COGS260 Face Recognition Using Deep Learning

Yan Sun
University of California, San Diego
La Jolla, California, USA
yas108@ucsd.edu

Abstract

Nowadays face recognition is an interested topic for research study and industrial application, which leads to the development of many deep learning models for this task. The recognition could be separated into detection, feature extraction and classification. As for the detection part, Multi-task Cascaded Convolutional Networks (MTCNN) and Single Shot MultiBox Detector (SSD) are well-known detection models that could detect human face in the input images. The MobileNet technique could make SSD model light weighted. For the feature extraction part, one of the most powerful methods is FaceNet published by Google. Then the extracted feature will be used to train classifier such as Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) models. Through this process, the face recognition could be achieved either for videos or images. The FaceNet model has 99.5% accuracy on LFW benchmark and 93.9% accuracy on chinese celebrity face dataset. More tests on more complex video situation are also applied to SSD, MTCNN and FaceNet model for comprehensive detailed analysis.

1. Introduction

In order to build a model that consists of detection, feature extraction and classification function, it is significant to select the specific method for each part. In this project, based on the feasibility of different models verified by previous study, the availability of computing resource and the time limitation, pretrained Multi-task Cascaded Convolutional Networks (MTCNN) [19] and Single Shot MultiBox Detector (SSD)[8] based MobileNet[5] models are utilized for the detection part of face recognition process. The MobileNet techinique combined with SSD models could make models light weighted. Pretrained FaceNet models[11] (Inception ResNet version 1 trained by VGGFace2[2] and CASIA-WebFace dataset[18]) are used for feature extraction for cropped face image obtained from detection model. Finally, Support Vector Machine (SVM)[4] and K-Nearest Neighbor (KNN)[10] models are trained by the feature vector generated by FaceNet model for the classification. After building the models, self-made videos, images from Labeled Faces in the Wild (LFW) dataset[6] and Chinese celebrities photos obtained from Faceplusplus web crawler[3] are used for testing.

2. Method

2.1. Multi-task Cascaded Convolutional Networks

2.1.1 MTCNN Overview

Multi-task Cascaded Convolutional Networks (MTCNN) is three-stage cascaded framework for face detection and alignment[19]. The given image will be resized to different scales for image pyramid as the input for networks.

The first network is called proposal network (P-Net), which could generate candidate facial windows and their box regression vectors. The candidates generated at this stage will be calibrated and merged based on the non-maximum suppression (NMS) process. The second network is named refine network (R-Net), which will further exclude a large amount of candidates for more accurate result by calibration and NMS process. The last network is output network (O-Net), which performs similar function for ultimate face region prediction. In addition to the face boundary prediction, this network will also predict the location of five face landmarks (left eye, right eye, nose, left mouse corner and right mouse corner).

The whole process is shown in Figure 1. The architectures for these three networks are in Figure 2, 3 and 4, respectively.

2.1.2 MTCNN Preprocess

In order to train the networks of MTCNN, the image obtained from the dataset should be preprocessed for training process.

First of all, random crop process is applied to the WIDER-FACE dataset to collect training regions. These

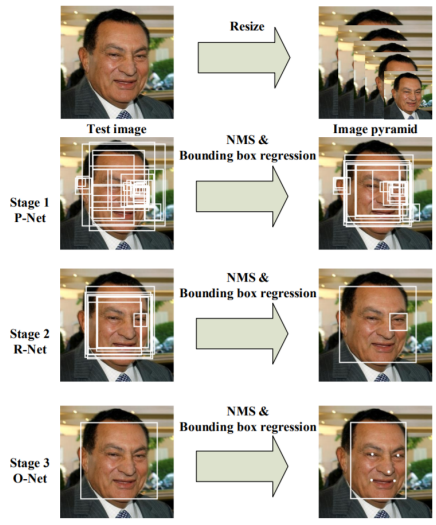


Figure 1: MTCNN Process[19]

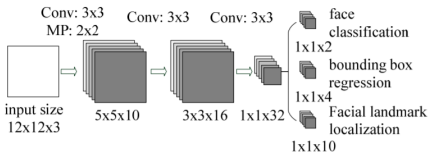


Figure 2: MTCNN Proposal Network[19]

