

Face recognition based on Siamese convolutional neural network using Kivy framework

Yazid Aufar¹, Imas Sukaesih Sitanggang²

¹Informatics Engineering, Politeknik Hasnur, Barito Kuala, Indonesia

²Department of Computer Science, Bogor Agricultural University, Bogor, Indonesia

Article Info

Article history:

Received Dec 14, 2021

Revised Mar 11, 2022

Accepted Mar 17, 2022

Keywords:

Convolutional neural networks

Face recognition

Kivy framework

One-shot learning

Siamese networks

ABSTRACT

Human face recognition is a vital biometric sign that has remained owing to its many levels of applications in society. This study is complex for free faces globally because human faces may vary significantly due to lighting, emotion, and facial stance. This study developed a mobile application for face recognition and implemented one of the convolutional neural network (CNN) architectures, namely the Siamese CNN for face recognition. Siamese CNN can learn the similarity between two object representations. Siamese CNN is one of the most common techniques for one-shot learning tasks. Our participation in this study determined the efficiency of the Siamese CNN architecture with the enormous quantity of face data employed. The findings demonstrated that the suggested strategy is both practical and accurate. The method with augmentation produces the best results with a total data set of 9,000 face images, a buffer size of 10,000, and epochs of 5, producing the minimum loss of 0.002, recall of 0.996, the precision of 0.999, and F1-score of 0.672. The proposed method gets the best accuracy of 98% with test data. The Siamese CNN model is successfully implemented in Python, and a user interface and executables are built using the Kivy framework.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Yazid Aufar

Informatics Engineering, Politeknik Hasnur

Barito Kuala, Indonesia

Email: yazid.aufar.ya@gmail.com

1. INTRODUCTION

Face and facial expression recognition have gotten a lot of interest from academic studies throughout the globe in the last five years [1]–[7]. Even now, face recognition research is being conducted on several new difficulties, and new approaches for various applications are being created [8]–[15]. The human face is regarded as the essential feature of the body. According to studies, even a face can communicate and has various phrases for different emotions. Face recognition systems, which are based on extraction of features and dimension reduction, are often employed to validate human identity. Face recognition systems have been created in various ways, with varying degrees of success. Face recognition remains a difficult challenge in real-world applications, despite several face recognition algorithms operating effectively in diverse situations. Currently, no technique provides a reliable solution to the various conditions and applications faced by face recognition. The face recognition challenge is divided into two groups. The first is a one-to-one matching challenge known as the face verification job. Face verification is used, for example, when people unlock their phone with their face. Passengers should pass through a system that scans their passport and face to ensure proper in certain airports. The second assignment is a facial recognition task, which requires humans to figure out who this individual is. It is an issue of one-to-many matching.

The performance of several complex tasks, such as face verification and face detection, has dramatically improved since convolutional neural network (CNN) based algorithms have been utilized. One-shot learning is another method for completing the goals listed above and a technique for learning representations from a model. This study aims to have the best performance Siamese CNN in face recognition and develop this model in the mobile application. This study creates encodings of the given input image in Siamese CNN. Then, it takes an image as an input from a different individual and calculates its encoding with the same network without changing any network parameters. Following these calculations, we may compare the two photos to see whether they are comparable. Face recognition, signature verification, and object tracking have been effectively accomplished using Siamese CNN in computer vision [16]–[22]. This study develops a building application of facial recognition systems using a Siamese CNN and the Kivy framework.

2. METHOD

2.1. Image datasets

The images in this study were saved in the .jpg file format. Each input batch of data contains three images: the anchor, positive, and negative images used to train a face recognition model utilizing Siamese CNN. Our existing face, identity A, is the anchor. The second image, which also includes the face of person A, is our positive image. On the other hand, the negative image does not share the same identity as the positive image and may be associated with person B, C, or even Y. The argument is that the anchor and positive image are from the same person/face. However, the negative image is not the same as the face image in the anchor and the positive image. The anchor photos and positive images were captured using a 250×250 resolution camera, yielding 550 and 400 images. Face image samples as shown Figure 1 for anchor images and positive images are shown in Figures 1(a) and (b). We utilized the labeled faces in the wild (LFW) face database [23] in link <http://vis-www.cs.umass.edu/lfw/> for the negative image. The database contains 13233 images of people’s faces, each labeled with their names. Five thousand seven hundred forty-nine persons in the LFW database have two or more different images. An example of the labeled faces in the wild (LFW) images dataset is shown in Figure 2.

