

Extraction of Eye and Mouth Features for Drowsiness Face Detection Using Neural Network

Elis Fitriainingsih^{*1}, Endang Setyati², Luqman Zaman³

^{1,2,3}Sekolah Tinggi Teknik Surabaya

oyrasile1@gmail.com^{*1}, endang@stts.edu², luqmanz@gmail.com³

Abstract

Facial feature extraction is the process of searching for features of facial components such as eyes, nose, mouth and other parts of human facial features. Facial feature extraction is essential for initializing processing techniques such as face tracking, facial expression recognition or face shape recognition. Among all facial features, eye area detection is important because of the detection and localization of the eye. The location of all other facial features can be identified. This study describes automated algorithms for feature extraction of eyes and mouth. The data takes form of video, then converted into a sequence of images through frame extraction process. From the sequence of images, feature extraction is based on the morphology of the eyes and mouth using Neural Network Backpropagation method. After feature extraction of the eye and mouth is completed, the result of the feature extraction will later be used to detect a person's drowsiness, being useful for other research.

Keywords: Facial Feature Extraction, Frame Extraction, Neural Network, Backpropagation

1. Introduction

Facial recognition and facial expression recognition are essential in areas such as access management, human-computer communication, production control, e-learning, drowsiness detection and robotic expression methods. Emotional facials are categorized depending on the data input through video sequences or static images. The algorithms used in facial recognition are quite numerous and varied, but they all have the same three basic stages: (1) face detection; (2) facial extraction; and (3) facial recognition (face recognition). The face detection stage, as the initial stage of facial recognition, is the stage for determining and separating parts of a defined image as a face. The following stage is the facial feature extraction stage, extracting facial features and retrieving information. The last stage is facial recognition, identifying human face based on information obtained at the stage of facial feature extraction. The facial feature extraction stage is an important stage. Information which is a unique feature of every human face will be searched.

2. Research Method

2.1 Facial Feature Extraction

One of the processes of facial recognition is through feature extraction. Facial feature extraction is the process of extracting features of facial components such as eyes, nose, mouth, and others from human face image. Facial feature extraction is essential for initializing processing techniques such as face tracking, facial feature recognition or facial recognition [1]. "Facial features such as eye and mouth detection are important issues in facial image processing which will be used for many areas of research such as emotional detection and face identification" (Teo, Silva, & Vadakkepat, 2004) [2]. The main purpose of feature extraction is used to take the essential features of a face object. Some problems arise in the process of extraction, commonly called robusts. These problems are related with alterations in the position or angle of the image, in the intensity of light occurring at the time of facial image captured, in facial expression and so on resulting misrepresentation by the system [3]. The level of accuracy of the feature extraction process on the face will determine the quality of the results, simplifying the next process.

Feature Extraction is a feature retrieval feature of a form which will obtain the value analyzed for the next process. Feature extraction is used to analyze and change the input information contained in a parameter pattern. The goals are to eliminate redundant information, improve precision recognition, reduce quantity calculations and increase calculation speed. The

accurate features should have severability, stability and freedom. Severability refers to the fact that there are discrepancies between the features of different classes. The bigger the difference is recognized, the better. Stability means that the features of different patterns of the same class should be close. The closer are the patterns the better. Moreover, features are less affected by random noises. Finally, freedom refers to the fact that each feature selected in one pattern is irrelevant to each other [4].

2.2 Backpropagation of Artificial Neural Network

Neural Network (NN) is a network of a group of small processing units which are modeled on the basis of human neural networks. ANN is one of the information processing systems designed by mimicking the workings of the human brain in solving a problem by doing the learning process through the synapse of weight changes. ANN is able to recognize activities based on data. The data will be studied by ANN so as to have the ability to make decisions on data which has not been studied [5].

