Modeling Emotion Recognition System from Facial Images Using Convolutional Neural Networks

Jasen Wanardi Kusno[1] and Andry Chowanda[2*]
[1]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University
Jakarta, Indonesia 11480
[1,2]Computer Science Department, School of Computer Science, Bina Nusantara University
Jakarta, Indonesia 11480
Email: [1]jasen.kusno@binus.ac.id, [2]achowanda@binus.edu

***Abstract*—Emotion classification is the process of identifying human emotions. Implementing technology to help people with emotional classification is considered a relatively popular research field. Until now, most of the work has been done to automate the recognition of facial cues (e.g., expressions) from several modalities (e.g., image, video, audio, and text). Deep learning architecture such as Convolutional Neural Networks (CNN) demonstrates promising results for emotion recognition. The research aims to build a CNN model while improving accuracy and performance. Two models are proposed in the research with some hyperparameter tuning followed by two datasets and other existing architecture that will be used and compared with the proposed architecture. The two datasets used are Facial Expression Recognition 2013 (FER2013) and Extended Cohn-Kanade (CK+), both of which are commonly used datasets in FER. In addition, the proposed model is compared with the previous model using the same setting and dataset. The result shows that the proposed models with the CK+ dataset gain higher accuracy, while some models with the FER2013 dataset have lower accuracy compared to previous research. The model trained with the FER2013 dataset has lower accuracy because of overfitting. Meanwhile, the model trained with CK+ has no overfitting problem. The research mainly explores the CNN model due to limited resources and time.**

***Index Terms*—Emotion Recognition, Facial Images, Convolutional Neural Networks**

I. Introduction

EMOTIONS and moods are mental states in humans. Emotions are generally often associated with affective behaviors and feelings. During interactions, emotions and moods can be one of the social signals used to communicate with other humans. Emotions can be captured and processed using Artificial Intelligence (AI) and machine learning techniques. Most techniques process visual data from input and search for general patterns present in human faces (i.e., facial expressions) from images or videos. Face detection and recognition can also be used for surveillance purposes by law enforcers and crowd management.

Moreover, emotion recognition models can be implemented in several real-world applications, such as affective systems, chatbots, virtual humans, stress detectors, and depression recognition. Emotions can be recognized from several social signals or cues, such as body gestures, facial expressions, and speech. With the rise of the deep learning method, the accuracy of facial expressions has tremendously advanced. For example, in previous research [1], Fusion Convolutional Neural Networks (CNN) and texture features for emotion recognition use Support Vector Machines (SVM). CNN networks have multiple layers, such as convolutional, activation, and pooling. They result in a map called a feature map. Many researchers have proposed different architectures to solve different problems in recognition tasks. However, CNN training and classification process require tremendous resources to execute. Hence, devices with small or limited computational power are not suitable for implementing this deep learning method.

FACS, which stands for Facial Action Coding System, is a comprehensive system and has been widely implemented to describe facial activity objectively. Previous research [2] develops the FACS, which is used for describing facial expressions by Action Units (AUs) and is revised in another previous research [3].

Received: Aug. 16, 2022; received in revised form: Feb. 13, 2023; accepted: Feb. 13, 2023; available online: Oct. 04, 2024.
*Corresponding Author

In the revision, 32 atomic are specified as facial muscle actions, called AUs, and 14 Action Descriptors (ADs) in addition that are used for various actions such as head pose, jaw thrust, and many more. The research only discusses AUs because they describe the muscle-based atomic facial actions. When a person expresses happiness, lips are pulled back (which detects AU12), cheeks are raised (which detects AU6), and wrinkles are detected around the eyes area (which detects AU1 and AU2). With training and testing through machine learning techniques, when all those patterns are detected, they will be automatically mapped into emotions.

The research uses Facial Expression Recognition 2013 (FER2013) and Extended Cohn-Kanade (CK+) datasets to detect seven types of emotions. The FER2013 dataset classifies facial expressions from 38,685 examples of 48×48-pixel greyscale images, and CK+ is a cropped version of CK [4], which contains 48 datasets with 5 emotions. Both FER2013 and CK+ datasets are commonly used datasets for FER.

In addition, the research aims to explore a deep learning model (i.e., CNN). By exploring the CNN model, the research will improve FER's accuracy. The model that will be proposed is then hyperparameter tuned and compared with the existing model. The model proposed is a greyscale channel known as a one-dimensional channel.

The motivation for the research is obtained from much research on emotion detection. The performance that other studies have done also varies. Therefore, by conducting the research, it is hoped that it can improve the research performance that other researchers have done. The research mainly explores the usage of the CNN model for the FER topic.