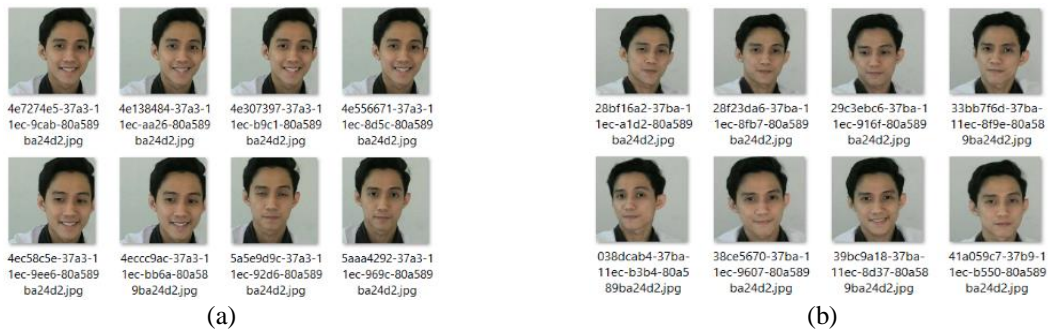


Figure 1. Face images sample for (a) anchor images and (b) positive images



Figure 2. Example of LFW images dataset for negative images

2.2. Augmentation techniques

We attempted five different augmentation techniques: random brightness, random contrast, random left-right flip, random jpeg quality, and random saturation. Figure 3 depicts their augmentations. This augmentation aims to replicate data to make the categorization process easier. After using the augmentation approach, the anchor and positive images' data increase to 5,590 and 4,000, respectively.



Figure 3. Example image used augmentation techniques

2.3. Siamese convolutional neural network

The typical model is Siamese CNN with L layers each with N_l units, where $h_{1,l}$ denotes the hidden vector in layer l for the first twin, and $h_{2,l}$ means the same for the second twin. In the first $L - 2$ layers, we only employ rectified linear (ReLU) units, whereas the subsequent layers use sigmoidal units. The model comprises a series of convolutional layers, each of which employs a single channel with different-sized filters and a fixed stride of one. The number of convolutional filters is given as a multiple of 16 to improve performance. The resultant feature maps are subjected to a ReLU activation function, which is optionally followed by max-pooling with filter size and stride of 2. As a result, each layer's k th filter map looks like (1) and (2) [16].

$$a_{1,m}^{(k)} = \max - \text{pool}(\max(0, W_{l-1,l}^{(k)} * h_{1,(l-1)} + b_l), 2) \tag{1}$$

$$a_{2,m}^{(k)} = \max - \text{pool}(\max(0, W_{l-1,l}^{(k)} * h_{2,(l-1)} + b_l), 2) \tag{2}$$

We chose $*$ as the legitimate convolutional operation corresponding to returning just those output units resulting from complete overlap among each convolutional filter and the input feature maps, and $W_{l-1,l}$ is the 3-dimensional tensor encoding the feature maps for layer l [16]. In the final convolutional layer, all elements are flattened into a single vector. This convolutional layer is represented by a fully connected layer, and then another layer that computes the inspired distance metric between each Siamese twin and outputs it to a single sigmoidal output unit. The prediction vector is defined as $p = \sigma(\sum_j \alpha_j |h_{1,L-1}^{(j)} - h_{2,L-1}^{(j)}|)$, where σ denotes the sigmoidal activation function. This last layer assesses the similarity between the two feature vectors by inducing a metric on the learned feature space of the $(L-1)$ th hidden layer. The α_j is the extra parameter that the model learns during training and use to weigh the relevance of component-wise distance. This describes the network's last L th fully-connected layer, connecting the two Siamese twins. We provide one example in Figure 4, which is the most effective form of our explored model. The Siamese twin is not shown in Figure 4, but it joins just after the 4096-unit fully-connected layer, which computes the L1 component-wise distance between vectors. This network also performed the best on the verification job of any network [16].