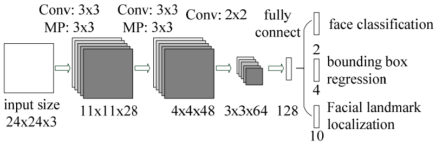


Figure 3: MTCNN Refine Network[19]

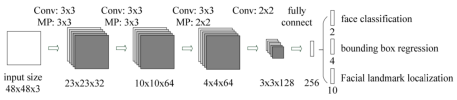


Figure 4: MTCNN Output Network[19]

regions are classified into three types based on Intersection-Over-Union (IOU) value. Regions with IOU less than 0.3, larger then 0.65 and within range [0.4,0.65] will be considered as negative sample, positive sample and part face, respectively. In addition, there is another type of samples with labeled facial landmark localization. The ratio of number of samples for these four types of samples is 3:1:1:2.

2.1.3 MTCNN Training

The prediction result consists of three separate parts for all networks, which is face classification, box regression and facial landmark localization (Figure 2, 3 and 4).

The face classification label is $y_i^{det} \in \{0, 1\}$ representing whether the sample is a face or not. Cross entropy loss is used for this part of classification problem (Equation 1) where p_i is the prediction result obtained from the network. The box regression part has four prediction values, which use Euclidean loss for the bounding boxes' left top location and boxes' height and width shown in Equation 2 where $\hat{y}_i^{box}$ is prediction location and y_i^{box} is the label location. The facial landmark localization has ten prediction values (location of left eye, right eye, nose, left mouse corner and right mouse corner), which also use Euclidean loss in Equation 3 where $\hat{y}_i^{landmark}$ is prediction location and $y_i^{landmark}$ is the label location.

After defining the loss for each part, the ultimate loss is summed up in Equation 4 where N is number of training data, α stands for the scale of different part ($\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 0.5$ for P-Net and R-Net and $\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 1$ for O-Net) and $\beta_i^j \in \{0, 1\}$ indicates whether i and j belongs to the same type or not. With all the training loss defined, Stochastic Gradient Descent (SGD) is used for training all the networks.

$$L_i^{det} = (y_i^{det} log(p_i) + (1 - y_i^{det})(1 - log(p_i))) \quad (1)$$

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2 \quad (2)$$

$$L_i^{landmark} = \left\| \hat{y}_i^{landmark} - y_i^{landmark} \right\|_2^2 \quad (3)$$

$$L = min \sum_{i=1}^{N} \sum_{j \in \{det, box, landmark\}} \alpha_j \beta_i^j L_i^j \quad (4)$$

Online hard sample mining technique is also used. Specifically, for each mini-batch, the loss for each data sample obtained from forward propagation is sorted and only top 70% of the hard samples will be used in the back propagation process, which will help strengthen the model for face classification task.

2.2. Single Shot MultiBox Detector

2.2.1 MobileNet Based Architecture

Single Shot MultiBox Detector (SSD) is another method for object detection, which could also be further used only for face detection[8]. SSD belongs to one-stage detection, using one single network to predict the box boundary and the object's category inside the box.

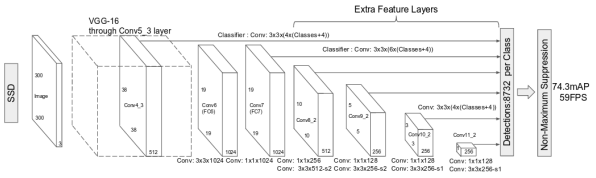


Figure 5: SSD300 architecture[8]

The basic SSD300 architecture is in Figure 5. The early layers in the architecture is named the base network. The base network could be the convolutional layer from VGG16 network[14]. The fully connected layer of original VGG16 network has been changed to convolutional layers in this architecture. In addition, extra convolutional layers are added. In order to make the total model light weighted, the base network could be replaced by MobileNet model[5] (Figure 7).

