Final Report of Final Year Project

HKU-Face: A Large Scale Dataset for Deep Face Recognition

HaoYu Li
3035141841

COMP4801 Final Year Project
Project Code: 17007
Abstract

Current research and development of face recognition mainly focus on three perspectives, the scale and purity of facial dataset, the effectiveness of loss functions and employment of effective backbone network structures[2, 6, 9, 13, 19, 34]. The scale and labelling errors of training dataset is usually the problem encountered by many researchers in institutions for face recognition research. This project introduces a dataset filtering and construction procedures for solving the problem with less human effort. Pre-trained face recognition models like Sphereface[19] and Facenet[28] are employed to extract deep features of face images. Then these representational features are used for noisy image filtering. Impressive result of automatic filtering and purity level have been achieved compared with the raw dataset. Except for the self-constructed dataset, another filtered dataset named CASIA-VGG merged from CASIA-Webface[40] dataset and VGG[23] Dataset is also researched and analyzed in this project.
Acknowledgments

I want to express my sincere thanks to Prof. Yu for his supervision and help in offering computational resources. I also want to thank Weifeng Ge for his instructive guide through this project. I would also want to thank all my classmates who help us in our project.
Contents

Abstract

Acknowledgements

Contents

List of Figures

List of Tables

1 Introduction 1
   1.1 Background ......................................................... 1
   1.2 Problem Definition .............................................. 2
   1.3 Objectives ........................................................... 3
   1.4 Outline of the Report ............................................ 4

2 Literature review 5
   2.1 Theoretical Background ........................................... 5
   2.2 Dataset construction overview .................................. 7
   2.3 Face Recognition Models ......................................... 7
   2.4 Benchmark for Face Recognition ................................ 14

3 Methodology and Approach 16
   3.1 Original Dataset Construction ................................... 16
   3.2 Dataset Filtering .................................................. 17
   3.3 Models in Automatic Filtering ................................... 19
   3.4 CASIA-VGG dataset ............................................... 20
   3.5 Evaluation of Face Recognition Performance ................... 22

4 Dataset Construction 23
   4.1 Construction of scratched face dataset ....................... 23
   4.2 Refinement in preprocessing CASIA-VGG dataset ............... 29

5 Face Recognition Experiments 32
   5.1 Result reproduction ............................................... 32
List of Figures

1.1 Illustration of dataset without filtering (in one class). ....................... 3

2.1 Triplet loss for classifying different classes ................................. 8

2.2 The sub-figure(a) shows 2D and 3D cases for angular softmax loss margin, where the orange color represents the class 1 and green color represents class 2, which are not only classified but also separated by a margin. The corresponding parameters are discussed in SphereFace paragraph. The right-upper graph illustrates the A-softmax margin effect when selecting $m = 4$ after projecting features into 3. The right-lower picture shows the distribution of angles under this setting ............................. 10

2.3 Weight and feature normalization are both implemented in arcface model. This setting is compared by the author to have a better performance[9] ................................................. 11

2.4 Traditional residual block used ReLU as activation function without batch normalization. The newly proposed structure applied parametric ReLU and batch normalization to increase learning efficiency ................................................. 13

2.5 Left image shows ReLU function and right image shows the PReLU. PReLU has another parameter $a_i$ to learn in back propagation process ................................................. 14

2.6 Decision boundary comparisons for softmax, sphereface, cosface and arcface. For softmax and Weight normalized softmax, only classification task is learned without enforcing a margin. The loss function does not have a good performance in solving open-set problem. The four rest loss functions all utilize parameter $m$ to enforce a margin, which makes the learned features more distinctive and separable ................................................. 15

2.7 Visualization for class margins of softmax, sphereface, cosface and arcface. As we can observe, adding parameter $m$ can enforce a margin between two classes ............................. 15

3.1 Illustration of occupational distribution of celebrity in MS-Celeb-1M[2] ........ 17

3.2 Five network structures experimented in sphereface. They are all ResNet based with weight normalization. The return is 512 hyper dimensional features that will be used for face recognition ................................................. 20

