Deep Learning for Face Identification

**Abstract.** The human identity authentication system which has high accuracy is needed in this era. Human identity authentication is often used called biometric systems with non-transferable characteristics. Face is one of biometric system that has a high level of challenge, with its variability such as ligthing, pose, and expression are very significant in classification. Facial identification uses computer algorithms that can generate face from camera capturing, then detect its similarities with face data that have been trained before, so that the computer can identify the face. In this paper used one method of deep learning for facial identification. Deep learning provides a powerful structure for supervised learning. A deep network can indicate functions of increasing complexity by adding more layers and more neurons within a layer. One method of deep learning is deep feedforward network. This paper uses 4 × 10 × 8 images data that have a resolution of 160 × 120 pixels. The target used is the identity of 4 people. The best model is obtained from the highest average performance of classification value testing using 10-fold cross validation. The conclusion of this paper are more number of hidden layer and more number of neurons within hidden layer not always make better performance of classification. But they can make time longer for training. In this paper, best model is 3rd fold for train data with 2 hidden layers and 75 neurons within hidden layer

1. Introduction

The human identity authentication system which has high accuracy is needed in this era. Human identity authentication used is called biometric system which advantages of non-transferable characteristics. Face is a biometric system that has a high level of challenge. It is in its variability, such as antiquity, pose, expression, lighting, etc, which is very significant. Antiquity is the time between images captured and image verified. Over time, the face also experiences aging and changes, start from children, adolescents, to adults. Pose is the angle of the face that is towards the camera such as directly facing the camera, head tilted left, right sloping, facing left or right, etc. Expression is a view lineament describing feelings such as happy, sad, embarrassed, etc. And then lighting is the spread of light that can create new shadow effects around the face [1].

Most of recognition algorithms rely on pattern recognition using pattern recognition using statistical leatning techniques. Face identification system usually does not use the ratio of distance between facial landmarks, such as length of nose or distance between two eyes, because this is not very characteristical. [1]. Face identification or face recognition uses a computer algorithm that can generate catching image face using camera, detect the pattern of it, also compare it with faces data that had been trained before, so computer can identify the face There are many research methods that process face identification. Face identification has been done by Turk [2] using eigenfaces for recognition, Zhao [3] using literature survey, and Ahonen [4] using local binary patterns. Method that used in this research is Deep Feedforward Network.

1. Literature Review

This part will explain about deep learning, deep feedforward network, and performance of classification.

* 1. Deep Learning

Deep learning methods aim featuring hierarchies with composition of lower level features form higher levels of features. They include learning methods for many deep architectures, including neural networks with many hidden layers [5] and also many neurons within it [6]. Theoretical about deep learning reviewed by Bengio [7] which the kind of complicated functions is suggested to need deep architectures for representing high level abstractions [8].

A deep learning architecture is stack of simple modules that many of them compute non-linear for mappings. That stack also called layer makes a multilayer. Each layer increases selectivity and constantly of representation of input. Which multiple non-linear layers, for example a depth of 5, a system can implement a very complicated functions that are simultaneously sensitive to minute details and insensitive to large variations such as lighting, pose, background, etc [9].

Deep learning provides a powerful structure for supervised learning. A deep network can indicate functions of increasing complexity by adding more layers and more neurons within a layer. It can make easy for a person to do most tasks about mapping input to output via deep learning rapidly, given enough large models and datasets of labeled training [10].

* 1. Deep Feedforward Network

One method of deep learning is deep feedforward network. Deep feedforward networks are the classical deep learning models. The aim this method is to approximate function of *f*. This method illustrates a mapping y = *f*(x; θ) and trains the value of parameters (θ) for result best function approximation. This model called feed forward because the algorithm or information flows from input to input, so there are no feedback connections (like Recurrent Neural Network) which outputs of the model are feedback into itself. The function will be evaluated from input, then hidden layer (or called intermediate connection for function of f), and the last in output layer [10]. Basically, a deep feedforward network has 3 principle layers, those are input layer, hidden layer, and output layer. This method can vary the amount of hidden layer and the amount of neurons within hidden layer [11]. Hidden layer gets weighted input from input layer and sends their results to the next layer.

|  |
| --- |
|  |
| **Figure 1.** Architecture of Deep Feedforward Network. |

* 1. Activation Function

In neural network, there is activation function which it is function that draw relation between input and output that result in linear form or non-linear form [12]. Activations function that used in this paper are shown below.

