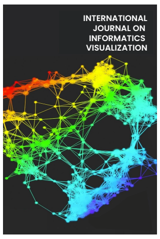
INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



The Effect of Layer Batch Normalization and Dropout of CNN model Performance on Facial Expression Classification

Norhikmah [a,*], Afdhal Lutfhi [a], Rumini [a]

[a] Faculty of computer Science, University AMIKOM Yogyakarta, Daerah Istimewa Yogyakarta 55283, Indonesia
Corresponding author: [*]hikmah@amikom.ac.id

***Abstract*— One of the implementations of face recognition is facial expression recognition, in which a machine can recognize facial expression patterns from the observed data. This study used two convolutional neural network models, model A and model B. The first model A was without batch normalization and dropout layers, while the second model B used batch normalization and dropout layers. It used an arrangement of 4 layer models with activation of ReLU and Softmax layers and two fully connected layers for five different facial expressions: angry, happy, normal, sad, and shock. The research method in this study is divided into data analysis, pre-processing, gray scaling, Convolutional Neural Network (CNN), and model validation testing. This study obtained an accuracy of 64.8% for training data and an accuracy of 63.3% for validation data. The dropout layers and batch normalization could maintain the training and validation data stability so that there was no overfitting. By dividing the batch size on the training data into 50% with 200 iterations, aiming to make the load on each training model lighter by using the learning rate to be 0.001, which works to improve the weight value, thus making the training model work to be fast without crossing the minimum error limit. Accuracy results in the classification of ekp facial receipts from the camera's distance to the face object about 30 cm in the room with the use of bright enough lighting by 78%.**

***Keywords*— Face recognition; facial expression; batch normalization; dropout lay*er*; convolutional neural network.**

Manuscript received 10 Jan. 2022; revised 25 Mar. 2022; accepted 27 Apr. 2022. Date of publication 31 Aug. 2022.

I. Introduction

One of the problems in computer vision, the solutions that have been being investigated for a long time, is the general classification of objects in the image. It concerns duplicating the human ability to understand image information so that the computer can recognize objects in the image like humans. The feature engineering process is limited as it can only be applied to certain datasets and cannot be generalized to any type of image due to differences in viewing angles, scale, lighting conditions, object deformation, and others [1].

The face is a multi-signal or multi-message system. The face can display one kind of signal but has many messages [2]. The texture of facial expressions can be classified into basic expressions: anger, pleasure, fear, surprise, disgust, and sadness [3]. Classification of facial emotions with the challenge of emotional intensity from smooth to sharp in a sequence of images or videos [4].

The Convolutional Neural Networks Architectural Model (CNN) is configured to initialize parameters to speed up the network training process. Use the Convolutional Neural Networks model for construction[5]. CNN consists of two phases, namely the training phase and the testing phase. This second phase only uses features in the form of eye locations to carry out the pre-processing stage, which includes spatial normalization, synthetic sampling, image cropping, sampling, and intensity normalization. These two phases also have slightly different stages where the testing phase does not go through the synthetic sample generation stage, namely the creation of duplicate images that have been given random noise such as translation, rotation, and slope. This aims to add to the database to increase accuracy in the phase of training. In addition, in the training phase, the output of CNN is in the form of CNN weights, which will later be used to provide output in the form of an expression type in the testing phase [6]. The system detects and recognizes faces at a distance of 0-40 cm. However, with a distance of more than 40 cm, the system cannot detect and recognize faces optimally [7]. The convolutional neural network method is used to recognize facial expressions, with the JAFFE dataset from several variations of parameters, using a 3-layer training architecture on the network [8]. Using the Convolutional Neural Network (CNN). Data distribution equalization is applied to improve the performance of the model. This paper presents a classification model that can be applied to find out emotions in a video [9].

Facial Expression Recognition (FER) has become the current topic of study in human-computer interaction, including digital entertainment, customer service, driver monitoring, and emotional robotics. Indeed, many extensive studies and methods have been developed to increase accuracy and reduce the amount of loss during model training, including adding a dropout screen and batch normalization. This study aims to identify the different effects of adding a dropout layer and batch normalization layer with and without using the two additional screens. The Convolutional Neural Network (CNN) method is a technique for processing two-dimensional data that are processed in only one screen dimension but in 2 screen dimensions [6]. Face Detection and Recognition in Real-Time Photos with Haar Cascade and Convolutional Neural Network is one of the machine learning methods from the development of Multi-Layer Perceptron (MLP), which is designed to process two-dimensional data [10]. A neural network model for a mechanism of visual pattern recognition [11]. A good image classification model must be invariant to the cross-product of all these variations [12]. Training Deep Neural Networks is a complicated by the fact[13]. Deep Learning has excellent skills in computer vision. One of them is on classifying objects in the image [1]. When a face is detected, the system will create a red Bounding Box, and then the image of the face will be saved into the feature extraction process. Classification using the CNN method is also relatively reliable for parameter changes as changes in the confusion level are not affected by the accuracy results. This study's ultimate goal is to help detect facial expressions of angry, happy, normal, sad, and shock in real-time and to determine the level of emotion accuracy based on facial expressions towards indoor lighting and camera distance.

