Convolutional Neural Network Models for Deep Face Recognition on Limitation and Interfering Factors in Image Dataset

Manop Phankokkruad
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand 10520
Email: manop@it.kmitl.ac.th

***Abstract*—Face recognition is one of effective method often used for personal identification, the accuracy of the face recognition depends on many factors typically implemented at different places in unconstrained environments. Not only, the amount of images in the dataset are affected to the accuracy of face recognition but also the quality of the images is also an impact. For this reason, this work proposed the convolutional neural networks model to improve the accuracy of the face recognition under an insufficient a number of images in dataset and the images that contains an interfering factors. The challenge of this work is the regulating and configuring of many parameters the network for its best performance and suite for this conditions. The experiment results shown that the CNN model gives encouraging accuracy of the face recognition. Furthermore, this work also compared the accuracy with the different face recognition techniques such as Fisherfaces, Eigenfaces, LBPH, and MLP neural networks. For these result, CNNs were used as an efficient solution for improving the rate of recognition accuracy on this conditions.**

I. Introduction

Face recognition is one of the most important research topics and widely used for personal identification due to it has feasibility to apply in many purposes [1]. However, there are many factors that affected to the face recognition accuracy such as age of people [2], face shape, face texture, glasses, hair, and light exposure, etc. In addition, an interfering factors, algorithms, and quality of image dataset are the common factors which has an effect on accuracy [3]. Thus, any controllable factors should be well regulated to have less impact on the face recognition system as described by Jafri [4], and Givens et al. [5]. These factors can cause missed face recognition in case of individuals who have a similar face appearance [6]. In the same way, an algorithms have an effect on the face recognition accuracy. Many publications reported that each algorithm has a characteristics and provide the different accuracy. From the prior work [7], the algorithm selection is an importance step for implementing the face recognition system in the classroom because it is very hard to control student's facial expressions, gestures, and some environmental factors. These factors are also highly affecting the face recognition accuracy. An intuitive way, the standard OpenCV library is the interesting solution because it is general used and perform mostly conditions. Now the problem becomes much harder, Jaturawat et al. [8] reported the face recognition accuracy could not improve the recognition rate in the case of low quality and a bit of images in the dataset. This research only focused on the classroom conditions. They tried to control any affecting factors, facial expression and environmental conditions but the recognition rate still could not improve it better.

Convolutional Neural Networks (CNNs) have become the most popular approach among researchers in many field. CNNs have achieved significant performances for computing, and many researches are applied to explore the advanced model structures to solve the problem. The great success of CNNs for computing has applied for solving many hard problems, such as object detection, image retrieval, image segmentation, image recognition [9], feature extractions [10], and so on. There are many literature works and publications on CNNs that demonstrate acceptably high face recognition accuracy. An interesting article written by Vinay et al. [11]. They proposed the CNNs for the face recognition in video and trivial objects. The CNNs were trained using the combination datasets of labeled faces. This method leads to the effective face recognition system.

For this reason, the objectives of this work is to study the perform techniques for improving the accuracy of the face recognition. This work will propose the CNNs model to improve the accuracy of the face recognition under an insufficient a number of images in dataset and the images that contains an interfering factors. The challenge of this work is the regulating and configuring of many parameters the network for its best performance and suite for this conditions. Furthermore, this work will compare the results with the existing standard face recognition library and the other face recognition techniques.

The rest of this paper is structured as follows. Section 2 describes the state of the problem, basic of the convolutional neural networks, and the details of the proposed model. Section 3 describes the experiments, the configuration of the model, and evaluation the precision. Section 4 reveals the results, and some discussion. Finally, Section 6 presents the summary, conclusion, contribution and the future work.

978-1-5386-5892-5/18/$31.00 ©2018 IEEE
ICIS 2018, June 6-8, 2018, Singapore

II. Background and Approach

In this section, the state of the problem, theoretical basis of convolutional neural networks, proposed methods and related backgrounds are described.