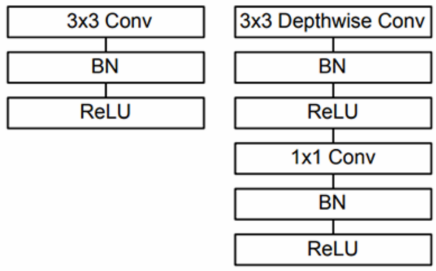


Figure 6: MobileNet Convolution Process[5]

The alternative convolution process to the standard convolution process is in Figure 6. Assume the input dimension is $D_F \times D_F$ for M channels, kernel size is $D_K \times D_K$ and output dimension is $D_G \times D_G$ for N channels. Then the standard convolution computation cost will be $D_K \times D_K \times M \times N \times D_G \times D_G$. However, if depthwise convolution plus one 1×1 pointwise convolution process is applied instead (the process at right side of Figure 6), the computation cost will be $D_K \times D_K \times M \times D_G \times D_G + M \times N \times D_G \times D_G$. The ratio of computation cost between standard convolution and alternative convolution is in Equation 5, which indicates that the computation will be 8 to 9 times less than standard convolution process when the kernel size is 3×3. This will contribute to less computation work while the reduction in model's accuracy is small[5].

$$\frac{D_K \cdot D_K \cdot M \cdot D_G \cdot D_G + M \cdot N \cdot D_G \cdot D_G}{D_K \cdot D_K \cdot M \cdot N \cdot D_G \cdot D_G} = \frac{1}{N} + \frac{1}{D_K^2} \quad (5)$$

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 300, 300, 3)	0	
conv0 (Conv2D)	(None, 150, 150, 32)	896	input_1[0][0]
conv1dw (SeparableConv2D)	(None, 150, 150, 32)	1344	activation_1[0][0]
conv1 (Conv2D)	(None, 150, 150, 64)	2112	activation_2[0][0]
conv2dw (SeparableConv2D)	(None, 75, 75, 64)	4736	activation_3[0][0]
conv2 (Conv2D)	(None, 75, 75, 128)	8320	activation_4[0][0]
conv3dw (SeparableConv2D)	(None, 75, 75, 128)	17664	activation_5[0][0]
conv3 (Conv2D)	(None, 75, 75, 128)	16512	activation_6[0][0]
conv4dw (SeparableConv2D)	(None, 38, 38, 128)	17664	activation_7[0][0]
conv4 (Conv2D)	(None, 38, 38, 256)	33024	activation_8[0][0]
conv5dw (SeparableConv2D)	(None, 38, 38, 256)	68096	activation_9[0][0]
conv5 (Conv2D)	(None, 38, 38, 256)	65792	activation_10[0][0]
conv6dw (SeparableConv2D)	(None, 19, 19, 256)	68096	activation_11[0][0]
conv6 (Conv2D)	(None, 19, 19, 512)	131584	activation_12[0][0]
conv7dw (SeparableConv2D)	(None, 19, 19, 512)	267264	activation_13[0][0]
conv7 (Conv2D)	(None, 19, 19, 512)	262656	activation_14[0][0]
conv8dw (SeparableConv2D)	(None, 19, 19, 512)	267264	activation_15[0][0]
conv8 (Conv2D)	(None, 19, 19, 512)	262656	activation_16[0][0]
conv9dw (SeparableConv2D)	(None, 19, 19, 512)	267264	activation_17[0][0]
conv9 (Conv2D)	(None, 19, 19, 512)	262656	activation_18[0][0]
conv10dw (SeparableConv2D)	(None, 19, 19, 512)	267264	activation_19[0][0]
conv10 (Conv2D)	(None, 19, 19, 512)	262656	activation_20[0][0]
conv11dw (SeparableConv2D)	(None, 19, 19, 512)	267264	activation_21[0][0]
conv11 (Conv2D)	(None, 19, 19, 512)	262656	activation_22[0][0]
conv12dw (SeparableConv2D)	(None, 10, 10, 512)	267264	activation_23[0][0]
conv12 (Conv2D)	(None, 10, 10, 1024)	525312	activation_24[0][0]
conv13dw (SeparableConv2D)	(None, 10, 10, 1024)	1058816	activation_25[0][0]
conv13 (Conv2D)	(None, 10, 10, 1024)	1049600	activation_26[0][0]
conv14_1 (Conv2D)	(None, 10, 10, 256)	262400	activation_27[0][0]
conv14_2 (Conv2D)	(None, 5, 5, 512)	1180160	activation_28[0][0]
conv15_1 (Conv2D)	(None, 5, 5, 128)	65664	activation_29[0][0]
conv15_2 (Conv2D)	(None, 3, 3, 256)	295168	activation_30[0][0]
conv16_1 (Conv2D)	(None, 3, 3, 128)	32896	activation_31[0][0]
conv16_2 (Conv2D)	(None, 2, 2, 256)	295168	activation_32[0][0]
conv17_1 (Conv2D)	(None, 2, 2, 64)	16448	activation_33[0][0]
conv17_2 (Conv2D)	(None, 1, 1, 128)	73856	activation_34[0][0]

Figure 7: MobileNet Based SSD Architecture

In Figure 7, the base network has been replaced by MobileNet with batch normalization and ReLU activation function after each separable convolution and standard convolution combination, which leads to the effective decrease of trainable parameters from 25,765,497 of previous VGG16 based SSD to 8,599,161 of current MobileNet based SSD[17]. In this way, the total SSD model becomes light weighted.

