speed.

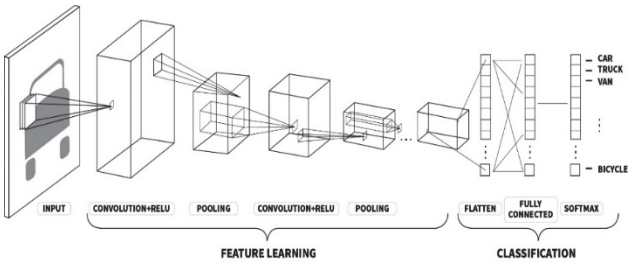


**Figure 2** Working principle of CNN [60]

There are various pooling methods, including Max Pooling, which selects the maximum value in the block as the representative value; Average Pooling, which selects the average value; and Stochastic Pooling, which converts the block into selection probabilities and selects based on those probabilities. However, Max Pooling is generally used, and thus, this paper also applies Max Pooling (Fig. 3).

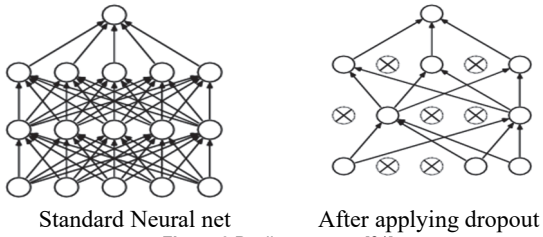


**Figure 3** Pooling process [61]

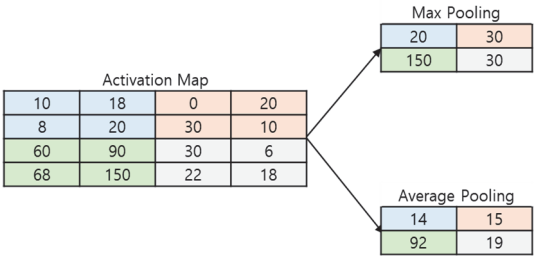


**Figure 4** Dropout process [63]

Dropout is used to reduce overfitting in deep learning models. As shown in Fig. 4, it involves cutting off some of the nodes connected between the input layer, hidden layers, and output layer in the initial neural network, preventing them from connecting [62]. This process results in the creation of a new neural network structure that differs from the initial model. Dropout often yields better results than using regularization methods.

The dataset used in this study consists of facial expression images, which were preprocessed to prepare them for training and testing. The following steps were performed:

① Normalization: Pixel values of the images were normalized to the range [0, 1] by dividing each pixel value by 255. This ensured that the data was standardized and improved the convergence rate of the models during training.

② Shuffling: The dataset was shuffled to ensure a balanced distribution of data during training.

③ Splitting: The entire dataset was split into training and test sets in an 80:20 ratio. This split ensured a sufficient number of samples for training while reserving a portion for evaluating the models.

The performance of the models was validated using a holdout validation strategy. Specifically, the dataset was divided into a training set (80%) and a test set (20%).The test set was used to evaluate the accuracy of the models after training, ensuring unbiased performance assessment.

The experimental model uses CNN-based models (Conv2D with Max Pooling (M1), Conv2D with Max Pooling& Dropout (M2), and EfficientNet-B0 (M3)) to compare with the accuracy determined by the evaluators. CNN is the most efficient method for image or video recognition, and its accuracy improves with the appropriate use of Max Pooling and dropout. This study uses the EfficientNet-B0 model as a comparative model because it outperforms the well-known ResNet model in image classification while requiring only about 1/5th of the parameters [2, 64].

This study utilizes the Tsinghua facial expression database constructed by Yang et al. [2], which demonstrates an accuracy of 79.08% through evaluator identification of facial emotional expressions.

Since this paper compares the identification accuracy of evaluators with that of deep learning models, the execution speed of the deep learning models is not considered. Additionally, the hyperparameter values used in the deep learning models, such as learning rate, batch size, and epochs, are uniformly set for fair comparison.

Fig. 5 shows the network structure of the deep learning models used to compare the identification accuracy with that of human evaluators.


**Figure 5** Network structure of deep learning models used in the experiment

In Fig. 5, Model M1 uses Conv2D as the basic network and applies only Max Pooling. The activation function is ReLU, and since the output layer needs to classify one of the eight expressions, softmax is used as the activation function. The first convolutional layer contains 16 filters of size 3 × 3, and the second convolutional layer contains 32 filters of size 3 × 3.

Model M2, similar to M1, uses Conv2D as the basic network and applies Max Pooling along with the Dropout technique. The activation functions for the input and hidden layers are ReLU, and since the output layer classifies one of the eight expressions, softmax is used as the activation function, as in M1.

Model M3 uses the EfficientNet-B0 network structure, which is a lightweight and efficient CNN architecture designed to achieve high accuracy with fewer parameters. The EfficientNet-B0 network also applies the Pooling technique in the output layer.

While hyperparameter values such as learning rate, batch size, and epochs were pre-selected based on common practices and verified settings from similar studies (Tab. 1), no exhaustive hyperparameter optimization was conducted in this study. Instead, the chosen values represent a balanced approach to achieving reliable results without extensive fine-tuning. Future studies could employ grid search or Bayesian optimization methods to systematically explore the hyperparameter space.

