Convolutional_Neural_Networks_for_Realtime_Multi-Faces_Verification_with_Occlusion

by Muhammad Fachrurrozi

Convolutional Neural Networks for Realtime Multi-Faces Verification with Occlusion

Saparudin
Department of Informatics Engineering
Universitas Sriwijaya
Indralaya, Indonesia
saparudinmasyarif@gmail.com

Reza Firsandaya Malik*
Department of Computer Engineering
Universitas Sriwijaya
Indralaya, Indonesia
rezafm@unsri.ac.id

Erwin
Department of Computer Engineering
Universitas Sriwijaya
Indralaya, Indonesia
erwin@unsri.ac.id

M. Fachrurrozi
Department of Informatics Engineering
Universitas Sriwijaya
Indralaya, Indonesia
mfachrz@unsri.ac.id

Sukemi
Department of Computer Engineering
Universitas Sriwijaya
Indralaya, Indonesia
sukemiku66@gmail.com

Puji Rachmawati
Department of Informatics Engineering
Universitas Sriwijaya
Indralaya, Indonesia
mowikatta@gmail.com

Ari Susanto
Department of Informatics Engineering
Universitas Sriwijaya
Indralaya, Indonesia
risusanto@outlook.com

Bambang Suprihatin
Department of Mathematics
Universitas Sriwijaya
Indralaya, Indonesia
bambangs@unsri.ac.id

***Abstract*— The face is a major component of living creature that becomes a feature to distinguish between one and another living creature and also between living creature with inanimate objects such as statues. In the digital era nowadays, faces are used as objects for identification and verification. But the existence of occlusion in the form of glasses has a potential to influence the verification process carried out by the system. Therefore, a system will be build to perform a real time multi-faces verification processes with occlusion using Convolutional Neural Networks. We propose a "Siamese" architecture with 37 convolutional layer, 10 pooling layer, and 3 fully connected layer. On the training and testing we used Labeled Faces in the Wild (LFW) dataset. The image were taken from 5749 people. We also took 165 images from 11 people for testing with image size 96x96 for each images. The verification accuracy achieved for the proposed method is 98%.**

Keywords*— Face Verification, Occlusion, Convolutional Neural Network, Real-Time, Multi-Faces

I. Introduction

Computer vision is a field of science that has become a concern and has continued to develop over the past 20 years. Much research can be done in this field of science, one of which is face recognition. Face recognition is personal identification based on geometric or statistical features obtained from images of human faces.

Face recognition is usually categorized into two tasks, namely verification and identification. Face identification is a face recognition study by comparing face image of unknown subjects with databases of faces of known people; that is, one-to-many matching. Face verification, on the other hand, is to determine whether two face images belong to the same person. This is a one-to-one match where an identity claimed by someone is accepted or rejected. In general, face verification tasks are more difficult than face needed to make decisions that distinguish between subjects [1]

Facial occlusion, such as glasses is one critical factor that affects the performance of face recognition. Face recognition with occlusion algorithm based on deep learning has been proposed by many researchers which achieved a good performance [2]. A study of face recognition with occlusion in the form of glasses and scarves using the PCA and LDA methods described in a paper entitled "Face Recognition with Occlusion" resulted in an accuracy rate of 99.6% for glasses and 100% for scarves [3]. But in that study, it was not based on real-time.

In this paper, we proposed a "Siamese" architecture of Convolutional Neural Network (CNN) method for multi-faces to verify multiple face with occlusion in real-time which is that architecture can learn similarity metric from training data. This similarity metric can later be used to match new face images from faces of people not seen during training (i.e previously unseen categories). Siamese architecture of convolutional neural network (CNN) achieving a verification accuracy of 2.5 EER on unseen data with unknown categories [4].

II. Related Work

In the case of the face recognition process, the initial stage is needed before we can recognize faces, the process is the face detection process. However, in general, the face images obtained have varied face sizes and shapes, have a non-uniform background, and varying or uncertain lighting conditions so that face detection becomes more difficult to detect. So in that case the faces in the picture generally have a very varied background shape.

To determine a face in a certain image, we need to define some of the structure of a human face such as eyes, nose, and mouth that will be used for face detection. The proposed algorithm for face detection is Viola-Jones method. The Viola-Jones face detector contains three main ideas that can run in real time: the image integral, classifier learning with AdaBoost, and the attentional cascade structure [5].

