IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.16, No.2, April 2022, pp. 159~168
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: 10.22146/ijccs.72493

Face Expression Classification in Children Using CNN

Yusril Ihza *[1], Danang Lelono[2]
[1]Master Program of Computer Science, FMIPA UGM, Yogyakarta, Indonesia
[2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]yusril.ihza@mail.ugm.ac.id**, [2]danang@ugm.ac.id

Abstrak

Ekspresi wajah dapat menyampaikan apa yang ada di pikiran orang, agar dapat mengenali emosi seseorang. Berdasarkan kesamaan dalam perubahan otot-otot wajah ada enam bentuk emosi yang diterima secara universal sejak tahun 1992 yaitu, marah, jijik, takut, bahagia, sedih dan terkejut. Melalui ekspresi wajah, maka dapat dipahami emosi yang sedang bergejolak pada diri individu. Jika dibandingkan dengan orang dewasa maka ekspresi wajah anak-anak lebih ekspresif untuk emosi positif dan ambigu untuk emosi negatif sehingga jauh lebih sulit dikenali. Ambigu dalam hal emosi negatif misalnya anak-anak ketika sedang marah terkadang menunjukkan wajah tanpa ekspresi, sehingga sulit untuk mengetahui emosi yang sedang dialami anak tersebut. Oleh karena itu, maka diusulkan penelitian menggunakan Convolutional Neural Network dengan arsitektur ResNet-50. CNN ResNet-50 dimanfaatkan untuk mengklasifikasikan enam ekspresi anak diantaranya, marah, takut, jijik, senang, sedih, dan terkejut dengan cara mengambil model yang paling optimum atau model yang memiliki nilai validation loss yang terendah pada saat proses training. Kemudian, model tersebut akan digunakan pada saat proses testing. Hasil pengujian yang dilakukan dengan mempertimbangkan antara hyperparameter epoch dengan batch size, dan hyperparameter learning rate dengan batch size. Eksperimen pengujian menggunakan hyperparameter epoch dengan batch size menghasilkan nilai presisi dengan nilai rata-rata presisi sebesar 0.9983 dengan tingkat akurasi sebesar 99.89%, sedangkan eksperimen menggunakan hyperparameter learning rate dengan batch size menghasilkan rata-rata nilai presisi 0.9975 dengan tingkat akurasi sebesar 99.73%.

Kata kunci*— Ekspresi, Anak, CNN, ResNet-50*

Abstract

Facial expressions can convey what's on people's minds, in order to recognize someone's emotions. Based on the similarities in changes in facial muscles there are six forms of emotion that are universally accepted since 1992 namely, anger, disgust, fear, happiness, sadness and surprise. Through facial expressions, it can be understood the emotions that are turbulent in the individual. When compared with adults, children's facial expressions are more expressive for positive emotions and ambiguous for negative emotions so that they are much more difficult to identify. Ambiguous in terms of negative emotions, for example, when children are angry, sometimes they show an expressionless face, making it difficult to know what emotions the child is experiencing. Therefore, it is proposed research using Convolutional Neural Network with ResNet-50 architecture. CNN ResNet-50 was used to classify six children's expressions including anger, fear, disgust, joy, sadness, and surprise by taking the most optimum model or the model with the lowest validation loss value during the training process. Then, the model will be used during the testing process. The results of the tests carried out by considering the hyperparameter epoch with batch size, and hyperparameter learning rate with batch size. Testing experiments using hyperparameter epoch with batch size resulted in precision values with an average precision value of 0.9983 with an accuracy rate of 99.89%,

Received January 24th,2022; Revised March 11th, 2022; Accepted April 22th, 2022

while experiments using hyperparameter learning rates with batch size resulted in an average precision value of 0.9975 with an accuracy rate of 99.73%.

Keywords— *Expression, Children, CNN, ResNet*

1. INTRODUCTION

The face is an important social stimulus that is often the focus of scientific research. Facial expression is one of the characteristics of behavior. Facial expressions can convey what is on people's minds, so they can recognize someone's emotions [1]. Based on the similarities in changes in facial muscles there are six forms of emotion that are universally accepted since 1992 namely, anger, disgust, fear, happiness, sadness and surprise. Through facial expressions, it can be understood the emotions that are turbulent in the individual.