## 4 EXPERIMENT RESULTS AND DISCUSSION

The Tsinghua facial expression database presented by Yang et al. [2] consists of images of eight facial expressions from 110 Chinese individuals, spanning both young and old age groups. The initial dataset included facial expressions from 67 young individuals interested in acting (average age of 23.82 years, standard deviation of 4.18 years, age range of 19-35 years, with 34 females) and 70 older individuals (average age of 64.40 years, standard deviation of 3.51 years, age range of 60-76 years, with 35 females). After evaluation and validation by judges, the final selection included facial expressions from 110 individuals. Specifically, 47 older adults (21 males, 26 females) and 63 younger adults (32 males, 31 females) were selected, with recognition accuracy ranging from a minimum of 70.19% to a maximum of 88.87%. The average accuracy was 79.08%, as previously mentioned. The deep learning models used in this study (Conv2D with Max Pooling, Conv2D with Max Pooling& Dropout, and EfficientNet-B0) were implemented using basic packages composed of Anaconda, TensorFlow, and Keras. The architectural details for each model are as follows:

- M1 (Conv2D with Max Pooling): This model consists of a sequential architecture starting with five Conv2D layers with filters of 16, 32, and 64, each followed by MaxPooling2D layers with a pooling size of (2, 2). A Flatten layer then connects the convolutional output to a Dense layer with 512 neurons, followed by a final Dense layer with 8 neurons and a softmax activation function for multi-class classification. The ReLU activation function is applied to all Conv2D layers.
- M2 (Conv2D with Max Pooling & Dropout): Similar to M1, but this model incorporates Dropout layers after the first Conv2D layer (dropout rate: 0.25) and after the Dense layer (dropout rate: 0.5) to prevent overfitting during training. The rest of the architecture is identical to M1.
- M3 (EfficientNet-B0): This model uses a pre-trained EfficientNet-B0 architecture for feature extraction, followed by a Dense layer with 8 neurons and a softmax activation for classification. The input images are resized to 224 × 224 pixels, and the model is fine-tuned on the Tsinghua dataset by freezing the initial convolutional layers and training only the final layers. Standard normalization techniques (mean and standard deviation) are applied during preprocessing.

The coding for experiments and analysis was conducted using Python version 3.7.6 [14, 15, 20, 21, 65-70].

In this study, a series of preprocessing steps was performed to convert the image data into a format suitable for deep learning models. First, the pixel values of all images were normalized to a range of [0, 1] by dividing them by 255. This step adjusts the range of the data to enhance training efficiency and improve the model's convergence rate. Next, the data was randomly shuffled to ensure a distribution that was favorable for training. Finally, the entire dataset was split into training and test sets in a ratio of 80:20, which were used for model training and evaluation.

The first experiment compares the accuracy of the three models (Conv2D with Max Pooling, Conv2D with Max Pooling & Dropout, and EfficientNet-B0) presented in Fig. 5 with the accuracy presented by Yang et al. [2].

**Table 1** The hyperparameter values commonly used across the three models

| Hyperparameter | Description | Value/Setting |
|---|---|---|
| NumOfRuns | Number of total experiment repetitions | 100 |
| EPOCHS | Number of times the entire dataset is trained | 100 |
| Conv2D Filters | Number of filters in each Conv2D layer | 16, 32, 64 |
| Conv2D Kernel Size | Size of the filter (kernel size) | (3, 3) |
| Activation Function | Activation function to add non-linearity and enhance the model's learning ability | 'relu', 'softmax' |
| MaxPooling2D Pool Size | Pooling size for the MaxPooling layer, used for downsampling feature maps | (2, 2) |
| Flatten Layer | Flattens multi-dimensional arrays into 1D to connect to the Dense Layer | N/A |
| Dense Layer Neurons | Number of neurons in the Fully Connected Layer | 512 |
| Loss Function | Loss function used for model optimization, suitable for multi-class classification problems | 'sparse_categorical_crossentropy' |
| Optimizer | Model training optimization algorithm that adjusts the learning rate automatically for efficient learning | 'adam' |
| Metrics | Metrics to monitor during model training and evaluation | 'acc' (accuracy) |
| EarlyStopping Monitor | Metric to monitor for early stopping during training | 'val_loss' |
| EarlyStopping Patience | Number of epochs to wait before early stopping when there is no improvement | 20 |

The hyperparameter values commonly used across the three models are presented in Tab. 1 below. These values were selected based on their widespread use in similar studies and their proven effectiveness in achieving stable training and high accuracy. For instance, the Conv2D filter sizes of 16, 32, and 64 and the kernel size of (3, 3) are commonly employed in CNN architectures to balance computational efficiency and feature extraction capability. The learning rate, batch size, and number of epochs were chosen through preliminary trials to ensure convergence without overfitting. Additionally, early stopping with a patience of 20 epochs was incorporated to prevent unnecessary training when validation loss ceased to improve, thus optimizing training efficiency.

The second experiment investigates which of the eight expressions were misclassified as other expressions by the three models (Conv2D with Max Pooling, Conv2D with Max Pooling & Dropout, and EfficientNet-B0) presented in Fig. 5. The confusion matrix generated for each model revealed patterns in misclassification, which are analyzed to identify similarities between expressions. This analysis aims to identify expressions that are likely to be grouped due to their similarities.

### 4.1 Accuracy and Statistical Analysis

The formula used to evaluate the accuracy is expressed as Eq. (1):

$$(TP + TN)/(TP + FP + TN + FN) \quad (1)$$

Eq. (1) can be explained using the confusion matrix presented in Tab. 2. The confusion matrix quantifies the number of instances where the actual values (the true class of the target) match the predicted values (the class predicted by the model). The correct classes are displayed in the rows, while the predicted classes are displayed in the columns.