A. Related Works

Several researchers proposed the CNN method for facial emotion recognition, like [5], with Japanese Female Face Expression (JAFFE) and FER2013. Previous research uses the CK+ datasets. With many researchers using CNN architecture, another previous research [6] adds a Gabor filter before inputting the image into CNN. In addition, it adds two Gabor filters in preprocessing, which then the output is fed into the convolution layer. CNN can also be implemented into a web application, where the user inputs an image into the web, and after a few seconds, the web will give an emotional result [7].

Another implementation [8] uses a facial emotion recognition application to help teachers detect students' emotions while they are presenting. Modifying CNN architecture can also gain higher accuracy compared to the simple CNN architecture. This method is done by proposing Venturi architecture [9].

Like [10–12], the other researchers use a simple CNN method for facial emotion recognition. For example, CNN and Recurrent Neural Networks (RNN) are proposed by combining models to extract the relations in images [13, 14]. The recurrent network is utilized for the temporal dependencies within the images to be considered during the classification. This experiment also uses two datasets which are JAFFE and M&M Initiative (MMI). Electroencephalogram (EEG) signals can also be recognized using CNN. Similar research contains EEG signals using the Database for Emotion Analysis using Physiological Signals (DEAP) dataset [15]. The dataset contains 32 subjects, and each subject watches around 40 one-minute music videos.

Transfer Learning (TL) can also be another method for facial emotion recognition. Researchers like [10, 16, 17] use the TL method for facial emotion recognition. In [16], they use VGG19, Resnet50, MobileNet, and Inception V3. Meanwhile, in [17], they use VGG16, VGG19, Resnet18, Resnet34, Resnet50, Inception V3, and DenseNet161. TL methods are pre-trained models trained with the EmotioNet dataset.

Attention is also another method for detecting emotion. The previous researchers propose BiLSTM with multimodal consideration to arrange a system for a more efficient FER [18]. First, each channel will provide physiological signals which are transformed into spectrogram images (which will capture time and frequency data). Next, the best temporal features are automatically learned by utilizing Attention-based Bidirectional Long Short-Term Memory-Recurrent Neural Networks (LSTM-RNN). With the LSTM-RNN method, deep features will be gained and fed into a Deep Neural Network (DNN), resulting in a prediction probability of emotional output for each channel. The final step is utilizing the decision-level fusion technique to predict the final emotion. A Dataset for Affect, Personality and Mood Research on Individuals and Groups (AMIGOS) datasets are used for the research [18].

There are many methods to detect facial emotion recognition, but the researchers will only use simple CNN architecture in the research. Several researchers, like [6], use a simple CNN method while adding a Gabor filter before training the dataset. At the same time, previous research [9] modifies simple CNN architecture and proposes Venturi architecture to improve accuracy and performance. Another method, like TL, can also be used and proven to have higher accuracy than the CNN model.

TABLE I
OVERVIEW OF THE FER2013 DATASET.

Emotions	Training	Test	Total
Angry	3,995	958	4,953
Disgusted	436	111	547
Fearful	4,097	1,024	5,121
Happy	7,215	1,774	8,989
Neutral	4,965	1,233	6,198
Sad	4,830	1,247	6,077
Surprised	3,171	831	4,002

TABLE II
OVERVIEW OF CK+ DATASET.

Emotions	Training	Test	Total
Angry	90	45	135
Contempt	36	18	54
Disgusted	118	59	177
Fearful	50	25	75
Happy	138	69	207
Sad	56	28	84
Surprise	166	83	249

II. RESEARCH METHOD

The model uses the CNN model as a baseline based on previous research [19], which will use hyperparameter tuning to gain better accuracy for the CK+ dataset. Meanwhile, the FER2013 model uses the model by [5] with the same setting as the CK+ model. The datasets used are FER2013 and CK+. Both datasets are being collected from Kaggle (the data can be access through these links (https://www.kaggle.com/datasets/shuvoalok/ck-dataset and https://www.kaggle.com/datasets/nicolejyt/facialexpressionrecognition). The FER2013 dataset comprises 35,887 face images with 48×48 in width and height. The dataset is annotated with six basic emotions (i.e., angry, fearful, happy, disgusted, surprised, and sad) plus one neutral emotion. The dataset is then split into 80% for training and 20% for testing and validation. Meanwhile, the CK+ dataset consists of 981 grayscale images with a 48×48 sized face in width and height with seven emotions (angry, contempt, disgusted, fearful, happy, sad, and surprised). The dataset is also split into training and testing. The overview of each dataset is shown in Tables I and II.

