Child face recognition system with deep learning

Shun Lei Myat Oo, Aung Nway Oo
University of Information Technology, Yangon, Myanmar
shunleimyatoo@uit.edu.mm, aungnwayoo@uit.edu.mm

Abstract

Deep learning is one of the popular techniques in machine learning and it gives state-of-art performance on a wide variety of tasks such as natural language processing (NLP), face recognition and speech recognition. Convolutional Neural Networks (CNNs) is a top performer on face recognition. In this paper, the accuracy and performance of three Convolutional Neural Networks (CNNs) such as VGG Face based on two architectures (VGG16 and ResNet50),and MobileFaceNet on child face dataset is tested. The experiments results are shown and evaluated. According to experiments results, MobileFaceNet on child face dataset provide better accuracy than others. Among three proposed methods, the best recognition accuracy is 99.75% from MobileFaceNet.

Key Words- Deep Learning, Convolutional Neural Network(CNN) , Face recognition, VGG16, ResNet50, MobileFaceNet

1. Introduction

Face recognition system is a system which can find and identify a specific person or people by detecting their faces from images or videos. Face recognition is one of the ongoing researches and various kinds of techniques and algorithms have been developed. Deep learning is a kind of state-of-art method which can give high performance on face recognition. Deep learning is a collection of layers of neurons and it perform like a human brain, i.e., it can instantly detect and recognize single or multiple faces.

Face recognition is a process of matching existing data and unknown data. Face recognition is applied non-contact biometric authentication and identification system. Generally, Convolutional Neural Network (CNN) is used for feature extraction and classification. CNN consists of three layers which are convolutional layer, pooling layer and fully connected layer. Convolutional layer and pooling layer are used for feature extraction. Fully connected layer is used for classification. Analyst and scientist need domain or business knowledge for visual recognition in traditional way. Convolutional Neural Networks (CNNs) eliminate the manual feature extraction and it appear as a form of automated feature engineering.

Face recognition system is applied in wide variety of applications such as school or public security system, finding missing children or adult and social network. Face recognition system includes two stages: face detection and face recognition. Face detection is process of finding faces from images or videos. Face recognition is a process of matching existing faces from database and unknown faces and decide who he is or who she is.

In this paper, three Convolutional Neural Networks (VGG16, ResNet50 and MobileFaceNet) were proposed. MobileFaceNet is a lightweight model compared to other two models. MobileFaceNet has less than 1 million parameters [1]. It works efficient and fast on embedded devices because of its depthwise separable convolutional layers. Models details are described in experiments and results section. Ensemble of Regression Trees (ERT) presented by Vahid Kazemi and Josephine Sullivan in 2014 is used for face detection and alignment as a preprocessing step.

2. Related Work

In early 1990s and late 2000s, holistic learning approach and local handcrafted approach dominated face recognition area respectively [2]. This approach faced worse result for unconstrained facial changes, lighting, expression and pose since it used two or three feature descriptors. In 2012, AlexNet [3] won the ImageNet competition by reducing the top-5 error from 26% to 15.3% on ImageNet, achieving a top-1 error rate of 37.5% using a deep learning technique and it overcomes the previous methods performance. Convolutional neural network (CNN) is one of the types of deep learning methods. I use multiple layers feature descriptors for feature extraction and transformation. Generally, early layers extract the basic features of face and later layers extract the detail features of face. DeepFace and DeepID achieve the high accuracy on LFW (Labeled Face in-the-Wild) dataset around 90% in 2012. In 2015, FaceNet [4] , was trained by triplet loss, achieves 99.63% on LFW and 95.12% on YouTube Faces DB. It achieves state-of-art face recognition performance and use 128 embeddings per face. In 2017, ResNet50 [5] use residual blocks and residual connections and train on VGGFace2, on MS-

Celeb-1M and on their union to assess face recognition performance. In 2018, MobileFaceNet achieves high accuracy with real-time high-performance face recognition and efficient on embedded devices.