**Table 2** Confusion Matrix

| Classification | PredictedPositive | Predicted Negative |
|---|---|---|
| ActualPositive | True Positive (*TP*) | False Negative (*FN*) |
| Actual Negative | False Positive (*FP*) | True Negative (*TN*) |

In this table, *TP* (True Positive) and *TN* (True Negative) represent the instances where the model correctly predicted the actual value, while *FP* (False Positive) and *FN* (False Negative) represent the instances where the model incorrectly predicted the actual value.

To compare the results with those of Yang et al. [2], the accuracy of the three selected models is evaluated by running each model 100 times.

For the first model, M1: Conv2D with Max Pooling, it is observed that when the number of epochs exceeds 49, the accuracy reaches 82.95% and does not improve further (Fig. 6).

For the second model, M2: Conv2D with Max Pooling & Dropout, it is observed that even under the same conditions as Model M1, the accuracy does not converge even after exceeding 100 epochs (Fig. 7).

Model M2, which adds the Dropout condition to the Max Pooling condition of Model M1, has a more complex network structure and longer execution time. However, its average accuracy is not only worse than that of Model M1 but also lower than the accuracy presented by Yang et al. [2]. Therefore, although increasing the epochs might improve the accuracy of Model M2, the execution time would also increase, making it less efficient.

The third model, M3: EfficientNet-B0, was also run 100 times under the same conditions as the other models to compare the results (Fig. 8). Model M3 exceeds 90% accuracy after 30 epochs, reaches a maximum accuracy of 94.3% at 46 epochs, and maintains an accuracy of 92.0% from 98 epochs onwards.

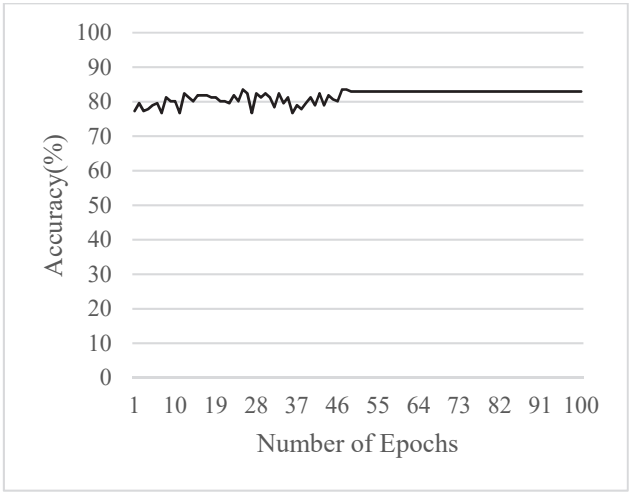


**Figure 6** Accuracy variation with epochs for model M1

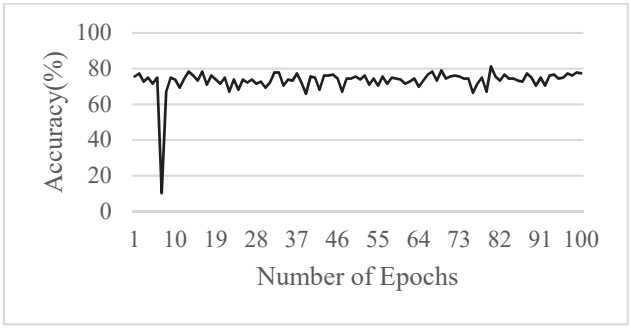


**Figure 7** Accuracy variation with epochs for model M2

The accuracy of the three models is in the order of M3 > M1 > M2, with models M3 and M1 achieving higher accuracy than that presented by Yang et al. [2]. However, despite longer execution times, Model M2 shows lower accuracy than Yang et al. [2], indicating lower efficiency.

In this study, the statistical significance of the data for three groups (M1, M2, M3) was evaluated through Exploratory Data Analysis (EDA), normality tests, one-way Analysis of Variance (ANOVA), post-hoc tests (Tukey HSD), and correlation analysis.

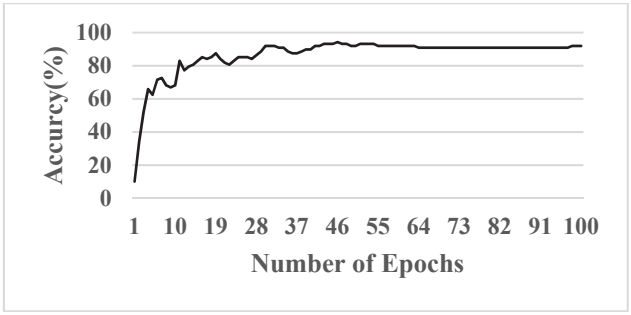


**Figure 8** Accuracy variation with epochs for model M3

The basic statistics for the three groups (M1, M2, M3) showed that the mean of M1 was 0.8176, M2 was 0.7374, and M3 was 0.8613. The standard deviations were 0.018, 0.076, and 0.092, respectively, indicating differences in data distribution among the groups. Histograms and box plots were used for data visualization, revealing that M1 and M2 were relatively narrowly distributed, while M3 showed a wider distribution. This suggests that the data distribution characteristics differ among the groups.

The Shapiro-Wilk test was conducted to evaluate whether each group followed a normal distribution. The *p*-values for M1 and M2 were 0.6303 and 0.5311, respectively, suggesting that normality could be assumed. However, the p-value for M3 was 0.0246, indicating that normality could not be assumed ($p < 0.05$). Thus, M3 is likely not normally distributed, suggesting the need to consider non-parametric methods in data analysis.