1. Rectified Linear Unit (ReLU). ReLU is the activation function that reduced vanishing gradient by equation 10. In other words, this function force the value less then 0 to 0 [12].

  (1)

With *pj* is input of each neuron, *qj* is output of each neuron after entering activation function, and *j* noted to number of neurons until *nh*.

1. Softmax. Softmax is a non-linear function that used real number for input and range 0 to 1 (probabilistic) for output. Sum of probabilistic of output is 1. Often, softmax is used for multiclass classification in output layer [12].

  (2)

With rk is input of each neuron in output layer,  is output of each neuron in output layer after entering activation function, and k is amount of classes.

* 1. Loss Function

Loss or loss function is function used as criterion which must minimize that can be solution for model comparing [13]. Loss is very closely related with activation function in the output layer. Softmax activation function will produce a probability value, where the error calculation between two probability distributions called cross-entropy [14]. Technically, cross-entropy estimates the difference between probability of output layer result and target in 0 and 1 form. Calculation of cross entropy is formed by equation 6.

  (3)

Where *yk* is target value for 0 and 1 and is predicted value

* 1. Optimization

The optimization used in this paper is Adam. Adam (adaptive moment estimation) is adaptive training optimization algorithm that specially designed for training in deep learning. Adam optimization is combination of Adagrad and RMSprop to get parameter which can get best performance [15]. In optimization parameter used Adam, the first one we must do is calculate the gradient from loss function to parameter such as bias and weight. To calculate the gradient, we can use a chain rule to get partial derivative [18]. After that, the second one is updating gradient exponential moving averages () and squared gradient () requiring hyper-parameters *β*1 and *β*2 (in default : *β*1 = 0,9 and *β*2 = 0,999). Moving averages are estimating first moment (mean) and second moment (un-centered variance) from gradient initialize with  and  Equations for calculating the value of and  are in below [15].

  (4)

  (5)

Where  is *gradient* for loss function that calculated from step one. While  is the multiplication of each vector element. *t* is defined by step of iteration [15]. The third one, the result of  and  are corrected to fix bias value. If the moments are not fixed, they can make small value of gradient of weight. Calculation of and  are in equation below [15].

  (6)

* 1. Classification performance

The performance of classification method is very important to get the best model. The classification model will produce results in discrete or continuous forms. Discrete forms will predict the class label of testing, yet continuous forms will represent an estimate of the prediction class probability. The results in discrete form are represented in the form of a confusion matrix [16]*.*

**Table 1.** Confusion matrix

|  |  |  |
| --- | --- | --- |
| ActualClass | Prediction Class | Total |
| *C*­1 | *C*­2 | *C*­3 | ... | *C*­*k* |
| *C*­1 | *n*11 | *n*12 | *n*13 | ... | *n*1*k* | *n*1 |
| *C*­2 | *n*21 | *n*22 | *n*23 | ... | *n*2*k* | *n*2 |
| ... | ... | ... | ... | ... | ... | ... |
| *C*­*k* | *nk*1 | *nk*2 | *nk*3 | ... | *nkk* | *nk* |
| Total | *n*1 | *n*2 | *n*3 | ... | *nk* | *N* |

Classification performance for multiclass classificationis defined by average of classification performances from each class (*Ck*­ where *k* = 1,2, …, *K* with *K* is number of class [17]. In this paper, classification performances used are accuracy, precision, sensitivity, and Fscore. *Accuracy* is amount of observation that are exactly classified. Fscore is a combination between precision and sensitivity. *Precision* is amount of observation which true positive predicted in all positive predicted, and sensitivity is amount of observation that correct classified to its category [17]. Formula of accuracy, precision, sensitivity, and Fscoreare in equations below.