A. State of the problem

Face recognition has become a popular and widely used personal identification technique. In the implementation, this technique uses only a computer with a camera, thus it easy to install and can be applied to any systems such as an access control system, surveillance, etc. In the prior work that it was described in Jaturawat et al. [8], the face recognition system was implemented in a classroom. Even though the face recognition system worked well for the purpose of the system, the recognition accuracy of system was lower than expectations, being only 75.18%. This accuracy produced by testing students in an actual classroom environment. This experiments were conducted and created a training model and test the accuracy by three face recognition algorithms, including Eigenfaces [12], Fisherfaces [13], and Local Binary Pattern Histograms (LBPH).

The recognition system was operated by manually taking the student pictures and proceeding to face recognition. There are many problems when conducting face recognition experiments in an actual classroom. The facial expressions, light level and a number of images are the problem which highly impact to the accuracy of the face recognition. While the students images are captured by the camera, it is in the nature of human behavior to have different facial expressions. The moving people made the different facial expression, sometimes they did not direct address to the camera. The motion of people sometimes caused of blurred images and reduced the quality of the input image. Moreover, light levels in the classroom that change everyday can also affect the quality of the input images. Therefore, the quality of images of students and interfering factors in the dataset are an important factors that effects face recognition accuracy. In existing dataset has limited of number of images that it has 8-15 images per a student, and collected without controlling facial expressions, light level, quality of images and other interfering factors.

For this reason, this work intends to improve the accuracy of face recognition by using the deep learning technique on the face recognition. The constraint on the limit of quality of images, and the interfering factors will be overcome. Consequently, convolutional neural networks is a good way to improve the face recognition accuracy when a limitation of dataset and interfering factors are occurred. This result of this work will be helpful for the face recognition system with a limit of number of image, quality of images by interfering of the various of factors, and the another face recognition systems perform under the similar conditions.

B. Convolutional Neural Networks

Convolutional neural networks (CNNs) are biologically-inspired variants of multi-layer perceptron (MLP) neuron networks. Not only CNNs were proposed to address all problems of simple neural networks, but also CNNs have the different kinds of layers. Also each different of layer performs different from the usual MLP neuron networks layers. Since CNNs are regarded as a deep learning application which has been very attracted because of their effective results [14].

The convolution layer calculates the convolution of the input feature maps with convolution kernels, and adds a bias. This operation can be formulated as depicted in Eq.(1).

$$x_j^l = f\left(\sum_i w_{ij}^l * x_i^{l-1} + b_j^l\right) \tag{1}$$

where * represents the operation of convolution. x_i^{l-1} is the i_{th} feature map in the $l-1_{th}$ layer and x_l^j is the j^{th} feature map in the l^{th} layer. The trainable parameters w_{ij}^l and b_j^l are the weight connecting x_i^{l-1} to x_j^l and the bias of the x_j^l, respectively.

CNNs are more complex than the standard MLP neural networks, thus this work will start by creating a simple structure. Subsequently the structure will be adjusted for state of the art results. The summarizes the network architecture are the input layer, convolutional layers, pooling layers, dropout layer, rectified linear unit layers, fully-connected layers, and output layers.

The first layer is the input layer. This layer feed the data into the network. The neurons of the input layer are passive thus they can not modify any data. Each value from the input layer is duplicated and sent to the next layer.

Convolutional layers comprise a set of filters with fixed size. it is used to operate convolution on the data. This generating is called the feature map, that it can highlight some patterns.

Pooling layers summarize the data by sliding a window across the feature maps. This layer applies either linear or non-linear operations on the data within the window.

Dropout layer, A dropout [15] is a regularization technique. During training, the randomly selected neurons are ignored. The effect is the network becomes less sensitive to the specific weights of neurons. This make the results in the network is capable of better generalization and less likely to over the training data. Dropout can be applied to input layer and the first hidden layer.

Rectified Linear Unit (ReLU) layers are the standard way for applying a non-linear function. Where f is a function of its input x is with $f(x) = tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$. CNNs with ReLUs train several times faster than their equivalents with $tanh$ units in all network layers as described in [16].