A one-way ANOVA was performed to test the differences in means among the three groups. The *F*-statistic was 81.468, with a *p*-value of 6.22e−29, indicating statistically significant differences in the means among the three groups. Furthermore, post-hoc testing with Tukey HSD showed significant differences between all group comparisons: the mean difference between M1 and M2 was −0.0802 ($p = 0.001$), between M1 and M3 was 0.0437 ($p = 0.0468$), and between M2 and M3 was 0.1238 ($p = 0.001$). In all cases, the p-value was less than 0.05, confirming that the differences among the groups were statistically significant.

To assess the correlations among variables, Pearson correlation coefficients were calculated. The correlation coefficient between M1 and M2 was −0.0560, indicating a very weak negative correlation, and between M1 and M3 was −0.0670, also indicating a weak negative correlation. The correlation coefficient between M2 and M3 was 0.0041, suggesting no correlation. These results indicate that there is no strong correlation among the three variables.

### 4.2 Evaluation of Facial Expression Errors

Among the three CNN-based models, M1 and M3 achieve higher accuracy than Yang et al. [2], while Model M2 shows lower accuracy. Therefore, the facial expression error evaluation focuses on Models M1 and M3, which have higher accuracy than Yang et al. [2].

Using Model M1, we conducted experiments and analyzed facial expression errors at an accuracy close to 78.98%, which is like the 79.08% accuracy reported by Yang et al. [2]. Tab. 3 shows the number of misclassifications for each facial expression. Tab. 4 shows the heatmap of facial expression recognition errors for Model M1 based on the data described in Tab. 3.

**Table 3** Facial expression recognition error data for model M1

| | Anger | Content | Disgust | Fear | Happiness | Neutral | Sadness | Surprise | Total |
|---|---|---|---|---|---|---|---|---|---|
| Anger | 14 | 0 | 0 | 1 | 0 | 2 | 2 | 0 | 19 |
| Content | 1 | 16 | 0 | 1 | 0 | 2 | 1 | 0 | 21 |
| Disgust | 2 | 0 | 16 | 0 | 2 | 0 | 1 | 0 | 21 |
| Fear | 0 | 1 | 0 | 16 | 0 | 1 | 0 | 1 | 19 |
| Happiness | 0 | 0 | 1 | 0 | 25 | 0 | 0 | 0 | 26 |
| Neutral | 1 | 2 | 0 | 2 | 0 | 12 | 1 | 0 | 18 |
| Sadness | 4 | 3 | 0 | 0 | 0 | 2 | 15 | 0 | 24 |
| Surprise | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 25 | 28 |

**Table 4** Heatmap for facial expression recognition errors of model M1

| | Anger | Content | Disgust | Fear | Happiness | Neutral | Sadness | Surprise |
|---|---|---|---|---|---|---|---|---|
| Anger | 0.74 | 0.00 | 0.00 | 0.05 | 0.00 | 0.11 | 0.11 | 0.00 |
| Content | 0.05 | 0.76 | 0.00 | 0.05 | 0.00 | 0.10 | 0.05 | 0.00 |
| Disgust | 0.10 | 0.00 | 0.76 | 0.00 | 0.10 | 0.00 | 0.05 | 0.00 |
| Fear | 0.00 | 0.05 | 0.00 | 0.84 | 0.00 | 0.05 | 0.00 | 0.05 |
| Happiness | 0.00 | 0.00 | 0.04 | 0.00 | 0.96 | 0.00 | 0.00 | 0.00 |
| Neutral | 0.06 | 0.11 | 0.00 | 0.11 | 0.00 | 0.67 | 0.06 | 0.00 |
| Sadness | 0.17 | 0.13 | 0.00 | 0.00 | 0.00 | 0.08 | 0.63 | 0.00 |
| Surprise | 0.00 | 0.00 | 0.00 | 0.11 | 0.00 | 0.00 | 0.00 | 0.89 |

The heatmap in Tab. 4 indicates that the darker the color, the stronger the connection. For each facial expression, the highest accuracy is achieved when recognizing happiness as happiness, with an accuracy of 96%. The happy facial expression is misclassified as disgust only 4% of the time among the eight facial expressions, and the rest are correctly identified as happiness, indicating very high accuracy.

The lowest accuracy is for recognizing sadness as sadness, with an accuracy of 63%. When sad facial expressions are misclassified, they are most mistaken for anger (17%), content (13%), and neutral (8%). Sad facial expressions are not misclassified as disgust, fear, happiness, or surprise.

These misclassifications may result from similarities in facial muscle movements between expressions. For example, 'sadness' and 'neutral' or 'anger' and 'sadness' share subtle facial muscle contractions that the model might confuse. Such misclassifications highlight the need for more refined models that can better differentiate these nuances. Moreover, these findings suggest that in real-world applications, such as customer service or psychological analysis, reliance solely on facial expressions for emotion recognition may lead to inaccuracies, particularly for similar expressions. Thus, integrating additional data sources (e.g., voice tone, and body language) could improve overall emotion recognition performance.

Although the accuracy for recognizing sad facial expressions is the lowest at 63%, the accuracy for neutral facial expressions is also very low at 67%. Neutral expressions are misclassified as content and fear at 11% each, and as anger and sadness at 6% each.

Next, we analyze the facial expression recognition errors for Model M3. Tab. 5 shows the facial expression recognition error data for Model M3. Tab. 6 shows the heatmap of facial expression recognition errors for Model M3 based on the data described in Tab. 5.