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

The Viola-Jones method has a higher accuracy than other face detection methods (such as segmentation, Euclidean distance, and simple artificial neural networks), but the Viola-Jones method also has a weaknesses in the form of difficulties in determining faces on non-frontal faces (images not perpendicular towards the camera). The facial position determines the success of Viola-Jones method on detecting faces [6]. The figure below shows the flow of Viola-Jones method in detection faces.

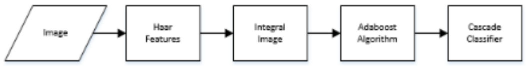


Fig. 1. The flow of Viola-Jones method [7].

The problem of face recognition under occlusions caused by scarves and sunglasses using PCA and Support Vector Machine Resulted in accuracy rate of 54.17% for sunglasses and 81.25% for scarf [8].

Face verification, is to determine if two face images belong to the same person. This is a one-to-one matching in which the identity claimed by a person is accepted or denied [1]. Many techniques have been proposed for face verification; some examples are system using Adapted Generative Models [9], Deep CNN [10], Hybrid Deep Learning [11], FaceNet [12], and Face Verification Based on Convolutional Neural Network and Deep Learning [13].

III. LIRETATURE REVIEW

A. Convolutional Layer

In Convolutional neural networks, input is a multi-dimensional array data and weights (w) in neural networks will then be called the kernel, the kernel is a multidimensional array which is a parameter that will be implemented using a learning algorithm such as backpropagation. Both multidimensional arrays, namely input and kernel are generally referred to as tensors. The following equation describes the convolution operation with I as input and K is a two-dimensional kernel:

$$(I * K)_{(i,j)} = \sum_{m=0}^{k1-1} \sum_{n=0}^{k2-1} I(i-m, j-n) K(m,n) \quad (1)$$

The above equation also applies commutatively so that it can be written as follows:

$$(I * K)_{(i,j)} = \sum_{m=0}^{k1-1} \sum_{n=0}^{k2-1} I(m,n) K(i-m, j-n) \quad (2)$$

The commutative form is obtained because in the convolution operation the K kernel will be flipped based on input I, where if m continues to increase the index in the input will increase, but the index in the kernel continues to decrease. Implementation of convolution operations when building program code, or technically often done by convolution without flipping the kernel. The operation is called the cross-correlation, as can be seen in the following equation:

$$(I * K)_{(i,j)} = \sum_{m=0}^{k1-1} \sum_{n=0}^{k2-1} I(i+m, j+n) K(m,n) \quad (3)$$

B. Pooling Layer

The pooling layer consist of filters with specific sizes and steps that will shift to the entire map feature area. Pooling that is commonly used is Max Pooling and Average Pooling. For example in Max Pooling 2x2 with step 2, then for each filter change, the maximum value in the area of 2x2 pixels will be selected, while Average Pooling will choose the average value.

The pooling layer will reduce the spatial size and number of parameters in the network and accelerate overfitting calculations and controls. Pooling layers work with spatial blocks that move along the size of a feature pattern. The size of the block shift is generally the size of the block dimension (HxH) itself so there is no overlap as in the convolutional layer. This layer has no parameters because the parameters are predetermined (fixed). Pooling layers have several types, including average pooling, max pooling, and Lp Pooling.

Pooling function replaces the output value in several locations with a summary statistic of the nearby outputs. Pooling is needed to overcome if there are differences in translation in the input. By overcoming the difference in translation at the input, it means that if you change the value slightly in the input, the value that is successfully obtained from the pooling process does not change.

C. Fully Connected Layer

Fully connected layer calculates the class scores, which will produce volume. As with ordinary neural networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume. In this study, the fully connected layer will be a layer to classify objects such as faces. This layer will verify the face on the input.