II. Materials and Methods

The procedure of this study is presented in Figure 1 below:

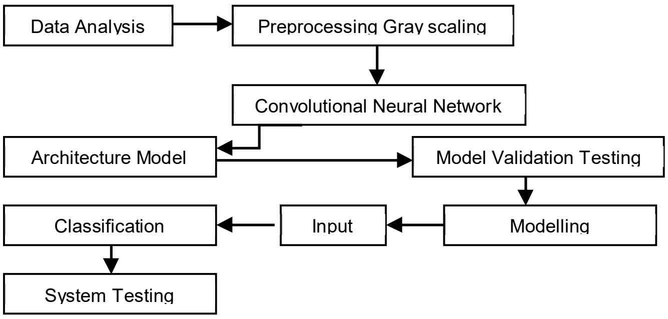


Fig. 1 The Procedure of the Study

A. Data Analysis

This present study used face expression recognition data from a site (https://www.kaggle.com/c/challenges in representation learning facial expression recognition challenge). The data consisted of two columns of class and an array in which the class presented a class of images consisting of angry, happy, sad, shocked, and normal, while the array shows the number of pixels of each image. The data consists of 24256 records for training and 3006 for validation. The data were collected randomly by angle, position, and light intensity. Table I below shows facial expression class data.

TABLE I
Facial expression class data

Angry	Happy	Sad	Shocked	Normal
4005	7255	4965	4830	3201

Processed data were divided into two, namely training and validation data. The result of the produced model was taken for the best accuracy and stored in a file with the extension of h5 containing the weights of the evaluated parameters. Data for validation were tested to compare the produced model.

B. Pre-processing Gray scaling

Before the facial expression detection process was done, the first process was to normalize the image to a 1.0 rotational scale, change the color to grayscale, and resize it into 64x64 pixels.

C. Convolutional Neural Network (CNN)

Machine learning and deep learning have produced much good performance in image recognition [23]. The CNN working procedure is almost the same as MLP, but each neuron in CNN is represented in a two-dimensional form, while in the MLP, it is only one dimension. On CNN, the propagated data on the network is in the form of two-dimensional data in which the linear operation uses a convolution process, and data weights are a collection of four-dimensional convolution filters consisting of 25-input neurons, output neurons, height, and width. Due to the nature of the convolution process, CNN can only be used on two-dimensional data. Based on the LeNet architecture, a CNN consists of four layers: convolution layer, subsampling layer, fully connected layer, and activation function [1]. CNN is a type of neural network where this method can be used on images [19].

Parameter implementation of Convolutional Neural Network (CNN) was divided into 2:

- CNN test value parameter: (a) Training is Dropout Layer, Batch Normalization and Non-Dropout Layer, Non-Batch Normalization, (b) Learning rate is le3(0.001), (3) epoch is 50.100 and 20.
- CNN fixed value parameters: (a) Layer Hidden is 2 Layer, (b) Training Algorithm is Adam Method, (c) Batch Size is 128/500, (d) Filter is 32, 64, 128, 256, (e) Filter size is 3x3 and 5x5

D. Architecture Model

This study used the architectural model of two screens. The first used the normalization batch screen and the dropout screen, and the second used two screens of non-batch normalization screen and a non-dropout screen. These two models were compared for the accuracy and training error levels. The form of the CNN architecture is presented in figures 2 and 3 below by using the equation (1) to (3):