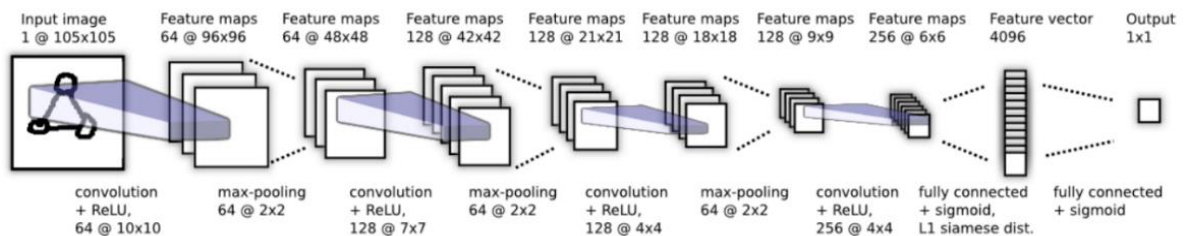


Figure 4. The convolutional architecture for the verification was selected

Training for Siamese CNN is done in mini-batch sizes. In order to create an effective network train, the system selects image pairings at random for training while avoiding an imbalanced amount of similar and

dissimilar image pairs in a mini-batch production. The anchor in distinct classes is random as the mini-batch size of images, while the paired images are controlled as half the same class and half the different class. The weights are then updated throughout each mini-batch of training iterations using an adaptive moment estimate optimizer (Adam) [24]. With an initial learning rate of 0.0001, the Adam optimizer is employed for model training and optimization. The encodings of input images are computed in Siamese CNN, and then the results perform the same thing with the same network, calculating the encoding image of a different individual. We may compare two encodings after doing computations to see whether they are comparable. Images' encodings serve as representations of their latent features. The encoding comparison reveals that the photos belong to the same individual. In the network's training, an anchor image was used and compared to its examples of positive and negative images. The gap between anchor-positive and anchor-negative must be modest, but the gap between anchor-negative and anchor-positive must be substantial.

$$L = \max(d(a,p)d(a,n) + \text{margin}, 0) \quad (3)$$

The following (3) is known as the triplet loss function, and it may be used to compute gradients. Where “*a*” represents an anchor image, “*n*” represents a negative image, and “*p*” represents a positive image. Another variable is known as margin. The margin indicates the significance of the gap of similarity. For example, if we pick margin=0.3 and $d(a,p)$, then $d(a,n)$ must be greater than 0.8. It helps in locating the supplied photos. The triplet loss function is used to compute gradients, which are then used to update the parameters of the Siamese CNN.

2.4. Performance evaluation

We assess our model using four metrics: accuracy, recall, precision, and F1-score. Accuracy in (4) is the fraction of forecasts that exactly match the actual data. Precision in (5), also known as positive predictive value (PPV), is the percentage of the main face image successfully validated out of an overall optimistic prediction. Recall or sensitivity in (6), often known as true positive rate (TPR) in facial recognition applications, refers to the percentage of verified main face photos correctly classified as positive. Specificity, also known as the true negative rate (TNR), is the fraction of all facial characteristics not in the primary face image classified as negative. To illustrate, the percentage of the direct face image correctly identified is not another person's face. The F1-score in (7) indicates the harmonic mean derived by taking the weighted average of precision and recall [25].

$$\text{Accuracy} = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (4)$$

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (5)$$

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (6)$$

$$\text{F1score} = \frac{2(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \quad (7)$$

2.5. Kivy framework

Kivy is a cross-platform Python toolkit that may quickly construct apps with novel interfaces. Kivy is a sophisticated Python-based framework for developing mobile apps featuring natural user interfaces (NUI) [26]. Kivy has the following features: support for numerous inputs such as tangible user interface objects (TUIO), multi-touch, mouse, and keyboard; robust APIs for most smartphones; a single application for several operating systems; and compatibility for networking protocols and remote login. Many widgets and multi-touch assistance Kivy is used to customize widgets [27]. Face recognition using Siamese CNN is suggested in this study and implemented in Python. The Kivy used to create the user interface in this paper is Kivy 2.0.0.

