Development of a person's facial expression recognition system using a convolutional neural network

B Siregar1, J S Wirtjes1, E B Nababan1 and Fahmi2

1 Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara
2 Department of Electrical Engineering, Faculty of Engineering, Universitas Sumatera Utara

E-mail: baihaqi@usu.ac.id

Abstract. Humans naturally and intuitively use important and powerful facial expressions to communicate and show their emotions during social interactions. In that interaction, the effort to interpret someone's emotional condition is important in good communication. A person's emotional condition is reflected in the form of words, gestures, and especially facial expressions. Although humans can recognize expressions well, research on expression recognition carried out by machines is continuously being carried out in order to be able to carry out expression recognition in human and computer interactions. This research conducted facial expression recognition using the Convolutional Neural Network method. To perform facial expression recognition, the Convolutional Neural Network is trained with expression image data. The training process is carried out using different optimizer values, batch size, and epoch to get the best model. To overcome overfitting, data augmentation was carried out on training data and validation data. The experimental results in this research indicate that the system method built is able to recognize a person's facial expressions with an accuracy rate of 80%.

1. Introduction
Humans naturally and intuitively use important and powerful facial expressions to communicate and show their emotions in social interactions. In that interaction, interpreting emotional conditions is important in good communication. Emotional conditions are reflected in speech, gestures, and especially facial expressions. Although humans can recognize expressions well, research expression recognition carried out by machines continues to be carried out in order to carry out expression recognition in human and computer interactions. The recognition of facial expressions from an image is one area that has attracted the attention of many researchers in recent times. Facial expression recognition can be applied to computer vision and pattern recognition.

Several studies related to expression recognition were carried out by Samarawickrame and Mindya in 2013 using Active Shape Models and Support Vector Machine (SVM) to classify expressions based on points of facial landmarks. The working stage begins with face detection using Haar Classifiers. The points of landmarks are extracted into a module to then be classified by SVM [1]. Subsequent research was carried out by Mliki et al. in 2013 using the Decision Tree. To solve the problem of variation and intensity of facial expressions, the researcher proposed a different feature extraction method.
Components of facial features such as eyes, eyebrows and mouth are detected and segmented automatically. The next stage is the extraction of points of interest and classification using SIPINA, the Decision Tree analysis software, based on a pre-existing set of rules [2]. In 2016, Voisan et al. carried out facial expression recognition using Constrained Local Models (CLM) and SVM. This study uses CLM to extract the points of facial landmarks, then uses a representation based on distance to make it more flexible when processing images that represent facial changes and SVM is used to detect emotions from facial expressions [3]. Another study conducted by Shan et al. did expression recognition in 2017 using Convolutional Neural Network (CNN). They performed face detection and histogram equalization feature extraction with the JAFFE and CK+ datasets, and compared their experiments using the K-Nearest Neighbor [4]. Other studies using CNN were conducted by Zufar and Setiyono in 2016. They used CNN as face recognition in real time with OpenCV for multi face detection [5].

2. Method
The data used for this study consisted of three types of data, namely training data used in the learning process, validation data used in the process of testing learning outcomes during training (validation) and testing data for testing after training. The data used in this study were 24,021 images consisting of four categories, namely anger, happiness, sadness, and surprise. The data used for the training process are 19,211 images spread over four categories. Data validation is data used to test accuracy when carrying out the training process. The data used for the validation process were 2,430 images spread over four categories. While testing data is data used to carry out the testing process of learning facial expressions that have been stored in a model. The data used is in the form of images from facial expressions that are different from the images that have been used in the training and validation processes. The testing data used has different color and size modes. In table 1, it can be seen that the proportions for training data and test data for each category of human facial expression.

| No. | Category | Training Data | Testing Data |
|-----|----------|---------------|--------------|
| 1   | Anger    | 3995          | 467          |
| 2   | Happiness| 7215          | 895          |
| 3   | Sadness  | 4830          | 653          |
| 4   | Surprise | 3171          | 415          |
| Total |         | 19211        | 2430         |

2.1. Facial expression recognition system
The method used to recognize facial expressions consists of several stages. The stages taken are pre-processing, the training stage, the classification stage using CNN, and the system testing stage. The general architecture used to help carry out this research can be seen in figure 1.