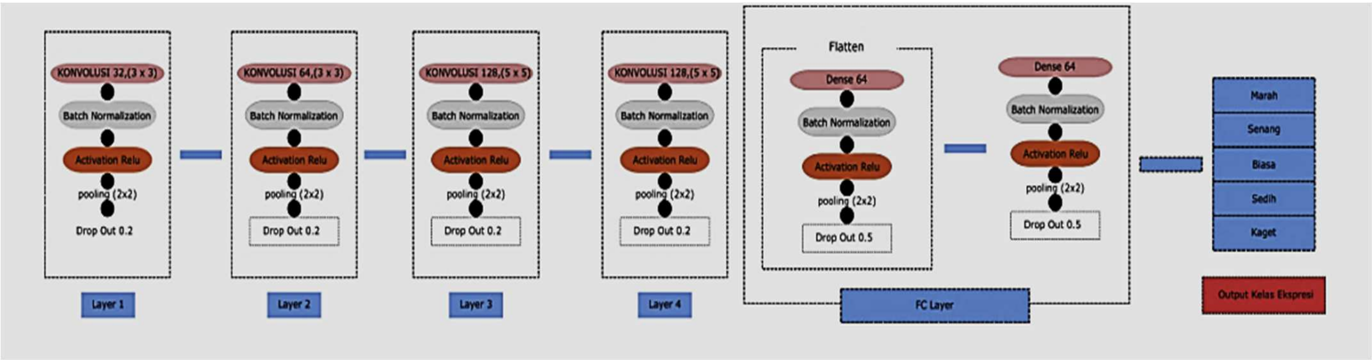


Fig. 2 Structure of Model A using Batch Normalization Screen and Dropout Screen

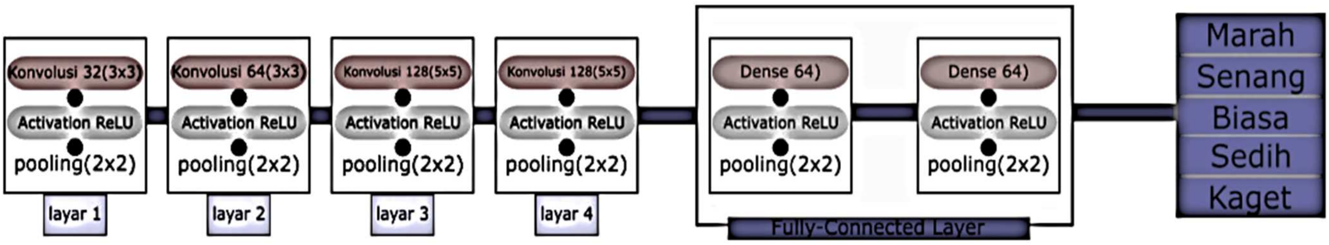


Fig. 3 Structure of Model B using Non-Batch Normalization Screen and Non-Dropout Screen

1) Convolution: The convolutional stage is to multiply the input data with a function (filter) repeatedly to obtain a feature map. The filter value used in this study was randomly initiated following a normal distribution with the size and number of filters as described in Figures 3 and 4 above.

2) Fully Connected layer (FC): The feature map produced from feature extraction is in the form of a multidimensional array so that it should be flattened or reshaped into a vector so that it can be used as the input of the fully connected layer (FC). The FC layer is the layer where all the activity neurons of the layer are previously connected with the next neuron layer as an ordinary artificial neural network. Each dimension can be connected to all neurons in the fully connected layer. The FC layer is usually used in the multi-layer perceptron method and aims to process data so that they can be classified based on the differences between layers. The use of several layers affects the time and accuracy of test data [15].

3) ReLU Activation: ReLU processes the convergence performed by stochastic gradient descent faster than sigmoid and tan h. ReLu can also be implemented only by setting a threshold at zero. However, ReLU can be fragile during training and cause unit malfunction. For example, the study found that 40% of the networks were not functioning because the learning rate was initialized higher from lr=0.01 to lr=0.001.

The equation of ReLu[15]:

$$F(x) \;=\; x, \text{If } x \geq 0$$

$$0, \text{If } x <= 0 \qquad (1)$$

as it only makes a limit on zero number, it means that if x 0, then x = 0 and if x > 0, then x = x [20].

4) Pooling Layer: Pooling is an important concept of CNN. It lowers the computational burden by reducing the number of connections between convolutional layers. This section introduces some recent pooling methods that used two hundred CNN [17].

The process is the same as the convolution screen that uses a filter that the movement can be changed as described above and the result of the changes can be seen from the following equation [18]:

$$\frac{n-f}{s} + 1 \qquad (2)$$

The notation in the above equation can be defined as follows: n = input size of the map feature, f = filter size, and s = number of strides.

The main goal of Pooling Layer is to reduce the number of parameters of the input tensor to help reduce overfitting, extract representative features from the input tensor, reduce computation, and thereby aid efficiency [21].

5) Layer Batch Normalization: We refer to the change in the distributions of internal nodes of a deep network, in the course of training, as Internal Covariate Shift. Eliminating it offers a promise of faster training. We propose a new mechanism called Batch Normalization [13].

To reduce internal covariate shifts and increase the accuracy of a model so that the produced model can be efficient, it can use a Batch Normalization layer in which, after passing the convolution layer process, the BN layer will work to distribute the value automatically at the time of layer activation. To reduce the internal covariate shift problem, a BN layer was added in each of the dimensions, because between each of the connected BN layers, the input distribution range of each layer remains the same regardless of changes in the previous layer, as presented in the following equation.

In training, we refer to the change in the distributions of internal nodes of a deep network as Internal Covariate Shift and eliminating it offers a promise of faster training. We propose a new mechanism, which we call Batch

Normalization, which takes a step toward reducing internal covariate shift, and in doing so dramatically accelerates the training of deep neural nets. It accomplishes this via a normalization step that fixes the means and variances of layer inputs. Batch Normalization also benefits the gradient flow through the network by reducing the gradients' dependence on the parameters' scale or initial values. This allows us to use much higher learning rates without the risk of divergence. Furthermore, batch normalization regularizes the model and reduces the need for dropout [13].

$$\hat{X}^{(k)} = \frac{x^{(k)})-E[x^{(k)}]}{\sqrt{var[x^{(k)}]}} \qquad (3)$$

The equation above can be represented by [4]:
$x^{(k)}$ = input of k-neuron
E $[x^{(k)}]$ = Population of mean input from each k-neuron
Var $[x^{(k)}]$ = Variance of input on each k-neuron

6) Dropout: Dropout is a technique that addresses both of these issues [14]. The regularization in the training model can be overcome using a dropout layer which can minimize the occurrence of too many training errors. In principle, each neural forces others to connect after passing through activation but causes over-fitting data and a large training error. Dropout will disable random networks connected to neurons with a rating scale as presented in the equation below. In which p is the value of the hidden layer that is connected to other layers with the standard parameter for the dropout screen is 0.5. the optimal probability of retention is usually closer to 1 than 0.5[14]. The dropout rate is higher than in offline classes because students have to manage and control study time without the professor or manager's help. Therefore, professors and managers must support students in a timely act to avoid the risk of dropout online university classes [24].