In the FER2013 dataset, happy is the majority class with a total of 8,989 data (7,215 for training and 1,774 for testing). Moreover, the disgusted is the minority class with a total of 547 data (436 for training and 111 for testing). Meanwhile, in the CK+ dataset the majority class is surprise class with a total of 249 data (with 166 for training and 83 for testing). Moreover, contempt is the minority class with a total of 54 data (36 for training and 18 for testing).

In the research to predict facial emotion, two CNN models are proposed. The model is proposed by exploring the deep learning model from the baseline paper [5, 19]. The research also uses deep learning models that are not too deep by changing the filter of convolutional layers and adding a dropout layer, max pooling layer, and batch normalization layer. Dropout layers are used to ignore random neurons in the training process. Then, max pooling layers are operations that calculate a maximum value in feature maps and create pooled feature maps. Meanwhile, batch normalization layers are used to normalize the input of each layer.

The optimization algorithm that will be used for training neural networks is the Adaptive Moment estimation (Adam) optimizer which is used to reduce losses by changing the weight and learning rate of neural networks. Adam optimization works in first and second-order momentum. Adam optimizers need to decrease the velocity a little bit for a careful search and not roll too fast. Then, it will result in jumping to the minimum. The $M(t)$ will also be kept by the Adam optimizer. The $M(t)$ represents the bias-corrected first moment (mean) capturing the direction of the gradients and $V(t)$ represents the bias-corrected second moment (variance), scaling the learning rate based on the magnitude of the gradient. The equations for $M(t)$ and $V(t)$ are shown in Eqs. (1) and (2).

If the mean of $M(t)$ and $V(t)$ are taken, the update of parameters θ_{t+1} is updated uses Eq. (3). It shows η as the step size or learning rate, and the ϵ normalizes the parameter updates to prevent division by zero (generally set to 10^{-8})

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \tag{1}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}, \tag{2}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}\hat{m}_t. \tag{3}$$

Model A (the first model) consists of five convolutional layers, and model B (the second model–the eighth model) has four convolutional layers. There is only one model in Model A because the model is too complex, causing overfit, while Model B mainly explores the layer pixel. In model B, the convolutional layers are hyperparameter tuned, which results in seven other models with different filter sizes.

Hopefully, the eight proposed models will gain better accuracy than the previous model. Therefore, the proposed models are evaluated for effectiveness, and their performance and accuracy are compared. On the other hand, FER systems have five steps: illumination normalization, face detection, face registration, feature

TABLE III
OVERVIEW OF THE PROPOSED MODELS.

Model	Layer
1st	5 convolutional layers (32, 64, 128, 256, 512), 4 max pooling layers, 1 Fully Connected (FC) layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.
2nd	4 convolutional layers (32, 64, 128, 256), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.
3rd	4 convolutional layers (64, 64, 128, 128), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.
4th	4 convolutional layers (64, 32, 64, 32), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.
5th	4 convolutional layers (32, 64, 32, 64), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.
6th	4 convolutional layers (32, 64, 128, 128), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.
7th	4 convolutional layers (32, 64, 128, 256), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for the FC layer.
8th	4 convolutional layers (32, 32, 64, 64), 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization.

TABLE IV
CONVOLUTIONAL NEURAL NETWORKS (CNN) ARCHITECTURE IN THE THIRD MODEL FOR FER2013 DATASET.

Layer (type)	Output Shape	Parameter Number
conv2d (Conv2D)	(None, 46, 46, 64)	640
conv2d_1 (Conv2D)	(None, 44, 44, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 22, 22, 64)	0
dropout (Dropout)	(None, 22, 22, 64)	0
conv2d_2 (Conv2D)	(None, 20, 20, 128)	73856
max_pooling2d_1 (MaxPooling2DP)	(None, 10, 10, 128)	0
dropout_1 (Dropout)	(None, 10, 10, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
batch_normalization (BatchNormalization)	(None, 1024)	4096
dropout_3 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 7)	7175

TABLE V
CONVOLUTIONAL NEURAL NETWORKS (CNN) ARCHITECTURE IN THE FOURTH MODEL FOR FER2013 DATASET.