2.2. Pre-processing stage
At this pre-processing stage, the resize and grayscale process occurs. Resizing stage is needed to adjust the pixel size of the image that will be processed at the testing stage. The image input used in this study has a different pixel size. Therefore, it is necessary to carry out the resizing process so that each training data and testing data has the same dimension size and value range. The greater the number of pixels, the more time it takes to process the image. In this research, the dimensions of the image that were originally very diverse were changed to a size of 48 x 48 pixels. Next is to convert images that have RGB mode into grayscale mode, this is done because color characteristics are not the topic of this research.
2.3. **Classification stage**

The classification stage consists of several sub processes as follows:

2.3.1. **Determination of filter, pool size, stride, and padding.** The filter used is 3 x 3, while the pool size used is 2 x 2. The stride and padding settings are set to the default values, namely 1 and 0.

2.3.2. **Determination of the activation function.** The calculation results between input, weight and bias will be calculated again with the equation of the activation function to get the output of each layer. In this study, the ReLU activation function on the convolutional layer and the Softmax activation function on the output layer is used to get results in the form of categorical data. Systematically, the ReLU activation function can be seen in Equation 1.

\[
\sigma(a) = \max(0, a) \tag{1}
\]

---

**Figure 1.** General architecture.
Whereas the Softmax function follows Equation 2.

$$\sigma(a_j) = \frac{\exp(a_j)}{\sum_{k=1}^{m} \exp(a_k)}$$  \hspace{1cm} (2)

2.3.3. **Optimizer determination.** Optimizer is an algorithm for determining optimal weight. In this study, the optimizers used were Adam, SGD and RMSProp.

2.3.4. **Batch size determination.** The batch size is used to determine the number of observations made before changing the weight, which is determined relative to computer specifications. In this study, several batch sizes were used, namely the values 32 and 128.

2.3.5. **Epoch determination.** Epoch is the amount of literacy used to repeat the learning process. The larger the number of epochs will increase the level of learning outcomes. There are several epoch values used in this study, namely 50, 100, 200, and 300.

2.3.6. **Training process.** In the training process, the input image measuring 48 x 48 x 1 will first be processed by the first convolutional layer with a 64-kernel filter measuring 3 x 3 x 1 and the result is 46 x 46 x 64. In the first pooling layer, the dimensions are reduced from 46 to 23. In each pooling layer, the parameters do not change. Then, the results from the first convolutional layer measuring 23 x 23 x 64 will be processed by the second convolutional layer with a 128-kernel filter measuring 3 x 3 x 64 and the results are 21 x 21 x 128. In the second pooling layer, the dimensions are reduced from 21 to 10.

The results from the second convolutional layer with a size of 10 x 10 x 128 will be processed by the third convolutional layer with a 256-kernel filter measuring 3 x 3 x 128 and the results are 8 x 8 x 256. In the third pooling layer, the dimensions are reduced from 8 to 4. After the convolutional layer process, 2 fully connected layers were generated which had 256 neurons in the first layer and 4 neurons in the second layer. The training process at each layer using Deep CNN can be seen in table 2.

| Layer    | Size          | Parameter                          |
|----------|---------------|------------------------------------|
| Input    | 48 x 48 x 1   | 0                                  |
| Conv 1+ ReLU | 46 x 46 x 64 | (3 x 3 x 1 + 1) x 64 = 640        |
| Pool 1   | 23 x 23 x 64  | 0                                  |
| Norm     | 23 x 23 x 64  | 64 x 4 = 256                       |
| Conv 2+ ReLU | 21 x 21 x 128 | (3 x 3 x 64 + 1) x 128 = 73,856   |
| Pool 2   | 10 x 10 x 128 | 0                                  |
| Norm     | 10 x 10 x 128 | 128 x 4 = 512                      |
| Conv 3+ ReLU | 8 x 8 x 256 | (3 x 3 x 128 + 1) x 256 = 295,168 |
| Pool 3   | 4 x 4 x 256   | 0                                  |
| Norm     | 4 x 4 x 256   | 256 x 4 = 1024                     |
| Conv 4+ ReLU | 2 x 2 x 512 | (3 x 3 x 256 + 1) x 512 = 1,180,160|
| Pool 4   | 1 x 1 x 512   | 0                                  |
| Norm     | 1 x 1 x 512   | 512 x 4 = 2048                     |
| Dropout  | 1 x 1 x 512   | 0                                  |
| Flatten  | 512           | 0                                  |
| FC + Softmax | 256          | (512 + 1) x 256 = 131,328         |
| Dropout  | 256           | 0                                  |
| Output   | 4             | (256 + 1) x 4 = 1028              |