A lot of research on face recognition has been done. One of the commonly used methods is Artificial Neural Network (ANN). Some research studies face recognition using ANN method such as in Sigit Kusmaryanto's research, this research employs face recognition method of texture based feature extraction. According to the test results, face recognition system using PCA method as feature extraction and GLVQ can be well conducted in face recognition, obtaining the average level of face recognition accuracy of 97.5% for testing 120 facial images, 92.5% for 160 face image and 91% test with 200 facial images [6]. In Husdi's research of recognizing facial expressions of e-learning users, it suggested that the introduction of facial expressions of learners based on eye and mouth parameters managed to obtain better accuracy, so it can be used to provide input in Adaptive e-Learning System engineering. The test conducted on backpropagation Neural Network classification with feature of LBP and GLCM extension of highest accuracy of 88,89% shows better result than the classification result with backpropagation Neural Network with Gabor Wavelet extraction feature with highest accuracy obtained of 85,19% [7]. According to Zaenal Abidin's research, which developed facial expression recognition system (case study on MUG database), it is concluded that from several variations, it was found that optimal artificial neural network to recognize facial expressions is to set the number of hidden neurons by 12 and learning rate by 0.25. The results of the introduction made on the image of the MUG database obtained the results of recognition rate of 98.09% and false positive of 5 [8]. Based on the research conducted by Sigit Kusmaryanto in performing face recognition method of texture based feature extraction, it was found that face image pattern with preprocessing using histogram-based feature extraction method can be identified with ANN using backpropagation network. Moreover, best ANN test resulted on the number of units in hidden layer of 6, number of input units of 225, number of output units of 10, maximum epoch of 2500, target error of 0.001 and learning rate of 0.9 with a recognizable average of 95% [9].

Backpropagation Artificial Neural Network model is widely used to be applied to solve a problem related to identification, prediction, pattern recognition and so on. In repetitive practices, this algorithm will produce better performance. It means that the "interconnection weight" of ANN is closer to the proper weight (Research Neural Network, 1993). Another advantage of this ANN is its ability to learn (adaptive) and immune to the error (Fault Tolerance) with the advantages of ANN can realize a robust system consistently working well.

Backpropagation method was first introduced by Paul Werbos in 1974, reiterated by David Parker in 1982 and then popularized by Rumelhart and McClelland in 1986. In this backpropagation algorithm, network architecture uses multiple layers of network.

Backpropagation training method known as Generalize Delta Rule (GDR) is a supervised training where for each input pattern there is a target output pair. It is actually a gradient descent method to minimize the total square error in the output of the network calculation. The basic idea can be described with a simple relationship pattern: if the output gives results that do not match the unwanted target, then the weights will be corrected so that the error can be minimized and then the network response is expected to be closer to the appropriate value. In general, the objective of artificial neural networks in the training process is to obtain balancing between the network's ability to correctly respond to input patterns during training (it can be said of the ability to remember) and the ability to provide a proper assessment of a similar pattern of other entries. Hence, from the training process, a weighting value which will be used as a digging factor of the pattern will be formed. In this method, there are three stages in the training process, namely:

advanced feeding process from training input pattern, calculation and backpropagation of error and weight value adjustment.

At this stage of the training, a neural network will be practiced, namely: advanced feed process from the pattern of training input, calculation and backpropagation of errors which occur from weighting adjustment values, by changing the weight of the connection, both the connection weight between the input layer and hidden layer as well as between the hidden layer and output layer. If there is more than one hidden layer, then there is also a weight between the hidden layer itself. Solving new problems will be conducted if the training process is completed. The phase is the process of using/testing by using weights which have generated from the training process which has been completed.

2.2.1 Feedback Training Phase (Backpropagation)

Backpropagation Network consists of several layers in which each unit on a single layer is fully connected with each unit on the layer above or below it. However, the bias is only fully connected with the upper layer units presented in Figure 1.

Figure 1 shows a network which has one hidden layer with input layer X, hidden layer Z, output layer Y, giving bias value, an enter with the same fixed value, that is 1.