3. State of art face recognition system

The process of face recognition is generally preprocessing, feature extraction and classification (see in Figure 1). Preprocessing step includes face detection, face alignment or face landmark localization to convert raw image data to trainable form. Ensemble of regression tree is used for estimate face's landmark positions directly from spare subset of pixel intensities [6]. 68 landmarks points are used for face alignment. A convolutional neural network (CNNs) is mainly used for feature extraction. Classification is performed by fully connected layer of CNNs or other classification algorithms such as KNN, SVM.

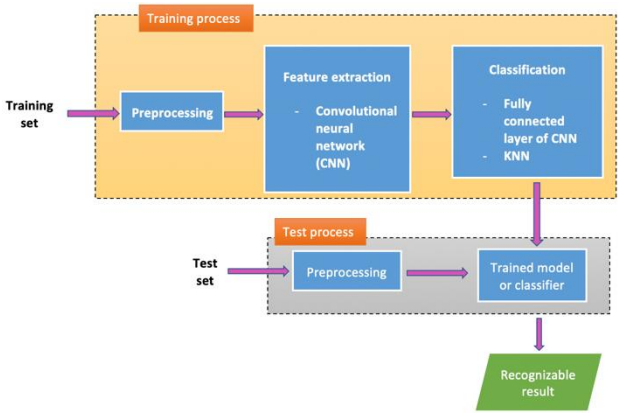

Figure 1. Flow of face recognition system

3.1. Face recognition using VGGFace (VGG16)

Keras_vggface, proposed by Oxford's VGGFace classifier, was used [7]. This model was trained on child dataset with fine tuning approach . Some earlier layers was frozen and some latter layers of model were trained. Modification was made on original model by dropping a last layer, adding some convolutional layers, dropout layers and fully connected layer. Stochastic gradient descent optimizer (SGD) was used with small learning rate 0.0001. This model consists of mainly three type of layers namely convolution layer, pooling layer (max pooling) and fully connected layer (See in Figure 3). In preprocessing step, face detection and face alignment of some parts of non-frontal face such as eyes, eyebrow, nose, mouth, chin were included. Image was resized to 224x224 which is the acceptable input size of model. Since this model accepts the RGB image as input, they were not needed to change to grayscale. Model accepts preprocessed image and then make feature extraction layer by layer. Pooling layers perform down sampling along with spatial dimensions (width x height). RELU layers perform elementwise activation function, max (0, x). Finally, fully connected layer of model makes classification task. All layers start from input layer to 'pool5' layer is original of VGG16. Three dense layers and two dropout layers are added after 'pool5' layers

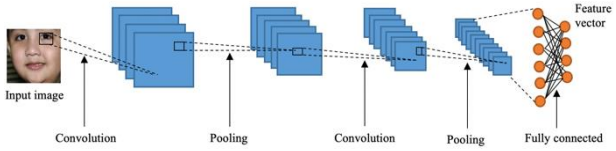

Figure 2. Flow of layers in Convolutional Neural Network (CNN)

3.2. Face recognition using VGGFace based on ResNet50

ResNet50 architecture which consists of 50 layers residual network was used. Generally, in deep learning neural networks, several layers are arranged sequentially for learning low, medium and high-level features of inputs. In residual learning, it tries to learn some residual connection instead of learning some features. ResNet uses residual blocks and residual connections which directly connecting from input to some nth layer or from nth layer to some latter layers of nth layer (see in figure 3) [8]. So, Resnet is easier to train than other simple deep neural network; training time is fast and degrading accuracy is reduced.

In original network, 3x3 filters are mostly used in network, down sampling with stride 2, global average pooling layer and fully connected layer with softmax in the end. There are two kinds of residual blocks according to input and output dimensions. The identity shortcuts are used when input and output dimensions are same. When input and output dimensions are different, firstly, the shortcut performs identity mapping and then the projection shortcut performs to match the dimension. Layers of original networks after average pooling layers were dropped. New layers were added the orginal network: two dropout layers and a fully connected layer with softmax activation. Stochastic gradient descent optimizer (SGD) with learning rate 0.001. The input image size is 224x224. Preprocessing is same as the process which made in VGG16.