Fully-connected layers are used for understanding the patterns which is created by the previous layers. The neurons in this layer have full connections to all activation in the previous layer. Fully connected layers are defined using the dense class, and it takes a number of neurons and activation function as arguments. The benefit of with the fully-connected neural networks on CNNs is used for reducing a number of parameters to be learned. Likewise, the convolutional layers

are made of the small size kernels for extracting features, then it is fed to fully-connected layers. The training step of CNN is performed through the back-propagation and stochastic gradient descent as clearly described by Rumelhart et al. [17]. The misclassification error is reacted to the weights update of both fully-connected layers and convolutional layers.

In the output layer, a softmax activation function is utilized to output a probability of the network predicting each of the class values. In this case, a softmax is used for choosing the output with the highest probability, which can be used to make a classification value. In this case that wants to handle multiple classes, a softmax regression is applied. By assuming the labels is a binary: $y^{(i)} \in \{0, 1\}$. This paper uses a classifier to distinguish between two person images. A Softmax regression allows to handle $y^{(i)} \in \{1, \ldots, K\}$ where K is the number of classes.

The model implementation will be done using the deep learning framework such as Tensorflow, Keras, and cuDNN, etc. These toolboxs based on Python language that allowed for a fast and scalable prototyping. The experiment use of NVIDIA GPU for deep learning the neural network with stochastic gradient descent(SGD) [18] and provide efficient compiled functions for predicting with the model. This paper set the learning rate to 0.01.

III. Experiments

In this section, this paper discusses the experiment details, and configuration of the proposed model.

A. Prepare Image Dataset

This paper has created the image dataset, that the images are collected from 148 students in the actual classroom. The images were captured by the web camera on the different PC, and mostly of the collected images have various image qualities. This dataset collected the images with uncontrolled facial expressions, facial viewpoint and different light levels. An example of the images in first the dataset are shown in Fig. 1. This image data was ever used in the work of Jaturawat el al [8]. These factors rendered the database quality inadequate thus, this research created a new reference database with prescribed facial expressions and viewpoints that accorded with the possible face in the face recognition system. Therefore, this paper has eliminated, adjusted and improved the image dataset appropriate to the CNNs model. This dataset contains around 2,180 RGB images with 640x480 pixels, subdivided in 148 classes, containing between 7 and 15 images per class. Firstly, all images were applied the face detection by Haar-like features[19], and it was cropped and kept it is only face area. Moreover, these face images have been size-normalized and centered in a fixed size image of 46x56 pixels. Each face has been labeled with the name of the person pictured. To be suitable for this proposed model, this paper converted all images into grayscale. The examples of image in the new dataset are depicted in Fig. 2. The dataset is divided into two different datasets. The first set is used to train the CNNs, it contains 1,346 images. Likewise, the second set is used for testing, it contains 843 images.

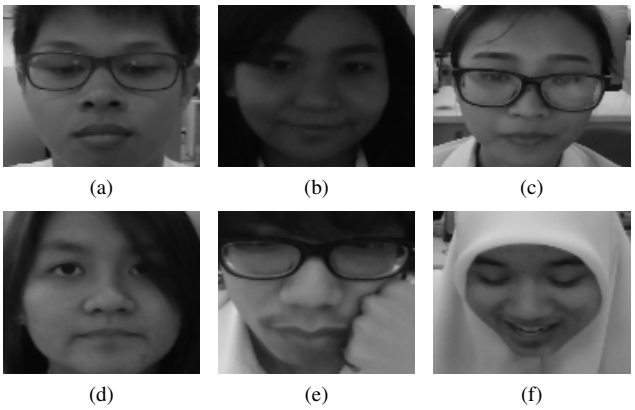


Figure 1: The example of images in the first dataset with an interfering factors include (a) close eyelid (b) low light (c) big eyeglass, (d) hair covering face, (e) hand covering face and (f) bow the head, respectively

B. Model and Configuration

As described in the previous section that a structure of CNNs composes of multi-layer. Consequently, this paper creates the CNN model includes one input layer, two convolutional layers, two hidden layers, one dropout layer, two pooling layer and output layer. The details of the different types of layers are illustrated in Fig. 3. This paper defines the model as a sequential model of layers, and adds layers one at a time with network structure. Furthermore, the configuration of the parameters and more details of each layer are the following description.