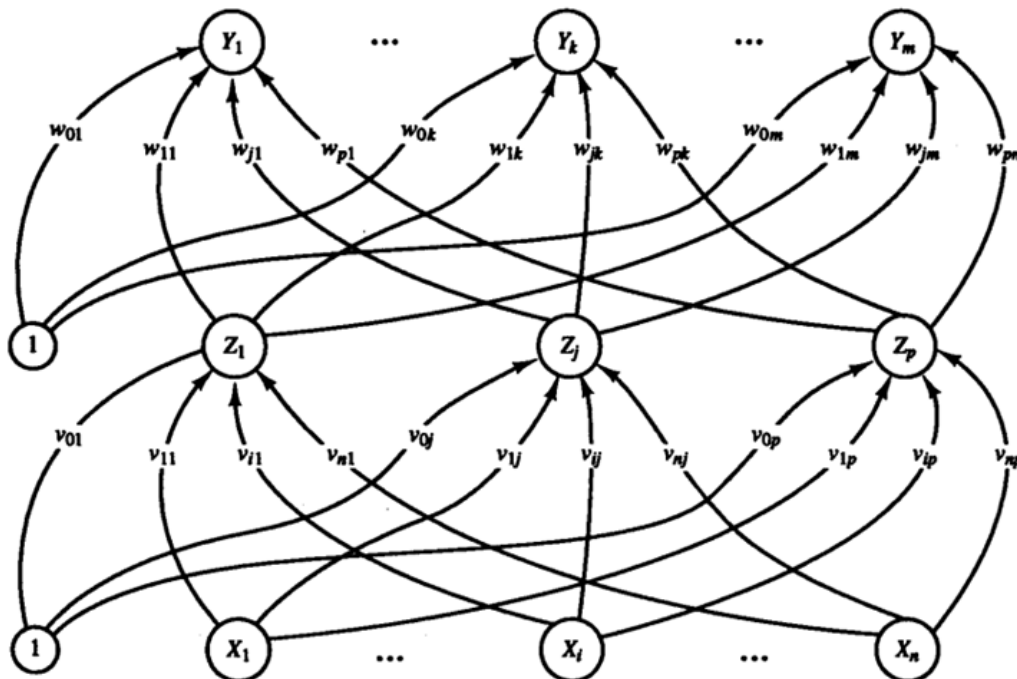


Figure 1. Artificial Neural Network with One Hidden Layer

Backpropagation algorithm consists of two processes, feed-forward and error backpropagation. During feed-forward, each input unit receives X or an input signal from the outside, then the signal is propagated to each unit on the hidden layer Z. Each hidden unit is calculated according to its activation function. Afterwards, the signal will be sent to each unit on the output layer and will be calculated according to the activation function as well, producing the output signal as the network response with the existence of the input pattern.

In backpropagation, each unit output is compared with the result of activation calculation Y with target value t to get error. After obtaining the error, it will be calculated by value of δ_k , then the error value at unit output will be backpropagated to each unit in hidden layer. Furthermore, the error is used to fix the weighting factor between the unit output with the hidden unit; thus, the error will be searched from the hidden output to improve the weighting factor between the input unit. For more details, it can be explained as follows:

Step 0: Initializing the weighing factor (given a small value randomly)

Step 1: Repeating steps 2 through 9 until the stop condition is met

Step 2: Performing step 3 through 8 for each pair of training pairs

2.2.2 Feedforward

The calculation of feedforward is illustrated by Equations 1 to 13. Each of which can be described as follows:

Step 3: Receiving the input signal X_i for each input unit ($X_i, i = 1, \dots, n$), and the signal is Propagating the signal to the upper unit of the hidden layer (hidden units)

Step 4: Summing up the weighing factor for each hidden layer

$$Z - in_j = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (1)$$

Afterwards, it is calculated according to the activation function:

$$Z_j = \int (Z - in_j) \quad (2)$$

Due to the application of sigmoid function then the following equation is used:

$$Z_j = \frac{1}{1 + \exp(-z - in_j)} \quad (3)$$

Then the signal will be sent to all units above it (output unit).

Step 5: Summing the weighing factor for each output unit ($Y_k, k = 1, 2, 3 \dots m$).

$$Y - in_k = W_{ok} + \sum_{j=1}^p Z_j W_{jk} \quad (4)$$

It then is calculated according to the activation function

$$y_k = \int (y - in_k) \quad (5)$$

2.2.3 Backpropagation of the existing Error

Step 6: Receiving the target pattern according to the input pattern during training and calculates the error for each output unit ($Y_k, k = 1, \dots, M$).

$$\delta_k = (t_k - y_k) \int' (y - in_k) \quad (6)$$

Because of $f'(y - in_k) = y_k$ by using the sigmoid function, then:

$$\int' y - in_k = \int' (y - in_k)(1 - \int y - in_k) \quad (7)$$

The correction of the weighing factor is calculated (then fixes W_{jk}).

$$\Delta W_{jk} = \alpha \cdot \delta_k \cdot Z_j \quad (8)$$

Then the correction is calculated by:

$$\Delta W_{ok} = \alpha \cdot \delta_k \quad (9)$$

And the value of δ_k will be sent to the unit layer below it.