Research on facial expressions has been carried out by [2], [3] to classify adult faces using the Independent Component Analysis (ICA), Genethic Algorithm (GA) and Neural Network (NN) methods with an average accuracy of 98.85%, while [3] using the Convolutional Neural Network (CNN) which produces a validation accuracy of 96.24%. In a study conducted by [4], [5] classified children's facial expressions using the Principal Component Analysis and Support Vector Machine methods with an accuracy of 93%, while [5] using Mean Supervised Deep Boltzman and Random Decision Forest with an accuracy of 56%. Based on previous research, most researchers used facial expressions of adults, whereas children had more expressive facial expressions in terms of positive emotions, and ambiguous in terms of negative emotions [6]. Ambiguous in terms of negative emotions, for example, children when they are angry sometimes show neutral expressions, making it difficult to know which emotions the child is experiencing.

Research on children's facial expressions by [5] using the Mean Supervised Deep Boltzmann Machine as a feature extraction process and Random Forest as a child expression classification resulting in an accuracy of 56% using a CAFE dataset, most of which are misclassified sad expressions which have almost the same variation as neutral expressions so that most sad expressions are classified as a neutral expression. Then, in the feature extraction process used, there are several facial features that are almost similar or cannot be distinguished, some of the classes of fearful expressions show similarities to surprised expressions, causing miss-classification. The classification of child expressions produces inadequate results due to the large variety of children's expressions, but the dataset used is still limited.

Based on the problems above, the author will build a classification system for children's facial expressions using a fully connected Convolutional Neural Network (CNN) using ResNet-50 architecture proposed by [7] and the Children's Spontaneous facial Expressions (LIRIS-CSE) as dataset used proposed by [8].

2. METHODS

2.1 Systems Analysis

This study aims to create a system that can classify children's expressions. In a study conducted by [5] produces an accuracy of 56% which is mostly from misclassification of sad expressions which have expressionless variations so that the system predicts incorrectly. Then, in the feature extraction process used, there are several facial features that are almost similar or cannot be distinguished, some of the classes of fearful expressions show similarities to surprised expressions, causing miss-classification. The cause of the miss-classification is that the dataset presented is still inadequate, while the expressions of the children vary widely. Children's facial xpressions vary widely, as can be seen from several characteristics such as:

a. Angry expression that tends to be characteristic, low eyebrows close to the eyes and clearly knotted, Mouth clenched into a line with a hard descending line at an angle.

b. An expression of fear that tends to characterize, the eyes begin to widen with distended pupils, and the mouth stretches in nervousness.
c. Sad expression that tends to be characterized, one corner of the mouth is pressed, the eyebrows are neutral, the eyes are relaxed, and the pupils touch.
d. Disgusted expression that tends to be characteristic, slightly curved nose, slightly parted lips, and slightly narrowed eyes.
e. A happy expression that tends to be characteristic, has eyes closed as if to hide complacency, lower lip pressing upwards, which narrows the eyes more, and has a wide smile.
f. Expression of surprise that tends to be characteristic, eyebrows raised or one eyebrow raised higher, eyes become alert and focused, mouth slightly open.

Based on these varied facial expressions required:
a. Retrieve a face object from the input image.
b. Equalizes the pixel size of the input image.
c. Grayscaling: Convert Red Green Blue (RGB) color image to Grayscale.
d. Perform histogram equalization on image objects that have been converted to grayscale so that the distribution of gray degree values in an image is made even [9].
e. Normalize the intesity of each image that has been histogram equalized.
f. Training data that has been normalized.

The expression image retrieval process described above is processed in such a way that it involves an artificial intelligence-based digital image process with stages so that the sample can be classified. These stages include:
a. Models generated during the training process.
b. Classification of the expression the child is experiencing.

2.2 System Design

The next stage is to design the flow of the child's expression classification system. The stages of the system flow design, can be seen in Figure 1.