**Table 5** Facial expression recognition error data for model M3

| | Anger | Content | Disgust | Fear | Happiness | Neutral | Sadness | Surprise | Total |
|---|---|---|---|---|---|---|---|---|---|
| Anger | 19 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 22 |
| Content | 0 | 16 | 0 | 0 | 0 | 2 | 0 | 0 | 18 |
| Disgust | 0 | 0 | 23 | 0 | 0 | 0 | 0 | 0 | 23 |
| Fear | 0 | 0 | 0 | 20 | 0 | 1 | 0 | 1 | 22 |
| Happiness | 0 | 0 | 0 | 0 | 22 | 0 | 0 | 0 | 22 |
| Neutral | 2 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 20 |
| Sadness | 2 | 1 | 0 | 0 | 0 | 5 | 14 | 0 | 22 |
| Surprise | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 27 |

**Table 6** Heatmap for facial expression recognition errors of model M3

| | Anger | Content | Disgust | Fear | Happiness | Neutral | Sadness | Surprise |
|---|---|---|---|---|---|---|---|---|
| Anger | 0.86 | 0.05 | 0.00 | 0.05 | 0.00 | 0.00 | 0.05 | 0.00 |
| Content | 0.00 | 0.89 | 0.00 | 0.00 | 0.00 | 0.11 | 0.00 | 0.00 |
| Disgust | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Fear | 0.00 | 0.00 | 0.00 | 0.91 | 0.00 | 0.05 | 0.00 | 0.05 |
| Happiness | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 |
| Neutral | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 | 0.90 | 0.00 | 0.00 |
| Sadness | 0.09 | 0.05 | 0.00 | 0.00 | 0.00 | 0.23 | 0.64 | 0.00 |
| Surprise | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

In the heatmap in Tab. 6, the recognition accuracy for facial expressions of disgust, happiness, and surprise is 100%, while the recognition accuracy for sad expressions is the lowest at 63%. Sad expressions are misclassified as neutral 23% of the time, as anger 9% of the time, and as content 5% of the time. These misclassifications can be attributed to overlapping features, such as the slight downward turn of the mouth in both 'sadness' and 'neutral' expressions. Such errors indicate that these CNN models, while accurate, may require enhancement for more subtle emotional distinctions, suggesting further development in feature extraction and model training to address these weaknesses. In practical applications, like real-time customer service analysis, relying solely on facial expression data could mislead interpretations, especially in culturally diverse environments. The recognition accuracy for anger and content facial expressions is 86% and 89%, respectively. Although these accuracies are much higher than the lowest accuracy for sad expressions (63%), they are the second lowest in terms of recognition accuracy. Anger expressions are misclassified as content, fear, and sadness, each with a misclassification rate of 5%. Content expressions show an 11% misclassification rate as neutral and are not misclassified as any other expressions.

### 4.3 Computational Complexity Analysis

The computational complexity of the three CNN-based models (M1, M2, and M3) is evaluated qualitatively based on their relative structural complexity and training efficiency.

M1 (Conv2D with Max Pooling): This model has a straightforward architecture, relying on pooling operations to reduce feature map dimensions. Its simplicity makes it the least computationally demanding among the three models.

M2 (Conv2D with Max Pooling & Dropout): By incorporating Dropout layers, M2 enhances generalization, but incurs additional computation during training. This results in longer execution times without a significant improvement in accuracy, making it less efficient than M1.

M3 (EfficientNet-B0): Leveraging pre-trained weights and an optimized architecture, M3 achieves the highest accuracy while maintaining reasonable computational efficiency. Although more complex than M1 and M2, M3 demonstrates the best balance of accuracy and training efficiency.

## 5 CONCLUSIVE REMARKS

This paper classifies human facial expressions into eight categories and utilizes the facial expression data from Yang et al. [2], which showed an accuracy of 79.08% through empirical evaluations by a group of young and old evaluators. Three CNN-based deep learning algorithms (Conv2D with Max Pooling, Conv2D with Max Pooling& Dropout, EfficientNet-B0) were used, and the results obtained from these models were compared and analyzed against the results of Yang et al. [2].

The main research findings of this paper are as follows:

1) Novel Contributions in CNN-Based Algorithms**:** The study introduces a comparative analysis of three distinct CNN-based deep learning algorithms (Conv2D with Max Pooling, Conv2D with Max Pooling& Dropout, EfficientNet-B0) applied to facial expression recognition. Unlike previous studies, this research demonstrates the differential effectiveness of these models, particularly highlighting the advantages of EfficientNet-B0 in handling complex facial expressions. This represents a significant step forward in identifying optimal CNN architectures for nuanced emotion recognition tasks.

2) Comparison with Yang et al. [2] and Expansion of Existing Knowledge: Using the database presented by Yang et al. [2], Model M1 (Conv2D with Max Pooling) achieved an average accuracy of 81.68%, and Model M3 (EfficientNet-B0) achieved an average accuracy of 86.47%, both of which were higher than the accuracy reported by Yang et al. [2]. However, Model M2 (Conv2D with Max Pooling& Dropout) showed a lower average accuracy of 73.25%, indicating less efficiency compared to Yang et al. [2]. The findings expand upon the existing work by not only validating the effectiveness of simpler CNN models but also demonstrating that more sophisticated models like EfficientNet-B0 can achieve higher accuracy with fewer parameters, thereby setting a new benchmark for future studies.

