2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)

Human Face Recognition Using Combination of ZCA Feature Extraction method and Deep Neural Network

Naing Min Tun
Bauman Moscow State Technical University
Moscow, Russia
naingminhtun52@gmail.com

Alexander I. Gavrilov
Bauman Moscow State Technical University
Moscow, Russia
alexgavrilov@mail.ru

Pyae Phyo Paing
Bauman Moscow State Technical University
Moscow, Russia
ppaing12@gmail.com

***Abstract*—Nowadays, there are deep learning-based face recognition systems with reliable accuracy rates. But these systems need high computational powers because they should do feature extraction. This research presents the consequences of recognition rates on the AT&T face database, using the hybrid methods of ZCA feature extraction and ANN in order to reduce computational power usage.**

Keywords—face recognition, ZCA (zero-phase component analysis), ANN (artificial neural network), feature extraction

I. Introduction

Face recognition is the process of identifying people by matching query face image with the template face image in the facial database due to the calculation of their pixel data. Face recognition is widely used in the biometric field because of its benefits such as low cost and non-intrusive nature. Face recognition has been started in psychology and engineering works of literature in the 1950s and 1960s. After the seminal work of Kanade, automatic face recognition systems have been started since the 1970s [1]. Thereafter, a lot of researchers have been explored many approaches to handle this task. But it still needs to reach human performance. Nowadays, there are a lot of methods of face recognition systems with reliable accuracy rates are used in various purposes, such as machine learning, computer vision and pattern recognition. There are many face recognition-based applications which exist in our sophisticated environment. The most recently used face recognition-based applications can be found in the surveillance system, access control, credit-card verification, mug shot searching, criminal identification and security of computer systems [2].

Basically, the face recognition process consists of four main stages: face detection, feature extraction, feature selection and classification. There are various facial recognition strategies, such as verification or identification. In verification, the target face is compared to a set of faces of a similar target. In identification, the target person is compared to each person in the database. Three mandatory criteria to choose the most viable of facial recognition technology. First, an image representation scheme for extracting useful data. Second, issues related to the difference in position or orientation of the person and, third, the extracted data are calculated in statistical form and an appropriate analysis is carried out [3].

This paper is focused on the reduction of computational power consumption using the combination of ZCA feature extraction and ANN. The rest of this paper is organized as follows. In Section II literature review is provided. Section III describes the methodology of the research. Next, the experimental results and conclusions are given in Sections IV and V, respectively.

II. Related Work

There is an extensive corpus of facial recognition and verification works. Consideration of this is beyond the scope of this article, so we will only briefly discuss the relevant works. The early learning-based face recognition approaches are Eigenfaces [4] and Fisherfaces [5] using statistical methods of data. Later, feature extraction methods, histogram-oriented gradients (HoG) [6] and local binary structures (LBP) [7], became popular for encoding facial images for facial recognition.

To make a face recognition system automatically, the first thing we need to do is face detection. The face in the images can be mixed with other persons or objects in a real situation. A very famous face detector has been developed by Viola and Jones in 2004 [8], which uses AdaBoost combined with Haar features as a machine learning method [9].

The most face recognition algorithms required full facial image, but in [10] partial face parts are used instead of full-face images as input data. These facial patches are processed by Gabor ternary patterns (GTP) and SIFT to create feature vectors. These vectors become the inputs of a face recognition algorithm using a sparse coding scheme. Very high dimensional LBP filter is utilized to extract the feature vectors and the Joint Bayesian method is applied as the classifier in [11].

In recent years, algorithms using multilayer neural network architectures named deep learning have become increasingly reliable for facial recognition. The development of deep neural network-based learning has contributed to massive success in recognition abilities. Especially many researchers paid a lot of attention to convolutional neural networks (CNN) [12] and Deep belief networks (DBN) [13]. In [14], multiple CNNs are trained for face identification using more than 10K subjects to perform general face representations which are to be applied in face verification. Taigman et al. [15] presented a deep

978-1-5386-8119-0/19/$31.00 ©2019 IEEE

convolutional neural network (CNN) system, which demonstrated remarkable performance using a dataset of 4 million sample faces. All works [16, 17, 18] use a complex multistage system combining the output of a deep convolutional network with PCA for dimension reduction and SVM for classification.

The most effective techniques are usually based on AI algorithms and utilize intelligent computations [19,20,21,22]. Moreover, artificial neural networks are an essential part of intelligent systems, for which face recognition plays an important role. Nowadays, The most advanced performances of face recognition were obtained using deep learning techniques. For example, FaceNet [23] achieved 95.12% accuracy rate on YouTube Faces database and 99.63% on LFW dataset respectively, in [24] provided a comprehensive survey of the recent developments on deep FR.

