ICIC Express Letters
Volume **13**, Number **7**, July **2019**

 ISSN 1881-803X
pp. **569–574**

CONVOLUTIONAL NEURAL NETWORK FOR FACE RECOGNITION IN MOBILE PHONES

Andry Chowanda and Rhio Sutoyo

Computer Science Department
School of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ achowanda; rsutoyo }@binus.edu

Received December 2018; accepted March 2019

Abstract. *This paper presents the implementation of deep learning models for face recognition in a mobile phone by using CoreML. A dataset with a total of 1632 images from sixteen faces with various poses and angles was trained by using two popular existing Convolutional Neural Network (CNN) architectures: VGG-19, and Google's Inception V3 using Caffe deep learning framework. The models then were converted to a light CNN model (i.e., CoreML Model) by using CoreMLTools and Python. The converted models were implemented to an iOS-based application developed using Swift and CoreML SDK. The results show that both models resulted in an accuracy score of 93.2% and 93.3% for VGG-19 and Google's Inception V3 architectures respectively. The average accuracy score achieved by VGG-19 was 78.7% and Google's Inception V3 was 78.8%. There is no statistically significant difference between model trained by VGG-19 and Google's Inception V3 architecture. This almost certainly happened due to the fact that the number of images in the dataset is not significantly large enough to give a significant difference between those two architectures.*
Keywords: CNN, Mobile phone, Face recognition, CoreML

1. **Introduction.** Mobile phone technology has been improving tremendously in this ubiquitous era today. The power of computing (CPU), graphics (GPU) and video camera have been advancing expeditiously past these years. This opens the opportunity to implement algorithms that require abundant power of computing to process. A mobile phone has become a necessity for us to be connected to the world. The major improvement of mobile phone's computing power would also lead to the opportunity for the improvement of the user experience in a mobile phone. One of the methods to improve the experience within a mobile phone is to allow the mobile phone to accurately recognize a person. Recognizing one's identity is a basic operation to improve the user experience [19, 27] as this will allow a personalized user experience when user interacts with the mobile phone.

Face recognition is principally a classic problem in the computer vision research field. Similar to the other computer vision problems, face recognition problems revolve around in pose, lighting, and obstacles in faces. A number of researches have been done to solve the problems, and one of the popular approaches is by using deep learning. Today's best deep learning architecture for face recognition method is the Convolutional Neural Network (CNN). CNN allows the network to learn independently the images' features layer by layer with features maps that convolve around the images. The initial layers learn simpler features, while the latter layers learn more complex features [11]. A large neural-network architecture posed several problems if implemented to a device or system that has limited resources of computing and storage. The computing power required to train abundant images is extremely huge. Moreover, the model resulted from the training

DOI: 10.24507/icicel.13.07.569

process generally is extremely large. However, a couple of giant techs corporations have introduced API that can dramatically reduce the size of the trained models. This could be used to deal with the storage problem in the limited resources devices such as a mobile phone.

This paper presents the development of face recognition models by using Convolutional Neural Network (CNN) for mobile phones. Two popular CNN architectures were used to train the model and deploy it on a mobile phone. The models then were converted to CoreML Models[1] and deployed to a mobile phone. The results show the models achieved a maximum score of 93.2% and 93.3% for VGG-19 and Google's Inception V3 architectures respectively. The models then were converted to light CNN models using CoreMLTools. The converted models were deployed to an iOS-based mobile phone by using Swift and CoreML SDK.

2. Recent Work.

2.1. Face recognition.

This research applies Convolutional Neural Network (CNN) to solving face recognition problem in mobile phones. CNN feature extractor uses deep learning, a method that independently learns and discovers features in the provided image datasets. Before CNN, there are several other approaches of face recognition techniques which are categorized as local image descriptors such as: Histograms of Oriented Gradient (HOG) [21, 26], Local Binary Pattern (LBP) [6, 24], and Scale-Invariant Feature Transform (SIFT) [4, 5]. In these previous approaches, the researchers manually determine image features from the datasets. Then, it extracts a small portion of the images using image descriptors. In the later step, a pooling mechanism is used to generate the bigger picture [18]. Other facial recognition approaches can also be found here [15].