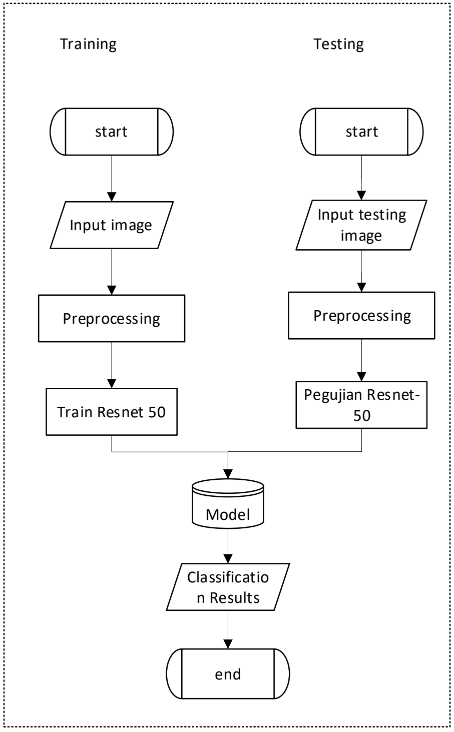


Figure 1 System flow design

Figure 1 is the process of designing a system that is built, namely classification of children's expressions. There are two processes including the training process (left) and testing (right). The initial stage in the training process is image input in the form of video which is then processed at the data pre-processing stage which has been described in Section 2.3. The result of the data pre-processing stage is in the form of normalized image data. The next stage is the CNN Resnet-50 architecture training process which is described in more detail in Section 2.4. The next stage after the training process is done, it produces a model that will be saved into a file (.hdf5).

2. 3 Preprocessing

The video dataset that has been collected is then pre-processed for data. The flow of data pre-processing can be seen in Figure 2.

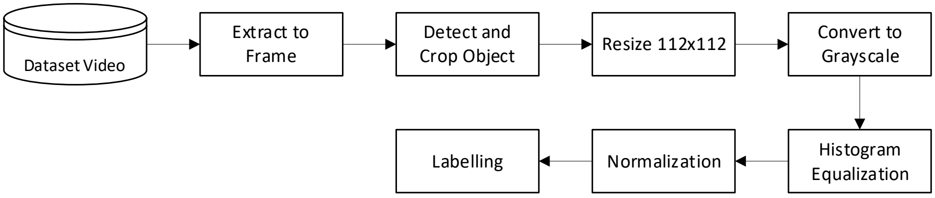


Figure 2 Flow of pre-processing

Figure 2 shows the stages of data pre-processing which consists of loading the video dataset from [8], then extracting it to frame because the input process from the research carried out is by using images. The next stage is detection and cropping of objects in order to separate face and non-face objects using the haarcascade classifier library proposed by [10]. The results of cropping face detection have different sizes so it is necessary to equalize pixels from the results of cropping face detection with a size of 112 x 112 pixels so that each image has the same dimensions to be used as input parameters for ResNet-50 input image and can speed up the training process. The next step is the image that has been cropped is converted to grayscale using Equation (1) where Y is a grayscale image, R is intensity value of red channels input image, G is intensity value of Green channels and B is intensity value of Blue channel, then the image converted to grayscale is then the histogram equalized so that the distribution of the gray degree values in an image is made even. After passing the histogram equalization stage, the image will be normalized and labeled according to the label determined by the dataset provider.

$$Y = 0.2989R + 0.5870G + 0.1141B \quad (1)$$

Next steps, calculate histogram equalization of image that have been converted to grayscale. Histogram equalization is used to even out the contrast of the image so that the image is neither too bright nor too dark. An image equalization histogram can be created through the following steps:

1. The process of grouping the same pixel intensity
2. The number of intensities divided by the number of image pixels
3. Insertion of pixel intensity values into the image

After histogram equalization calculated, every image normalized and used as input parameters on CNN.

2. 4 Design CNN Architecture ResNet-50

In this study, the classification of children's facial expressions was carried out using the CNN architecture ResNet-50. The CNN Residual Network-50 (ResNet-50) architecture used in this study can be seen in **Error! Reference source not found.**.

Figure 3 shows the stages of the process of classifying children's facial expressions using the ResNet-50 architecture. Basically, the CNN architecture is divided into two major parts, namely, feature learning and classification. The Resnet-50 architecture consists of 4 stages as shown in Figure 3. This architecture, the network can take input images that have height, width multiples of 32 and 1 as channel size.