2.6. The novelty of the proposed method

The main novelty is the idea of using the Siamese CNN to study facial recognition and its application on mobile using the Kivy framework. Due to the considerable interclass similarities and intraclass variances. To address this, we propose in this research employing a Siamese CNN to provide the computer with the capacity of similarity learning and, as a result, lower the interclass similarity and intraclass variation of the non-linear representation of pictures of each face. The computer may use this Siamese design to reduce interclass similarity and intraclass variances.

3. RESULTS AND DISCUSSION

We used the following method to test the effectiveness of our proposed Siamese CNN model in face recognition instances. We began by assessing our model for the different approaches, quantity of data gathered with and without augmentation, buffer size, and epoch count. We used up to 900 face photos in Method I to train the Siamese CNN without augmentation. Methods II and III collected 9,000 data face images with augmentation. We explored how the sample count influences the training task. We tested two synthetic samples: face images without enhanced datasets and face images with augmented datasets. Due to Methods II and III containing a significant number of datasets, we decided that the buffer size should be 10,000 bigger than the Method I's, which is 1,024. We can also examine the influence of the epoch in each technique. Methods I, II, and III used 50, 5, and 32 epochs, respectively. The outcomes of our approaches are shown in Table 1.

Table 1. Performance evaluation of our method without augmentation and with augmentation

	Anchor	Positive	Negative	Buffer size	Epoch	Loss	Recall	Precision	F1-Score
Method I (without augmentation)	300	300	300	1,024	50	1.788	1	1	0.997
Method II (with augmentation)	3,000	3,000	3,000	10,000	5	0.002	0.996	0.999	0.672
Method III (with augmentation)	3,000	3,000	3,000	10,000	32	0.694	1	0.506	0.997

The findings show that our network can perform relatively well on augmented datasets. The loss function Method II outperforms the others by yielding the most negligible value of 0.002, indicating that the loss function adequately expresses the amount of misclassification. Method I has a more significant loss function than the other methods while having a high recall, precession, and F1-score. This suggests that Method I have a high misclassification rate. Figure 5 depicts the loss, recall, and precision in each epoch of procedure I. Method II has the best loss function, 0.002, while having a smaller loss, recall, and precision than Method I. Figure 6 depicts the loss, recall, and precision in each epoch of procedure II. Figure 7 shows that recall and precision are equal, but the graph dips again at the fifth epoch. The sole difference between Method III and Method II is the number of epochs. Although having a higher Recall and F1 score than Method II, Method III has a lower Loss and Precision. Figure 7 depicts the drop in the graph.

After obtaining the Siamese CNN model using Methods I, II, and III, we used the Kivy framework to incorporate the model into a mobile application. For verification, we utilized 50 photos from the positive image collection. This verification image is conducted to compare to the input image. The outcomes of detection, validation and verification results are used to test Method I, Method II, and Method III models for face detection application. Each approach is compared using a different detection threshold and the same verification threshold. Method A uses a detection threshold of 0.1 and a verification threshold of 0.8. Method B uses a detection threshold of 0.5 and a verification threshold of 0.8. A detection threshold is a statistic that determines if a prediction is positive. The fraction of positive predictions divided by the total number of positive samples is the verification threshold. Table 2 shows the results of the comparison of each approach for implementing face recognition using the Kivy framework. Method II (A) got the best accuracy of 98%. Figure 8 shows the results of the confusion matrix model for the Siamese CNN architecture with data testing. From the 50 samples data tested, 9 sample data were misclassified in Figure 8(a), 8 sample data were misclassified in Figure 8(b), 1 sample data were misclassified in Figure 8(c), and 25 sample data were misclassified in Figures 8(d), (e), and (f). We can see that Figure 8(b) has one False Image identification error, the least of the others, and no false image identification error.