III. Results and Discussion

Dropout and batch normalization are well-recognized approaches to tackling these challenges [22]. Training and validation testing in classifying the detection of emotional expressions on a person's face used hardware consisting of AMD-87410 Processor, 12GB Ram, AMD Radeon R5 2.2 GHz graphics, and software of Google Colab, Visual Studio Code Anaconda, with hardware libraries, NumPy, and OpenCV.

A. Model Validation Testing

This study used two models, A and B. Model A does not use batch normalization and dropout screens, and model B uses batch normalization and dropout screens. The comparisons of how the two models work are presented in Tables II and III by using the equation (1) to (3):

TABLE II
Model A without batch normalization and dropout screen

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	320
activation_1 (Activation)	(None, 48 ,48, 32)	0
Max_pooling2d_1 (MaxPooling2)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
activation_2 (Activation)	(None, 24, 24, 64)	0
Max_pooling2d_2 (MaxPooling2)	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	204928
activation_3 (Activation)	(None, 12, 12, 128)	0
Max_pooling2d_3 (MaxPooling2)	(None, 6, 6, 128)	0
conv2d_4 (Conv2D)	(None, 6, 6, 256)	819456
activation_4 (Activation)	(None, 6, 6, 256)	0
Max_pooling2d_4 (MaxPooling2)	(None, 3, 3, 256)	0
Flatten_1 (Flatten)	(None,2304)	0
dense_1(Dense)	(None, 64)	147520
activation_5 (Activation)	(None, 64)	0
dense_2(Dense)	(None, 64)	4160
activation_6 (Activation)	(None, 64)	0
dense_3(Dense)	(None, 5)	325
activation_7 (Activation)	(None, 5)	0
Total param	**: 1,195,205**	
Trainable params	**: 1,195,285**	
non-trainable params	**: 0**	

TABLE III
Model B using batch normalization dan dropout screen

Layer (Type)	Output Shape	Param#
Conv2d_1 (Conv2D)	(None, 48, 48, 32)	320
batch_normalization_1	(None, 48, 48, 32)	128
activation_1 (Activation)	(None, 48, 48, 32)	0
max_pooling2d_1(MaxPooling2)	(None, 24, 24, 32)	0
dropout_1(Dropout)	(None, 24, 24, 32)	0
Conv2d_1 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_2	(None, 24, 24, 64)	256
activation_2 (Activation)	(None, 24, 24, 64)	0
max_pooling2d_2(MaxPooling2)	(None, 12, 12, 64)	0
dropout_2(Dropout)	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	204928
batch_normalization_3	(None, 12, 12, 128)	512
activation_3 (Activation)	(None, 12, 12, 128)	0
max_pooling2d_3(MaxPooling2)	(None, 6, 6, 128)	0
dropout_3(Dropout)	(None, 6, 6, 128)	0
conv2d_4 (Conv2D)	(None, 6, 6, 256)	819456
batch_normalization_4	(None, 6, 6, 256)	1024
activation_4 (Activation)	(None, 6, 6, 256)	0
max_pooling2d_4(MaxPooling2)	(None, 3, 3, 256)	0
dropout_4(Dropout)	(None, 3, 3, 256)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 64)	147520
batch_normalization_5	(None, 64)	256
activation_5(Activation)	(None, 64)	0
dropout_5(Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 64)	4160
batch_normalization_6	(None, 64)	256
activation_6(Activation)	(None, 64)	0
dropout_6(Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 5)	325
activation_7(Activation)	(None, 5)	0
Total Params	**: 1,197,637**	
Trainable Params	**: 1,196,412**	
Non-Trainable params	**: 1.216**	

Based on Figure 4 for model A, all networks are active or connected, but they will have an impact on backpropagation in the form of difficulty in updating each weight and causing overfitting in which the training value is good but not good for data that have not been trained. And figure 5, model B used batch normalization, and the results showed that some networks are not activated to reduce constraints in updating the weighting process and prevent data overfitting