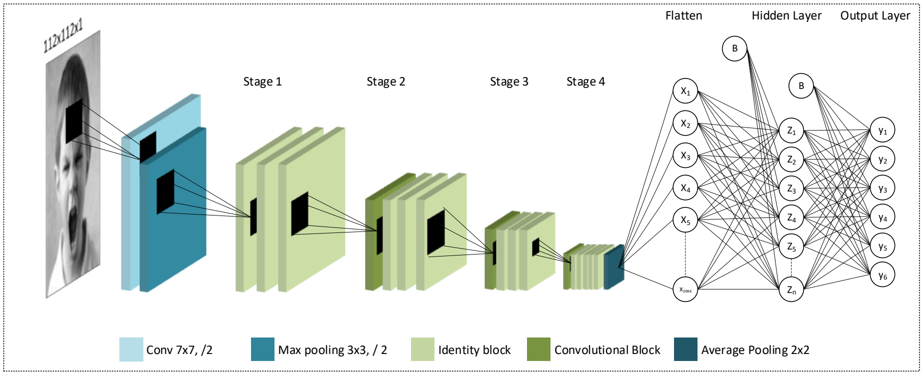


Figure 3 Resnet architecture

The input layer consists of a normalized grayscale image having 1 channel measuring 112 × 112 × 1. Each ResNet-50 architecture performs initial convolution and max pooling using kernel sizes of 7 × 7 and 3 × 3, respectively, which will be explained in more detail in section 2.4.1. The output of max pooling 3 × 3 is used as input for stage 1 which consists of 3 blocks where each block on stage 1 is an identity block (the size of the input and output dimensions is the same) and each block consists of 3 layers, regarding identity blocks described in section 2.4.2. Then, the output of the last block on stage 1 will be used as input for the first block of stage 2. Each first block on stage 2, stage 3 and stage 4 is a convolutional block (the size of the input dimension is not the same as the size of the output dimension) and the block after it is an identity blocks. The output of the 6th block on stage 4 is carried out with a max pooling operation of 3 × 3 with a total of 2048 channels, regarding convolutional block described in section 2.4.3.

2. 4.1 Convolution 7 x 7 and max pooling 3 x 3

The initial convolution process consists of a 7 × 7 kernel using random values, the number of channels is 64 stride 2 and padding 3. Next, max pooling is carried out with a kernel size of 3 × 3 stride 2 and padding 1. The convolution stage is 7 × 7 and max pooling 3 × 3 will reduce the dimensions of the image, to determine the results of dimension reduction can use the Equation (2), where n is the number of initial dimensions of the image, p is the amount of padding used, f is the size feature of each channel, and s is the number of strides used.

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

2. 4.2 Identity block

The identity block is the standard block used in ResNet and is suitable for cases where input activation has the same dimensions as output activation.

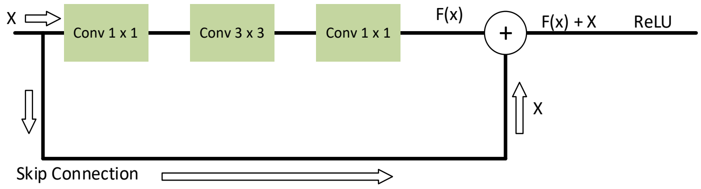


Figure 4 Identity block

Each identity block has 3 layers. The first layer of input is convoluted with a kernel size of 1 × 1, batch normalization and ReLU activation. The second layer, the output of the first layer is convoluted with a kernel size of 3 × 3, batch normalization and ReLu activation. The third layer in each block, the output of the second layer is convoluted 1 × 1 and batch normalization, at this stage ReLu activation is carried out after adding F(x) + X. In the identity block, the addition operation can be used directly because the number of dimensions of the input is equal to the number of dimensions of the output in the third layer. In Figure 4.14 there is an identity block process in each stage, all blocks in stage 1 are identity blocks, while stages 2, 3 and 4 in the first block use the convolutional block process (explained section 2.4.3) and the next block uses identity block.

2.4.3 Convolutional Block

Convolutional block is used when the input activation dimension is not the same as the output activation dimension. For example, the input dimension value is 28 × 28, while the result of the output dimension is 14 × 14 pixels, so in this case a convolutional block is used.

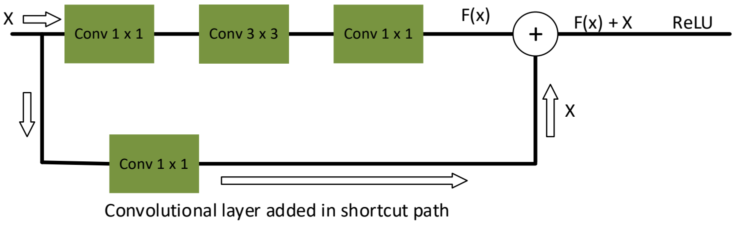


Figure 5 Convolutional block

Figure 5 is a convolutional block process by adding a 1 × 1 convolution operation to the skip connection so that the dimensions of the input X are the same as the dimensions of the output F(x), so that the sum of the dimensions of the input and output is the same. In Figure 4.14, the convolutional block process is used at stage 2, stage 3 and stage 4 in each first block.