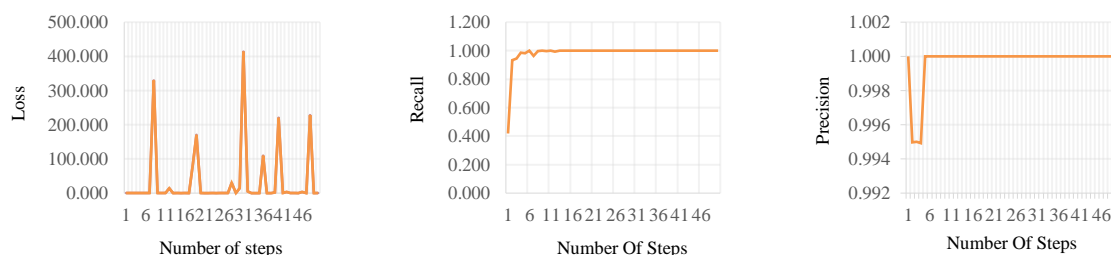


Figure 5. Loss, recall, and precision Method I

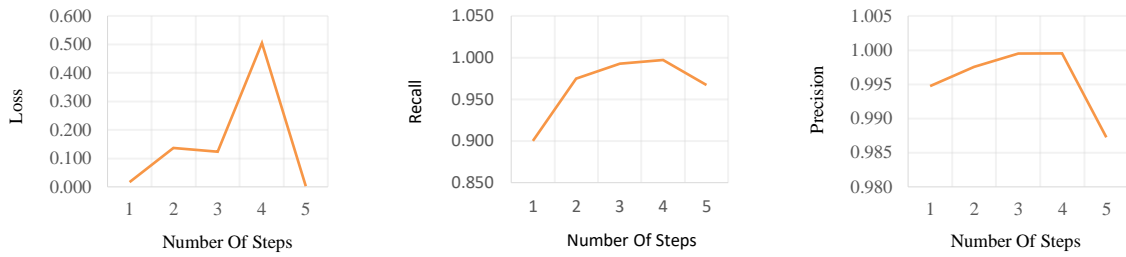


Figure 6. Loss, recall, and precision Method II

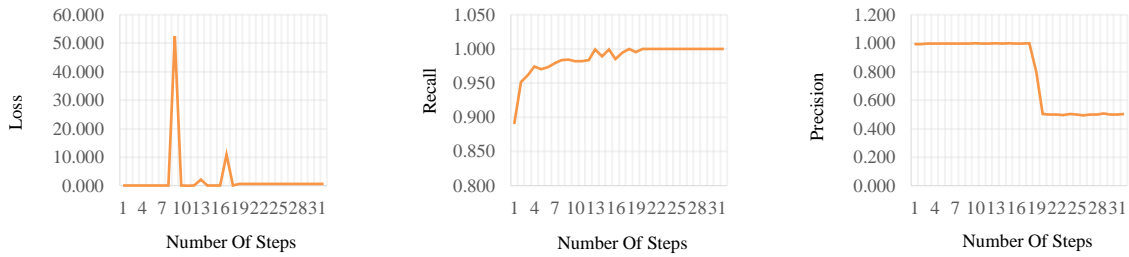


Figure 7. Loss, recall, and precision Method III

Table 2. Implementation face recognition using Kivy framework

Method Used	Accuracy
Method I (A)	82%
Method I (B)	84%
Method II (A)	98%
Method II (B)	50%
Method III (A)	50%
Method III (B)	50%