- At input layer, it can be specified by creating the input layer with setting to 2,576 input neurons. This value comes from the dimension of the input image.
- Convolutional layer 1 applies with 30 feature maps, and kernel size 5x5. In the same way, Convolutional layer 2 applied with 15 feature maps, and kernel size 3x3.
- Pooling layer 1 performs max pooling with with the pool size of 2x2 patches. Pooling layer 2 performs max pooling with the pool size of 2x2.
- Dropout layer is the set with a probability of 20%. This layer is used to reduce overfitting [15].

This paper uses a fully-connected network structure in two hidden layers, that is defined using the Dense class. In the same way, it can define the number of neurons in the layer as the first argument.

- The hidden layer 1 is the fully-connected layer with 128 neurons and rectifier activation. In the same way, the hidden layer 2 is also the fully-connected layer with 50 neurons and rectifier activation. It initializes the network weights to a small random number generated from a uniform distribution.

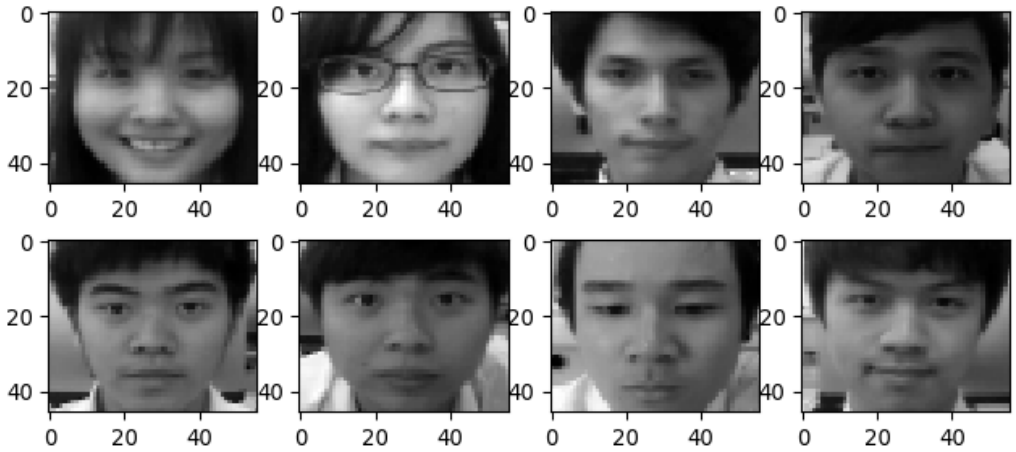


Figure 2: An example of image dataset

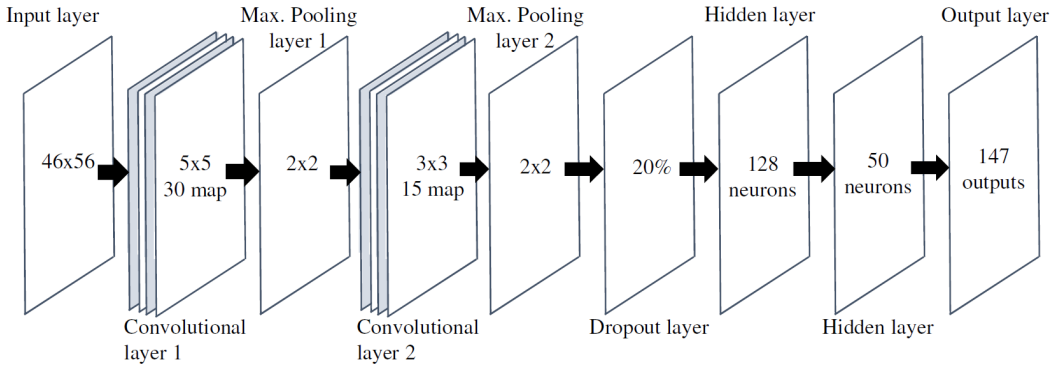


Figure 3: Topology of the convolutional neural network

- Finally, the output layer has 148 outputs. In addition, the sigmoid activation function in the output layer to ensure our network output is between 0 and 1 and easy to map to a probability of class.