Layer (type)	Output Shape	Parameter Number
conv2d (Conv2D)	(None, 46, 46, 64)	640
conv2d_1 (Conv2D)	(None, 44, 44, 32)	18464
max_pooling2d (MaxPooling2D)	(None, 22, 22, 32)	0
dropout (Dropout)	(None, 22, 22, 32)	0
conv2d_2 (Conv2D)	(None, 20, 20, 64)	18496
max_pooling2d_1 (MaxPooling2DP)	(None, 10, 10, 64)	0
dropout_1 (Dropout)	(None, 10, 10, 64)	0
conv2d_3 (Conv2D)	(None, 8, 8, 32)	18464
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 32)	0
dropout_2 (Dropout)	(None, 4, 4, 32)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 1024)	525312
batch_normalization (BatchNormalization)	(None, 1024)	4096
dropout_3 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 7)	7175

extraction, and classification regression [20]. The proposed models are evaluated for their effectiveness and compared for their performance. Table III illustrates the eight CNN architectures proposed to model emotion recognition using CK+ and FER2013 datasets.

Table IV demonstrates one of the CNN architecture examples for training emotion recognition models with the FER2013 dataset. The models for FER2013 consist of 4 convolutional layers with 64 filters in the first convolutional layer, 64 filters in the second convolutional layer, 128 in the third convolutional layer, and 128 in the fourth convolutional layer, 3 max pooling layers, 1 Fully Connected (FC) layer with 4 dropouts for convolutional layers, and 1 for the FC layer with the addition of batch normalization layer.

Moreover, Table V illustrates another one of the FER2013 architectures examples to train the emotion recognition models with the FER2013 dataset. The model for FER2013 consists of 4 convolutional layers with 64 filters in the first convolutional layer, 32 filters in the second convolutional layer, 64 in the third convolutional layer, and 32 in the fourth convolutional layer, 3 max pooling layers, 1 FC layer with 4 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization layer.

Table VI demonstrates one of the CNN architectures examples to train the emotion recognition models with the CK+ dataset. The models for CK+ consist of 4 convolutional layers with 32 filters in the first convolutional layer, 64 filters in the second convolutional layer, 128 in the third convolutional layer, and 256 in the fourth convolutional layer, 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for the FC layer with the addition of batch

TABLE VI
CONVOLUTIONAL NEURAL NETWORKS (CNN) ARCHITECTURE IN THE SECOND MODEL FOR CK+ DATASET.

Layer (type)		Output Shape	Parameter Number
conv2d (Conv2D)		(None, 46, 46, 32)	320
conv2d_1 (Conv2D)		(None, 44, 44, 64)	18496
max_pooling2d (MaxPooling2D)		(None, 22, 22, 64)	0
dropout (Dropout)		(None, 22, 22, 64)	0
conv2d_2 (Conv2D)		(None, 20, 20, 128)	73856
max_pooling2d_1 (MaxPool-ing2DP)		(None, 10, 10, 128)	0
conv2d_3 (Conv2D)		(None, 8, 8, 256)	295168
max_pooling2d_2 (MaxPool-ing2D)		(None, 4, 4, 256)	0
dropout_1 (Dropout)		(None, 4, 4, 256)	0
flatten (Flatten)		(None, 4096)	0
dense (Dense)		(None, 1024)	4195328
batch_normalization (BatchNor-malization)		(None, 1024)	4096
dropout_2 (Dropout)		(None, 1024)	0
dense_1 (Dense)		(None, 7)	7175

TABLE VII
CONVOLUTIONAL NEURAL NETWORKS (CNN) ARCHITECTURE IN THE SIXTH MODEL FOR CK+ DATASET.

Layer (type)		Output Shape	Parameter Number
conv2d (Conv2D)		(None, 46, 46, 32)	320
conv2d_1 (Conv2D)		(None, 44, 44, 64)	18496
max_pooling2d (MaxPooling2D)		(None, 22, 22, 64)	0
dropout (Dropout)		(None, 22, 22, 64)	0
conv2d_2 (Conv2D)		(None, 20, 20, 128)	73856
max_pooling2d_1 (MaxPool-ing2DP)		(None, 10, 10, 128)	0
conv2d_3 (Conv2D)		(None, 8, 8, 128)	147584
max_pooling2d_2 (MaxPool-ing2D)		(None, 4, 4, 128)	0
dropout_1 (Dropout)		(None, 4, 4, 128)	0
flatten (Flatten)		(None, 2048)	0
dense (Dense)		(None, 1024)	2098176
batch_normalization (BatchNor-malization)		(None, 1024)	4096
dropout_2 (Dropout)		(None, 1024)	0
dense_1 (Dense)		(None, 7)	7175

normalization layer.

Moreover, Table VII illustrates one of the CNN architectures examples to train the emotion recognition models with the CK+ dataset. The model for CK+ consists of 4 convolutional layers with 32 filters in the first convolutional layer, 64 filters in the second convolutional layer, 128 in the third convolutional layer, 128 in the fourth convolutional layer, 3 max pooling layers, 1 FC layer with 3 dropouts for convolutional layers, and 1 for FC layer with the addition of batch normalization layer.