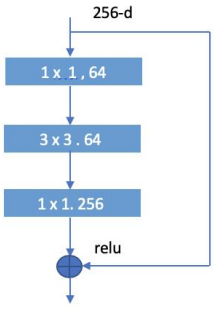

Figure 3. A ResNet block of ResNet50 with 3 layers deep

3.3. Face recognition using MobileFaceNet

MobileFaceNet is one of the lightweight models and which gives high-accuracy for real-time face recognition on mobile and embedded devices with 1 millions parameters. It achieves high accuracy compared with others high MB size models [1]. It performs more efficient and accuracy is higher than others lightweight models. Input child images are detected, aligned and resized as 112 x 112. Normalization process is performed by subtracting 127.5 and then divided by 128 of each pixel in RGB image. Output layer of the model is 128 features vectors per face. MobileFaceNet performs feature extraction process. K-Nearest Neighbors (KNN) was used for classification task. The output of 128 features vectors of MobileFaceNet was fed as input for KNN classifier. KNN constructs the graph of 128 feature vectors and calculate the distance of 128 feature vectors of an unknown image and 128 feature vectors of existing images on KNN graph. The nearest distance class is the answer class for an unknown image.

The architecture of MobileFaceNet is almost same as the architecture of MobileNetV2 [1]. MobileFaceNet adopts the bottleneck layers architecture of depth wise separable convolution found in MobileNetV2. But MobileFaceNet uses input size 112 x 112 with stride=1 instead of input size 224 x 224 with stride=2 found in MobileNetV2 [9] to overcome the degrading of accuracy in latter layers. MobileNetV1 [10] start to use depth wise separable convolutional with two types: depth wise convolutional and point wise convolution. Depth wise separable convolution reduces computation compared to traditional layers by almost a factor of k^2. MobileNetV2 [9] use inverted residuals block with linear bottleneck. A bottleneck residual block contains three layers with residual connection. Expansion factor of MobileNetV2 is 2 times bigger than MobileFaceNet. Batch normalization is used after expansion layer, depth wise convolution layer and projection layer of bottleneck residual block. A linear 1x1 kernel is used for expansion and projection layer. MobileFaceNet use global depth wise convolution layer to output a discriminative feature vector after a last convolutional layer of a face feature embedding. The detail architecture of MobileFaceNet see in Table 1.

Table 1. MobileFaceNet architecture [1]. t is the expansion factor of channel. c is the number of output channel. n is the blocked repeated time. s is the value of stride.

Input	Operator	*t*	*c*	*n*	*s*
$112^2 \times 3$	conv3x3	-	64	1	2
$56^2 \times 64$	depthwise conv3x3	-	64	1	1
$56^2 \times 64$	bottleneck	2	64	5	2
$28^2 \times 64$	bottleneck	4	128	1	2
$14^2 \times 128$	bottleneck	2	128	6	1
$14^2 \times 128$	bottleneck	4	128	1	2
$7^2 \times 128$	bottleneck	2	128	2	1
$7^2 \times 128$	conv1x1	-	512	1	1
$7^2 \times 512$	linear GDConv7x7	-	512	1	1
$1^2 \times 512$	linear conv1x1	-	128	1	1

3.4. Problems of child face recognition

Face recognition play a vital role in children life. Adult faces can better recognize than younger faces [11]. Morphological and soft tissue continue to change with age [11]. Skin textures are noticeable changes in adult faces while minimal changes are found in younger faces. As the skin's elasticity begins to degrade wrinkles form in adult ages, more notably in the eyelids, and the corners of the mouth. Fine facial lines will appear horizontally on the forehead, vertical lines between the forehead and thin lines around the outside corners of the eyes will appear. As the skull continues to change with age, the eyes appear smaller as they sink in deeper into orbits. The aging rate of adults differs heavily on the individual, which is not the case for non-adults .