2.2.2 SSD Prediction

The layers added after the base network have the function for multi-scale feature maps detection, each layer is able to generate a fixed set of prediction result. These layers contribution could be observed at the top of Figure 5.

For example, for one added feature layer with dimension

$m \times n$ for p channels and the predicting parameter for a potential prediction is 3×3 kernel either predict for the category score or the offset relative to default box. These default boxes are generated based on the feature map via the convolution method so that the default boxes' relative positions to their corresponded cells are fixed. As for each cell in feature map, if there are k default boxes, category scores for c object classes and 4 offsets relative to original default boxes shape are needed to be predicted. In this way, k(c+4) filters are required to be applied to each cell at $m \times n$ feature map and the total number of filters for this feature map will be kmn(c+4). The aforementioned process will be applied to different feature maps of different resolutions.

The selection of default boxes is significant. Suppose there are m feature maps needed for SSD model, the scale for each map will be defined in Equation 6 where $k \in [1, m]$, $s_{min} = 0.2, s_{max} = 0.9$. At each scale there should be 6 boxes defined. Five aspect ratio will be used here $\{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$ for determining the width and height for different boxes, where width $w_k^a = s_k\sqrt{a_r}$ and height $h_k^a = s_k/\sqrt{a_r}$. When the aspect ratio is 1, one extra default box with scale $s_k^{'} = \sqrt{s_k s_{k+1}}$ will be added. In this way, there are totally six default boxes for each cell in a feature map (Figure 8). The center of each default box will be defined as $(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|})$ where f_k is the map size for k-th feature map. The aforementioned process is obtained the SSD model provided by original paper[8]. The method for defining parameters of default boxes could be different for specific datasets.

After predicting all the category scores and offset values relative to default boxes, the non-maximum suppression (NMS) process will be applied to all the prediction box. The boxes remained will be the final prediction result.

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}(k - 1) \tag{6}$$

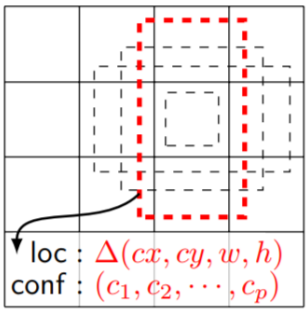


Figure 8: SSD Default Box[8]

2.2.3 SSD Training Configuration

The training loss for SSD model consists of two primary parts, which is localization loss and confidence loss (Equation 7). In this equation, the $x_{ij}^p \in 0, 1$ indicates whether the generated i-th default box matches with j-th ground truth box of category p or not judged by whether jaccard overlap of them is larger than 0.5 or not, c denotes the category, l represents the predicted box and g stands for the ground truth. N is the total number of matched default boxes. As for the localization loss in Equation 8, the specific calculation for each element for offset is defined in Equation 9. As for the confidence loss in Equation 10 where $\hat{c}_i^p = \frac{exp(c_i^p)}{\sum_p exp(c_i^p)}$, it is the softmax loss for multiple classes. With the loss function defined, the stochastic gradient descent (SGD) could be applied to update the weights in the model through back propagation.

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{7}$$

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k smooth_{L1}(l_i^m - \hat{g}_j^m) \tag{8}$$

$$\begin{aligned} \hat{g}_j^{cx} &= (g_j^{cx} - d_i^{cx})/d_i^w \\ \hat{g}_j^{cy} &= (g_j^{cy} - d_i^{cy})/d_i^h \\ \hat{g}_j^w &= log(\frac{g_j^w}{d_i^w}) \\ \hat{g}_j^h &= log(\frac{g_j^h}{d_i^h}) \end{aligned} \tag{9}$$

$$L_{conf}(x, c) = -(\sum_{i \in Pos}^{N} x_{ij}^p log(\hat{c}_i^p) + \sum_{i \in Neg} log(\hat{c}_i^0)) \tag{10}$$