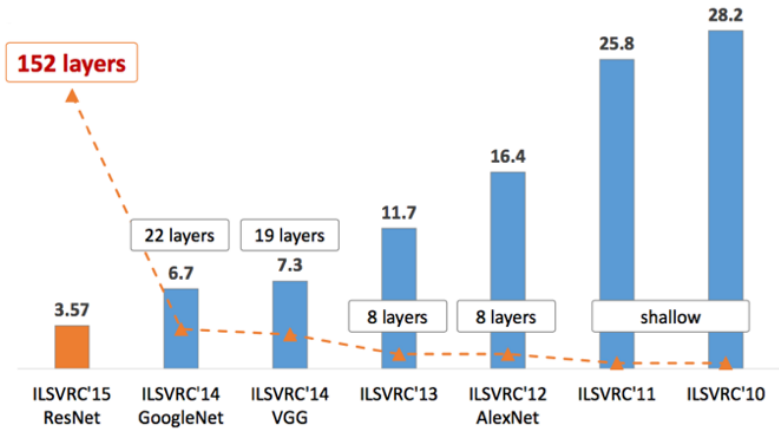


FIGURE 1. CNN architectures and its error rate result on ILSVRC [1]

2.2. Convolutional neural network.

Convolutional Neural Network (CNN) is a multi-layered neural network that focuses on visual or image processing [9, 20]. There are several architectures in CNN and it is still evolving. These architectures are usually introduced on ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The contest processes its large database, ImageNet, using CNN method to correctly classify and detect a visual object on the datasets. It can be clearly seen in Figure 1; the early model of CNN's architecture only has a shallow number of layers. Residual Neural Network (ResNet) introduced in the ILSVRC 2015 increased the number of layers into 152. In this research, we choose VGGNet Framework [22] introduced in the ILSVRC and developed by Simonyan and Zisserman as our CNN's architecture because it has medium size layers that

[1]https://developer.apple.com/documentation/coreml

is suitable for training model in mobile phones. One advantage of using CNN and deep learning technique compared with older learning algorithm is its scale with big amount of data (i.e., more data delivers improved result). In a conventional machine learning algorithm, more data generally delivers improved results until it reached some threshold point, where more data will not result in a better result. Nevertheless, it also becomes one of its cons because it also means the technique needs tremendous and various data to produce a satisfying result. Looking at the CNN's characteristic, this research only uses a small portion of face datasets and aims to combine the coreML API that can reduce the need for big-size of trained models.

2.3. **Deep learning in a mobile phone.** Research in deep learning area has been a popular topic past these years due to the major improvement of the Graphical Processing Unit (GPU) technology to process the computing problem in deep learning architectures. Modelling data with statistics, mathematics and graphics were linked back to the first neural network algorithm coined in 1940s [16]. However, the computing power back then was not powerful enough to cope with a numerous number of data or complex architecture. Nowadays, the computing power does not become a major problem in deep learning due to the enhancement of the Graphical Processing Unit (GPU) technology. Recently, deep learning models are being implemented to mobile phones as those devices have been improving tremendously these days. Their CPU and GPU computing power has exceeded the computer. This expands an immense number of opportunities to improve the mobile phone user experiences. Some researches have shown that deep learning implementation in a mobile phone has revolutionised mobile sensing [14] and has changed the way we interact with phones [13, 17].

Some researches have been done to implement deep learning to a mobile phone [7, 8, 12, 23]. Most of the approaches were to build the deep learning network efficiently, so the mobile phones are able to process the network fast and efficient, as the main concern to implement a deep learning network to a mobile phone is the battery (besides the computing power). Han et al. [7] implemented an interface engine to compress the deep neural network architecture and speeded up the process up to 1018 times faster and compressed 119,78 times smaller than the original architecture. Hinton et al. [8] used a small image resolution and eliminate the explaining away effects in the architecture to perform a faster deep learning. Finally, Lane et al. [12] used a software accelerator to escalate the power of mobile phone to process a deep neural network architecture in a mobile phone.