Step 7: Adding its input delta (from the layer unit above it) for each hidden unit ($Z_j, j=1, \dots, p$).

$$\delta - in_j = \sum_{k=1}^m \delta_k W_{jk} \quad (10)$$

Then it is multiplied by the derivative of its functionality to calculate the error.

$$\delta_j = \delta - in_j \int' (y - in_j) \quad (11)$$

The weighing correction is then calculated (used to fix V_{ij})

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (12)$$

The bias rectification is calculated (to fix V_{oj})

$$\Delta V_{oj} = \alpha \delta_j \quad (13)$$

2.2.4 Fixed the Weight and Bias

Improving the weight and bias is described by Equation 14 and 15, consisting of the following steps:

Step 8: Fixing bias and its weighing ($j = 0, \dots, p$) for each unit output ($Y_k, k = 1, \dots, m$)

$$W_{jk}(\text{new}) = W_{jk}(\text{old}) + \Delta W_{jk} \quad (14)$$

each hidden unit ($Z_j, j=1, \dots, p$) is fixed bias and its weighing ($j = 0, \dots, n$)

$$V_{jk}(\text{new}) = V_{jk}(\text{old}) + \Delta V_{jk} \quad (15)$$

Step 9: Testing the dismissal condition (the end of iteration).

3. Research Result and Discussion

3.1 Process Design

The feature extraction simulation system for face recognition has several process stages to facilitate usage visualization. Figure 2 is a flowchart of a feature extraction simulation system. The process steps of the flow diagram are:

1. Inputting data video
2. Converting the video to the frame (frame extraction) to become an image sequence.
3. Performing feature extraction based on the morphology of the eyes and mouth, from the sequence of images/images, to produce morphological and labeled features.
4. Labelling morphological features determined from the ground truth feature previously set. From the labeled morphological features, the NN training process is conducted to yield trained NNs.

The trained NN will be used to classify features because the trained NN contains parameters of the NN algorithm used for the next process, i.e. to detect drowsiness and so on.

For morphological features, the eye and mouth areas are identified using a neural network. The mouth identified as the feature vector N , illustrated by the Equation (16) [10].

$$M = (W_m, H_m, H)^T \quad (16)$$

The eye area identified as the feature vector M is illustrated by the Equation (17).

$$N = (W_{max}, H_{max}, H_e)^T \quad (17)$$

The output vector is shown as the following formula.

$$M_1 = [1, 0, 0]^T$$

$$M_2 = [0, 1, 0]^T$$

$$M_3 = [0, 0, 1]^T$$

$$N_1 = [1, 0, 0]^T$$

$$N_2 = [0, 1, 0]^T$$

$$N_3 = [0, 0, 1]^T$$

Figure 3 presents the model of the mouth and eye character dots, wherein:
 M1, Abbreviation of the closing of a person's mouth when he/she does not yawn or does not speak;
 M2, An abbreviation for a person's wide open mouth when he/she evaporates;
 M3, Abbreviation of the mouth opening, usually during speaking;
 N1, Abbreviation of driver's eyes closed and fallen asleep;
 N2, Abbreviation of driver's open eye and high spirited;
 N3, Abbreviation of driver's narrow and sleepy eye.