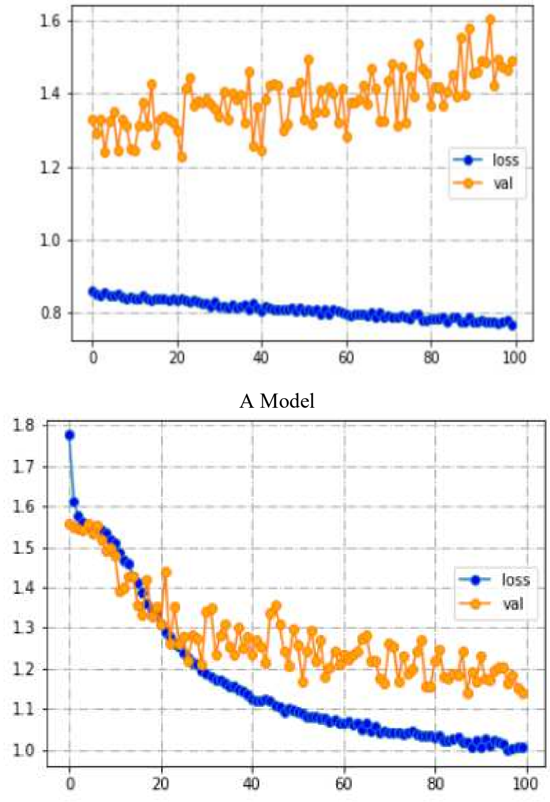
B. Comparison between the Training Models during Iteration

Model A and Model B were compared with 50 and 100 iteration stages. The results of the comparison between the two models can be seen in each iteration process in the following Figure 6.

TABLE IV
MODEL A AND B IN THE 100TH ITERATION STAGE

Model A	Model B
17s 88ms/step - loss:0.7416	16s 85ms/step - loss:0.9746
17s 88ms/step - loss:0.7474	16s 85ms/step - loss:0.9664
16s 86ms/step - loss:0.7432	16s 85ms/step - loss:0.9761
16s 87ms/step - loss:0.7526	16s 85ms/step - loss:0.9652
16s 87ms/step - loss:0.7459	16s 85ms/step - loss:0.9656
16s 87ms/step - loss:0.7418	16s 84ms/step - loss:0.9707
16s 87ms/step - loss:0.7480	16s 84ms/step - loss:0.9678
16s 86ms/step - loss:0.7513	16s 85ms/step - loss:0.9647
17s 87ms/step - loss:0.7492	16s 85ms/step - loss:0.9667
17s 88ms/step - loss:0.7426	16s 89ms/step - loss:0.9662

Based on Table IV, the model has reached the 100th iteration stage to check the suitability between the training data and the validation data to create a graph. Generally, the validation data error value is lower than the training data, and there is no big gap between the two. The graph can be seen below in Figure 7.



A Model

B Model

Fig. 4 Graph Comparison between model A and model B

Based on Figure 4, the addition of the batch normalization screen aims to normalize input by adjusting and scaling activation and reducing internal covariate shifts. The dropout screen functions to prevent overfitting and accelerate the learning process. The dropout refers to disabling the hidden and visible layers in the network

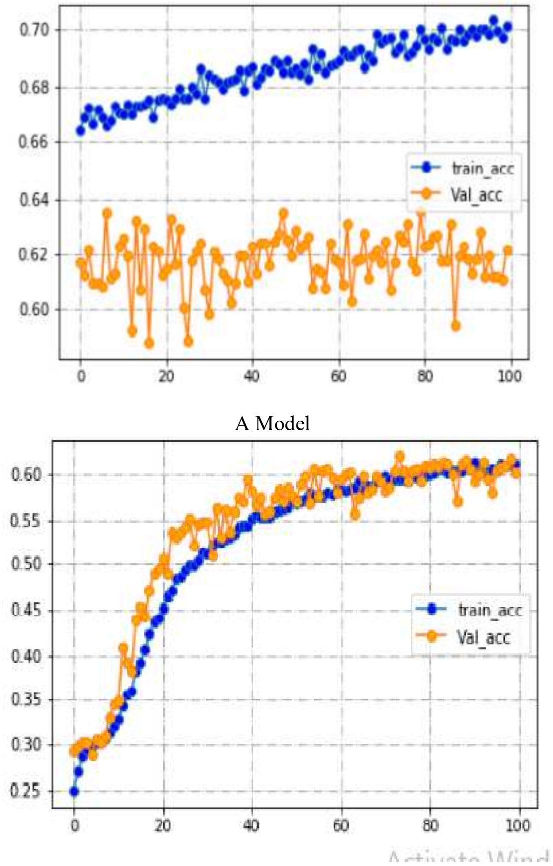
C. Comparison of Accuracy

Based on Table V, model A has a large difference between training and validation accuracy of about 6.7 points. Meanwhile, model b does not have a large difference between the training and validation accuracy (0.7 points).

TABLE V
COMPARISON BETWEEN THE RESULT OF MODEL A AND MODEL B

Stage	**Model A**		**Model B**	
	Training Accuracy (%)	**Validation Accuracy (%)**	**Training Accuracy (%)**	**Validation Accuracy (%)**
0	66.4	61.7	24.9	29.2
1	66.9	61.2	27.0	29.8
98	69.7	61.0	61.1	61.7
99	70.1	62.2	61.1	60.3
AVG	**68.58**	**61.77**	**52.29**	**53.84**
MAX	**70.3**	**63.6**	**61.3**	**62.0**
MIN	**66.4**	**58.8**	**24.9**	**28.8**

Thus, model A cannot be used for modeling an application's implementation. On the other hand, model B has ideal accuracy, which can be implemented into the application as presented in Fig. 5.