The proposed architecture receives input with $48\times48\times1$ tensor dimension, and all models are trained with Adam optimizer with 50 epochs and 0.0001 learning rate. The proposed architecture from Table IV has 590,599 trainable parameters (a total of 592,647 parameters and 2,048 non-trainable parameters). Then,

TABLE VIII
THE OVERALL RESULT OF THE FER2013 DATASET.

Model	Accuracy	Loss	VAcc	VLoss
1st	0.9200	0.2264	0.6342	1.4309
2nd	**0.9254**	**0.2136**	0.6295	1.4523
3rd	0.8985	0.2874	**0.6343**	1.3078
4th	0.6194	1.0142	0.5961	**1.0553**

Table V shows 2,366,407 trainable parameters (a total of 2,368,455 and 2,048 non-trainable parameters). The first convolutional layer extracts the feature from the input and passes it through the next convolutional layers and the max-pooling to reduce the dimension of the features, the Rectified Linear Unit (ReLU) activation function, and dropout layers. The next process is repeating the previous parsing. Then, the classification layers, which have a Softmax function for the activation layer and flatten, are normalized with batch normalization and a dropout layer. Table VI has 4,592,391 trainable parameters (total parameters of 4,594,439 and 2,048 non-trainable parameters). Finally, the architecture in Table VII has 2,347,655 parameters with a total of 2,349,703 and 2,048 non-trainable parameters.

III. RESULTS AND DISCUSSION

Two datasets (FER2013 and CK+) are used to model emotion recognition using eight architectures proposed in Table III. The models are being trained with NVIDIA GPU RTX 3070 8GB for all architectures, and some models, such as the first, second, and third models with the FER2013 dataset, are trained using Google Colab (using NVIDIA TESLA K80). Overall, the FER2013 dataset has a longer training time compared to the CK+ dataset. In addition, the models trained with the FER2013 dataset result in a relatively high overfitting problem. Hence, in the research, the training stops at the fourth model instead of using all eight models. Meanwhile, the CK+ dataset results in good models, and all eight architectures proposed in Table III are used to train the models. Table VIII illustrates the overall results of the FER2013 dataset, followed by model accuracy and model loss in Figs. 1 and 2.

In Table VIII, the model column indicates the model settings. The accuracy column represents the training accuracy, and the loss column is the training loss. Then, the VAcc column is the validation accuracy, and the VLoss column means the validation loss. The best training accuracy score and training loss are achieved by the second model with 0.9254, 0.2136, 0.62, and 1.4523 for training accuracy, training loss, validation accuracy, and validation loss, respectively. Meanwhile, the third model achieves the best validation accuracy

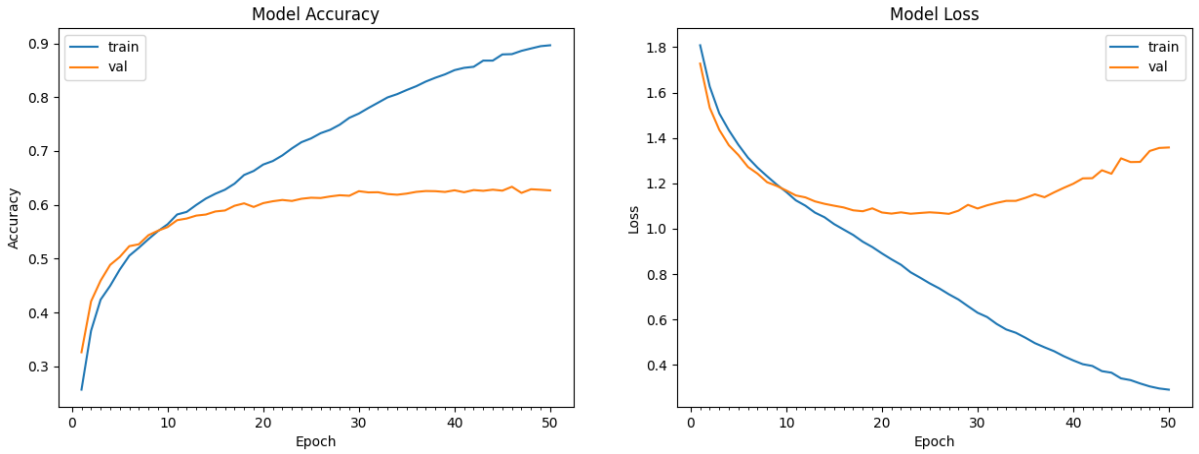

Fig. 1. Illustration model of accuracy and loss of the third model trained with the FER2013 dataset.