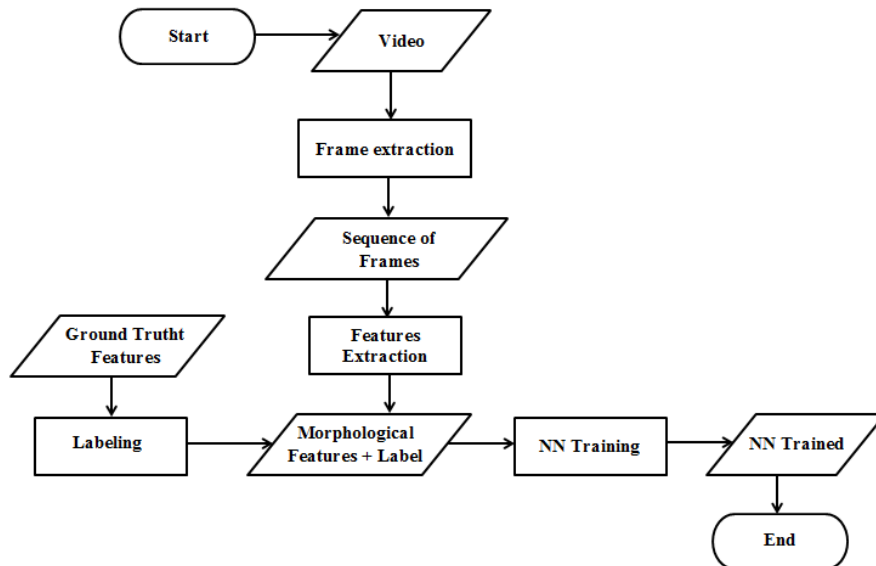


Figure 2. Flowchart of Feature Extraction

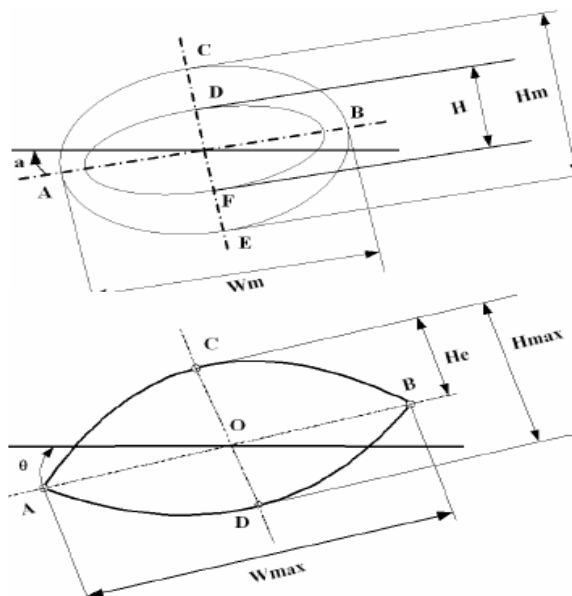


Figure 3. Model Dots of the Mouth and Eye Character

3.2 Research Result

A simulation experiment was performed using this arithmetic. The inputs are 10 video recordings, each of which is about 1 minute per video. The recording is done in 5 various conditions: including one who is speaking, laughing, yawning, smiling and having normal conditions (normal open eyes and mouth closing). The camera is placed right in front of the object (a person's face). The experiment results obtains each condition as presented in Figure 4.

From Figure 4, the following results are obtained:

The diagram of the training results of the Artificial Neural Network is shown in Figure 5, with the achievement of the output and targets shown in Figure 6. The best performance of the training process is presented in Figure 7, with the training state illustrated in Figure 8 and the regression results presented in Figure 9.

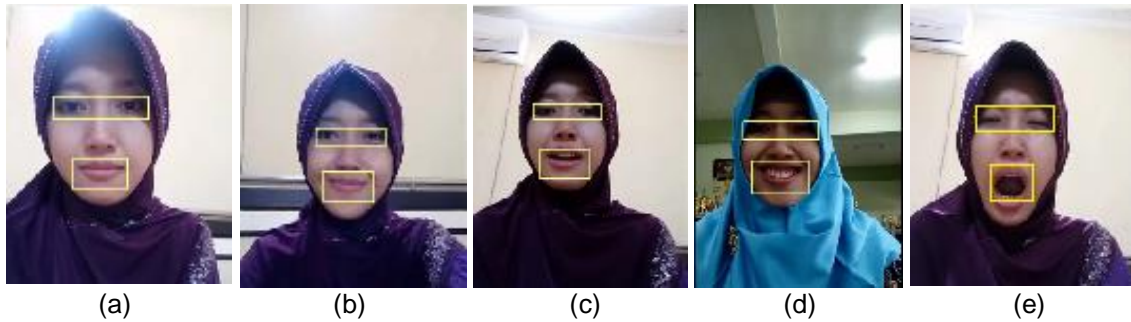


Figure 4. (a) Normal Condition; (b) Smile Condition; (c) Speech Condition; (d) Laughter; and (e) Yawning Conditions