4. Experiments and Results

In this section, performance of the proposed methods was evaluated on child face dataset. All images are aligned or landmark localization using 68 landmarks points based on the positions of eyebrows, eyes, nose, mouth and chin. All the proposed methods (VGG16, ResNet50, MobileFaceNet) based on Convolutional Neural Network (CNN) are implemented in python programming language with open-source neural network libraries such as Keras, tensorflow.

4.1. Datasets

Child face database (see in figure 4) contains 98 classes and each class contains from 10 to 25 images.

Images are taken under different light, pose and time variations. Different resolution cameras are used to take photos. Face distance from camera is also different. The face position towards the camera angle is vary from top to bottom and left to right side. There are different modes of faces such as laugh or without laugh, open or close eyes, open or close mouth. Photos of children were taken from schools and sport teams in MYANMAR. Some of the images of children are collected from social media in different countries. Age is range from 5 to 14 years.

Figure 4. Example of child face dataset

4.2. Experiments

The example of input child face dataset is shown in figure 4. Dlib was used for face landmark localization and alignment using 68 landmark points. Training and testing images were cropped as two type of sizes: 224 x 224 and 112 x 112. Finally, the input images were augmented such as flip, gamma correction. VGG16 and ResNet50 use 224 x 224, MobileFaceNet uses 112 x 112 as input image size. Training and testing images as numerical array are transformed and stored.

The preprocessed training dataset was fed into the proposed CNNs (VGG16 [12], ResNet50 [8], MobileFaceNet [1]). Stochastic gradient descent was used as optimizer for both VGG16 and ResNet50. Adam was used as optimizer with learning rate schedule start from 0.005 in MobileFaceNet. Since VGG16 and ResNet50 have fully connected layer as last layer, classification task is performed by fully connected layer. Unlike VGG16 and ResNet50, MobileFaceNet outputs the 128 feature vectors per face. K-Nearest Neighbours (KNN) classifier was applied as a classification layer of MobileFaceNet. KNN classifier was trained using the embeddings generated from MobileFaceNet model.

Testing data are used to test the accuracy of the model. Unlike MobileFaceNet, VGG16 and ResNet50 directly generated the classification result of input images. MobileFaceNet generated the embeddings of input images and then KNN generated the classification result when embeddings are fed into KNN. In this case, MobileFaceNet mainly perform feature extraction and KNN is for classification. In VGG16, down sampling layers used max-pooling method which kept the useful information and drop the amount of data which needed to be processed on the upper level. Max pooling method and average pooling method are used as down sampling layer in ResNet50. MobileFaceNet use global depth wise convolution layer instead of a global average pooling layer or a fully connected layer. Some dropout layers were added and softmax activation function was used at last output layer. Generally, the more convolutional steps CNNs have, the more complicated features are able to learn to recognize [13]. CNN in image classification learns simple pattern in upper layer and the latter layers learn more complex features. In this experiment, additional dense layers were added in VGG16 and ResNet50. The example of kernels of CNN is shown in figure 5.

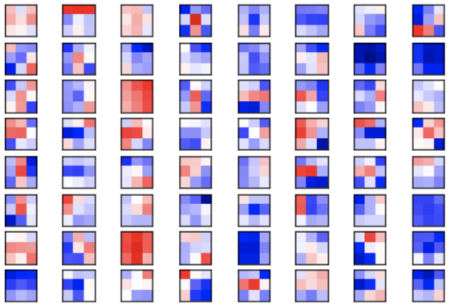

Figure 5. Example of kernels in a layer of CNN

4.3. Results

Confusion matrix was used to evaluate the proposed methods (VGG16, ResNet50, MobileFaceNet). Recall, specificity, precision, false acceptance rate, false rejection rate and accuracy of the proposed methods were calculated. The following formulas are used to evaluate the experimental results. Calculating result is shown in table 2 and 3.