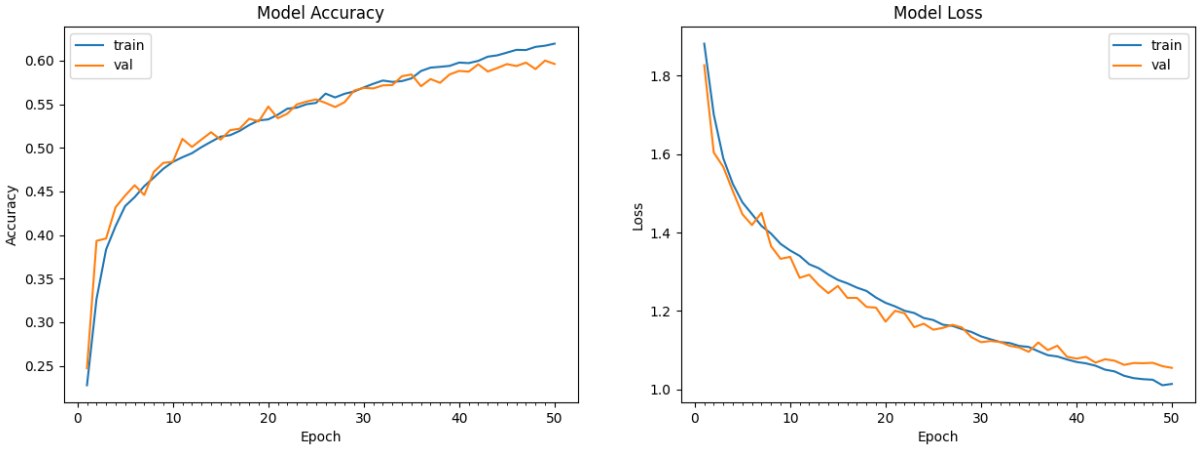

Fig. 2. Illustration model of accuracy and loss of the fourth model trained with the FER2013 dataset.

score with 0.8985, 0.2874, 0.6343, and 1.3078 for training accuracy, training loss, validation accuracy, and validation loss, respectively. However, the second and third models are highly overfitted.

From the results of the models proposed in the experiment using the FER2013 dataset, the model exhibits lower accuracy than the baseline, with Model A and Model B achieving scores of 70% and 67%, respectively. However, the proposed model obtains the highest score of 63%. When tested on the CK+ dataset, the sixth model records the highest accuracy at 98.98%. The fourth model performs worse than the FER2013 dataset, with an accuracy score of 61.94%.

The first and second models perform the best during training. However, both show signs of overfitting due to higher validation losses and lower validation accuracies to overview the experiment using the FER2013 dataset. The third model has slightly lower training performance, but it may generalize a bit better than the first two models based on validation loss. Then, the fourth model is the worst performer, with poor accuracy and high loss in both training and validation, indicating significant issues in learning the data. This result suggests that none of the models are perfectly balanced between training and validation, with most showing overfitting tendencies except for the fourth model, which underperforms overall.

In Fig. 1, it can be seen that the training loss is constantly decreasing, but the validation loss is not. It means that the model is complex enough to memorize the patterns in the training data. Regularization, like reducing the neural network layer, adding more dropouts and tuning its rate, and adding more data for training, may help to reduce the overfitting problem. In Fig. 2, the training and validation losses are relatively the same as the loss and accuracy. Therefore, it results