3. **Proposed Method.** Figure 2 illustrates the proposed research methods in this research. First dataset was collected using front camera from a smartphone (f/2.2, 2.87 mm, exposure time = 1/44). A burst picture (2320×3088) was collected from sixteen adults (14-21 images per-person covering several angles and poses) with a total of 272 images. In the data pre-processing phase, a face detector was applied to the images to crop the image. Due to the amount of the images, the collected dataset then was augmented in this phase. The images were augmented by rotating the images to 30, 60, 120, 150 degrees, sheering, and flipping the images horizontally, resulting in a total of 1632 images. For the next step, the dataset was trained by using two existing CNN architectures, VGG-19 [22] and Google's Inception V3 [25]. The dataset was trained in 15000 iterations using Caffe

FIGURE 2. Proposed research methods

[10] with step learning policy, base learning rate = 0.001, momentum = 0.9, step size = 10000, and gamma = 0.1 using GPU. The Caffe models then were converted to a light deep learning model, CoreML models using CoreMLTools[2].

Next step is to implement the converted models to a mobile phone. The mobile phone used was an iPhone X, with A11 Bionic Chip, Hexa-core 2.39 GHz processor, three-core graphics GPU, 3 GB RAM, and a dual camera 12 MP, f/1.8, 28 mm + 12 MP, f/2.4, 52 mm. Finally, the models were evaluated with a total of 180 images divided into 26 classes. Sixteen classes were from the original classes, and ten additional images were added to the testset. The architectures used to evaluate models were two existing CNN architectures used in the training, VGG-19 [22] and Google's Inception V3 [25]. Section 4 demonstrates the results and discussion from the training and evaluation of the models trained with both VGG-19 and Google's Inception V3.

4. **Results and Discussion.** A dataset collected with 1632 images was trained with two existing CNN architectures, VGG-19 [22] and Google's Inception V3 [25]. Figure 3 illustrates the example of training dataset images that have been processed in the data pre-processing phase. Caffe deep learning framework was used to train the models using VGG-19 and Google's Inception V3 architectures (see Figure 4 for a CNN architecture illustration). Both architectures receive input of $224 \times 224 \times 3$ (RGB) images to be convolved. The final layers of architecture are flattened, fully connected, and classified using Softmax. Please refer to the original papers for the details of the architectures [22, 25]. The models were trained using a computer with GPU powered for less than an hour each architecture.

FIGURE 3. Training dataset

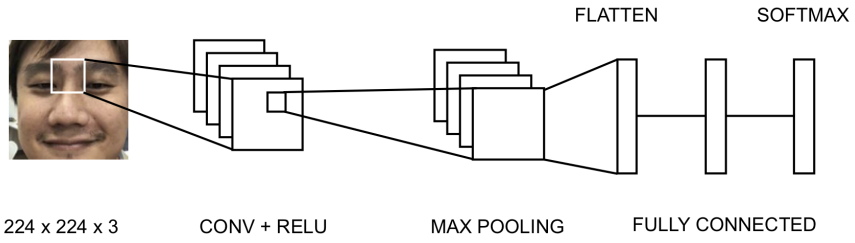
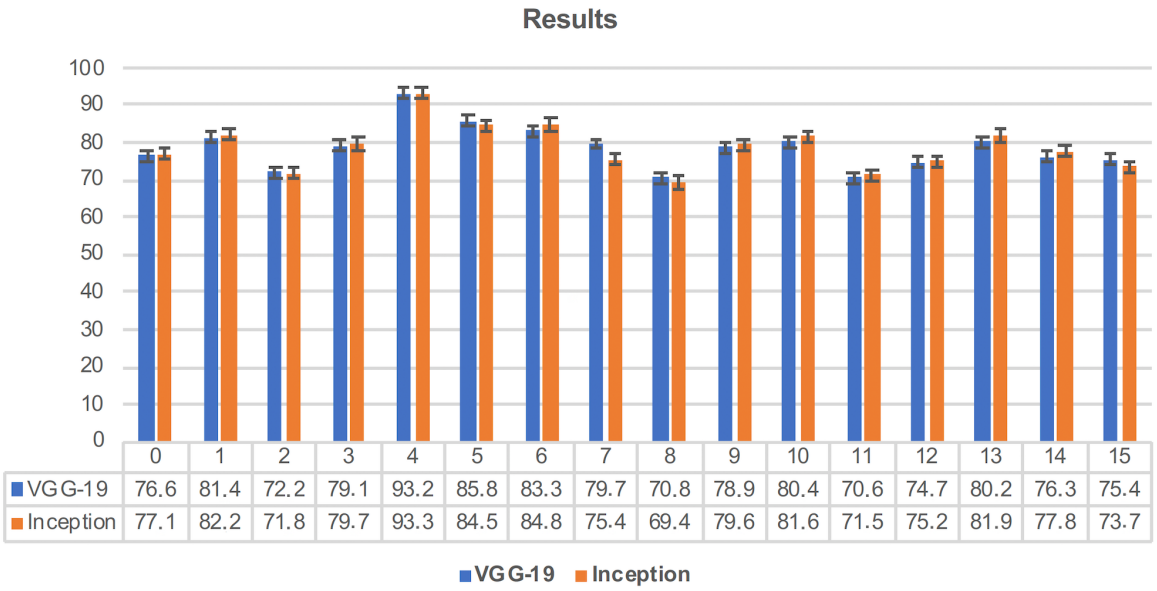


