KDS Knowledge Engineering and Data Science (KEDS)
Vol 4, No 1, July 2021, pp. 49–54

pISSN 2597-4602
eISSN 2597-4637

Face Images Classification using VGG-CNN

I Nyoman Gede Arya Astawa [a, 1, *], Made Leo Radhitya [b, 2],
I Wayan Raka Ardana [a, 3], Felix Andika Dwiyanto [c, 3]

[a] *Electrical Engineering Department, Politeknik Negeri Bali*
Kampus Jimbaran, Badung, Bali, 80361 Indonesia

[b] *Department of Informatics, STMIK STIKOM Indonesia*
Tukad Pakerisan 97, Denpasar, Bali, 80225 Indonesia

[c] *Association for Scientific Computing Electronics and Engineering (ASCEE)*
Jl. Janti, Karangjambe 130B, Banguntapan, Bantul, Yogyakarta, Indonesia

[1] *arya_kmg@pnb.ac.id*;* [2] *leo.radhitya@stiki-indonesia.ac.id;* [3] *rakawyn@pnb.ac.id;* [4] *felix@ascee.org*
** corresponding author*

ARTICLE INFO

Article history:
Received 4 March 2021
Revised 29 March 2021
Accepted 4 April 2021
Published online 17 August 2021

Keywords:
Classification
CNN
Face
Image
VGG

ABSTRACT

Image classification is a fundamental problem in computer vision. In facial recognition, image classification can speed up the training process and also significantly improve accuracy. The use of deep learning methods in facial recognition has been commonly used. One of them is the Convolutional Neural Network (CNN) method which has high accuracy. Furthermore, this study aims to combine CNN for facial recognition and VGG for the classification process. The process begins by input the face image. Then, the preprocessor feature extractor method is used for transfer learning. This study uses a VGG-face model as an optimization model of transfer learning with a pre-trained model architecture. Specifically, the features extracted from an image can be numeric vectors. The model will use this vector to describe specific features in an image. The face image is divided into two, 17% of data test and 83% of data train. The result shows that the value of accuracy validation (val_accuracy), loss, and loss validation (val_loss) are excellent. However, the best training results are images produced from digital cameras with modified classifications. Val_accuracy's result of val_accuracy is very high (99.84%), not too far from the accuracy value (94.69%). Those slight differences indicate an excellent model, since if the difference is too much will causes underfit. Other than that, if the accuracy value is higher than the accuracy validation value, then it will cause an overfit. Likewise, in the loss and val_loss, the two values are val_loss (0.69%) and loss value (10.41%).

This is an open access article under the CC BY-SA license
(https://creativecommons.org/licenses/by-sa/4.0/).

I. Introduction

Facial recognition is one of the most widely studied biometrics fields due to a high level of difficulty [1][2]. Specifically, image classification is part of facial recognition processes, which is an actual problem in computer vision [3]. The classification process helps accelerate the training process due to data that has been classified before performing the training process. The classification method selection also determines the level of accuracy in the training process [4].

Several popular classifications in the facial recognition process are Euclidean distance, KNN, SVM, PCA, and CNN [5][6]. Currently, studies that apply the deep learning method provide better results in facial recognition [7]. The most compelling image recognition method is Convolutional Neural Network (CNN) [8]. Recent researches results show that transfer learning solutions are the basis for image classification [7][9][10]. The research claimed that CNN provides significant results.

Moreover, each binary image classification, ReLU activation function, and Sigmoid classifier combination provide the best classification accuracy [11]. Other studies result that the activation function strongly influences the system's accuracy to identify and recognize mushroom images [12].

https://doi.org/10.17977/um018v4i12021p49-54
©2021 Knowledge Engineering and Data Science | W : http://journal2.um.ac.id/index.php/keds | E : keds.journal@um.ac.id
This is an open access article under the CC BY-SA license (https://creativecommons.org/licenses/by-sa/4.0/)
KEDS is Sinta 2 Journal (https://sinta.ristekbrin.go.id/journals/detail?id=6662) accredited by Indonesian Ministry of Research & Technology

This study aims to determine the CNN method's facial images classification. In this study, the pre-trained model used is the VGG-face model [8]. This significant result was obtained using 16-19 layer weights. The classification modeling in this study is changing the last layer on CNN.

II. Method

In this study, there are several processes to achieve the expected result. Those processes are data collection, map feature extraction, classification modeling, and result validation testing. Therefore, this study using KomNet dataset [13], and the face image used is 36600 face images with a size of 224×224 pixels. In computer vision, transfer learning is commonly expressed through the use of a pre-trained model. A typical implementation used is to import and use models from existing libraries.