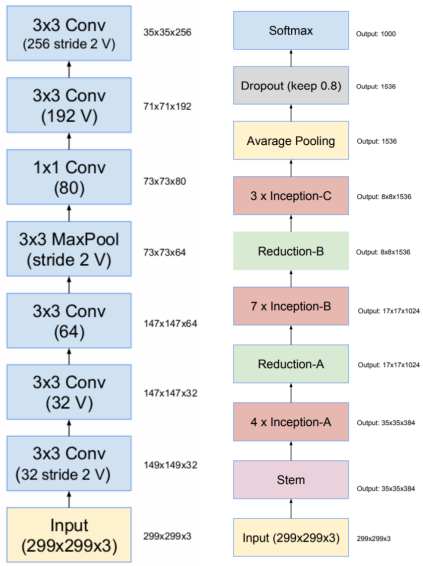
2.3. FaceNet

2.3.1 FaceNet Architecture

There exist many FaceNet models with structures in consideration of computation and accuracy trade off[11]. In this project, the pretrained Inception Resnet version 1 architecture[15] is used (Figure 9). At the final softmax process, 128 dimension is proved to be the optimal selection[11]. So a 128-dimension feature will be generated for the input image.

2.3.2 FaceNet Training

The loss function used in FaceNet training is triplet loss, which aims at minimizing the distance between anchor



(a) *Stem Architecture*. (b) *Total Architecture*.

Figure 9: Inception Resnet Version 1 Architecture[15]

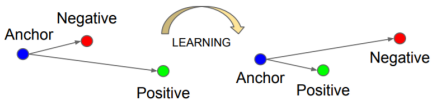


Figure 10: FaceNet Triplet Loss[11]

identity and positive identity and maximizing distance between anchor identity and negative identity (Figure 10). Assume the embedded feature is $f(x) \in \mathbb{R}^d$ and the embedding is constrained in d-dimension hypersphere, which means $\|f(x)\|_2 = 1$. The primary objective for FaceNet is the condition in Equation 11 where α is the margin used for constrain the positive and negative pairs. In this way, the expression of loss function should be Equation 12. In order to maintain the quality of training process, the selection of triplets is significant. One appropriate method to generate triplets for training process is online method. Specifically, selecting hard examples for both hard positive ($argmax_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$) and hard negative ($argmin_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$) situation in a mini-batch. About 40 faces are selected per identity per mini-batch. In addition to selecting hard examples, semi hard examples ($\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$) are also needed to avoid getting stuck in local minima[11]. With loss function defined and mini-batch used, stochastic gradient descent (SGD) will be utilzied for training FaceNet model.

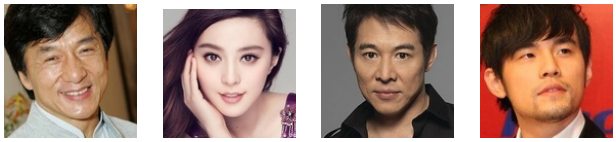
$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{11}$$

$$L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+ \tag{12}$$

3. Experiment

3.1. Datasets

Three datasets are used to test the models' performance in this project. The first dataset is LFW dataset[6], which contains more than 13,000 images of faces collected from Internet and each human face has labeled as the name of person in that image. The second dataset is Chinese celeberity face datasets, which is obtained from Faceplus-plus dataset via web crawler. The celebrities included in this datasets are based on Tencent star list and Baidu star list[16][1] and there are totally 3213 images in this dataset (Sample images in Figure 11). The last dataset is videos captured from laptop, which is used to test the processing speed, robustness and accuracy of models included in this project. All these three original data is the image contains human face without any preprocessing work.



(a) . (b) . (c) . (d) .

Figure 11: Chinese Celebrity Face dataset Sample Images

3.2. Detection Test

In this part of experiment, the detection models is tested for their processing speed and accuracy. Specifically, pretrained MTCNN model and pretrained MobileNet based SSD model are utilized for detection on chinese celebrity datasets, during which the detection model is responsible for finding the position of human face in the original images. For each model, every image in the dataset will be fed into the model once and make an output of the nunber of face detected in the picture as well as the value of frames per second (FPS) (Table 1). Note that this part experiment is run on the computing server using NVIDIA GeForce GTX 1080 Ti, which is crucial for the FPS performance.