III. Research Methodology

The face recognition workflow is shown in figure 1. For the implementation of the face recognition system, facial images of people are required. In this paper, the AT&T face database is used for experiments.

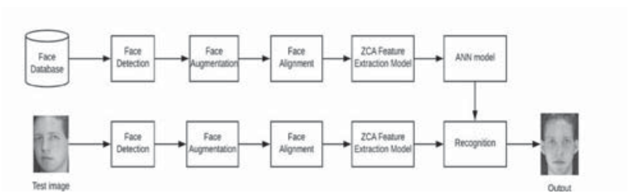

Fig. 1. Face recognition workflow

In order to improve the accuracy rate, before recognizing step we used preprocessing steps like face detection, face augmentation and face alignment. For face detection, a well-known haar cascade method is implemented. After face detection step, histogram equalization is applied to the images in order to enhance the contrast of images. To improve the accuracy, it is necessary to train the ANN model on a representative data set consisted of numerous data samples. But every data acquisition process is associated with a cost. This cost can be in terms of dollars, human effort, computational resources and time consuming. Therefore, we may need to augment existing data to increase the data size that we feed to our ANN classifier. In the Machine Learning context, data normalization process can promote better results. Thus, here face alignment can be seen as a form of data normalization to obtain a normalized rotation, translation, and scale representation of the face. Before we fed the preprocessed data to the ANN model, we extracted features from the individual images using ZCA method. Finally, we trained the extracted facial data on our ANN model in the vector forms.

A. *Feature Extraction*

Feature extraction is a dimensionality reduction process that is considered to transform the original source dataset into more manageable groups. Big data represent a large number of variables that require a lot of computing resources to process. Feature extraction is useful when we need to reduce the computing resources without losing important or relevant information in the following learning process. In this research, we used the ZCA whitening method as feature extraction.

1) *ZCA Whitening*

ZCA whitening is a linear transformation of data into a set of new variables whose covariance is the identity matrix. Hence ZCA whitening decorrelates features. ZCA whitening is calculated by the following:

Let the 2D image array to be represented by a matrix form:

$$X = \left[a_{ij} \right]_{M\times N}$$

Then, we calculate the covariance and its Eigen decomposition:

$$XX^T = LDL^T$$

Where, L is an orthogonal rotation matrix composed of eigenvectors. $D = diag\left(\lambda_1, \lambda_2, ..., \lambda_n\right)$ is the diagonal matrix of Eigenvalues.

$$LL^T = I$$

Where, I is identity matrix, ordinary whitening resorts to transforming the data into a space where the covariance matrix is diagonal:

$$\begin{aligned}\sqrt{D^{-1}}L^{-1}XX^T L^{-T}\sqrt{D^{-1}} &= \sqrt{D^{-1}}L^{-1}LDL^T L^{-T}\sqrt{D^{-1}} \\ &= I\end{aligned}$$

That means we can diagonalize the covariance by transforming the data according to

$$\tilde{X} = \sqrt{D^{-1}}L^{-T}X$$

This is an ordinary whitening with PCA. In ZCA a small epsilon is added to the Eigenvalues and transforms the data back.

$$\tilde{X} = L\sqrt{\left(D+\in\right)^{-1}}L^{-T}X$$

Figure 2 shows some examples of images before and after ZCA processing.