  (7)

  (8)

  (9)

  (10)

1. Result and Discussion

Dataset used in this paper is primary dataset capturing form webcam with 640$×$480 pixel in dimension. Capturing image held in Department of Statistics in Institut Teknologi Sepuluh Nopember, Surabaya on April 2019. Dataset that used are images from 4 people (undergraduate students who class of 2015). Each person took 10 images with determined pose, so initially there were 40 images. Because of the amount of data is little, we pre-process data such as resizing (from 640$×$480 to 160$×$120 pixels), changing type of image from RGB to grayscale, and adding image with image augmented.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |  |
| --- | --- | --- | --- |
| *D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG*Real Image | D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_brigthnessmin.JPG*brightness decreasing*  | D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_brigthnessplus.JPG*brightness* increasing | D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_contrastmin.JPGContrast decreasing |
| D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_contrastplus.JPG*Contrast* increasing | D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_hor.JPGhorizontal *flipping*  | D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_rotatedright.JPG5 degree left rotating  | D:\Tugas Akhir\Tugas Akhir\New1\ulf (2).JPG_rotatedleft.JPG5 degree right rotating |

 |
| **Figure 2.** Image Augmentation. |

After preprocessing, we classified them with deep feedforward network method which the number of hidden layer are 2,3,4 and 5, and the number of neurons within each hidden layer are 25,50,75, and 100. And then, these are results of classified every hidden layer with their architectures. Training and testing data are divided by K-folds cross validation method. This method used to decrease bias in partition train and test data, which divided in a value called fold. This paper used 10-folds cross validation which the ratio between train and test data is 288:32.

**Table 2.** Performance of Classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of hidden layer** | **Number of neurons** | **accuracy** | **precision** | **sensitivity** | **Fscore** |
| 2 | 25 | 0.83125 | 0.83668 | 0.83125 | 0.83395 |
| 2 | 50 | 0.92813 | 0.93425 | 0.92813 | 0.93117 |
| **2** | **75** | **0.92813** | **0.93459** | **0.92813** | **0.93134** |
| 2 | 100 | 0.92188 | 0.92966 | 0.92188 | 0.92575 |
| 3 | 25 | 0.89375 | 0.90997 | 0.89375 | 0.90175 |
| 3 | 50 | 0.91563 | 0.92220 | 0.91563 | 0.91889 |
| 3 | 75 | 0.91875 | 0.92456 | 0.91875 | 0.92164 |
| 3 | 100 | 0.92500 | 0.93343 | 0.92500 | 0.92918 |
| 4 | 25 | 0.77500 | 0.78244 | 0.77500 | 0.77864 |
| 4 | 50 | 0.86563 | 0.88705 | 0.86563 | 0.87607 |
| 4 | 75 | 0.88125 | 0.89466 | 0.88125 | 0.88783 |
| 4 | 100 | 0.91563 | 0.92134 | 0.91563 | 0.91847 |
| 5 | 25 | 0.75313 | 0.77551 | 0.75313 | 0.76390 |
| 5 | 50 | 0.80625 | 0.83637 | 0.80625 | 0.82065 |
| 5 | 75 | 0.83125 | 0.84485 | 0.83125 | 0.83794 |
| 5 | 100 | 0.82500 | 0.81335 | 0.82500 | 0.81838 |

The performances of classification in Table 2 are gotten by averages of testing value for accuracy, precision, sensitivity, and Fscore for 10 folds. In Table 2 we can see that not always more neurons in hidden layer make better performance of classification. Also it not always more hidden layer can make better performance of classification. But more hidden layer can make time of train the data longer than small number of hidden layer. Performances of classification obtained are all above 70% which indicated that the result is excellent for face classification. The best performance obtained is architecture with 2 in number of hidden layer 2 and 75 in number of neurons within hidden layer. They produce above 90% that indicated that this architecture is excellent for face identification. Accuracy value of 92.81%, precision value of 93.46%, sensitivity is 92.8%, and Fscore value is 93.13%. After that, we inspect further in every fold to get performance of classification for each fold. Table 3 is performance every fold in 2 hidden layers with 75 neurons.