3.3. Classification Test

This part of experiment focus on the performance of pretrained FaceNet model and corresponded KNN or SVM

Model	Accuracy	FPS
MTCNN	99.63%	21.38
MobileNet Based SSD	88.37%	76.37

Table 1: Detection Test Result

classifier. Specifically, all the original images in the datasets are processed by detection and alignment via MTCNN model (the images without detected face will be removed). All the detected parts that contain face in the original images will be cropped and aligned as the input for this part of experiment. NVIDIA GeForce GTX 1080 Ti is the device used for this part of experiment.

3.3.1 LFW Benchmark

For the classification for LFW benchmark, there exist 10 official separated subsets that have many matched pairs or mismatched pairs for validation. For these subsets, there is no need to train a corresponded classifier since the FaceNet model only need to determine whether one given image pair is anchor-positive pair or anchor-negative pair. Assume the classification threshold is $d = 1.242$. Then the result of two pretrained FaceNet models for LFW benchmark is in Table 2 where the definition of validation rate (VAL), false accept rate (FAR) could be found at Equation 13.

Model	Training Dataset	Acc avg	Acc std	VAL avg	VAL std	AUC	FAR
Inception ResNet v1	CASIA-WebFace	0.9913	0.00433	0.964	0.01636	0.999	0.00067
Inception ResNet v1	VGGFace2	0.9955	0.00342	0.986	0.00975	1.0	0.00100

Table 2: Classification Test for LFW Benchmark

$$VAL(d) = \frac{|\{(i,j) \in P_{same} \quad with \quad D(x_i, x_j) < d\}|}{|P_{same}|}$$
$$FAR(d) = \frac{|\{(i,j) \in P_{diff} \quad with \quad D(x_i, x_j) \leq d\}|}{|P_{diff}|} \tag{13}$$

3.3.2 Celebrity Dataset Validation

The Chinese celebrity dataset is split into training data and testing data with ratio of 2735:478 since there are totally 478 celebrities in the dataset. Each celebrity will have one test image. For each aligned training image in Chinese celebrity face dataset, the FaceNet model will predict its feature in 128 dimension space. With all the features generated, they will be utilized to train KNN or SVM model. Then the testing data will be input into the trained classifier for testing. If the detected person's name is identical to its labeled name then the prediction will be considered as correct. The corresponded result is in Table 3. Based on the result of this part, the rest of the experiment will use the FaceNet model trained by VGGFace2 dataset.

Index	FaceNetModel	Classifier	Training Dataset	Accuracy
1	Inception ResNet v1	KNN	CASIA-WebFace	0.7723
2	Inception ResNet v1	SVM	CASIA-WebFace	0.7897
3	Inception ResNet v1	KNN	VGGFace2	0.8988
4	Inception ResNet v1	SVM	VGGFace2	0.9396

Table 3: Classification Test for Chinese Celebrity Dataset

3.4. Comprehensive Test

In this part of experiment, NVIDIA GeForce MX150 is the used for the video capture input and Tensorflow framework. One sample video frame is in Figure 12.

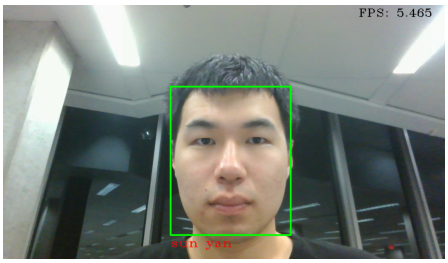


Figure 12: FaceNet Triplet Loss[11]

3.4.1 Basic FPS Test

Videos that contain no face and one face video is one short video that contains one face (Figure 13). The corresponded result obtained from video of 500 frames is in Table 4, which shows the mean and standard error of FPS.

(a) *Zero Face* . (b) *One Face*.

Figure 13: Zero Face and One Face FPS Test Sample Frame

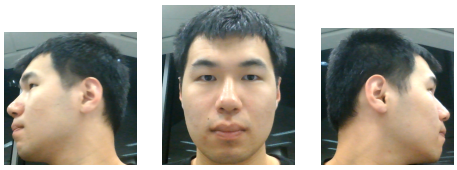
3.4.2 Side Face Test

In order to test models' robustness for 3D side face, one video contains face with different side angle (changed to

Index	Detection Model	No Face FPS	One Face FPS
1	SSD	8.95 ± 1.63	4.87 ± 0.76
2	MTCNN	9.06 ± 1.41	3.82 ± 0.56

Table 4: Basic FPS Test Result

different angles from left 90° to right 90° continuously) is fed into two models. Related sample frame is showed in Figure 5. The corresponded result is in Table 5 where detection rate is percentage of frames that have face detected and accuracy is percentage of frames that have face detected with correct prediction. Based on the observation, if the face side angle is too large the model cannot detect the human face or the classification result could be wrong.