D. Backpropagation

The backpropagation algorithm was first introduced in 1970, but it has not attracted much attention until there was popular research in 1986 by David Rumelhart, Geoffrey Hinton and Ronald Wiliams. The study suggests that backpropagation is much faster that other learning algorithms of the time. The steps in the backpropagation algorithm are as follows:

1. FeedForward

Feedforward on artificial neural networks has been explained in sub-section 2.4 where the feedforward process calculates the output of artificial neural networks with the following equation.

$$a^l = \sigma(w^l a^{l-1} + b) \quad (4)$$

2. Calculating the value of the error in the last layer

The error in the last layer is denoted as δ^L which is obtained through the following equation.

$$\delta^L = \nabla_a C \odot \sigma'(a^L) \quad (5)$$

3. Calculating the error value for each layer

After getting the error value from the last layer, the error value is also calculated for each layer on the network, this process is called backwardpass. The error value for each layer is denoted as δ^l. To calculate the value of δ^l, use the following equation.

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(a^l) \tag{6}$$

5. Change weight and bias value

To change the weight value w and bias b use the equation.

$$w^l = w^l - \frac{\eta}{m}\frac{\partial c}{\partial w} \tag{8}$$

$$b^l = b^l - \frac{\eta}{m}\frac{\partial c}{\partial b} \tag{9}$$

IV. Methodology

A. Dataset

In this research, we use 2 types of data are the primary data and the secondary data types. The primary data types we took from Computer Science students at the Faculty of Computer Science, Sriwijaya University. The secondary data types from Labeled Faces in the Wild (LFW) databases. LFW has 13233 images that were taken by 5749 people, and for the primary data, we took our own data from 11 people with 15 images of each. The sample of the LFW database is shown in figure 1 and the sample of primary dataset is shown in figure 2.

Fig. 2. *Sample of Secondary Data Types*

4. Calculate Partial Derivative for function cost for weight

The main process in backpropagation to calculate the effect of weight change w on changes in cost value is as follows.

$$\frac{\partial c}{\partial w_{jk}} = a_k^{l-1}\delta_j^l \tag{7}$$

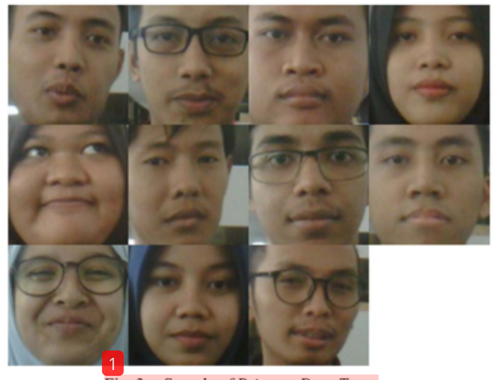


Fig. 3. *Sample of Primary Data Types*

B. Proposed Method

Convolutional Neural Networks, or also known as convolutional networks, are one type of neural networks. Convolutional neural networks indicate that the network implements mathematical operations, namely convolution. Convolution is a form of linear operation. Convolutional neural network is a neural network that uses convolution to replace at least one matrix multiplication in a layer arrangement [14].

In convolutional neural network, input is a multi-dimensional array data and weights (w) in neural network named kernel, kernel is a multi-dimensional array which is a parameter that will be implemented using a learning algorithm such as backpropagation. Both multidimensional arrays, namely input and kernel are generally referred to as tensors. We use facenet architecture in this research.

C. Performance Measurement

Confusion matrix is a method that summarizes the classification performance of several test data. This method consist of a two-dimensional matrix where the first index for the actual class of and object and the other for the class of classification. Example of this method, if there is only two classes in the dataset, positive and negative, then there are several terms commonly used as measurement notations.

- True Positive (T+) : data with positive classes that are positively classified
- True Negative (T-) : data with negative classes that are classified as negative
- False Positive (F+) : data with a negative class that is positively classified

- False Negative (F-) : data with a positive class that is classified as negative

The output of confusion matrix is accuracy, precision, recall, and f-measure with the following equation:

- Accuracy : how accurate the classification method is in determining class of faces.

$$Accuracy = \frac{(T+)+(T-)}{Total\ Citra} \quad (10)$$

- Precision : the fraction value of each class of faces that is correctly classified.

$$Precision = \frac{(T+)}{(T+)+(F+)} \quad (11)$$

- Recall : the fraction of the number of correct face classes obtained is divided by the number of face classes that should be obtained.

$$Recall = \frac{(T+)}{(T+)+(F-)} \quad (12)$$

- F-Measure : mean harmonic functions of precision and recall.

$$F-Measure = 2\frac{Precision.Recall}{Precision+Recall} \quad (13)$$

V. Resullt and Discussion

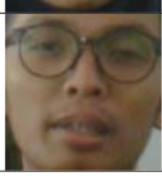
We train and test our experiments based on Convolutional Neural Network (CNN) using LFW face database, which contains 13233 images from 5749 classes from http://vis-www.cs.umass.edu/lfw/. We also took 165 images from 11 people in our faculty to test the proposed method. The results from the proposed method using LFW database is shown in Table. 1 and our own datasets is shown in Table. 2.