2.4.4 Illustration of fully connected on CNN ResNet-50

Basically, fully connected is an artificial neural network training algorithm with a backpropagation model which includes three stages, namely the feed forward stage of the input training pattern, backpropagation of related errors, and weight adjustments. The network is given a pair of patterns consisting of the desired input pattern and output pattern, when a pattern is given to the network, the weights are changed to minimize the difference in the network output pattern and the desired output pattern [11].

The architecture of the artificial neural network that is built consists of three layers, namely, the input layer, the hidden layer and the output layer. The features obtained from the flatten result are used as input to the Backpropagation neural network.

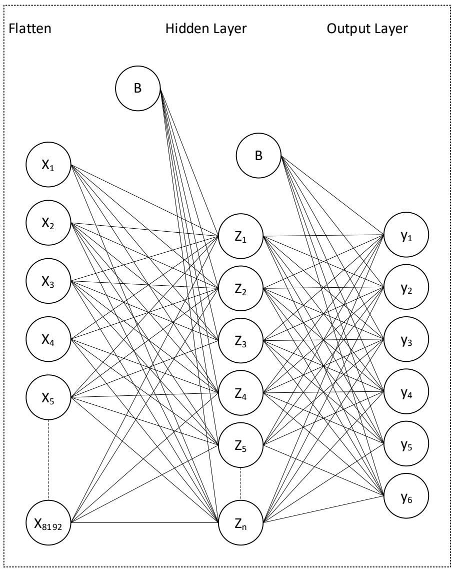


Figure 6 Backpropagation architecture

Description

B	:	Bias (Default 1).
X	:	Input Layer (Flatten) 8192 feature
Z	:	Hidden Layer
y	:	Output Layer 6 layer (Softmax)

3. RESULTS AND DISCUSSION

Testing of several hyperparameters is carried out to obtain optimal hyperparameters so that it is expected to improve performance in classifying expressions in children and also increase accuracy during the classification process. The CNN Resnet-50 architectural model has several hyperparameters. This study uses four hyperparameters, namely, the use of learning rate, number of epochs, batch size and dropout. The hyperparameter sub-sampling is shown in Table 1.

Table 1 Hyperparameter

No	Hyperparameter	Nilai
1	Epoch	50, 75, 100
2	Batch Size	32, 64, 128
3	Learning Rate	0.1, 0.001, 0.0001
4	Dropout	0.25

The first experiment was to test the hyperparameters using epochs of 50, 75, 100 with batch sizes 32, 64, 128. The epoch experiment was conducted to determine the effect of the number of epochs on achieving local optima, while the batch size experiment was carried out for each epoch due to hyperparameters. This relates to how much data is processed and how minimal the error rate is in learning. The experimental results of batch size and epoch are shown in Table 2.

Table 2 Result of experiment using hyperparameter batch size and epoch

Hyperparameter	Epoch 50		Epoch 75		Epoch 100	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Batch Size 32	99,70%	0,0214	99,79%	0,0115	99,86%	0,0156
Batch Size 64	99,64%	0,9944	99,83%	0,0135	99,76%	0,0193
Batch Size 128	96,49%	0,0270	99,89%	0,0121	99,73%	0,0138

The batch size and the number of epochs in the experimental results in Table 2 affect the accuracy and the resulting loss. Experiments with batch size 32 get the highest accuracy at epoch 100 with an accuracy of 99.86% and a loss of 0.0156. when the batch size is increased to 64, it gets an accuracy of 99.83% and a loss of 0.0135. The last experiment using a batch size of 128, obtained the highest accuracy of 99.89% and a loss of 0.0121 at epoch 75.