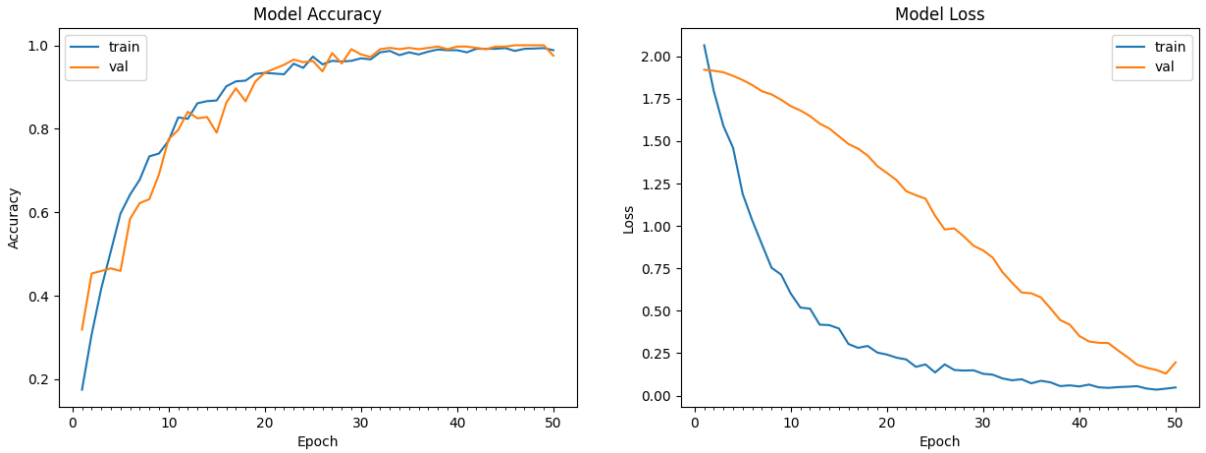

Fig. 3. Illustration model of accuracy and loss of the second model trained with CK+ dataset.

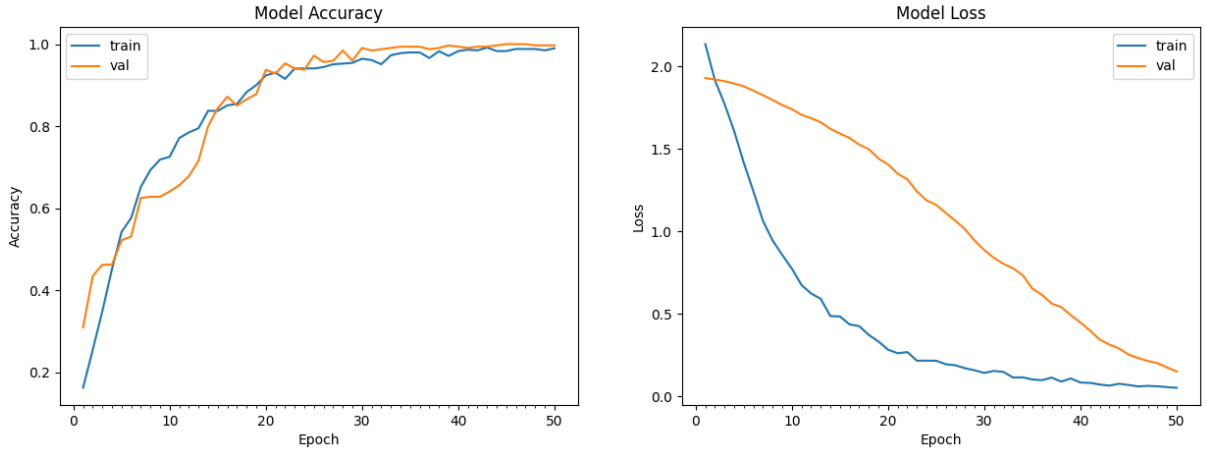

Fig. 4. Illustration model of accuracy and loss of the sixth model trained with CK+ dataset.

TABLE IX
THE OVERALL RESULTS OF THE CK+ DATASET.

Model	Accuracy	Loss	VAcc	VLoss
1st	0.8500	0.3882	0.8969	0.3539
2nd	0.9881	**0.0481**	0.9750	0.1957
3rd	0.9254	0.2415	0.9563	0.1698
4th	0.8695	0.3826	0.9281	0.5769
5th	0.8525	0.4328	0.9156	0.6499
6th	**0.9898**	0.0513	**0.9969**	**0.1504**
7th	0.9203	0.2458	0.9438	0.1952
8th	0.8932	0.3191	0.9625	0.4901

in a more stable result. Hence, the best result of the model trained with the FER2013 dataset is the fourth model with more stable results between training and validation accuracy.

Table IX illustrates the best results from the models trained with the CK+ dataset, followed by model accuracy and model loss (also shown in Figs. 3 and 4). The sixth model achieves the best model with 0.9898, 0.0513, 0.9969, and 0.1504 for training accuracy, training loss, validation accuracy, and validation loss, respectively. However, the second model achieves the best training loss with 0.0481 of training loss. In Figs. 3 and 4, the validation accuracy and accuracy are relatively the same, while the validation loss and loss have a different result. This problem may occur because of the overfitting problem.

IV. CONCLUSION

The research proposes several CNN architectures to train the emotion classification models and improve the model. The eight proposed architectures are evaluated with two datasets. The architecture model trained with the CK+ dataset gains higher accuracy compared with the previous model. In the FER2013 dataset, some models gain over fittings, such as the first, second,

and third models. After adding batch normalization in the fourth model, it reduces the overfitting. In the model with the FER2013 dataset, the model has lower accuracy compared with the baseline model, which gains 70% and 67% scores for Model A and Model B. Meanwhile, the proposed model gains the highest score of 63%. The sixth model achieves the best accuracy score with the CK+ dataset with a score of 0.9898, while the fourth model achieves worse accuracy than the FER2013 dataset with a score of 0.6194.