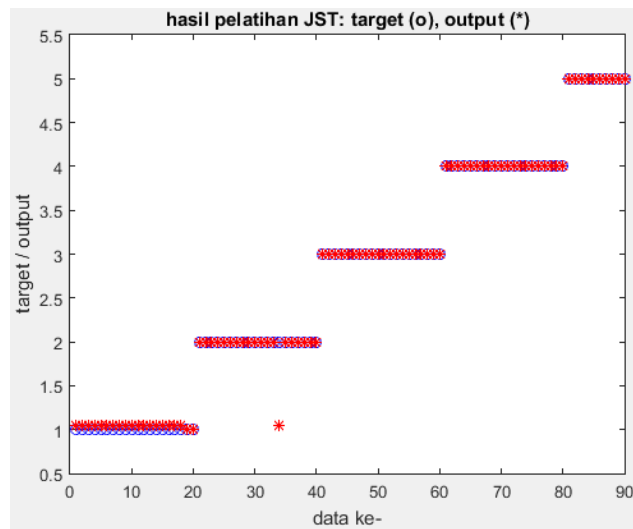


Figure 5. Diagram Result of Artificial Neural Network Training

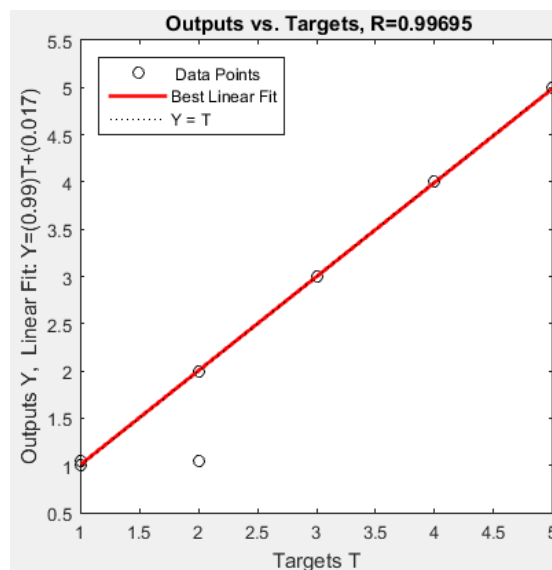


Figure 6. Diagram of Output and Target Achievement

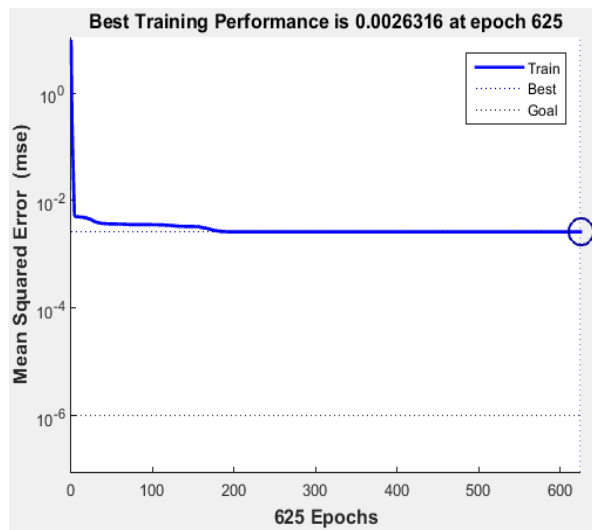


Figure 7. Performance Diagram of Best Training Process

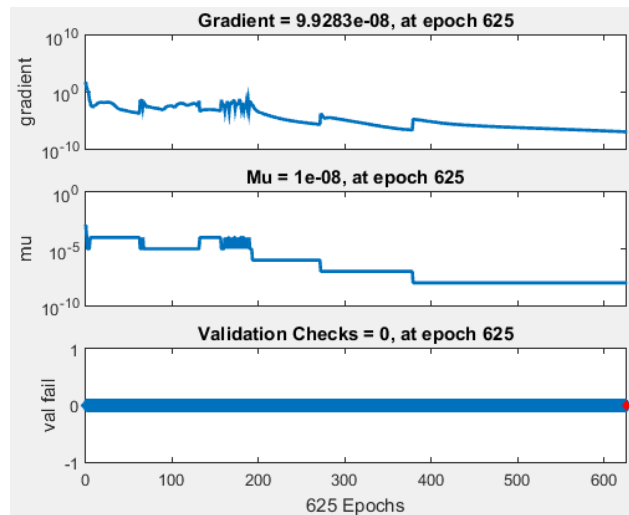


Figure 8. Diagram of Training State

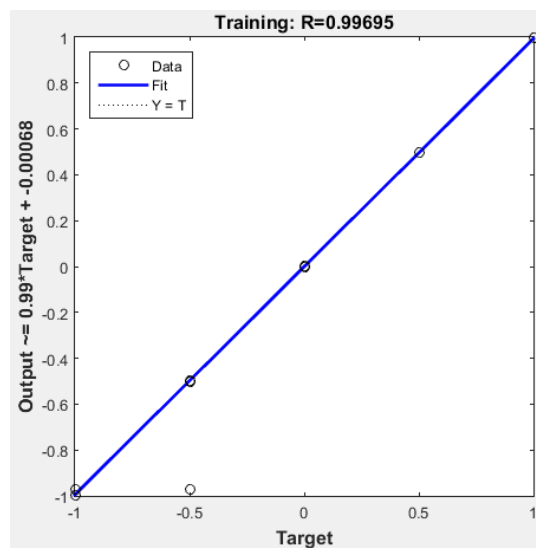


Figure 9. Diagram of Regression Result

4. Conclusion

This system monitors a person's face for a feature extraction, in which the presence of eyes and mouth is traced. This research uses Matlab to get feature extraction from the face, i.e. eye and mouth. From this study, a person's face can be detected as normal, talking, smiling, laughing or yawning. Further research which may be possible to conduct is detecting an individual's drowsiness.

Notations

x = Training data for input, $x = (x_1, \dots, x_i, \dots, x_n) \rightarrow i - n$.

t = Training data for output (target), $t = (t_1, \dots, t_k, \dots, t_m) \rightarrow k - m$.

α = Learning rate (usually between 0-1).

z = Hidden unit, $z = (z_1, \dots, z_j, \dots, z_p) \rightarrow j - p$.

V = Weight for the hidden unit, expressed in V_{ij} , where i : index input; j : index hidden, so V_{13} meaning the weight of input 1 leading to the 3rd hidden.

Y = Unit of output, $Y = (Y_1, \dots, Y_k, \dots, Y_m) \rightarrow k - m$.

W = Weight for output unit, expressed in w_{jk} , where j : index hidden; k : index output so that w_{21} means the weight of the 2nd hidden leading to the 1st output.

Bias = expressed as the 0th node, so w_{02} means the weight of the hidden layer bias to the 2nd output. (as well as for v).