(a) *left*. (b) *middle*. (c) *right*.

Figure 14: Side Face Test Sample Frames

Index	Detection Model	FPS	Detection Rate	Accuracy
1	SSD	6.10 ± 1.02	0.80	0.72
2	MTCNN	2.53 ± 0.29	0.77	0.73

Table 5: Side Face Test Result

3.4.3 Face Size Sensitivity Test

One image is used for making animation which make the face from very small size into normal size through zoom in operation. The animation is made by Microsoft Powerpoint and transformed into readable format for OpenCV. The same video is fed into two models for testing. Related result is in Table 6 where detection rate is percentage of frames that have face detected and accuracy is percentage of frames that have face detected with correct prediction. In addition, the test result also shows that the model will not be able to detect the face if the face size is too small.

3.4.4 Rotate Robustness Test

One image contains one face is used to create animation that rotates the image continuously for 360°. The same animation video is fed into two models for testing the rotate robustness. Related result is in Table 7 where detection rate is percentage of frames that have face detected and accuracy is percentage of frames that have face detected with correct prediction.

Index	Detection Model	FPS	Detection Rate	Accuracy
1	SSD	7.69 ± 2.19	0.84	0.84
2	MTCNN	2.75 ± 0.47	0.83	0.83

Table 6: Face Size Sensitivity Test

Index	Detection Model	FPS	Detection Rate	Accuracy
1	SSD	6.78 ± 2.56	0.45	0.40
2	MTCNN	7.06 ± 3.33	0.30	0.30

Table 7: Rotate Robustness Test Result

3.4.5 Dark and Light Test

One image contains one face is used to create animation which makes the image from dark to light. The same animation video is fed into two models for testing the sensitivity for dark and light. Related result is in Table 8 where detection rate is percentage of frames that have face detected and accuracy is percentage of frames that have face detected with correct prediction.

Index	Detection Model	FPS	Detection Rate	Accuracy
1	SSD	6.88 ± 2.08	0.75	0.7
2	MTCNN	6.53 ± 3.09	0.75	0.55

Table 8: Dark and Light Test Result

4. Discussion

4.1. Dataset Performance

For the LFW dataset, both FaceNet models trained by VGGFace2 or CASIA-Webface dataset could get good prediction accuracy for the pair prediction, which is identical to the result of original paper[11]. However, if these models are applied to Chinese celebrity face dataset, the prediction result cannot reach as high as 99% (Table 1). The reasons contribute to this could be explained in two main aspects.

First, the LFW benchmark is a pair prediction that only classify whether the distance between two faces is less than threshold value or not, which does not guarantee the ultimate prediction for whose face is in the given image. This makes the prediction more easy since it is only a binary classification problem.

Second, as for the Chinese celebrity face dataset, the models are responsible for predicting the person inside the image. Some celebrities may looks like same due to makeup, sunglasses or other external factors. In these situations, the distance for faces from these celebrities could be within the threshold but the ultimate identification process may not be reliable. So this kind of accuracy could be lower than the accuracy of pair prediction. The 4th model in Table 1 has the best classification result for aligned face so the rest test is based on this model. In addition, as for the FPS performance, it is obvious that MobileNet based SSD model has higher speed to process the input images but has lower ability to find the face in the images of Chinese celebrity face dataset. The MobileNet makes the model less weighted but sacrifice the detection accuracy, which is similar phenomenon to what the paper conclude[5].

4.2. Video Performance

The FaceNet model trained by VGGFace2 dataset combined with SSD or MTCNN detection model have been tested for various aspect in Section 3.4. At this part the GPU has been changed to NVIDIA GeForce MX150 so the FPS value is not as high as what it is in Section 3.2.

As for the FPS performance, SSD detection model has higher speed for most of the test tasks. Sometimes MTCNN detection model may have slightly better speed such as test in Section 3.4.1 and 3.4.5. These results proved that MobileNet indeed optimizes the model, making it faster to process the input information.