As described, the details of each layer in such layer type, and output shape are summarized in Table I. These model informations were extracted from Tensorflow and Keras. A number of iterations for training is crucial in getting the result to reach a good value of loss and accuracy on the test data. This paper finds the best number of iterations during the experiments by varying the number of iterations. It is also optimized most by using ReLU activation. From many experiments reveal that the 50 iterations are the optimized value. After also of optimizing, though there are around 148 classes to choose from our network successfully identified people with the high degree of confidence.

Furthermore, this work also conduct the experiment the face recognition by using MLP neural networks in order to compare the the performance of this CNN model. The MLP neural network consists of three different layers; input layer, output layer and hidden layer. For the input layer, we scale all image size down into 46x56 pixels, and it will be 2,576 neurons.

Table I: The details of layouts of each network

Layer type	Output shape
conv2d_1 (Conv2D)	64, 42, 52
max_pooling2d_1 (MaxPooling2)	64, 21, 26
conv2d_2 (Conv2D)	32, 19, 24
max_pooling2d_2 (MaxPooling2)	32, 9, 12
dropout_1 (Dropout)	32, 9, 12
flatten_1 (Flatten)	3456
dense_1 (Dense)	128
dense_2 (Dense)	50
dense_3 (Dense)	148

In hidden layer, this model is a simple neural network with one hidden layer with the same number of neurons. Thus, the number of neurons in hidden layer is 2,576 neurons, that it equal to the number of input layer. A rectified linear units (ReLUs) is used as the activation function. ReLU activations are the simplest non-linear activation function. That is, the

output is 0 if the input is less than 0, and the output is equal to the input otherwise. In output layer, we define the 148 outputs, which are the number of students in the dataset. Furthermore, a softmax activation function is used to turn the outputs into possible values and allow one of the 148 classes to be selected as the model's output prediction.

IV. Results and Discussion

This paper applies the CNNs to recognize the face images. The proposed method has an ability to learn the face image, that it is represented in the training dataset, and relates it to the best output variable.

Although the dataset has a bit number and low quality of images, the result reveals that this CNN model able to successfully classify the face into 148 classes accurately. In addition, student face are very similar patterns, and the differences of faces are proved difficultly, this model can perform very well. The results shown that accuracy of the face recognition is 99.58% and 0.42% of loss.

Fig. 4(a) shows the accuracy obtained in both subsets, approximately 97.77% for the training and 99.58% for the validation. For the validation set of images, the accuracy obtained with the configuration defined in the previous section with respect to the error function, its evolution is shown based on 50 iterations(epochs). From the plot of model loss as shown in Fig. 4(b), the model has comparable performance on both train and validation datasets.

To evaluate the model efficiency, the proposed model was applied to the Japanese female facial expression (JAFFE) dataset [20]. This dataset contains 217 photos of 10 Japanese female models posing various expressions. This work chose this dataset because it contains the facial expressions similar to the student dataset and any standards for emotional facial expressions. In this case, the results shown that accuracy of the face recognition is 100.00% and 0.00% of loss. The evaluation was done with the same configuration, and based on 50 epochs. The model accuracy and model loss of the JAFFE dataset are as depicted in Fig. 5(a) and 5(b), respectively.

Furthermore, this work compares the results with the another recognition techniques including Eigenfaces, Fisherfaces, LBPH, and MLP neural networks. The accuracy of Eigenfaces, Fisherfaces, and LBPH were done by Jaturawat el al [8], which are 46.88%, 75.45%, and 79.65% respectively. In the case of MLP neural network technique, the model accuracy performs based on 50 epochs, and one hidden layer. This technique gave the recognition accuracy at 94.06%. The comparison of accuracy on different techniques are shown in Fig. 6. The proposed CNN model achieves good performance in two datasets. Furthermore, this model performed better accuracy than the others face recognition techniques.