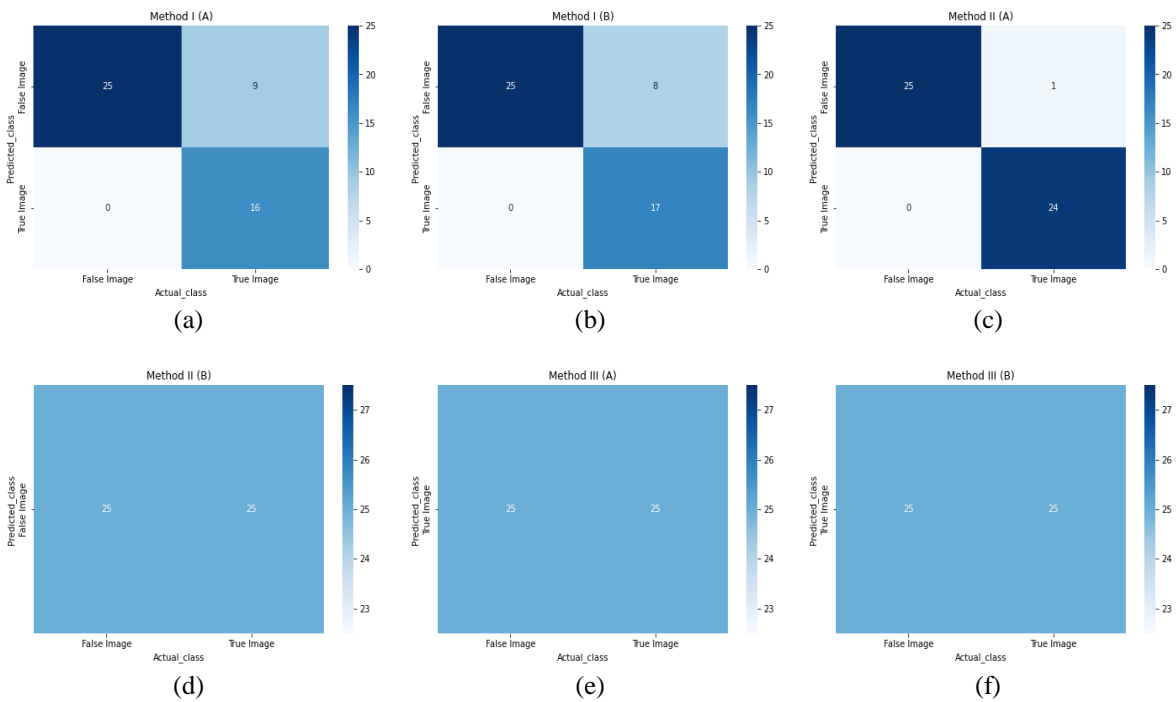


Figure 8. Confusion matrix of (a) Method I (A), (b) Method I (B), (c) Method II (A), (d) Method II (B), (e) Method III (A), and (f) Method III (B)

There are 50 images of faces that are tested. The first 25 face image was taken using a camera where the image is the same as the positive face image. The second 25 pictures were taken from the LFW images dataset, which differs from the face in the positive face image. Figure 9 shows the sample result of the system using the Kivy framework. Methods I (A) and (B) get pretty good accuracy, but Method II (A) is still better. Method I only uses a little training, and data does not use augmentation techniques. Method II (A) is the best among other methods by using a detection threshold of 0.1 and a verification threshold of 0.8. Method II (B) has low accuracy because it uses a detection threshold that is too high, namely 0.5. Methods III (A) and (B) get low accuracy because the model is overfitting.

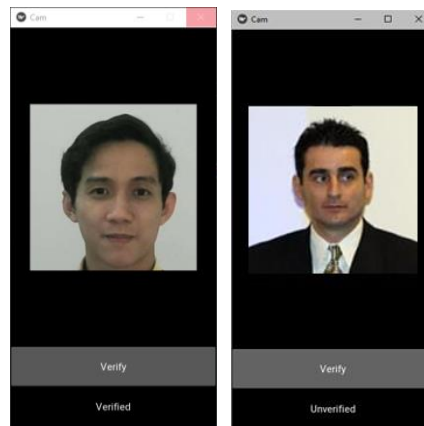


Figure 9. Sample face recognition using Kivy framework

The comparison results of implementation Siamese CNN and other recent works using the LFW dataset are shown in Table 3. As shown in Table 3, our work achieves better accuracy than most of the other recent work reported in this paper. The accuracy of the joint Bayesian method for the LFW dataset is 90.9%, the accuracy of the fisher vector faces method for the LFW dataset is 93%, the accuracy of the FR+FCN method for the LFW dataset is 93.6%, the accuracy of the principal component analysis (PCA), discrete cosine transform (DCT) method for CASIA-Web face and LFW dataset is 94.8%, the accuracy of the Face++ method for LFW Dataset is 97.2%, and the accuracy of the Siamese CNN method for LFW Dataset is 98%. In the end, it was proved that Siamese CNN was the best among others for face recognition.

Table 3. The comparison results of the proposed method Siamese CNN with existing methods

Method Used	Dataset	Accuracy
Joint Bayesian [28]	LFW Dataset	90.9%
Fisher Vector Faces [29]	LFW Dataset	93%
FR+FCN [30]	LFW Dataset	93.6%
PCA, DCT [31]	CASIA-web face and LFW dataset	94.8%
Face++ [30]	LFW Dataset	97.2%
Siamese CNN (Proposed)	LFW dataset	98%

4. CONCLUSION

In this paper, we have proposed a method to improve face detection performance using Siamese CNN. The experiments show that the proposed method for face detection using the augmentation technique presents superior results than not using augmentation. The objective was to improve accuracy, which is the goal of every face recognition system. The LFW images dataset was used to test the approach. The approach was tested on a total of 9,000 face images, with a classification accuracy rate of 98%. The rate of recognition confidence is influenced by the number of photos used for training. It shows that the Siamese CNN can be utilized for real-world face recognition using Kivy framework. The Kivy framework effectively constructs and tests the suggested facial recognition method and mobile application. The researchers want to validate the method with a variety of datasets in the future. By increasing the number of images used in the technique, the degree of accuracy may be further enhanced.