A Model

B Model

Fig. 5 Graph of the accuracy of model A and model B

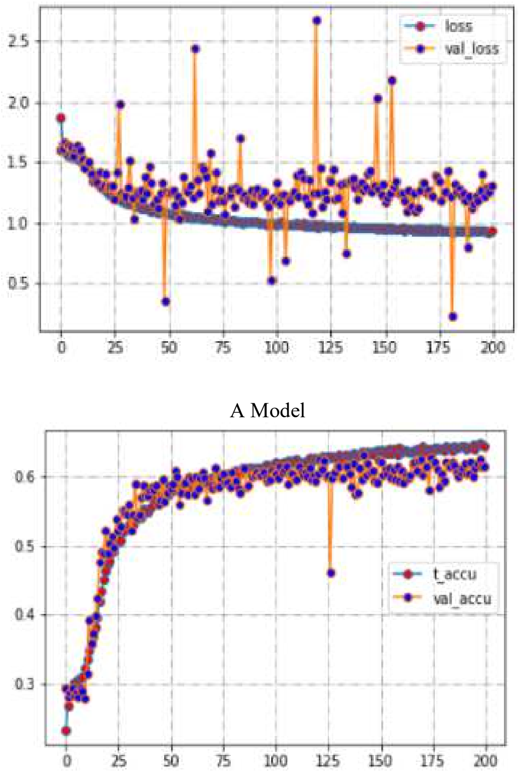
Figure 5 shows that model B, which uses batch normalization and dropout layers, got a maximum accuracy of 62% with no overfitting. To see the batch normalization and dropout in handling weight changes in iterations, a

Comparison of Each Accuracy as presented in Table VI below.

TABLE VI
COMPARISON OF EACH ACCURACY

Accuracy	Iteration 50	Iteration 100	Iteration 200
	45,23%	62,00%	63,30%

Table VI provides information regarding the differences in the accuracy of each iteration performed in this study. The iteration with the best accuracy is 63.3% in iteration 200.



A Model

B Model

Fig. 6 Graph of iteration 200

Based on Figure 6, the effect of the number of iterations, batch size, and the addition of batch normalization screens as backward propagation handling so that changes in weight in each training do not cause overfitting. The dropout screen is to deactivate neurons to reduce changes in weight values temporarily. Thus, adding weight to each neuron on a large scale can be adjusted by the model.

D. *Modeling*

This system is divided into two stages of classifying facial expressions as follows:

1) Input: At this stage, face recording was performed when the person was facing the camera in front. The system began with frame acquisition and then continued with gray scaling pre-processing. Then, it was continued with human form detection by using a frame difference that detects movement. The system returns to the video frame acquisition process if no motion is detected. However, if motion is detected, the system continues detecting faces using the Viola-Jones algorithm. After getting the human form, then it was followed by face detection using the Viola-Jones method of the Haar Cascade Classifier algorithm. When a face is detected, the system creates a red Bonding Box, which is then saved to enter the feature extraction process.

2) Classification: This study used a feature extraction of the Convolutional Neural Network method. Convolutional neural networks (CNNs) have shown great performance in various fields, such as image classification [16]. The result of the input stage continued with the process features stored as a model. However, it should go through the process of the gray scaling pre-processing stage, changing the size, extracting layers of each image data on the pre-trained model, and filters on the pre-trained obtained from the training process. The explanation of the stages of training and testing in the process of recognizing facial expressions is as follows:

- Image pre-processing, before the face detection process, the first process is changing the image into grayscale and changing the size.
- Feature Learning, after the face image passes through the pre-processing of the image data, extra features are added to each layer to get the learning of each image data to feature using the CNN method.
- After the feature learning process, classification is performed from each processed image feature.
- For face detection, the study used the Viola-Jones method.

E. *System Testing*

The testing was performed using the camera: Arcus -> Full HD 1080P, AWC-07.2K Resolution (2560x144) pixel.

1) Lighting Condition Testing: In testing the lighting conditions in the room and the camera's distance, a picture was taken using a webcam directly. The indoor lighting was divided into three light sources: sun, room light, and camera light. Meanwhile, the distance from the camera to the face is 30 cm and 60 cm.