The next step is to create a new convolutional neural networks (CNN) model for image classification using multiclass CNN. This image classification model is generated from the transfer learning approach, which is based on the CNN pre-trained model [14]. In general, CNN proved to be superior in a variety of computer vision tasks [15]. Convolutional Networks (ConvNets) have shown excellent performance in handwritten digital classification and face detection [16].

Figure 1 is an outline of the CNN processes in the system. The process begins by inputting the face image. The method used for transfer learning is the feature extractor preprocessor. This study uses an optimization model of transfer learning with a pre-trained model architecture, the VGG-face model. Mainly, the features extracted from an image can be numeric vectors. The model will use this vector to describe specific features in an image. The reason for the VGG-face model selection because it is perfect for producing facial feature extraction [17]. The Feature Extraction has a VGG-face 16 layer architecture. After performing the VGG-face model, the last layer of the VGG-face will be modified to achieve the maximum result.

Figure 2 presents the VGG-face architecture with the last three layers are the classifications to be modified. The first layer features are general, and the last layer features specific, so there must be a transition from general to specific somewhere on the network [18]. The pre-trained strategy leaves some initial layers unprocessed and trains the final layers to avoid overfitting [19]. The initial layer is for convolution or feature extraction, while the last layer is for classification. The last three layers are

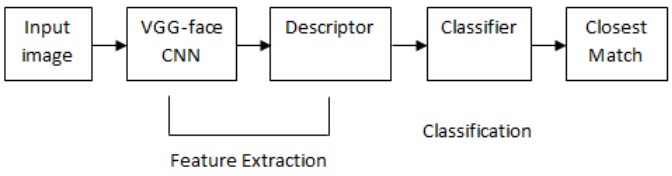


Fig. 1. CNN process

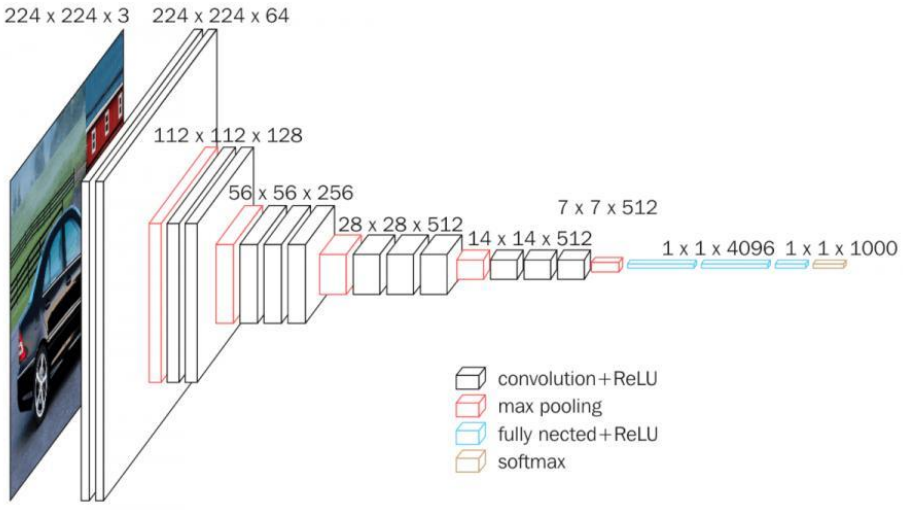


Fig. 2. VGG-face Architecture

fully connected + ReLU. The modification of the last three layers is required to provide better performance. Following is the pseudocode for the last three layers.

```
#LAST LAYER
classifier_model=Sequential()
classifier_model.add(Dense(units=100,input_dim=x_train.shape[1],kernel_initializer='glorot_uniform'))
classifier_model.add(BatchNormalization())
classifier_model.add(Activation('tanh'))
classifier_model.add(Dropout(0.3))
classifier_model.add(Dense(units=10,kernel_initializer='glorot_uniform'))
classifier_model.add(BatchNormalization())
classifier_model.add(Activation('tanh'))
classifier_model.add(Dropout(0.2))
classifier_model.add(Dense(units=24,kernel_initializer='he_uniform'))
classifier_model.add(Activation('softmax'))
classifier_model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(),optimizer='nadam',metrics=['accuracy'])
```

The last layer of the VGG-face model is the one that is wholly connected before the output layer. These layers will provide a complex set of features for describing an input image and provide useful input when training a new image classification model.

After the pre-trained model from the last VGG-face layer is loaded, the next step is to create a data train and data test. It consists of five stages, and the first stage is to change the existing wavelet feature in the train or test folder with a target size of 224 (224×224 pixels). At the second stage, it needs to change the image into an array. Next, the third stage is inputting the results into the last VGG-face. Then, the results are entered into the train data array and the test data array. After that, the last stage is to repeat step one until all face images in the train or test folder have been read

After the model is made, the next step is using epoch 100 in the training process. In this process, the weight value will be obtained, which is stored in a file in h5 format. Tests are performed to obtain a validation test of the results through training facial images from several devices. The results of image testing from several devices are displayed in graphical form.