REFERENCES

- [1] S. Tikoo and N. Malik, "Detection of face using viola jones and recognition using back propagation neural network," *International Journal of Computer Science and Mobile Computing*, vol. 5, no. 5, pp. 288–295, 2017, Accessed: Mar. 23, 2022. [Online]. Available: <http://arxiv.org/abs/1701.08257>
- [2] J. Efendi, M. I. Zul, and W. Yunanto, "Real time face recognition using eigenface and viola-jones face detector," *International Journal on Informatics Visualization*, vol. 1, no. 1, pp. 16–22, Mar. 2017, doi: 10.30630/joiv.1.1.15.
- [3] S. Suwarno and K. Kevin, "Analysis of face recognition algorithm: Dlib and OpenCV," *Journal of Informatics and Telecommunication Engineering*, vol. 4, no. 1, pp. 173–184, Jul. 2020, doi: 10.31289/jite.v4i1.3865.
- [4] Y. Pratama, L. M. Ginting, E. H. L. Nainggolan, and A. E. Rismanda, "Face recognition for presence system by using residual networks-50 architecture," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, pp. 5488–5496, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5488-5496.
- [5] M. Moussa, M. Hamila, and A. Douik, "Face recognition using fractional coefficients and discrete cosine transform tool," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 892–899, Feb. 2021, doi: 10.11591/ijece.v11i1.pp892-899.
- [6] D. S. A. ELminaam, A. M. Almansori, M. Taha, and E. Badr, "A deep facial recognition system using computational intelligent algorithms," *PLoS ONE*, vol. 15, no. 12 December, p. e0242269, Dec. 2020, doi: 10.1371/journal.pone.0242269.
- [7] C. E. Panjaitan, D. Hagayna, D. Prandi, R. Wiranto, P. T. Elektro, and F. Teknologi, "Integration face recognition and body temperature," *JITE (Journal of Informatics and Telecommunication Engineering)*, vol. 5, no. July, pp. 198–208, Jul. 2021, doi: 10.31289/JITE.V5I1.5315.
- [8] F. H. K. Zaman, J. Johari, and A. I. M. Yassin, "Learning face similarities for face verification using hybrid convolutional neural networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 3, pp. 1333–1342, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1333-1342.
- [9] M. J. Alam and T. C. M. ShahzahanAli, "A smart login system using face detection and recognition by ORB algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 2, pp. 1078–1087, Nov. 2020, doi: 10.11591/ijeecs.v20.i2.pp1078-1087.
- [10] K. D. Ismael and S. Irina, "Face recognition using Viola-Jones depending on Python," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1513–1521, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1513-1521.
- [11] A. Herlambang, P. W. Buana, and I. N. Piarsa, "Cloud-based architecture for face identification with deep learning using convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 811–820, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp811-820.
- [12] S. Yallamandaiah and N. Purnachand, "An effective face recognition method using guided image filter and convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1699–1707, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1699-1707.
- [13] H. R. V. Kumar and M. Mathivanan, "A novel hybrid face recognition framework based on a low-resolution camera for biometric applications," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 2, pp. 853–863, Nov. 2021, doi: 10.11591/ijeecs.v24.i2.pp853-863.
- [14] A. Archilles and A. Wicaksana, "Vision: A web service for face recognition using convolutional network," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 3, pp. 1389–1396, Jun. 2020, doi: 10.12928/TELKOMNIKA.v18i3.14790.
- [15] S. A. Baker, H. H. Mohammed, and H. A. Aldabagh, "Improving face recognition by artificial neural network using principal component analysis," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 6, pp. 3357–3364, Dec. 2020, doi: 10.12928/TELKOMNIKA.v18i6.16335.
- [16] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015.
- [17] W. Cui, W. Zhan, J. Yu, C. Sun, and Y. Zhang, "Face recognition via convolutional neural networks and Siamese neural networks," in *Proceedings - 2019 International Conference on Intelligent Computing, Automation and Systems, ICICAS 2019*, Dec. 2019, pp. 746–750, doi: 10.1109/ICICAS48597.2019.00161.
- [18] M. Shorfuzzaman and M. S. Hossain, "MetaCOVID: A Siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients," *Pattern Recognition*, vol. 113, p. 107700, May 2021, doi: 10.1016/j.patrec.2020.107700.
- [19] M. A.-Naji, H. Mahdavinataj, and A. Aghagolzadeh, "Alzheimer's disease diagnosis from structural MRI using Siamese convolutional neural network," in *4th International Conference on Pattern Recognition and Image Analysis, IPRIA 2019*, Mar. 2019, pp. 75–79, doi: 10.1109/PRIA.2019.8786031.
- [20] S. C. Hsiao, D. Y. Kao, Z. Y. Liu, and R. Tso, "Malware image classification using one-shot learning with Siamese networks," *Procedia Computer Science*, vol. 159, 2019, pp. 1863–1871, doi: 10.1016/j.procs.2019.09.358.
- [21] L. Torres, N. Monteiro, J. Oliveira, J. Arrais, and B. Ribeiro, "Exploring a Siamese neural network architecture for one-shot drug discovery," in *Proceedings - IEEE 20th International Conference on Bioinformatics and Bioengineering, BIBE 2020*, Oct. 2020, pp. 168–175, doi: 10.1109/BIBE50027.2020.00035.
- [22] L. Vizvary, D. Sopiak, M. Oravec, and Z. Bukovcikova, "Image quality detection using the Siamese convolutional neural network," in *Proceedings Elmar - International Symposium Electronics in Marine*, Sep. 2019, pp. 109–112, doi: 10.1109/ELMAR.2019.8918678.
- [23] G. B. Huang, M. Ramesh, T. Berg, and E. L.-Miller., "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," 07–49, 2007, doi: 10.1109/ICB.2016.7550057.
- [24] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [25] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," Oct. 2020, doi: 10.48550/arxiv.2010.16061.
- [26] C. Vani and G. R. Reddy, "Development of internet of things and cloud based applications using Kivy for mobile and smartphone integration," *Materials Today: Proceedings*, Mar. 2021, doi: 10.1016/j.matpr.2021.01.753.
- [27] A. Bhoyarkar, A. Solanki, and A. Balbudhe, "Application development using Kivy framework," *Ijarcce*, vol. 8, no. 2, pp. 53–58, Feb. 2019, doi: 10.17148/ijarcce.2019.8209.
- [28] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7574 LNCS, no. PART 3, Springer Berlin Heidelberg, 2012, pp. 566–579.