V. Conclusion

This work proposed CNN model to improve the accuracy of the face recognition under a limitation of quantity and interfering factors that affected to the quality of images in the dataset. The challenge of proposed model is the regulating and configuring of many parameters the network for its efficient and suite for environmental conditions. The experiment results shown that CNN model gives the higher recognition accuracy than the prior techniques, and also performs with computational speed. Furthermore, this work compared the accuracy with the different face recognition techniques such as Fisherfaces, Eigenfaces, LBPH, and MLP neural networks. For these result, CNNs were used as an efficient solution for improving the rate of recognition accuracy on this conditions.

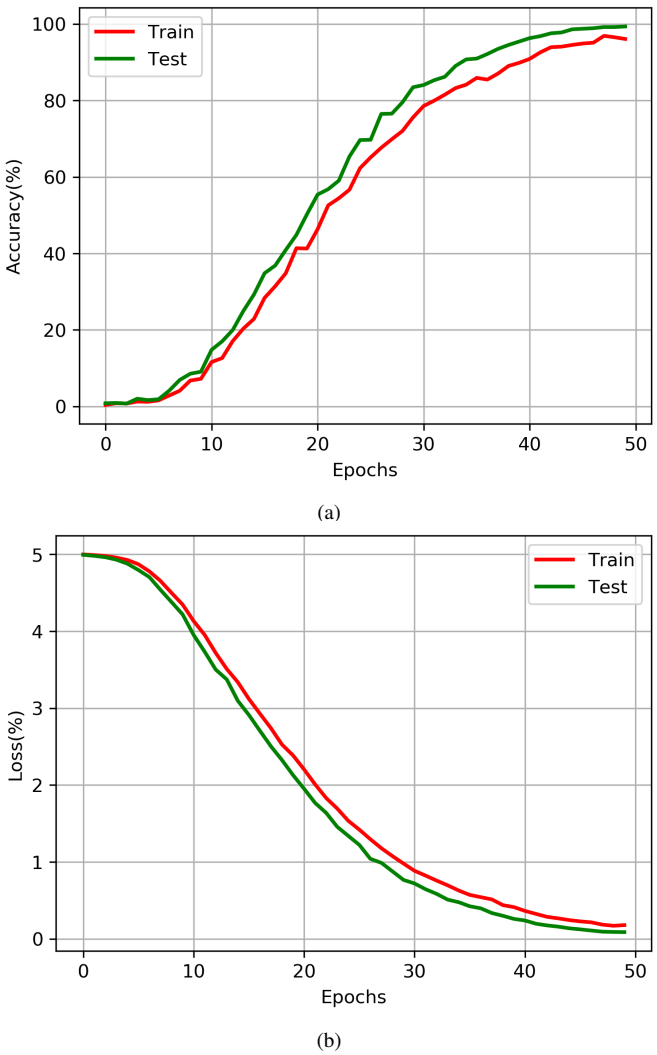


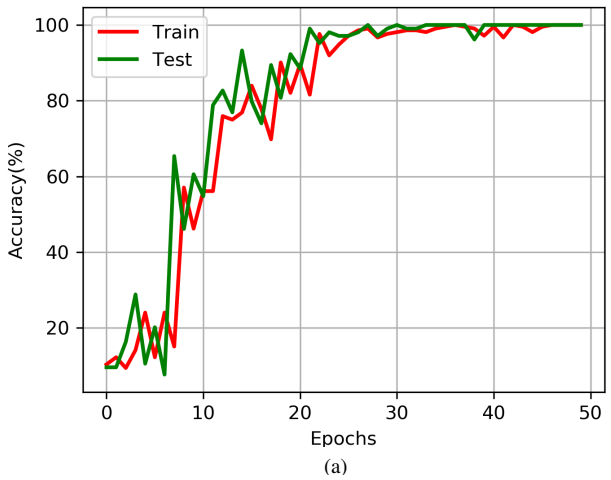
Figure 4: (a) Model accuracy and (b) model loss

The contributions of this work is a guideline of using CNN model to improve the accuracy of face recognition in case of a bit number of images and interfering factors in the image dataset, which highly impact to the accuracy.