FIGURE 4. CNN architecture

Figure 5 demonstrates the highest accuracy performed by both CNN architectures, VGG-19 and Google's Inception V3 with the dataset collected. The average accuracy score achieved by VGG-19 was 78.7% and Google's Inception V3 was 78.8%. There is no statistically significant difference between model trained by VGG-19 and Google's Inception V3 architecture. This almost certainly happened due to the fact that the number of images in the dataset is not significantly large enough to give a significant difference

[2]https://pypi.org/project/coremltools/



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
VGG-19	76.6	81.4	72.2	79.1	93.2	85.8	83.3	79.7	70.8	78.9	80.4	70.6	74.7	80.2	76.3	75.4
Inception	77.1	82.2	71.8	79.7	93.3	84.5	84.8	75.4	69.4	79.6	81.6	71.5	75.2	81.9	77.8	73.7

FIGURE 5. Training results

between those two architectures. The lowest accuracy score achieved by VGG-19 belongs to Class 11 with a score of 70.6%, while Google's Inception V3 achieved a score of 69.4% (Class 8). Moreover, the highest accuracy score achieved by both architectures belongs to Class 4, with an accuracy score of 93.2% and 93.3% from VGG-19 and Google's Inception V3 architecture respectively. The models then were converted to CoreML Model by using CoreMLTools and Python before the models were implemented to a mobile phone. An iOS application to recognize people face (using the converted models) then was developed by using Swift and CoreML SDK.

5. **Conclusion and Future Work.** Two models were trained with two existing CNN architectures, VGG-19 [22] and Google's Inception V3 [25] using Caffe deep learning framework [10]. The average accuracy scores achieved by both models were 78.7% and 78.8% for VGG-19 and Google's Inception V3 architectures respectively. The highest accuracy score belongs to Class 4 for both architectures (93.2% for VGG-19 and 93.3% for Google's Inception V3 architecture). The trained models then were converted using CoreMLTools and were implemented to an iOS-based mobile phone application using Swift and CoreML SDK. With the converted models, the application is able to recognize people face in real time.

Next steps for the research plan are to collect more faces data and train the models with more images, and to implement the models to a bigger system such as a virtual human [2, 3, 27] that is able to recognize people and enhance the social interaction between the virtual human and the users [3]. With the light version of the converted models, the system can be implemented to mobile phones.

Acknowledgment. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. We also would like to express our gratitude to all the participants who participate in the faces data collection.

REFERENCES

[1] *CNN Architectures: LeNet, AlexNet, VGG, GoogleNet, ResNet and More*, 2017.
[2] A. Chowanda, P. Blanchfield, M. Flintham and M. Valstar, ERiSA: Building emotionally realistic social game-agents companions, in *Intelligent Virtual Agents (IVA 2014), Lecture Notes in Computer Science*, T. Bickmore, S. Marsella and C. Sidner (eds.), Springer, Cham, 2014.