δ_k = Error correction factor for w_{jk} weights.

δ_j = Error correction factor for v_{ij} weights.

References

- [1] S. R. Benedict and J. S. Kumar, "Geometric shaped facial feature extraction for face recognition," IEEE Int. Conf. Adv. Comput. Appl. ICACA, Pp. 275–278, 2017.
- [2] S. Alexander, "Application Program on Face Recognition in Real Time Using Backpropagation and Wavelet Haar Methods," 2013.
- [3] D. Susantyo, "Feature Extraction for Face Recognition on Mongoloid Race Using Principal Component Analysis (PCA)," Pp. 1–9.
- [4] G. He and J. Gan, "A Method for Singular Value Feature Extraction of Face Image," Pp. 37–40, 2004.
- [5] S. N. Luthfie, "Implementation of Artificial Network Nerves Backpropagation on Face Recognition Application Using Various Distance Using MATLAB 7.0," 2007.
- [6] I. Hermawan, "Development of Face Recognition System Using Generalized Learning Vector Quantization (GLVQ)," Pp. 6–8, 2015.
- [7] Husdi, "Facial Recognition on E-learning Users Using Artificial Neural Network with Extraction Feature of Local Binary Pattern and Gray Level Co-Occurrence Matrix," J. Ilm. Ilk., Vol. 8, Pp. 212–219, Desember 2016.
- [8] Z. Abidin, "Development of Facial Expression Recognition System Using Artificial Network Nerves Backpropagation (A Case on MUG Database)," J. Mat. Murni dan Terap., Vol. 5, No. 1, Pp. 21–30, 2011.
- [9] S. Kusmaryanto, "Artificial Network Nerves Backpropagation for Face Recognition with Extraction Feature Based Histogram Method," Vol. 8, No. 2, Pp. 193–198, 2014.
- [10] Y. D. Q. J. Ying, S. Jing, and Z. Wei, "The Monitoring Method of Driver's Fatigue Based on Neural Network," No. c, Pp. 3555–3559, 2007.