TABLE I. Results from the Proposed Method Using LFW Database

Dataset	Accuracy
LFW (Labeled Faces in the Wild)	98%

TABLE II. Results from the Proposed Method Using Own Datasets

Images	Name	F-Measure	Accuracy
	adi	1	100%
	ajrul	0.875	97.59%
	ari	0.933333	98.79%
	dinda	1	100%

	dwita	0.842105	96.38%
	farhan	1	100%
	ilham	1	100%
	imam	1	100%
	puji	0.933333	98.79%
	rizka	1	100%
	rusdi	0.769231	96.38%

For the proposed method we achieved 98% for the level of accuracy using LFW databases and FaceNet architecture.

VI. Conclusion

In this paper, we propose a multi-faces verification with occlusion using Convolutional Neural Network (CNN) based on realtime. Our proposed method achieved high performance (98%) on LFW dataset. In the real-life, our method can be use in various fields such as campus attendance systems, system security, and others.

References

[1] M. Khalil-Hani and L. S. Sung, "A convolutional neural network approach for face verification," *High Perform. Comput. Simul. (HPCS), 2014 Int. Conf.*, no. 3, pp. 707–714, 2014.

[2] S. Saypadith and S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," no. November, pp. 1318–1324, 2018.

[3] Y. Su, Y. Yang, Z. Guo, and W. Yang, "Face recognition with occlusion," *Proc. - 3rd IAPR Asian Conf. Pattern Recognition, ACPR 2015*, no. November 2015, pp. 670–674, 2016.

[4] S. Chopra and R. Hadsell, "Learning a Similarity

Metric Discriminatively , with Application to Face Verification," 2005.

[5] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[6] M. D. Putro, T. B. Adji, and B. Winduratna, "Sistem Deteksi Wajah dengan Menggunakan Metode Viola-Jones," *Tugas Akhir*, pp. 1–5, 2012.

[7] A. Suharso, U. S. Karawang, T. Informatika, and T. Timur, "Pengenalan Wajah Menggunakan Metode Viola-Jones dan Eigenface Dengan Variasi Posisi Wajah Berbasis Webcam," vol. 1, pp. 19–30, 2016.

[8] R. Min and S. Antipolis, "Improving the Recognition of Faces Occluded by Facial Accessories," 2011.

[9] F. Cardinaux, C. Sanderson, and S. Bengio, "Face verification using adapted generative models," *Proc. - Sixth IEEE Int. Conf. Autom. Face Gesture Recognit.*, pp. 825–830, 2004.

[10] X. L. B, Y. Wang, W. Zhang, S. Ding, and W. Jiang, "Deep CNNs for Face Verification Deep CNNs for Face Verification," no. October, 2018.

[11] Y. Sun, X. Wang, and X. Tang, "Hybrid Deep Learning for Face Verification," vol. 8828, no. c, pp. 1–14, 2015.

[12] F. Schroff and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," 2015.

[13] A. Lebedev, V. Khryashchev, A. Priorov, and O. Stepanova, "Face Verification Based on Convolutional Neural Network and Deep Learning," 2017.

[14] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," *MIT Press*, p. 1, 2016.

Convolutional_Neural_Networks_for_Realtime_Multi-Faces_Verification_with_Occlusion

ORIGINALITY REPORT

98%	12%	98%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Saparudin Saparudin, Reza Firsandaya Malik, Erwin Erwin, M. Fachrurrozi et al. "Convolutional Neural Networks for Realtime Multi-Faces Verification with Occlusion", 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), 2019 Publication	93%
2	Submitted to School of Business and Management ITB Student Paper	3%
3	Submitted to Liverpool John Moores University Student Paper	2%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		