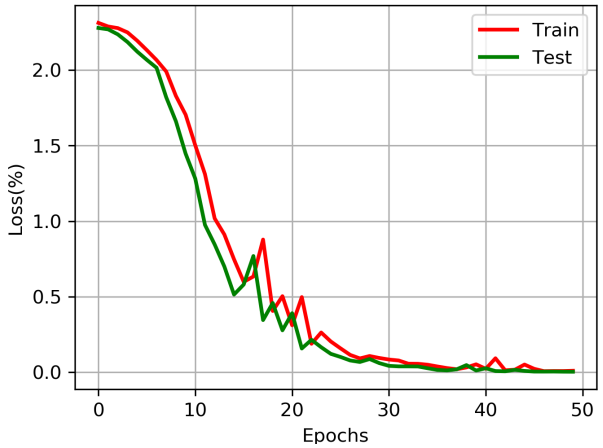
In the future, this work will study more effective technique to obtain fastest way for computation, and also interest to applied the CNN model to a real-time face recognition on video streaming.

References

[1] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, 2nd ed. Springer Publishing Company, Incorporated, 2011.



(a)



(b)

Figure 5: (a) Model accuracy and (b) model loss for JAFFE dataset

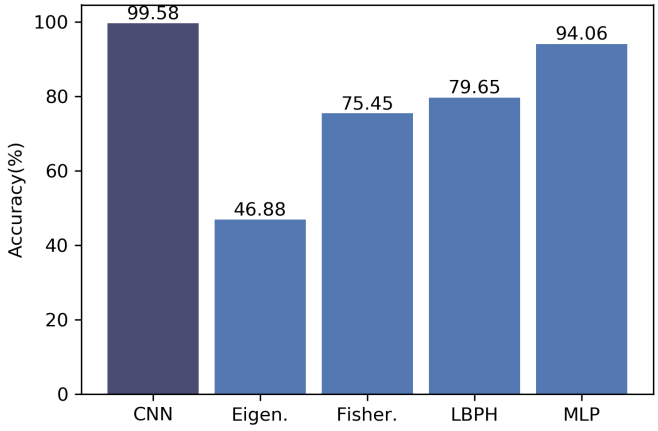


Figure 6: A comparison of accuracy by five face recognition techniques

[2] H. Ling, S. Soatto, N. Ramanathan, and D. W. Jacobs, "A study of face recognition as people age," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[3] K. Shi, S. Pang, and F. Yu, "A real-time face detection and recognition system," in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, April 2012, pp. 3074–3077.

[4] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.

[5] G. Givens, J. R. Beveridge, B. A. Draper, P. Grother, and P. J. Phillips, "How features of the human face affect recognition: a statistical comparison of three face recognition algorithms," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, June 2004, pp. II–381–II–388 Vol.2.

[6] P. Kocjan and K. Saeed, *Face Recognition in Unconstrained Environment*. New York, NY: Springer New York, 2012, pp. 21–42.

[7] M. Phankokkruad, P. Jaturawat, and P. Pongmanawut, "A real-time face recognition for class participation enrollment system over webrtc," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 10033, 2016.

[8] P. Jaturawat and M. Phankokkruad, "An evaluation of face recognition algorithms and accuracy based on video in unconstrained factors," in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov 2016, pp. 240–244.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[10] L. Hertel, E. Barth, T. Käster, and T. Martinetz, "Deep convolutional neural networks as generic feature extractors," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–4.

[11] A. Vinay, D. A. Mundroy, G. Kathiresan, U. Sridhar, K. N. B. Murthy, and S. Natarajan, "Dominant feature based convolutional neural network for faces in videos," in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, March 2017, pp. 17–22.

[12] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Jan 1991.

[13] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul 1997.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1026–1034.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[18] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *ICML 2004: Proceedings of the twenty-first International Conference on Machine Learning*, 2004, pp. 919–926.

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. 511–518.

[20] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Apr 1998, pp. 200–205.