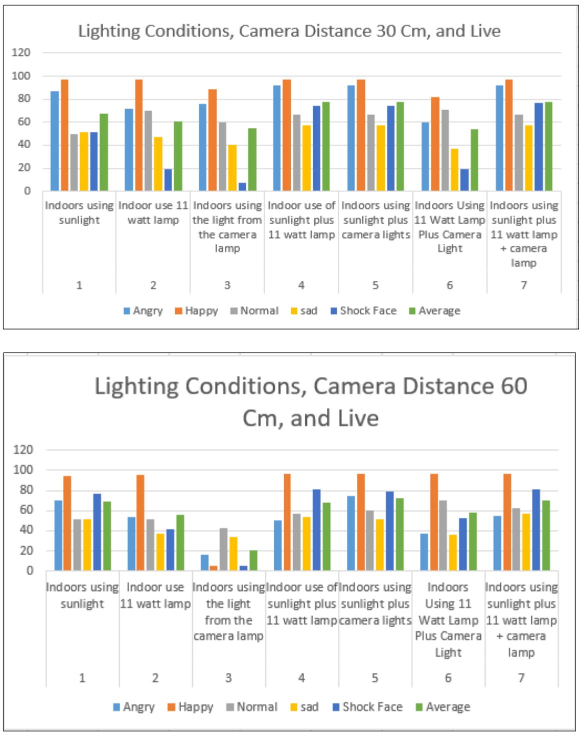


Fig. 7 Data Testing Based on Lighting Conditions and Camera Distance

Based on Figure 7, the best results are obtained for indoor conditions using sunlight, 11-watt lamps, and camera light, with the camera-to-face distance being 30cm with an average of 78%.

2) *Facial Expression Testing*

The test was based on facial emotional expressions consisting of angry, happy, normal, sad, and shocked by conducting direct testing using a webcam and the camera's distance to the face is 30 and 60 cm. The results showed that the highest average is in happy facial expressions (93.7%) and angry (81.5%) at a distance of 30 cm. The detailed results can be seen in Figure 8.

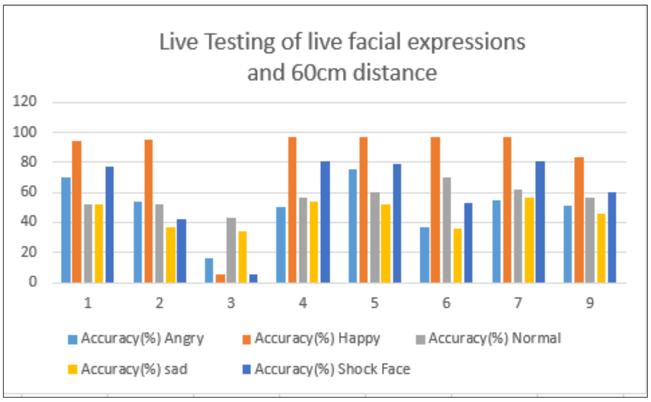
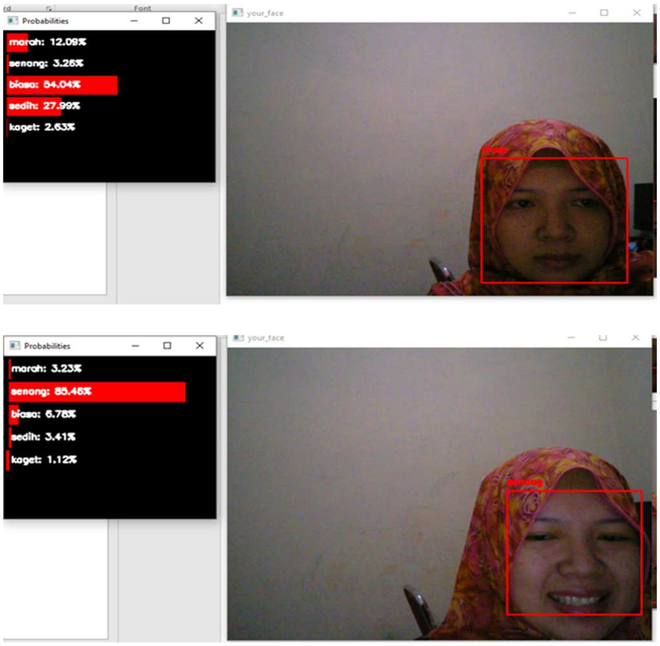


Fig. 8 Testing the Facial Expression Class based on Lighting Condition and Distance to the camera

3) Testing the Running Application: In testing the running application, the application can detect objects and focuses on faces. Thus, the application can show facial expression information according to the object presented in Figure 9.



```
marah: 12.09%
senang: 3.26%
biasa: 84.04%
sedih: 27.99%
kaget: 2.63%
```

```
marah: 3.23%
senang: 85.46%
biasa: 6.78%
sedih: 3.41%
kaget: 1.12%
```

Fig. 9 Testing the Facial Expression Application

IV. Conclusion

Of the two models used in the study, the first model did not use the dropout layer and batch normalization layer, and the second used both. The first model cannot maintain data stability from overfitting while the second model. Can maintain the stability of the training data and validation data during testing so that data do not overfit. Model validation Testing, Obtained an accuracy of 64.8% for training data and accuracy of 63.3% for validation data. The dropout layers and batch normalization could maintain the training and validation data stability so that there was no overfitting. With the testing process dividing the batch size of the training data into 50% with 200 iterations, aims to make the load on each training model lighter. Using the learning rate of 0.001 to correct the weight value can make the training model work faster without crossing the minimum error limit. The accuracy results in classifying facial expressions from the camera distance of 30 cm to the object in the room with the use of bright lighting reach 78%. The suggestions for further research using data augmentation and collaboration deep learning methods to get a higher level of accuracy and use high camera quality

Acknowledgment

We thank the Directorate of Service and Research, AMIKOM University Yogyakarta for funding this study.