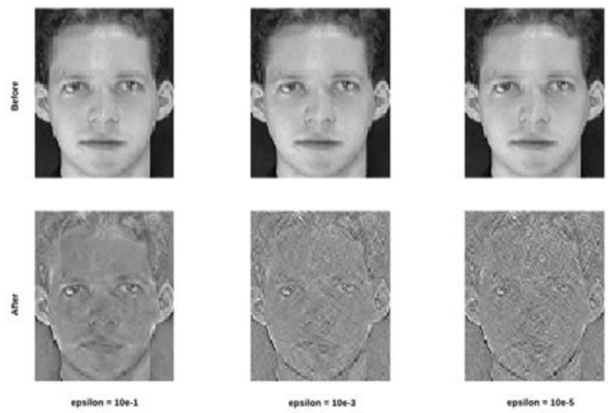

Fig. 2. Images before and after ZCA processing with different epsilon values

B. Deep Neural Network

Neural Network is an algorithm, imitated the human brain, designed to recognize the pattern in the form of numerical data sets. The neural network is a structure for many different machine learning algorithms for collaboration and processing complex input data. The goal of a neural network is to estimate the output result of any input data by manipulating some parameters and functions. There are many types of data sets forms in the real world like texts, audio, images, video, etc. This data must be converted to numeric vectors for use in ANN [25, 26, 27, 28]. A neural network (MLP) usually consists of three types (input, hidden, output) of layers, and the layers made up of artificial neurons called nodes. The nodes are assigned some weights to examine the input data. These weights will determine the influence of the input data. The output of the node is calculated by applying the activation function to the weighted sum of input data and the threshold biases. The general form of neural networks and the workflow of a node are represented in figure 3 and 4 respectively.

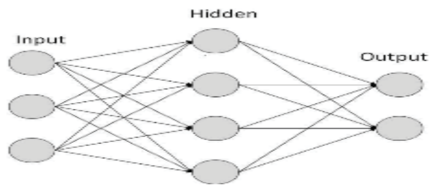

Fig. 3. Structure of neural network [29].

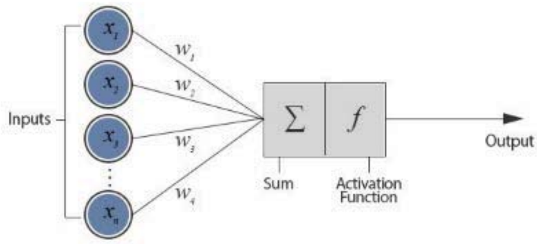

Fig. 4. The workflow of an artificial neuron (node) [30].

The function of an artificial neuron can be described mathematically as follows:

$$y = f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

where, y – the output of a neuron, b – the bias, x – the input to the neuron, w – the weights, n – the number of inputs from the incoming layer, i – a counter from 0 to n.

Finally, the errors of each output node in the output layer are getting as follows:

$$E\left(y_i, \ldots, y_N\right)$$

where, E – the error of each output node. So, we need to minimize errors by differentiating it.

There is a way to minimize the error E rate called "loss or cost function". That way is called a gradient descent method. Gradient descent method tries to minimize the error rate by updating the parameters (weights) of the neural network in the direction of the backward pass as follows:

$$\nabla E\left[\vec{w}\right] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_N}\right],$$

$$w_{new} \leftarrow w - \alpha \frac{\partial E}{\partial w},$$

$$b_{new} \leftarrow b - \alpha \frac{\partial E}{\partial b}$$

There are many gradients-based optimization algorithms, such as Adagrad, RMSprog, Adadelta, Momentum, Adam and etc. [31]. We use Adam algorithm as an optimizer in our work to optimize the neural network.

The utilizing neural network design and its classification results are described in the next section.

IV. Experimental Results

In this section, we have conducted three different epsilon's values of the ZCA feature extraction method. For training and testing datasets, we used AT&T database which consists of 40 samples, each represented by 10 images with different orientations. Before we fed the data to the neural network, we have performed the augmentation function on these data as shown in section III (8 times for each image) and converted it into the vector form. The database is split into two parts: 80% for training dataset and 20% for testing dataset respectively. The idea is implemented in Python 3.6.5 (CPU core i5 64 bits) system. For implementing the neural network framework, we have used Keras (open source neural network library). The structure of the ANN model is shown in figure 5.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 512)	11520512
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 40)	20520

Total params: 12,066,344
Trainable params: 12,066,344
Non-trainable params: 0

Fig. 5. The architecture of ANN.

We have used relu activation function in the hidden layers and softmax activation function for the output layer in our ANN model [32]. The simulation results, including accuracy and loss of ANN model, based-on ZCA method's epsilon values, are shown in fig 6,7. Confusion matrices of experiments are displayed in fig 8, 9, 10 respectively, and precision, recall and the f1 score of the ANN model are presented in table 1:

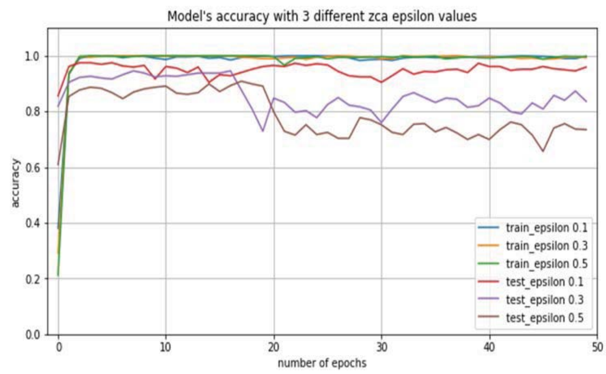

Fig. 6. Training and testing accuracies

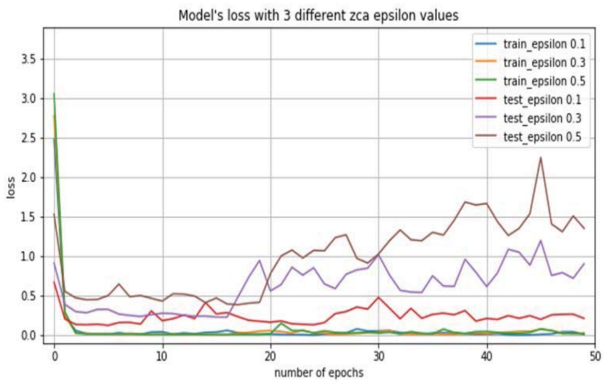

Fig. 7. Training and testing losses

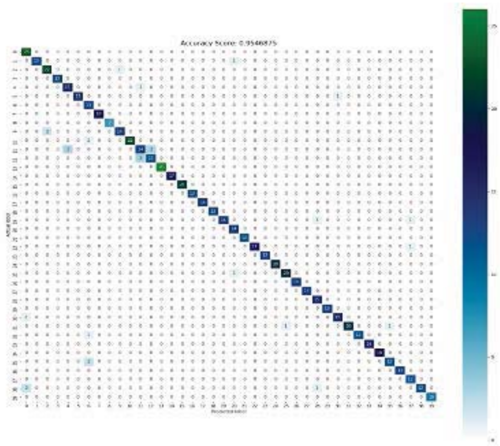

Fig. 8. Confusion matrix of experiment 1: ZCA's $\in = 10e^{-1}$.

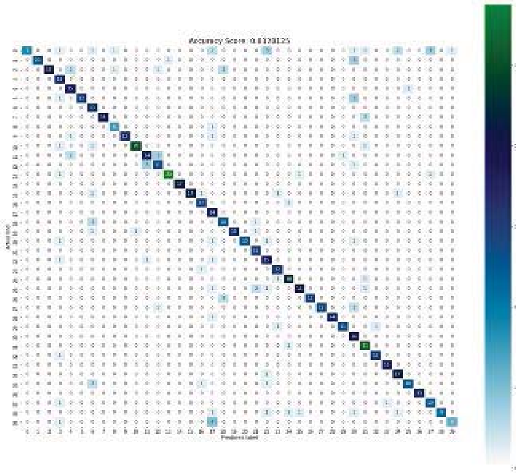

Fig. 9. Confusion matrix of experiment 2: ZCA's $\in = 10e^{-3}$.

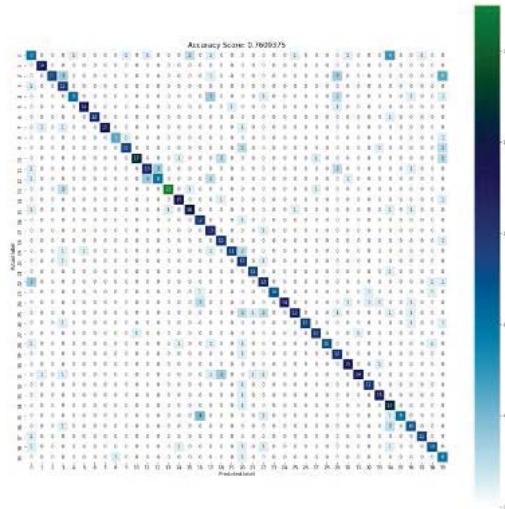

Fig. 10. Confusion matrix of experiment 3: ZCA's $\in = 10e^{-5}$.

TABLE I. PERFORMANCE EVALUATION

ZCA's values		Precision	Recall	F-1 score
$\in = 10e^{-1}$	micro avg	0.95	0.95	0.95
	macro avg	0.96	0.96	0.96
	weighted avg	0.96	0.95	0.95
$\in = 10e^{-3}$	micro avg	0.83	0.83	0.83
	macro avg	0.86	0.84	0.83
	weighted avg	0.87	0.83	0.83
$\in = 10e^{-5}$	micro avg	0.76	0.76	0.76
	macro avg	0.80	0.78	0.77
	weighted avg	0.81	0.76	0.77

V. Conclusion

Simulation of the proposed algorithms using publicly available AT&T face database with different ZCA feature extraction values demonstrates that increasing of epsilon's values leads to reduced recognition error and increase the accuracy.