2.4. **Testing stage**

Neural networks that have been modelled and generated from the training process will be tested at this stage. This stage is carried out to determine the effectiveness of the convolutional neural network method in recognizing facial expressions.
2.5. Output
The final output of all stages is accuracy and loss information at the training and testing stages per epoch.

3. Result and Discussion
The dataset used is in the form of images provided in the Challenges in Representation Learning: Facial Expression Recognition Challenge activity held by Kaggle in 2013. The dataset consists of grayscale images of human faces with a size of 48 x 48 pixels. The dataset is then categorized into four emotional expressions, namely anger, happiness, sadness, and surprise. Examples of expressions of human faces in the dataset used can be seen in figures 2 to 5.

Figure 2. Image of face with angry expression.

Figure 3. Image of face with sad expression.
Figure 4. Image of face with happy expression.

Figure 5. Image of face with surprised expression.
3.1. Testing systems built
At this stage, the test data and the system built are tested. Tests were carried out on 467 test data for angry expression images, 895 images of happy expressions, 653 images of sad expressions, and 415 images of shocked expressions using training data as many as 3995 images of angry expressions, 7215 images of happy expressions, 4830 images of sad expressions, and 3171 images of shocked expressions.

Tests were carried out using the epoch value parameters of 50 and 100, learning rate = 0.001, output layer = 4. Adjustments were also made by changing the batch size value, optimizer, and the number of convolutional layers. The batch size values used are 32 and 128. The optimizers used are Adam, Stochastic Gradient Descent, and RMSprop. The number of convolutional layers tested were 2 and 4 layers. The results of the test can be seen in figure 6 and figure 7.

Based on the results of the first test using two convolutional layers as shown in figure 6, it was found that the accuracy achieved was still not good. In this result, overfitting also occurred, which was due to the insufficient amount of training data and validation data which affected the level of accuracy, with the result that the accuracy of the training data was much higher than the test data. Therefore, the second test was carried out, namely performing data augmentation on the model test using 4 convolutional layers and reducing the learning rate during the training process.

Data augmentation is a technique for transforming images and using original and transformed data as training data. In the picture above, you can see an example of augmentation data in one of the angry expression data. The data that is transformed is actually the same data, so that the computer can learn, that the image is still an angry expression, regardless of any angle. By doing data augmentation, the model built can recognize large amounts of image data so that it is more efficient.

Figure 6. The first test results with 2 convolutional layers
Figure 7. The second test results with 4 convolutional layers

| Parameter       | Training | Validation |
|-----------------|----------|------------|
| SGD, BS, 32, epoch 100 | ![Graph](image.png) | ![Graph](image.png) |
| SGD, BS, 128, epoch 200 | ![Graph](image.png) | ![Graph](image.png) |
| RMSprop, BS, 100 | ![Graph](image.png) | ![Graph](image.png) |
| RMSprop, BS, 128, epoch 200 | ![Graph](image.png) | ![Graph](image.png) |
| Adam, BS, 32, epoch 100 | ![Graph](image.png) | ![Graph](image.png) |
| Adam, BS, 128, epoch 200 | ![Graph](image.png) | ![Graph](image.png) |
| Adam, BS, 128, epoch | ![Graph](image.png) | ![Graph](image.png) |

**Table 3. Model test results**

|        | Angry | Happy | Sad | Surprise |
|--------|-------|-------|-----|----------|
| Angry  | 1200  | 110   | 390 | 75       |
| Happy  | 150   | 2900  | 170 | 100      |
| Sad    | 380   | 170   | 1800| 100      |
| Surprise | 58    | 84    | 80  | 1300     |

Furthermore, the value of True Positive (TP), False Positive (FP), False Negative (FN), Recall, and Precision can be calculated, the results can be seen in table 4.