Table 3 Best experimental results use hyperparameter batch size 128 and number of epochs 75

No	Actual	Predicted								
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total	Precision	Accuracy
1	Angry	17	0	0	0	0	0	17	1,0000	1,0000
2	Disgust	0	308	0	0	0	0	308	0,9968	1,0000
3	Fear	0	0	699	0	0	1	700	0,9971	0,9986
4	Happy	0	1	0	1142	0	2	1145	1,0000	0,9974
5	Sad	0	0	1	0	732	0	733	1,0000	0,9986
6	Surprise	0	0	1	0	0	699	700	0,9957	0,9986
Average									0,9983	0,9989

Experiment shown in Table 3 was carried out with a total of 6 classes including angry, disgusted, afraid, happy, sad and happy. Angry expressions get a precision value of 1 and 100% accuracy with a total of 17 images, disgust expressions as many as 308 images get a precision value of 0.9968 and 100% accuracy, a fear expression as many as 700 images get a precision value of 0.9971 and an accuracy of 99.86% because there are 1 miss-classified data. happy expressions Experiments on happy expressions were carried out with a total of 1145 images, 3 of which were misclassified, namely 1 data classified as disgusted and 2 data classified as happy expressions. The precision of the happy expression is 1 and the accuracy is 99.74%. Another experiment is the expression of sadness with a total of 733 image data, only 1 data is misclassified in the expression of fear. The precision of the sad expression is 1 and the accuracy is 99.86%. Other experimental data with a happy target class of 700 images, and only 1 of them misses the classification on the expression of fear with a precision level of 0.9957 and an accuracy of 99.86%. From the good classes that have been tested, the average accuracy rate is 99.89%.

Another experiment was carried out by testing hyperparameters using learning rates of 0.01, 0.001 and 0.0001 with batch sizes 32, 64, 128. Learning rate experiments were carried out to determine the effect of the learning rate used during the training process until it reached a convergent level, while the batch experiment size is done every epoch because this hyperparameter relates to the amount of data that is processed and how minimal the error rate in learning is. The experimental results of batch size and epoch are shown in Table 4

Table 4 Experimental results of hyperparameter learning rate and batch size

Hyperparameter	LR 0,01		LR 0,001		LR 0,0001	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
Batch Size 32	99,86%	0,0156	96,53%	0,0372	69,77%	0,5815
Batch Size 64	99,76%	0,0193	87,24%	0,1175	50,43%	1,5920
Batch Size 128	99,73%	0,0138	90,67%	0,1749	16,67%	1,9141

The learning rate and batch size in the experimental results in **Error! Reference source not found.** greatly affect the accuracy and the resulting loss. The experimental results obtained the lowest loss value using a hyperparameter learning rate of 0.01 on a batch size of 128 with a loss value of 0.0138 and an accuracy rate of 99.79%. Another experiment using a learning rate of 0.001 resulted in the smallest loss value in batch size 32 with a loss value of 0.0372 and an accuracy of 96.53%, while the highest loss value was at a learning rate of 0.0001 with a batch size of 1.9141 and an accuracy of only 16.67% due to the very high learning rate used. small so that the training process requires a large number of epochs and the training process will run very slowly. This can be concluded in the experiments carried out, when the batch size is increased, it will affect the accuracy results produced.

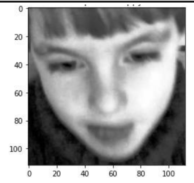
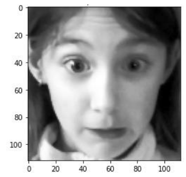
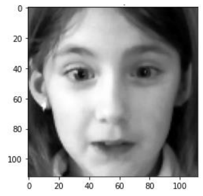
Table 5 is the best experimental result using a hyper parameter learning rate of 0.01 and a batch size of 128. The angry class was tested on 17 images with a precision value of 1 and an accuracy of 100%. Disgust class testing was carried out as many as 308 test images with a precision value of 1 and an accuracy of 100%. The fear class test uses 700 images, 3 of which miss classification on happy, sad and happy expressions. The precision value for the fear class is 0.9957 and the accuracy is 99.57%. Another test with a happy class as many as 1145 data, 2 of which are misclassified and both of the data are classified as happy. The precision value generated in the happy class is 0.9965 with an accuracy rate of 99.83%. The test results with the sad class as many as 733 images with a precision value of 0.9986 and 100% accuracy. The last class in this experiment is happy with 700 images, 7 of which are misclassified, namely 3 images classified as expressions of fear, 3 images classified as happy expressions and 1 image classified as sad expressions.