References

[1] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.

[2] A. A. Marunduh and Lina, “Klasifikasi Emosi Pada Wajah Dari Rekaman Aplikasi Video Conference Dengan Menggunakan Metode Convolutional Neural Network” *J. Ilmu Komput. dan Sist. Inf.*, vol. 09, pp. 27–31, 2021.

[3] N. Inzani, R. Amalia, and J. Y. Sari, “Mengidentifikasi Mood Mahasiswa Berdasarkan Ekspresi Wajah dengan Menggunakan Discrete Wavelet Transform dan Fuzzy K-Nearest Neighbor,” vol. XI, no. 1, pp. 34–38, 2019.

[4] Lia Farokhah, “Implementasi Convolutional Neural Network untuk Klasifikasi Variasi Intensitas Emosi pada Dynamic Image Sequence,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 1070–1076, 2020, doi: 10.29207/resti.v4i6.2644.

[5] M. Zufar and B. Setiyono, “Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time,” *J. Sains dan Seni ITS*, vol. 5, no. 2, p. 128862, 2016, doi: 10.12962/j23373520.v5i2.18854.

[6] A. T. Lopes, E. De Aguiar, and T. Oliveira-Santos, "A Facial Expression Recognition System Using Convolutional Networks," *Brazilian Symp. Comput. Graph. Image Process.*, vol. 2015-Octob, no. August 2005, pp. 273–280, 2015, doi: 10.1109/SIBGRAPI.2015.14.

[7] A. W. Wibowo, A. Karima, Wiktasari, A. Yobioktabera, and S. Fahriah, “Pendeteksian dan Pengenalan Wajah Pada Foto Secara Real Time Dengan Haar Cascade dan Local Binary Pattern Histogram,” *JTET (Jurnal Tek. Elektro Ter.*, vol. Vol. 9 No., pp. 6 – 11, 2020.

[8] L. M. R. Rere, R. Dalam, and K. Baru, “Studi Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network,” vol. 3, 2019.

[9] P. Daru, S.'U,Shofwatul."Penentuan Emosi Pada Video Dengan Convotional Neural Network" JISKA (Jurnal Informatika Sunan Kalijaga), vol. 5, no. 1, p. 23, 2020, doi: 10.14421/jiska.2020.51-04.

[10] N. Sofia, "Convolutional Neural Network," *Medium.com*, 2018. https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b.

[11] K. Fukushima, "Neocognitron: A self-organizing neuralnetwork model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.

[12] Cs231n.stanford.edu, "CS231n: Convolutional Neural Networks for Visual Recognition," 2021. https://cs231n.github.io/classification/

[13] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning, 2015, 448-456.

[14] S. Nitish and H. Geoffrey, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Phys. Lett. B*, vol. 299, no. 3–4, pp. 345–350, 1993, doi: 10.1016/0370-2693(93) 90272-J.

[15] A. Yusuf, R. Cahya Wihandika, and C. Dewi, "Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network," 2019. [Online]. Available: http://j-ptiik.ub.ac.id.

[16] F. M. Qotrunnada and P. H. Utomo, "Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker," *Prosiding Seminar Nasional Matematika*, vol. 5, pp. 799–807, 2022, [Online].Available:https://journal.unnes.ac.id/sju/index.php/prisma.

[17] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, 2017, doi: 10.1016/j.neucom.2018.09.038.

[18] Dharmadi.R."Mengenal Covutional Layer dan Pooling Layer," 2018 https://medium.com/nodeflux/mengenal-convolutional-layer-dan-pooling-layer-3c6f5c393ab2.

[19] Warnia,N , "CNN Modelling Untuk Deteksi Wajah Berbasis Gender Menggunakan Python," *Jurnal Politeknik Caltex Riau*, vol. 2, pp. 192–199, 2020.

[20] Nurkhasanah and Murinto, "Klasifikasi Penyakit Kulit Wajah Menggunakan Metode Convolutional Neural Network Classification of Facial Skin Diseases Using the Method of the Convolutional Neural Network," *SAINTEKS*, vol. 18, no. 2, 2021, [Online]. Available: https://www.kaggle.com/datasets.

[21] A. Kholik, "Klasifikasi Menggunakan CNN pada Tangkapan Layar Halaman Instagram" *JDMSI*, vol. 2, no. 2, pp. 10–20, 2021.

[22] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, vol. 79, no. 19–20, pp. 12777–12815, May 2020, doi: 10.1007/s11042-019-08453-9.

[23] D. Gede, T. Meranggi, N. Yudistira, and Y. A. Sari, "Batik Classification Using Convolutional Neural Network with Data Improvements,", Joiv: Int. J. Inform Visualization,vol 6, no 1,March 2022 6-11. Available:http://joiv.org/index.php/joiv/article/view/716/402.

[24] P.S.Hee.Y.J.Seong, "Early Droput Prediction in Online Learning Of University Meachine Learning, Joiv:: Int. J. Inform Visualization, vol 5, no 2, December 2021, 347-353 Available: http://joiv.org/index.php/joiv/article/view/732/375.