**Table 4. TP, FP, FN, recall, and precision values of recognized expressions**

|        | TP  | FP  | FN  | Recall | Precision |
|--------|-----|-----|-----|--------|-----------|
| Angry  | 1200| 588 | 575 | 67.60% | 67.11%    |
| Happy  | 2900| 364 | 420 | 87.35% | 88.85%    |
| Sad    | 1800| 640 | 650 | 73.47% | 73.77%    |
| Surprise | 1300| 275 | 222 | 85.41% | 82.54%    |
| Total  | 7200| 1867| 1867| 79.41% | 79.41%    |

Based on the results of the second test as shown in Figure 7, with a batch size of 128, Adam's optimizer, using the 4 convolutional layer model which has 256 neurons and 200 epochs, obtained a higher accuracy value of 80%. In testing with an epoch of 300, it was found that the system accuracy decreased to 71% because the learning rate value decreased during the training process.
| No. | Test Image | Actual Label | Label Result of Classification |
|-----|------------|--------------|-------------------------------|
| 1   | ![Happy](image1) | Happy        | Happy                         |
| 2   | ![Angry](image2) | Angry        | Angry                         |
| 3   | ![Happy](image3) | Happy        | Happy                         |
| 4   | ![Sad](image4)   | Sad          | Sad                           |
| 5   | ![Happy](image5) | Happy        | Happy                         |
| 6   | ![Sad](image6)   | Sad          | Angry                         |
| 7   | ![Sad](image7)   | Sad          | Sad                           |
| 8   | ![Surprise](image8) | Surprise | Surprise                     |
| 9   | ![Happy](image9) | Happy        | Happy                         |
| 10  | ![Sad](image10)  | Sad          | Sad                           |
| ... | ...          | ...          | ...                           |
| 100 | ![Angry](image11) | Angry       | Happy                         |
The test results with the best model from the above experiment can be seen in table 3. The label on the column side of the table shows the classified expression label, while the label on the row side of the table shows the actual expression label. There were 1,200 angry expressions classified as angry, 390 happy expressions, and 75 sad expressions. There were 2,900 happy expressions classified as happy, 150 angry expressions, 170 sad expressions, and 100 shocked expressions. There were 1,800 classified expressions of sadness, 380 angry expressions, 170 happy expressions, and 100 shocked expressions. There were 1,300 shocked expressions classified as shocked expressions, 58 angry expressions, 84 happy expressions, and 80 sad expressions. By using this model, testing was carried out using 100 random images with different image formats, sizes, and modes. Some of the results of the tests that have been carried out can be seen in table 5.

Based on the results of tests that have been carried out on the facial expression recognition system using CNN, an average value of 80% accuracy is obtained after going through the training process for an average of 10 seconds / epoch and it takes 33 minutes 18 seconds to train 200 epochs.

4. Conclusion
The conclusions that can be drawn from the results of this research are as follows: The CNN method is able to classify facial expressions quite well and has an accuracy of up to 80% of 100 facial expression images; In the training process, data augmentation is necessary to reproduce the data in order to avoid overfitting which affects the accuracy of the training data and test data. The selection of the batch size, the number of convolutional layers, optimizers and epoch also affects the accuracy obtained; After going through several tests, the batch size used must be in accordance with the specifications of the device used, the amount of epoch used must be adjusted to other parameters. The increasing number of convolutional layers can make the accuracy in the learning process better; Activation functions that are good for this research are ReLU and Softmax.

References
[1] Samarawickrame and Mindya S 2013 Facial expression recognition using active shape models and support vector machines ICTer 51-55
[2] Mliki H, Fourati N, Smaoui S and Hammami M 2013 Automatic facial expression recognition system ACS International Conference on Computer Systems and Applications 1-4
[3] Voisan E, Precup R and Dragan F 2016 Facial expression recognition system based on a face statistical model and support vector machines The 11th IEEE International Symposium on Applied Computational Intelligence and Informatics
[4] Shan K, Guo J, You W, Lu D and Bie R 2017 Automatic facial expression recognition based on a deep convolutional neural network structure The 15th IEEE International Conference on Software Engineering Research, Management and Applications
[5] Zufar M and Setiyono B 2016 Convolutional neural networks untuk pengenalan wajah secara real time Jurnal Sains dan Seni ITS