**Table 3.** Perfomance of Classification Every Fold

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **fold** | **accuracy** | **precision** | **sensitivity** | **Fscore** |
| 1 | 0.93750 | 0.94444 | 0.93750 | 0.94096 |
| 2 | 0.93750 | 0.94444 | 0.93750 | 0.94096 |
| **3** | **0.96875** | **0.97222** | **0.96875** | **0.97048** |
| 4 | 0.90625 | 0.92222 | 0.90625 | 0.91417 |
| 5 | 0.93750 | 0.94444 | 0.93750 | 0.94096 |
| 6 | 0.90625 | 0.90526 | 0.90625 | 0.90575 |
| 7 | 0.93750 | 0.93750 | 0.93750 | 0.93750 |
| **8** | **0.96875** | **0.97222** | **0.96875** | **0.97048** |
| 9 | 0.87500 | 0.88095 | 0.87500 | 0.87797 |
| 10 | 0.90625 | 0.92222 | 0.90625 | 0.91417 |
| **Averages** | **0.92813** | **0.93459** | **0.92813** | **0.93134** |

In Table 3, the best performances of classification are 3rd fold and 8th fold. Both of them get same performance such as 96.88% of accuracy, 97.22% of precision, 96.88% of sensitivity, and 97.05% of Fscore. It means if we want to predict new image which that image must include target of training data, we can use model of 3rd fold or 8th fold. But, we can see value of loss functions in two of them to know the the best for model.

|  |
| --- |
|  |
| **Figure 3.** Architecture of Deep Feedforward Network. |

From Figure 3, it can be seen that loss function of 8th fold is more constant than 3rd fold. But if we look the value of loss function, the 3rd fold is smaller than 8th fold. Best model can have predicted by seeing the smallest value of loss function. So, we used 3rd fold to predicted a new image.

1. Conclusion

Based on result and discussion, the conclusion of this paper are more number of hidden layer and more number of neurons within hidden layer not always make better performance of classification. But they can make time longer for training. In this paper, best model is 3rd fold for train data with 2 hidden layers and 75 neurons within hidden layer. So, future research can use different of number of hidden layer and number of neurons within them to get the optimum value of performance of classification and minimum value of loss function.

1. References

[1] Dunstone T and Yager N 2009 *Biometric System and Data Analysis* Berlin: Springer Science

[2] Turk M and Pentland A 1991 Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* **3** 71-86

[3] Zhao W, Chellappa R, Phillips P J and Rosenfeld A 2003 Face recognition : A literature Survey *ACM Computing surveys* *(CSUR)* **35** 399-458

[4] Ahonen T, Hadid A, and Pietikainen M 2004 Face recognition with local binary patterns *In European Conference on Computer Vision* 469-481 Springer,Berlin Heidelberg

[5] Vincent, P , Larochelle, H , Bengio, Y , and Manzagol, P A 2008 Extracting and composing robust features with denoising autoencoders In *Proceedings of the 25th international conference on Machine learning* 1096-1103

[6] Karsoliya, S 2012 Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture *International Journal of Engineering Trends and Technology* **3** 714-717

[7] Bengio, Y 2009 Learning deep architectures for AI  *Foundations and trends® in Machine Learning* **2** 1-127

[8] Glorot X and Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks *In* *Proceedings of the thirteenth international conference on artificial intelligence and statistics* 249-256

[9] LeCun Y , Bengio Y , and Hinton G 2015 Deep Learning *Nature* **521** 436-444

[10] Goodfellow I, Bengio Y, and Courville A 2016 Deep Learning [Online]: https://www deeplearningbook.org/contents/mlp.html

[11] Sarle W 1994 Neural Networks and Statistical Models *The Nineteenth Annual SAS Users Group International Conference*  NC USA: SAS Institute Inc

[12] Nwankpa C, Ijomah W, Gachagan A, and Marshall S 2018 *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*  [Online]: https://arxiv org/abs/1811 03378

[13] Russell R , and Marksll R 1999 *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks* Cambridge: The MIT Press

[14] GoodFellow I , Bengio Y , and Courville A (2016) *Deep Learning*  Cambridge: The MIT Press

[15] Kingma D and Ba J 2014 Adam : A Method for Stochastic Optimization *International Conference on Learning Representations 2015* San Diego

[16] Tharwat A 2018 Classification Assessment Method *Applied Computing and Informatics* doi:doi.org/10.1016/j.aci.2018.08.003

[17] Sokolova M and Lapalme G 2009 A systematic analysis of performance measures for classification tasks *Information Processing and Management* **45** 427-437

[18] Bishop C 2006 *Pattern Recognition and Machine Learning* New York: Springer Science