3) Implications from Heatmap Analysis: The heatmap of facial expression recognition results indicated varying

accuracy across different expressions. For Model M1, the expression 'Happiness' was recognized with the highest accuracy of 96%. Misclassification occurred as 'Disgust' only 4% of the time, with the rest being correctly recognized as 'Happiness'. The lowest accuracy was for recognizing 'Sadness' as 'Sadness', with an accuracy of 63%. Misclassification for 'Sadness' occurred as 'Anger' (17%), 'Content' (13%), and 'Neutral' (8%), and it was not misclassified as 'Disgust, Fear, Happiness, Surprise'. The accuracy for recognizing 'Neutral' expressions was also low at 67%, with misclassifications occurring as 'Content' and 'Fear' (11% each), and 'Anger' and 'Sadness' (6% each). For Model M3, the recognition accuracy for 'Disgust, Happiness, Surprise' was 100%, while the recognition accuracy for 'Sadness' was the lowest at 63%. Misclassifications for 'Sadness' occurred as 'Neutral' (23%), 'Anger' (9%), and 'Content' (5%). The recognition accuracy for 'Anger' and 'Content' was 86% and 89%, respectively. Although these accuracies are higher than 63% for 'Sadness', they were the second and third lowest. Misclassifications for 'Anger' occurred as 'Content, Fear, Sadness' (5% each), and for 'Content' as 'Neutral' (11%), with no other misclassifications. The analysis reveals critical insights into the nature of misclassifications, such as overlapping features in expressions like 'Sadness' and 'Neutral'. These findings underscore the need for more sophisticated feature extraction and model training techniques to address subtle emotional distinctions, especially in real-world applications where cultural and demographic variability can impact recognition performance.

The limitations of this study and future research directions to address them are as follows:

1) Quality and Quantity of Training Data: Deep learning techniques require a large amount of training data with high accuracy in feature representation. For facial expression recognition, the data should clearly distinguish between different expressions, which can be achieved through exaggerated acting in the creation of training data. However, since most people do not display exaggerated facial expressions, effective deep learning techniques need to be developed to recognize emotional states even with subtle differences in facial expressions. Future research could focus on incorporating data augmentation techniques such as rotation, flipping, and color jittering to create a more diverse datasets. Additionally, collecting facial expression data across different demographics and cultural contexts could provide a more comprehensive training set.

2) Analysis of Various Angles and Occlusions: This study analyzed eight facial expressions viewed from the front. However, facial expressions can be partially visible from side views or due to occlusions such as masks. Therefore, additional research is needed to analyze facial expressions viewed from various angles and those partially visible due to occlusions. Future studies could explore 3D facial expression recognition or multi-angle CNN models to improve recognition under varied conditions.

3) Database Expansion: The experiments in this study used the database constructed by Yang et al. (2020), which includes eight facial expressions but is not sufficiently large. Future work should focus on building a more comprehensive database with a wider variety of facial expressions and more data to improve the performance of deep learning models. Collaborating with institutions to gather larger datasets that include dynamic and spontaneous expressions could provide more realistic data for model training.

In conclusion, this study contributes to the field of facial expression recognition by providing a comparative analysis of CNN models, highlighting the potential of EfficientNet-B0 for higher accuracy and lower computational cost, and offering insights into the implications of misclassifications in practical applications. Future research will benefit from expanding on these findings to address the identified challenges and further advance the development of robust facial expression recognition systems. Future research will benefit from exploring various CNN architectures, leveraging more diverse datasets, and refining the models to address the identified challenges more effectively, thereby advancing the development of robust facial expression recognition systems.

**Acknowledgments**

This study was conducted by research funds from Gwangju University in 2024.

## 6 REFERENCES

[1] Palermo, R. & Rhodes, G. (2007). Are you always on my mind? A review of how face perception and attention interact. *Neuropsychologia, 45*(1), 75-92. https://doi.org/10.1016/j.neuropsychologia.2006.04.025

[2] Yang, T., Yang, Z., Xu, G., Gao, D., Zhang, Z., Wang, H., Liu, S., Han, L., Zhu, Z., Tian, Y., Huang, Y., Zhao, L., Zhong, K., Shi, B., Li, J., Fu, S., Liang, P., & Banissy, M. J. (2020). Tsinghua facial expression database - A database of facial expressions in Chinese young and older women and men: Development and validation. *PLoS ONE, 15*(4), 1-14. https://doi.org/10.1371/journal.pone.0231304

[3] Ahn, D., Kim, S., & Ko, B. C. (2021). Vision transformer based dynamic facial emotion recognition. *Proceedings of the 2021 Korea Software Congress*, 476-478.

[4] Rhyou, S. Y., Kim, H. J., & Cha, K. E. (2019). Development of access management system based on face recognition using ResNet. *Journal of Korea Multimedia Society, 22*(8), 823-831.https://doi.org/10.9717/kmms.2019.22.8.823

[5] Ahmad, A., Albalas, F., Tawfik, A., Younis, L. B., & Bashayreh, A. (2021). Masked face recognition using deep learning: A review. *Electronics, 10*(21), 1-35. https://doi.org/10.3390/electronics10212666

[6] Won, C. & Lee, B. K. (2018). Detection of face expression based on deep learning. *Journal of Korea Multimedia Society, 21*(8), 917-924. https://doi.org/10.9717/kmms.2018.21.8.917

[7] Yoon, K. S. & Lee, S. W. (2019). Music player using emotion classification of facial expressions. *Proceedings of the 2019 Korea Society of Computer and Information Winter Conference*, *27*(1), 243-246.

[8] Lee, M. K. & Song, B. C. (2019). Recent research trends of facial expression recognition. *Proceedings of the 2019 Korean Society of Broadcast and Media Engineers Autumn Conference*, 128-130.