In the research, the models are only trained in one channel (i.e., the Grayscale channel). Therefore, with the data gained from this experiment, the goals in the research have been fulfilled, which are creating a CNN model for FER while exploring other previous models. Then, the proposed model with the CK+ dataset gains higher accuracy compared to the previous method. The other problem that occurs in the research is the overfitting problem, which can be resolved by regularization (i.e., reducing the layer of the neural network, adding more dropouts, and adding more datasets for training).

The model proposed in the research mainly explores the CNN model by only changing the convolutional layer filter and training with one channel dataset due to limited resources and time. Therefore, in future works, the model proposed can be trained by using different datasets (i.e., RGB images). Moreover, the model can also be implemented in the device (i.e., helping teachers to monitor their students).

Author Contribution

Writing—original draft, J. K.; Methodology, J. K.; Data curation, J. K.; Review and editing, A. C.; Providing resources, A. C. All authors have read and agreed to the published version of the manuscript.

References

[1] J. Sujanaa and S. Palanivel, "Fusion of deep-CNN and texture features for emotion recognition using support vector machines," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 5, pp. 286–291, 2021.

[2] P. Ekman and W. Friesen, *Facial action coding system: A technique for the measurement of facial movement*. Consulting Psychologists Press, 1978.

[3] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial action coding system*. Human Face, 2002.

[4] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, 2010, pp. 94–101.

[5] A. Jaiswal, A. K. Raju, and S. Deb, "Facial emotion detection using deep learning," in *2020 International Conference for Emerging Technology (INCET)*. IEEE, 2020, pp. 1–5.

[6] M. M. T. Zadeh, M. Imani, and B. Majidi, "Fast facial emotion recognition using convolutional neural networks and Gabor filters," in *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*. IEEE, 2019, pp. 577–581.

[7] M. A. Shejwadkar, C. M. D. Souza, V. B. M. Raj, and V. J. Fernandes, "Facial emotion recognition using convolutional neural network," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 7, pp. 288–290, 2021.

[8] I. Lasri, A. R. Solh, and M. El Belkacemi, "Facial emotion recognition of students using convolutional neural network," in *2019 Third International Conference on Intelligent Computing In Data Sciences (ICDS)*. IEEE, 2019, pp. 1–6.

[9] A. Verma, P. Singh, and J. S. R. Alex, "Modified convolutional neural network architecture analysis for facial emotion recognition," in *2019 International Conference on Systems, Signals And Image Processing (IWSSIP)*. IEEE, 2019, pp. 169–173.

[10] S. Modi and M. H. Bohara, "Facial emotion recognition using convolution neural network," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2021, pp. 1339–1344.

[11] R. Jadhav, J. Bhuke, and N. Patil, "Facial emotion detection using convolutional neural network," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 5, pp. 1077–1082, 2019.

[12] E. Pranav, S. Kamal, C. S. Chandran, and M. H. Supriya, "Facial emotion recognition using deep convolutional neural network," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2020, pp. 317–320.

[13] N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali, and M. Zareapoor, "Hybrid deep neural networks for face emotion recognition," *Pattern Recognition Letters*, vol. 115, pp. 101–106, 2018.

[14] A. Chowanda, "Separable convolutional neural networks for facial expressions recognition," *Journal of Big Data*, vol. 8, pp. 1–17, 2021.

[15] H. Yang, J. Han, and K. Min, "A multi-column

CNN model for emotion recognition from EEG signals," *Sensors*, vol. 19, no. 21, pp. 1–12, 2019.

[16] M. K. Chowdary, T. N. Nguyen, and D. J. Hemanth, "Deep learning-based facial emotion recognition for human–computer interaction applications," *Neural Computing and Applications*, vol. 35, no. 32, pp. 23 311–23 328, 2023.

[17] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, "Facial emotion recognition using transfer learning in the deep CNN," *Electronics*, vol. 10, no. 9, pp. 1–19, 2021.

[18] C. Li, Z. Bao, L. Li, and Z. Zhao, "Exploring temporal representations by leveraging attention-based bidirectional LSTM-RNNs for multi-modal emotion recognition," *Information Processing & Management*, vol. 57, no. 3, 2020.

[19] D. Y. Liliana, "Emotion recognition from facial expression using deep convolutional neural network," in *2018 International Conference of Computer and Informatics Engineering (IC2IE)*, vol. 1193. IOP Publishing, 2019, pp. 1–5.

[20] T. Mitchell, *Machine learning*. McGraw Hill, 1997.