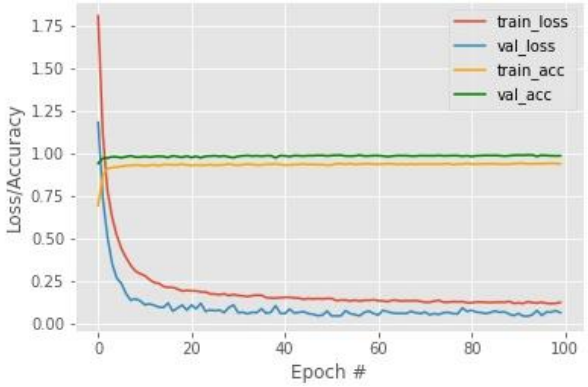
III. Results and Discussions

The use of massive data is necessary to produce an ideal result. Moreover, the pre-trained model is a conversion model provided by TensorFlow or Keras. This pre-trained model can be used directly from the VGG-face Keras library. After the model is made, the next step is the training process with epoch 100. Epoch 100 limits the iteration of large amounts of data that takes a long time to train in one training session. However, the deep learning method has a weakness with the long training process when using a server computer. It can be overcome using Graphical Processing Unit (GPU) technology [9][20]. This study using GPU, which Google Colab owns for the training process. The results of the training process are the weight value which is stored in an h5 format file. The train results with epoch 100 with three sources of face images are presented in Table 1.

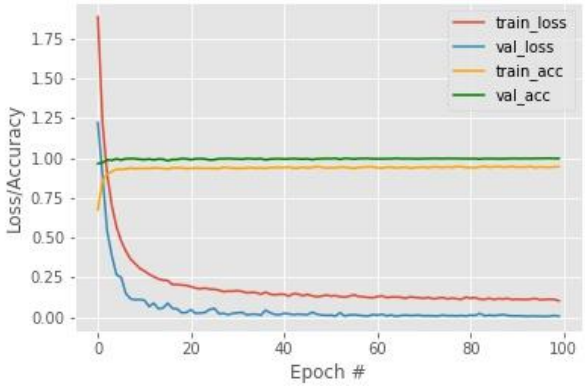
Table 1 shows that the results of facial image training are from three devices at epoch 100. In the training process, facial images are divided into two, which are 17% as data test and 83% as data train. The accuracy value, which are val_accuracy, loss, and val_loss values, are impressive. However, the best training result is the image that comes from a digital camera with a modified classification. The val_accuracy result is very high (99.84%), not too far from the accuracy result (94.69%). The difference in value that is not too significant indicates a great model. It is because if the difference is too far, it will cause an under fit. Other than that, if the accuracy value is higher than the accuracy validation value, it will cause an over fit. Moreover, the val_loss result is very low (0.69%) and the loss value is 10.41%. The error or loss value is the smallest compared to the others, which means that the model is ideal and proper to be used as a prediction.

Table 1. Training result of three image sources on epoch 100

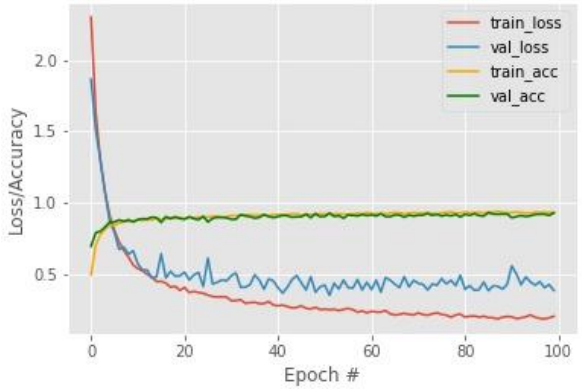
Image source	Number of face images		Epoch 100			
	Train	Test	Accuracy (%)	Val_Accuracy (%)	Loss (%)	Val-loss (%)
Mobile phone	11,000	2,200	94.05	98.69	12.32	6.29
Digital camera	11,000	2,200	94.69	99.84	10.41	0.69
Social media	11,000	2,200	93.02	92.75	20.07	38.20



(a)



(b)



(c)

Fig. 3. Graph of training results on epoch 100 with modification and facial image classification sourced from (a) mobile phones, (b) digital cameras, and (c) social media

The training results from start to finish are presented in a graphic image. Figure 3 shows a graph of the facial image training result on Epoch 100 with modified classification. Figure 3 shows that the model (the modification of the last three layers) is great and ideal since the value differences are insignificant. Likewise, the difference between val_loss and loss is relatively small, and the values are close.