- [29] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2013, pp. 1960–1967, doi: 10.1109/ICCV.2013.246.
- [30] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Recover canonical-view faces in the wild with deep neural networks," *arXiv:1404.3543*, Apr. 2014, doi: 10.48550/arxiv.1404.3543.
- [31] L. Yao, "A compressed deep convolutional neural networks for face recognition," in *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2020*, Apr. 2020, pp. 144–149, doi: 10.1109/ICCCBDA49378.2020.9095672.

BIOGRAPHIES OF AUTHORS



Yazid AUFAR    Obtained his Bachelor Computer degree in Computer Science from Lambung Mangkurat University, South Borneo, Indonesia in 2015. He received his master, in Computer Science from IPB University, Bogor, Indonesia, in 2019. Now, he served as a lecturer in Informatics Engineering at Politeknik Hasnur, South Borneo, Indonesia. His primary research interest is in image processing, pattern recognition, machine learning, and deep learning. He can be contacted at email: yazid.aufar.ya@gmail.com.



Imas Sukaesih SITANGGANG    received the Ph.D. degree in computational intelligence from the Universiti Putra Malaysia, in 2013, and the M.Kom degree from Gadjah Mada University, in 2002, respectively. She is a lecturer in Computer Science Department, IPB University, Indonesia. Her research interest is spatial data mining and smart agriculture. She can be contacted at email: imas.sitanggang@apps.ipb.ac.id.