Table 2. Recall, specificity and precision comparison

Model	Recall	Specificity	Precision
VGG 16	0.414	0.994	0.414
ResNet50	0.560	0.995	0.560
MobileFaceNet	0.881	0.999	0.881

Table 3. False Acceptance Rate, False Rejection Rate and accuracy comparison

Model	False Acceptance rate	False Rejection rate	Accuracy
VGG16	0.006	0.586	98.80
ResNet50	0.005	0.440	99.10
MobileFaceNet	0.001	0.119	99.75

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{specificity} = \frac{TN}{TN + FP}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{false acceptance rate} = \frac{FP}{FP + TN}$$

$$\text{false rejection rate} = \frac{FN}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Where, TP = True positive, TN = True negative
FP = False positive, FN = False negative

In three proposed methods, MobileFaceNet is the lightest weight model (See in table 4) among them. Experiment result shows that MobileFaceNet can process in shortest amount of time compared with other models (VGG16 and ResNet50) (See in table 12). MobileFaceNet achieves highest accuracy (See in table 3). Figure 6 and 7 show that accuracy and loss of VGG16 train on child face training dataset respectively. Figure 8 and 9 shows that accuracy and loss of ResNet50 train on child face training dataset respectively. Figure 10 and 11 shows the training accuracy and loss of MobileFaceNet respectively. Figure 12 shows the feature vector generated from MobileFaceNet.

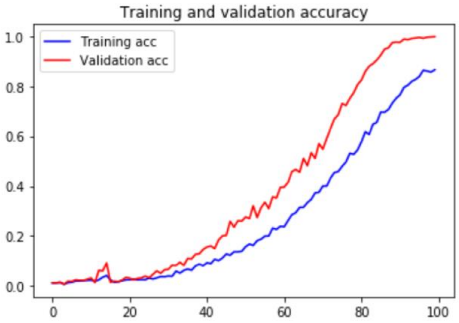

Figure 6. Accuracy of VGG16. Column represents accuracy score and row represents number of epochs.

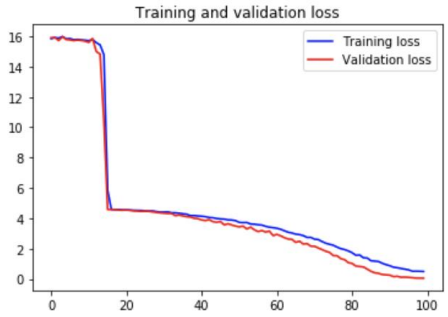

Figure 7. Loss of VGG16. Column represents loss score and row represents number of epochs.

Figure 8. Accuracy of ResNet50. Column represents accuracy score and row represents number of epochs.

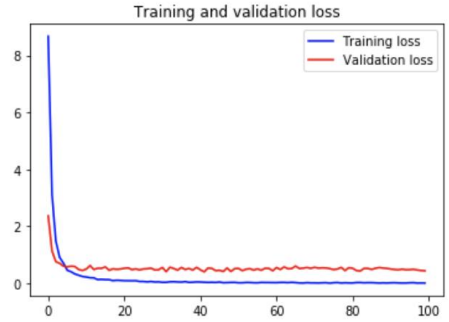

Figure 9. Loss of ResNet50. Column represents loss score and row represents number of epochs.

Table 4. Comparison of size and parameters of models

Model	Size	Parameters
VGG16	528 MB	138 million
ResNet50	98 MB	25 million
MobileFaceNet	4 MB	0.99 million

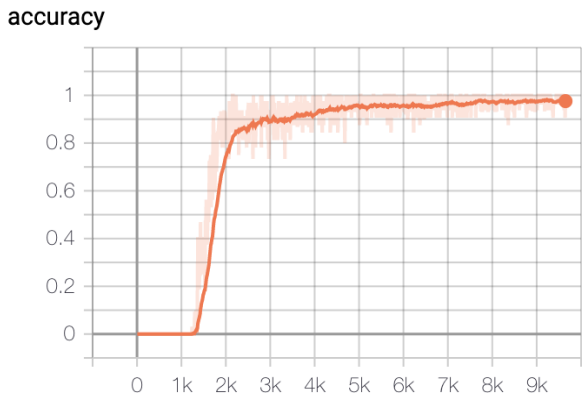

Figure 10. Accuracy of MobileFaceNet. Column represents accuracy score and row represents number of iteration steps.