As for the accuracy performance for different parts of test, the results show that both SSD and MTCNN model remained optimization.

First, as for the side face test, both models show reasonable robustness for the side face from left 90° to right 90° with detection rate 80%. However, the classification accuracy drops to 70%, which is a huge decrease compared to the accuracy in Table 3. The side face problem is hard to deal since when the side angle is too large some important face landmark will not be observed in the video. In order to solve this problem, Deep Face Feature and related face alignment method [7] could be able to capture more global structure information of human face improve the robustness for 3D rotation situation.

Second, as for the face size sensitivity test, the result shows that SSD and MTCNN model has similar performance to detect small face. In addition, once both models detect the face in the image, the accuracy for classification is 100% , which shows that these models have powerful function for classification once the face is detected. The small size object detection is one of the primary problem nowadays. In order to improve the small object detection performance, Class Activation Mapping method[20] could be used for localization detection. Another fine-tuned VGG-16 model could also be utilized for more detailed outline information detection[9].

Third, the rotate robustness test gives the poor result for the rotated image since both SSD and MTCNN models have detection rate and classification accuracy lower than 50%, which means that there is lots of work remained to be done for solving rotated face problem. Data augmentation is one feasible method to solve this problem with enough given data and computing resources. In Conference on Computer Vision and Pattern Recognition (CVPR) 2018, progressive calibration networks (PCN) was proposed for solving rotation-invariant faces in given image, which also has promising performance compared to data augmentation and divide-and-conquer strategy[13].

Finally, the dark and light test shows that SSD models and MTCNN model have similar performance for detection rate on dark image or light image. However, the accuracy for MTCNN is lower than SSD model. The varying illumination problem is another challenge that attracts lots of attention. Original Pixel Preservation Model (OPPM) [12] could be an alternative algorithm that helps prerpocess the input image and alleviate the influence caused by variance of illumination.

5. Conclusion

In this project, MobileNet based Single Shot Multibox Detector (SSD) model and Multi-task Cascaded Convolutional Networks (MTCNN) model are used as detection model, Google's FaceNet model is used as feature extraction model, SVM and KNN model are used as classifier for face recognition problem. For the detection problem, both MTCNN and SSD model could reach 100% detection rate for LFW images while 99.63% and 88.37% detection rate for Chinese Celebrity face dataset, respectively. For the Labeled Faces in the Wild (LFW) benchmark, the FaceNet could reach 99.5% accuracy for the triplet loss pair analysis. Then the comprehensive video test shows that both detection model remained optimized for the video of complex situation (3d side angle, rotate and darkness) and MobileNet based SSD model performs better (detection rate, classification accuracy and FPS) than MTCNN model for most of complex situation.

More future work may focus on improving model's robustness for aforementioned complex situation.

References

[1] Baidu. Baidu star list, http://news.baidu.cn/f/. 5

[2] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. *arXiv preprint arXiv:1710.08092*, 2017. 1

[3] Faceplusplus. Faceplusplus web crawler. Technical report, https://github.com/qibinlou/FacePlusPlus-Stars-Library-Images-Crawler. 1

[4] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998. 1

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 3, 8

[6] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. 1, 5

[7] B. Jiang, J. Zhang, B. Deng, Y. Guo, and L. Liu. Deep face feature for face alignment and reconstruction. *CoRR*, abs/1708.02721, 2017. 8

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2, 3, 4

[9] M. Menikdiwela, C. Nguyen, H. Li, and M. Shaw. Cnn-based small object detection and visualization with feature activation mapping. 8

[10] L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009. 1

[11] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 4, 5, 6, 7

[12] J. H. Shah, M. Sharif, M. Raza, M. Murtaza, and Saeed-Ur-Rehman. Robust face recognition technique under varying illumination. *Journal of Applied Research and Technology*, 13(1):97 – 105, 2015. 8

[13] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen. Real-time rotation-invariant face detection with progressive calibration networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2295–2303, 2018. 8

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 4, 5

[16] Tencent. Tencent star list, http://ent.qq.com/c/all_star.shtml/. 5

[17] J. Wang. Single shot multibox detector (ssd), https://magi003769.github.io/2018/02/12/ssd/. 3

[18] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 1

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 1, 2

[20] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015. 8