IV. Conclusion

This study performed a pre-trained model using the VGG-face architecture to modify the last three layers or the classification section. The model provides a result of very high accuracy. Also, the resulting loss is shallow. It is indicated that the model is great and ideal for prediction. Moreover, the image data for training are obtained from three sources. Based on the three image sources, the best source is from the digital camera with accuracy = 94.69%, and loss = 10.41%. Therefore, further research needs to focus on the quality of camera image sources to improve the classification performance optimally.

Acknowledgment

Politeknik Negeri Bali and STIKI Indonesia supported this research. We thank everyone who contributed to the completion of this paper. Hopefully, this research significantly contributes to knowledge development, especially in face image classification.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information is available at http://journal2.um.ac.id/index.php/keds.

Publisher's Note: Department of Electrical Engineering - Unversitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

[1] M. Andrejevic and N. Selwyn, "Facial recognition technology in schools: critical questions and concerns," *Learn. Media Technol.*, vol. 45, no. 2, pp. 115–128, Apr. 2020, doi: 10.1080/17439884.2020.1686014.

[2] C. M. Cook, J. J. Howard, Y. B. Sirotin, J. L. Tipton, and A. R. Vemury, "Demographic Effects in Facial Recognition and Their Dependence on Image Acquisition: An Evaluation of Eleven Commercial Systems," *IEEE Trans. Biometrics, Behav. Identity Sci.*, vol. 1, no. 1, pp. 32–41, Jan. 2019, doi: 10.1109/TBIOM.2019.2897801.

[3] Y. Lin and H. Xie, "Face Gender Recognition based on Face Recognition Feature Vectors," in *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Sep. 2020, pp. 162–166, doi: 10.1109/ICISCAE51034.2020.9236905.

[4] M. Imani and H. Ghassemian, "Fast feature selection methods for classification of hyperspectral images," in *7'th International Symposium on Telecommunications (IST'2014)*, Sep. 2014, pp. 78–83, doi: 10.1109/ISTEL.2014.7000673.

[5] Y. Zhu, C. Zhu, and X. Li, "Improved principal component analysis and linear regression classification for face recognition," *Signal Processing*, vol. 145, pp. 175–182, Apr. 2018, doi: 10.1016/j.sigpro.2017.11.018.

[6] A. Raikwar and J. Agrawal, "A Review of Face Recognition Using Feature Optimization and Classification Techniques," in *Information Management and Machine Intelligence. ICIMMI 2019. Algorithms for Intelligent Systems*, D. Goyal, V. E. Bălaş, A. Mukherjee, C. de A. V. Hugo, and A. K. Gupta, Eds. Singapore: Springer, 2021, pp. 595–604.

[7] A. Bilgic, O. C. Kurban, and T. Yildirim, "Face recognition classifier based on dimension reduction in deep learning properties," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, May 2017, pp. 1–4, doi: 10.1109/SIU.2017.7960368.

[8] T. Purwaningsih, I. A. Anjani, and P. B. Utami, "Convolutional Neural Networks Implementation for Chili Classification," in *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, Aug. 2018, pp. 190–194, doi: 10.1109/SAIN.2018.8673373.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[10] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv Prepr. arXiv1409.1556*, Sep. 2014.

[11] K. Chauhan and S. Ram, "Image classification with deep learning and comparison between different convolutional

neural network structures using tensorflow and keras," *Int. J. Adv. Eng. Res. Dev.*, vol. 5, no. 02, pp. 533–538, 2018.

[12] A. Fadlil, R. Umar, and S. Gustina, "Mushroom Images Identification Using Orde 1 Statistics Feature Extraction with Artificial Neural Network Classification Technique," *Journal of Physics: Conference Series*, vol. 1373, p. 012037, Nov. 2019.

[13] I. N. G. A. Astawa, I. K. G. D. Putra, M. Sudarma, and R. S. Hartati, "KomNET: Face Image Dataset from Various Media for Face Recognition," *Data Br.*, vol. 31, p. 105677, Aug. 2020, doi: 10.1016/j.dib.2020.105677.

[14] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.

[15] Y. Bengio, "Learning Deep Architectures for AI," *Found. Trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009, doi: 10.1561/2200000006.

[16] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer, 2014, pp. 818–833.

[17] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, May 2018, pp. 67–74, doi: 10.1109/FG.2018.00020.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *arXiv Prepr. arXiv1411.1792*, Nov. 2014.

[19] P. Marcelino, "Transfer learning from pre-trained models," 2018.

[20] Y. E. Wang, G.-Y. Wei, and D. Brooks, "Benchmarking TPU, GPU, and CPU Platforms for Deep Learning," *arXiv Prepr. arXiv1907.10701*, Jul. 2019.