[9] Lee, M. K., Kim, D. H., & Song, B. C. (2018). Visual scene-aware hybrid neural network architecture for video-based facial expression recognition. *Proceedings of the 2018 IEEE International Conference on Automatic Face and Gesture Recognition*, 1-8. https://doi.org/10.1109/FG.2019.8756551

[10] Lee, M. K., Kim, D. H., & Song, B. C. (2020). Visual scene-aware hybrid and multi-modal feature aggregation for facial

expression recognition. *Sensors, 20*, 1-24. https://doi.org/10.3390/s20185184

[11] Savchenko, A. V. (2021). Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)*, 119-124. https://doi.org/10.1109/SISY52375.2021.9582508

[12] Sung, S. H., Lee, K. B., & Park, S. H. (2020). Research on Korea text recognition in images using deep learning. *Journal of the Korea Convergence Society, 11*(6), 1-6.

[13] Lee, D. S. & Kwon, S. K. (2021). Methods of classification and character recognition for table items through deep learning. *Journal of Korea Multimedia Society, 24*(5), 651-658.

[14] Cho, T. H. (2020). *Python Machine Learning (2nd ed.)*. Gilbut Publishing.

[15] Cho, T. H. (2024). *Python Machine Learning (3rd ed.)*. Gilbut Publishing.

[16] Kim, J. H., Kim, N., & Won, C. S. (2022). Facial expression recognition with Swin Transformer. *arXiv Preprint*.

[17] Narayan, K., Vibashan, V. S., Chellappa, R., & Patel, V. M. (2024). FaceXFormer: A unified transformer for facial analysis. *arXiv Preprint.*

[18] Vats, A. & Chadha, A. (2023). Facial expression recognition using squeeze and excitation-powered Swin Transformers. *arXiv Preprint*.

[19] Qin, L., Wang, M., Deng, C., Wang, K., Chen, X., Hu, J., & Deng, W. (2023). SwinFace: A multi-task transformer for face recognition, expression recognition, age estimation, and attribute estimation. *arXiv Preprint*.

[20] Park, H. S. (2023). *Coding: Python to Awaken Your Brain*. Hanbit Media.

[21] Park, H. S. (2023). *Learning Generative AI by Building It*. Hanbit Media.

[22] Hur, K. M. & Kang, S. M. (2014). Facial expression recognition technology. *Journal of Institute of Control, Robotics and Systems, 20*(2), 39-45.

[23] Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(6), 681-685. https://doi.org/10.1109/34.927467

[24]Matthew, N. D., Garrison, W., Curtis, P., & Ralph, A. (2002). EMPATH: A neural network that categorizes facial expressions. *Journal of Cognitive Neuroscience, 14*(8), 1158-1173. https://doi.org/10.1162/089892902760807177

[25] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 971-987. https://doi.org/10.1109/TPAMI.2002.1017623

[26] Kang, H., Lim, K. T., & Won, C. (2017). Learning directional LBP features and discriminative feature regions for facial expression recognition. *Journal of Korea Multimedia Society, 20*(5), 748-757. https://doi.org/10.9717/kmms.2017.20.5.748

[27] Lim, K. T. & Won, C. (2016). Face image analysis using Adaboost learning and non-square differential LBP. *Journal of Korea Multimedia Society, 19*(6), 1014-1023. https://doi.org/10.9717/kmms.2016.19.6.1014

[28] Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of 7th International Conference on Computer Vision (ICCV'99)*, 1150-1157. https://doi.org/10.1109/ICCV.1999.790410

[29] Lowe, D. G. (2004). Distinctive image features from scale-invariant key points. *International Journal of Computer Vision, 60*(2), 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[30] Ahonen, T., Rahtu, E., Ojansivu, V., & Heikkilä, J. (2008). Recognition of blurred faces using local phase quantization. *2008 19th International Conference on Pattern Recognition, Tampa, FL, USA*, 1-4. https://doi.org/10.1109/ICPR.2008.4761847

[31] Viola, P. & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1*, 511-518. https://doi.org/10.1109/CVPR.2001.990517

[32] Kim, S. J. (2015). Design of efficient gradient orientation bin and weight calculation circuit for HOG feature calculation. *Journal of The Institute of Electronics and Information Engineers, 51*(11), 66-72. https://doi.org/10.5573/ieie.2014.51.11.066

[33] Ozuysal, M., Fua, P., & Lepetit, V. (2007). Fast keypoint recognition in ten lines of code. *2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA*, 1-8. https://doi.org/10.1109/CVPR.2007.383123

[34] Ozuysal, M., Calonder, M., Lepetit, V., & Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(3), 448-461. https://doi.org/10.1109/TPAMI.2009.23

[35] Froba, B. & Ernst, A. (2004). Face detection with the modified census transform. *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea*, 91-96. https://doi.org/10.1109/AFGR.2004.1301514

[36] Zabih, R. & Woodfill, J. (1994). Non-parametric local transform for computing visual correspondence. *ECCV '94: Proceedings of the Third European Conference on Computer Vision (Vol. II)*, 151-158. https://doi.org/10.1007/BFb0028345

[37] Freund, Y. & Schapire, R. E. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence, 14*(5), 771-780.

[38] Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297. https://doi.org/10.1007/BF00994018

[39] Vapnik, V. (2000). *The nature of statistical learning theory (2nd ed.)*. Springer-Verlag.

[40] Choi, S. U., Kim, J. D., Lee, S. H., & Ko, H. S. (2022). Facial expression recognition using convolution neural network applied with attention module and multi-feature fusion. *2022 Korean Institute of Communications and Information Sciences (KICS) Winter Conference, 77*(1), 1677-1678.