Table 5 Best experimental results on hyperparameter learning rate 0.01 and batch size 128

No	Actual	Predicted								
		Angry	Disgust	Fear	Happy	Sad	Surprise	Total	Precision	Accuracy
1	Angry	17	0	0	0	0	0	17	1,0000	1,0000
2	Disgust	0	308	0	0	0	0	308	1,0000	1,0000
3	Fear	0	0	697	1	1	1	700	0,9957	0,9957
4	Happy	0	0	0	1143	0	2	1145	0,9965	0,9983
5	Sad	0	0	0	0	733	0	733	0,9986	1,0000
6	Surprise	0	0	3	3	1	693	700	0,9943	0,9900
Average									0,9975	0,9973

Table 6 is some experimental results that are misclassified. Experiments on the first data in Table 6 also produce predictive values that are different from the actual values, where the classification results generated by the system are expressions of surprise but the target (actual value) is happy. The main cause of missing data classification is image input that is not clear (blur) so that the system cannot identify it correctly. Furthermore, experiments on the second data in Table 6.7 also produced predictions that were different from the target (actual). The system is difficult to identify these expressions because there are several facial similarities between the expressions of fear and surprise such as the eyebrows and mouth tend to open when experiencing an expression of fear or surprise. Likewise, on the contrary, which is shown in the fourth data in Table 6, the system identifies that the image data produces a surprised expression but the target of the image is an expression of fear.

Table 6 The results of the miss-classification experiment with a hyperparameter learning rate of 0.01 and a batch size of 128

No	Input	Aktual	Prediksi
1		Senang	Terkejut
2		Takut	Terkejut
3		Terkejut	Takut

4. CONCLUSIONS

Based on the test results obtained, conclusions can be drawn are tests using hyperparameter epoch and batch size can be concluded that the more loss value is generated, the accuracy will increase. This test experiment using hyperparameters produces a precision value with an average precision value of 0.9983 with an accuracy rate of 99.89%

Testing using hyperparameter learning rate based on batch sizes 32, 64 and 128 it can be concluded that the smallest loss value is found in the experiment using batch size 128 and learning rate 0.01 with a loss value of 0.0138 resulting in lower accuracy than the experiment with batch size 32 and learning rate of 0.01 which results in a loss value of 0.0156. Testing experiments using hyperparameter batch size 128 and learning rate 0.01 resulted in an average precision value of 0.9975 with an accuracy rate of 99.73%.

REFERENCES

[1] R. Ali, “Detektor Ekspresi Wajah Manusia,” J. Inform., vol. 16, no. 1, pp. 78–84, 2016.

[2] A. Garg and R. Bajaj, “Facial Expression Recognition & Classification using Hybridization of ICA, GA, and Neural Network for Human-Computer Interaction,” J. Netw. Commun. Emerg. Technol., vol. 2, no. 1, pp. 49–57, 2015.

[3] T. Ahmed, S. Hossain, M. Hossain, R. Islam, and K. Andersson, “Facial Expression Recognition using Convolutional Neural Network with Data Augmentation,” Apr. 2019. doi: 10.1109/ICIEV.2019.8858529.

[4] S. Anwar and M. Milanova, “Real Time Face Expression Recognition of Children with Autism,” Int. Acad. Eng. Med. Res., vol. 1, no. 1, pp. 1–8, 2016.

[5] S. Nagpal, M. Singh, M. Vatsa, R. Singh, and A. Noore, “Expression Classification in Children Using Mean Supervised Deep Boltzmann Machine,” 2019, pp. 0–0. Accessed: Aug. 28, 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2019/html/AMFG/Nagpal_Expression_Classification_in_Children_Using_Mean_Supervised_Deep_Boltzmann_Machine_CVPRW_2019_paper.html

[6] C. Herba and M. Phillips, “Development of facial expression recognition from childhood to adolescence: Behavioral and neurological perspectives,” J. Child Psychol. Psychiatry, vol. 45, pp. 1–14, Jan. 2004.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” ArXiv151203385 Cs, Dec. 2015, Accessed: Feb. 07, 2022. [Online]. Available: http://arxiv.org/abs/1512.03385

[8] R. A. Khan, C. Arthur, A. Meyer, and S. Bouakaz, “A novel database of Children’s Spontaneous Facial Expressions (LIRIS-CSE),” Image Vis. Comput., vol. 83–84, pp. 61–69, Mar. 2019, doi: 10.1016/j.imavis.2019.02.004.

[9] R. C. Gonzalez, R. E. Woods, and B. R. Masters, “Digital Image Processing, Third Edition,” J. Biomed. Opt., vol. 14, no. 2, p. 029901, 2009, doi: 10.1117/1.3115362.

[10] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, May 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.

[11] A. Hermawan, Jaringan Syaraf Tiruan Teori dan Aplikasi. Yogyakarta: Penerbit Andi, 2006.