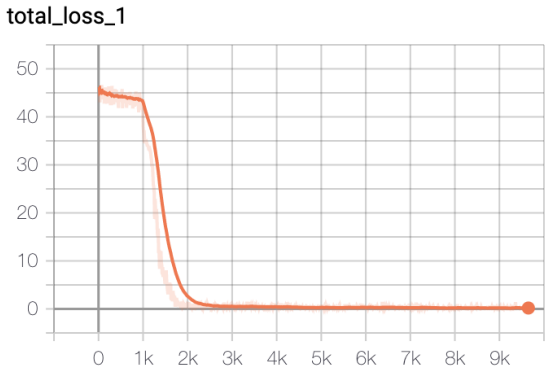

Figure 11. Loss of MobileFaceNet. Column represents loss score and row represents number of iteration steps.

Figure 12. Visualize feature vectors of child face data using 3D graph

Table 12. Overall computational time comparison

Model	Total training time	Number of steps	Batch size	Time per step
VGG16	58m 20s	5800	32	0.60s/step
ResNet50	35m	5800	32	0.36s/step
MobileFaceNet	19m 31s	9650	32	0.12s/step

5. Conclusion

In this research, experimental results of the proposed CNNs methods (VGG16, ResNet50, MobileFaceNet) were shown. Overall performances of modes are obtained using child face dataset. Among three proposed CNN methods, MobileFaceNet gives higher accuracy than others and processing time is faster than other two. VGG16 has largest model size and MobileFaceNet is a smallest model size. Two size of input images were used: 224 x 224 and 112 x 112. Overall model is run on GPU 1050Ti.

6. Acknowledgement

I would like to express my thanks to everyone who helps me for this research. I thank Codigo company for their Codigo child face dataset and other supports throughout my research.

7. References

[1] Sheng Chen, Yang Liu, Xiang Gao and Zhen Han, “MobileFaceNets:Efficient CNNSs for Accurate Real-Time Face Verification on Mobile Devices”, arXiv, 2018, pp.1-7.

[2] Mei Wang and Weihong Deng, “Deep Face Recognition: A Survey”, arXiv, 2019, pp. 1-3.

[3] Alex Krizhevsky , Ilya Sutskever , Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks “, 2012, pp. 4-5.

[4] Florian Schroff, Dmitry Kalenichenko , James Philbin , “FaceNet: A Unified Embedding for Face Recognition and Clustering” , CVPR 2015, pp. 1-4

[5] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi and Andrew Zisserman , “VGGFace2: A dataset for recognising faces across pose and age “, arxiv, 2017, pp. 4-7.

[6] Vahid Kazemi and Josephine Sullivan, “One Millisecond Face Alignment with an Ensemble of Regression Trees”, CVPR, 2014, pp. 1-2.

[7] Omkar M. Parkhi , Andrea Vedaldi , Andrew Zisserman, “Deep Face Recognition”, BMVC , 2015, pp. 5-8.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun , arxiv, 2015, pp. 2-6.

[9] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov and Liang-Chieh Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, arXiv, 2018, pp. 1-5.

[10] Andrew G. Howard Weijun Wang, Menglong Zhu Tobias Weyand, Bo Chen Macro Andreetto and Dmitry Kalenichenko

Hartwig Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Version Applications”, arXiv, 2017, pp. 1-4.

[11] Karl Ricanek Jr., Shivani Bhardwaj, & Michael Sodomsky, “A Review of Face Recognition against Longitudinal Child Faces”, sematicscholar, 2015, pp.1-4.

[12] Joseph Luttrell IV, Zhaoxian Zhou, Yuanyuan.zhang, Chaoyang Zhang, Ping Gong , Bei Yang, Runzhi Li ,” A Deep Transfer Learning Approach to Fine-Tuning Facial Recognition Models”, research gate, 2018, pp. 1-6.

[13] Patrik KAMENCAY, Miroslav BENCO, Tomas MIZDOS, Roman RADIL , “A New Method for Face Recognition Using Convolutional Neural Network “, research gate, 2017, pp. 6-8.