[41] Bengio, Y. (2009). Learning deep architectures for AI. *Now Foundations and Trends in Machine Learning, 2*(1), 1-127. https://doi.org/10.1561/2200000006

[42] Kim, H. I., Moon, J. Y., & Park, J. Y. (2018). Research trends for deep learning-based high performance face recognition technology. *Electronics and Telecommunications Trends, 33*(4), 43-53. https:// 10.22648/ETRI.2018.J.330405

[43] Hwang, W. J. (2017). Research trends in deep learning-based face detection, landmark detection, and face recognition. *Broadcasting and Media Magazine, 22*(4), 41-49.

[44] Lee, J. Y., Lee, S. W., Won, J. M., & Shin, D. R. (2017). Face recognition system using machine learning. *Proceedings of the Korean Society of Computer Information Conference, 25*(21), 137-140.

[45] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84-90. https://doi.org/10.1145/3065386

[46] Son, D. Y. & Lee, K. K. (2017). A study on the recognition of face based on CNN algorithms. *Korean Journal of Artificial Intelligence, 5*(2), 15-25. https://doi.org/10.24225/kjai.2017.5.2.15

[47] Kyung, J. H. (2023). Comparison of deep learning models for judging business card image rotation. *Journal of the Korea Institute of Information and Communication Engineering, 27*(1), 34-40. https://doi.org/10.6109/jkiice.2023.27.1.34

[48] Lee, D., Sun, Y. G., Kim, S. H., Sim, I., Lee, K. S., Song, M. N., & Kim, J. Y. (2020). CNN-based image rotation correction algorithm to improve image recognition rate. *The Journal of the Institute of Internet, Broadcasting and Communication, 20*(1), 225-229. https://doi.org/10.7236/JIIBC.2020.20.1.225

[49] Ahn, J. S. & Cho, K. H. (2019). Facial expression recognition using CNN. *Proceedings of the 2019 Institute of Control, Robotics and Systems Conference*, 399-400.

[50] Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations (ICLR 2015)*, 1-14.

[51] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[52] Huang, G., Liu, Z., Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks**.** *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 4700-4708. https://doi.org/10.48550/arXiv.1608.06993

[53] Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 1492-1500. https://doi.org/10.48550/arXiv.1611.05431

[54] Choi, I. K., Ahn, H. E., Song, H., & Ko, M. S. (2016). CNN-based facial expression recognition**.** *2016 Korean Society of Broadcast Engineers (KSBE) Summer Conference*, 271-272.

[55] Choi, I. K., Song, H., Lee, S., & Yoo, J. (2017). Facial expression classification using deep convolutional neural network. *Korean Institute of Broadcast and Media Engineers*, *22*(2), 162-171. https://doi.org/10.5909/JBE.2017.22.2.162

[56] Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Placid, NY, USA, 1-10. https://doi.org/10.48550/arXiv.1511.04110

[57] Jung, H., Lee, S., Yim, J., Park, S., & Kim, J. (2015). Joint fine-tuning in deep neural networks for facial expression recognition. *2015 IEEE International Conference on Computer Vision*, 2983-2991. https://doi.org/10.1109/ICCV.2015.341

[58] Lopes, A. T., Aguiar, E., & Oliveira-Santos, T. (2015). A facial expression recognition system using convolutional networks. *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, Salvador, Brazil, 273-280. https://doi.org/10.1109/SIBGRAPI.2015.14

[59] Hamester, D., Barros, P., & Wermter, S. (2015). Face expression recognition with a 2-channel convolutional neural network. *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 1-8. https://doi.org/10.1109/IJCNN.2015.7280539

[60] Math Works. (2024). Convolutional neural network. Retrieved from https://kr.mathworks.com/discovery/convolutional-neural-network.html?ef_id=EAIaIQobChMI64na0L6xhgMVtm4PAh272QbTEAAYASAAEgKXOvD_BwE:G:s&s_kwcid=AL!8664!3!650716963097!p!!g!!convolutional%20neural%20network&s_eid=psn_136154680972&q=convolutional%20neural%20network&gad_source=1&gclid=EAIaIQobChMI64na0L6xhgMVtm4PAh272QbTEAAYASAAEgKXOvD_BwE.

[61] IDPLab-Konkuk. (2024). CNN structure. Retrieved from https://idplab-konkuk.tistory.com/13.

[62] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(56), 1929-1958. https://doi.org/10.5555/2627435.2670313

[63] Wikidocs. (2014). Dropout in neural networks. Retrieved from https://wikidocs.net/196790

[64] Tan, M. & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 2019 36th International Conference on Machine Learning*, PMLR 97:6105-6114.

[65] Kim, H. J. & Yu, S. H. (2023). *Machine Learning and Deep Learning for Artificial Intelligence with Python*. Gilbut Publishing.

[66] Park, H. S. (2020a). *Machine Learning Textbook with Python, Scikit-Learn, and TensorFlow*. Gilbut Publishing.

[67] Park, H. S. (2020b). *Self-Study Machine Learning + Deep Learning*. Hanbit Media.

[68] Park, H. S. (2021). *Learning Generative AI by Building It (2nd ed.)*. Hanbit Media.

[69] Ban, B. H. (2021). *Easy Deep Learning: Understanding Without Math and Statistics*. Saengneung Publishing Co.

[70] Cho, J. M. (2021). *Big Data Analysis and Artificial Intelligence with Python*. Infinity Books.

**Contact information:**

**Kyoungjong PARK**
(Corresponding author)
Department of Business Administration,
Gwangju University, 277 Hyodeok-ro, Nam-gu,
Gwangju 61743, Korea
E-mail: kjpark@gwangju.ac.kr