Utilization of feedforward ANN architecture instead of convolutional neural networks (CNN) makes it possible to reduce the computational power usage when vectorization form is used as input data for ANN classifier. Therefore, the proposed approach can be used efficiently for the solution of different classification tasks when CPU power is limited for any reason.

Further work could be focused on practical applications of the proposed models and algorithms.

ACKNOWLEDGMENT

I would like to thank my parents and friends who raised me up during my difficult time. I would also like to express my special thanks to my supervisor Alexander Gavrilov who helped me in doing the research and to know about so many scientific thinking skills.

References

[1] Sujata G. Bhele and V. H. Mankar, "A Review Paper on Face Recognition Techniques," Int. Journal of Advanced Research in Computer Engineering & Technology, vol. 1, is. 8, pp. 339–346, 2012.

[2] Nilima B. Kachare and Vandana S. Inamdar, "Survey of Face Recognition Techniques," Int. Journal of Computer Applications, vol. 1, no. 19, pp. 30–34, 2010.

[3] Jherna Devi and Sajida Parveen, Nadeem Naeem and Nida Husan Abbas, "Face Recognition: Literature Review," Int. Journal of Advanced, vol. 5, is. 5, pp. 2258–2263, 2017.

[4] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," CVPR, 1991.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711–720, 1997.

[6] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using hog–ebgm," Pattern Recognition Letters, vol. 29, no. 10, pp. 1537–1543, 2008.

[7] T. Ahonen, A. Hadid, and M. Pietikainen, "Face recognition with local binary patterns," ECCV, 2004.

[8] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," CVPR, 2014.

[9] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. CoRR, abs/1412.1265, 2014.

[10] F. Schroff and D. Kalenichenko, "FaceNet: A Unified Embedding for Face Recognition and Clustering," 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 815–823, 2015.

[11] Mei Wang and Weihong Deng, "Deep Face Recognition: A Survey," School of Information and Communication Engineering, 2018.

[12] NEURONUS.com. [Online]. Avaliable: https://neuronus.com/stat/1271-nejronnye-seti-iskusstvennyj-intellekt.html.

[13] Technology Made Easy. [Online]. Avaliable: https://medium.com/technologymadeeasy/for-dummies-the-introduction-to-neural-networks-we-all-need-c50f6012d5eb

[14] X.-P. Yu, X.-R. Zhao, H. Sun, and A. Gavrilov, "Weld width prediction based on artificial neural network," Hanjie Xuebao, vol. 26, no. 5, 2005.

[15] E. A. Gladkov, A. V. Maloletkov, R. A. Perkovskii, and A. I. Gavrilov, "Evaluating the quality of argon-arc welded joints using neural network models with regression input," Welding International, is. 14, no. 7, pp. 559–564, 2000.

[16] A. Alfimtsev, S. Sakulin, and V. Devyatkov, "Web personalization based on fuzzy aggregation and recognition of user activity," Int. Journal of Web Portals, is. 4, no. 1, pp. 33–41, 2012.

[17] K. A. Neusypin, A. V. Proletarsky, K. Shen, and R. Liu, "Aircraft self-organization algorithm with redundant trend," Nanjing Li Gong Daxue Xuebao, vol. 38, no. 5, pp. 602–607, 2014.

[18] M. S. Selezneva and K. A. Neusypin, "Development of a measurement complex with intelligent component," Measurement Techniques, vol. 59, no. 9, pp. 916–922, 2016.

[19] B. I. Shakhtarin, K. Shen, and K. A. Neusypin, Modification of the nonlinear Kalman filter in a correction scheme of aircraft navigation systems.

[20] K. Shen, R. Liu, R. Guo, and K. A. Neusypin, "Technology of error compensation in navigation systems based on nonlinear Kalman filter," Guofang Keji Daxue Xuebao, vol. 39, no. 2, pp. 84–90, 2017.

[21] F. Ke and K. A. Neusipin, "Research progress of intelligent control systems of aircrafts," Binggong Xuebao, vol. 31, no. 7, pp. 939–949, 2010.

[22] Towards Data Science. [Online]. Avalaible: Avaliable:https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," IEEE Conf. on CVPR, 2014.

[24] Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover canonicalview faces in the wild with deep neural networks. CoRR, abs/1404.3543, 2014.

[25] Sebastian Ruder. [Online] Avaliable: http://ruder.io/optimizing-gradient-descent/index.html#adam