[3] A. Chowanda, P. Blanchfield, M. Flintham and M. Valstar, Computational models of emotion, personality, and social relationships for interactions in games, *Proc. of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp.1343-1344, 2016.
[4] C. Geng and X. Jiang, Face recognition using SIFT features, *The 16th IEEE International Conference on Image Processing (ICIP)*, pp.3313-3316, 2009.
[5] C. Geng and X. Jiang, SIFT features for face recognition, *The 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, pp.598-602, 2009.
[6] Z. Guo, L. Zhang and D. Zhang, Rotation invariant texture classification using LBP variance (LBPV) with global matching, *Pattern Recognition*, vol.43, no.3, pp.706-719, 2010.
[7] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz and W. J. Dally, EIE: Efficient inference engine on compressed deep neural network, *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp.243-254, 2016.
[8] G. E. Hinton, S. Osindero and Y.-W. Teh, A fast learning algorithm for deep belief nets, *Neural Computation*, vol.18, no.7, pp.1527-1554, 2006.
[9] S. Hoo-Chang, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura and R. M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, *IEEE Trans. Medical Imaging*, vol.35, no.5, pp.1285-1298, 2016.
[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional architecture for fast feature embedding, *Proc. of the 22nd ACM International Conference on Multimedia*, pp.675-678, 2014.
[11] A. Krizhevsky, I. Sutskever and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, pp.1097-1105, 2012.
[12] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, L. Jiao, L. Qendro and F. Kawsar, DeepX: A software accelerator for low-power deep learning inference on mobile devices, *Proc. of the 15th International Conference on Information Processing in Sensor Networks*, 2016.
[13] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi and F. Kawsar, An early resource characterization of deep learning on wearables, smartphones and Internet-of-Things devices, *Proc. of the 2015 International Workshop on Internet of Things towards Applications*, pp.7-12, 2015.
[14] N. D. Lane and P. Georgiev, Can deep learning revolutionize mobile sensing?, *Proc. of the 16th International Workshop on Mobile Computing Systems and Applications*, pp.117-122, 2015.
[15] Z. Mahmood, N. Muhammad, N. Bibi and T. Ali, A review on state-of-the-art face recognition approaches, *Fractals*, vol.25, no.2, 2017.
[16] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *The Bulletin of Mathematical Biophysics*, vol.5, no.4, pp.115-133, 1943.
[17] R. Miotto, F. Wang, S. Wang, X. Jiang and J. T. Dudley, Deep learning for healthcare: Review, opportunities and challenges, *Briefings in Bioinformatics*, 2017.
[18] O. M. Parkhi, A. Vedaldi, A. Zisserman et al., Deep face recognition, *BMVC*, vol.1, 2015.
[19] Y. L. Prasetio, R. Wijaya, M. P. Sjah, M. R. Christian and A. Chowanda, Location-based game to enhance player's experience in survival horror game, *Procedia Computer Science*, vol.116, pp.206-213, 2017.
[20] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, CNN features off-the-shelf: An astounding baseline for recognition, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp.806-813, 2014.
[21] C. Shu, X. Ding and C. Fang, Histogram of the oriented gradient for face recognition, *Tsinghua Science and Technology*, vol.16, no.2, pp.216-224, 2011.
[22] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*, 2014.
[23] D. Suryani, V. Ekaputra and A. Chowanda, Multi-modal Asian conversation mobile video dataset for recognition task, *International Journal of Electrical and Computer Engineering (IJECE)*, vol.8, no.5, 2018.
[24] R. Sutoyo, J. Harefa and A. Chowanda, Unlock screen application design using face expression on Android smartphone, *MATEC Web of Conferences*, vol.54, 2016.
[25] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, *AAAI*, vol.4, pp.4278-4284, 2017.
[26] X. Wang, T. X. Han and S. Yan, An HOG-LBP human detector with partial occlusion handling, *2009 IEEE 12th International Conference on Computer Vision*, pp.32-39, 2009.
[27] W. Zhu, A. Chowanda and M. Valstar, Topic switch models for dialogue management in virtual humans, *International Conference on Intelligent Virtual Agents*, pp.407-411, 2016.