3.3 Complete training-testing procedures in sphereface. In the training phase, features from Fully connected layer will be used in Angular-softmax loss function for updating network weights in backpropagation. In test phase, test images will be passed into the network and deep features will be extracted for verification or recognition purpose ................................................. 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Noisy images existed in merged dataset</td>
</tr>
<tr>
<td>3.5</td>
<td>Duplicated images of same identity</td>
</tr>
<tr>
<td>3.6</td>
<td>Gray and color images coexist for same identity</td>
</tr>
<tr>
<td>4.1</td>
<td>Illustration of clustering problem. Different colors represent different people</td>
</tr>
<tr>
<td>4.2</td>
<td>Purity distribution of benchmark set using Facenet generated features</td>
</tr>
<tr>
<td>4.3</td>
<td>Image in CASIA-VGG mtcnn aligned</td>
</tr>
<tr>
<td>4.4</td>
<td>Image in CASIA-Webface mtcnn aligned after cropping</td>
</tr>
<tr>
<td>4.5</td>
<td>Sample raw image without similarity transformation and resizing</td>
</tr>
<tr>
<td>4.6</td>
<td>Image after similarity transformation and resizing</td>
</tr>
<tr>
<td>5.1</td>
<td>The 36-layer residual based network used in our experiments. For instance, [3x3,64] means one convolutional layer with 64 filters and 3x3 kernel size. S2 means stride equal to 2 and the 512 in bottom means the embedding feature size</td>
</tr>
<tr>
<td>5.2</td>
<td>Raw images for an identity</td>
</tr>
<tr>
<td>5.3</td>
<td>Bounding box and landmarks calculated by mtcnn for the same identity</td>
</tr>
<tr>
<td>6.1</td>
<td>Two persons in the left image are both called Steve McQueen and the identities in the right image are both called Michael Douglas. These two identities are in the wiki celebrity list</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Face recognition datasets</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Newly proposed residual block. In Batch Normal layer explanation, i.i.d means</td>
<td></td>
</tr>
<tr>
<td></td>
<td>independently identically distributed. This new residual block structure incorporates batchnorm, PReLU to reduce training time and avoid degradation.</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Statistics of benchmark for auto filtering</td>
<td>25</td>
</tr>
<tr>
<td>4.2</td>
<td>Purity level of benchmark dataset</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>Facenet pre-trained model, Sphereface pre-trained model and self-trained model comparison in image filtering performance</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>Compare the reproduction experiment results of Sphereface and Arcface with results in paper.</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Comparison of model performance being trained using randomly picked 1k, 5k id and manually purified 1k id dataset.</td>
<td>35</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of model performance trained using randomly picked 782, 1k, 5k, manually purified 782 id from HKU-Face and 782, 1k, 5k randomly picked id from of CASIA-VGG.</td>
<td>36</td>
</tr>
<tr>
<td>5.4</td>
<td>Networks are trained in dataset with different preprocessing settings. MTCNN is used for face detection and alignment. Small margin, no margin and similarity transformation(ST) settings are set as experiment target. Sphereface used 36-layer residual network and Arcface used 100-layer improved version of resNet as discussed in chapter 2.</td>
<td>38</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Face recognition usually means identifying or verifying person from digital image or video source. Face recognition is a representative subdivision for the application of deep learning. Deep learning models can extract representational features of human faces that have lower intra-personal distance and higher inter-personal distance, which is the essential point for face recognition. Traditional CNN-based face recognition research trained network structure on labeled face dataset and select an intermediate bottleneck layer as identities’ feature representation [38, 32]. In recent years, Human beings have been beaten in some academic benchmarks after the emergence of state-of-the-art models such as FaceNet, Sphereface and Arcface [28, 19, 9]. Face recognition has also been applied in various areas like electronic devices unlocking in smart phone and criminal location. Current research has two emerging trends as softmax-like training pipeline such as Sphereface [19], Cosface [33], Arcface [9] and triplet loss-like training pipeline like contrastive loss [3], facenet(triplet loss) [28] and center loss [35].
CHAPTER 1. INTRODUCTION

1.2 Problem Definition

In spite of the outstanding achievements of Face Recognition so far, several limitations continue to exist. Firstly, current academic models are quite sensitive to variations like illumination, different perspective of faces, and potential obfuscation thus they cannot perform well in front of a large number of test cases. Secondly, the industrial companies have to solve the issue of adversarial samples which can be easily designed to crack the face authentication system. As a research project, we focus on the potential mitigation strategies on the first one.

The issue of sensitivity results from the nature of the current models. Almost all of the models are trained by supervised learning which uses dataset annotated by human beings. Two following drawbacks contribute to the problem of sensitivity: (1) limited size of dataset due to limited human effort; (2) accuracy problem resulted from human perceptual bias.

Parkhi, Vedaldi, Zisserman, et al.\cite{24} discussed the limitation on the small training dataset, showing that giant companies hold private face database with larger size of data, while other research institution could only get access to public but much smaller database for training such as CASIA-WebFace\cite{40}, which acts like a barricade to even higher performance (See the comparison in Table 1.1). However, the largest private database is still small compared with the size of images on the Internet because it can be inferred impossible to annotate face database in trillion size.

In Microsoft paper\cite{2}, the second problem has been addressed in an indirect manner by fault-tolerance classifier models. Therefore, training dataset in MS-Celeb-1M\cite{2} is only obtained from popular search engine without any filtering or labeling checking. As Figure 1 shows, these images should belong to one identity but obviously, there are men’s and women’s faces and even a non-face image. The performance of the classifier is highly likely to be better if the dataset were purified.
CHAPTER 1. INTRODUCTION

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Availability</th>
<th>identities</th>
<th>images</th>
</tr>
</thead>
<tbody>
<tr>
<td>IJB-A</td>
<td>public</td>
<td>500</td>
<td>5712</td>
</tr>
<tr>
<td>LFW</td>
<td>public</td>
<td>5K</td>
<td>13K</td>
</tr>
<tr>
<td>YFD</td>
<td>public</td>
<td>1595</td>
<td>3425 videos</td>
</tr>
<tr>
<td>CelebFaces</td>
<td>public</td>
<td>10K</td>
<td>202K</td>
</tr>
<tr>
<td>CASIA-WebFace</td>
<td>public</td>
<td>10K</td>
<td>500K</td>
</tr>
<tr>
<td>VGG-Face</td>
<td>public</td>
<td>2622</td>
<td>982K</td>
</tr>
<tr>
<td>VGG2-Face</td>
<td>public</td>
<td>9131</td>
<td>3.31M</td>
</tr>
<tr>
<td>MS-Celeb-1M</td>
<td>public</td>
<td>100K</td>
<td>about 10M</td>
</tr>
<tr>
<td>Facebook</td>
<td>private</td>
<td>4K</td>
<td>4400K</td>
</tr>
<tr>
<td>Google</td>
<td>private</td>
<td>8M</td>
<td>100-200M</td>
</tr>
</tbody>
</table>

Table 1.1: Face recognition datasets

Motivated by the aforementioned limitation in dataset size, this project is intended to tackle the issue from the perspective of dataset construction. At first, this project tried to build a large-scale dataset, called HKU-Face, from extracting images online to image dataset filtering. A general face dataset development procedure will be proposed as well as a dataset and trained models for the evaluation. The procedure should be able to construct large-scaled face datasets efficiently with controllable noise. Then several advanced face recognition models were researched through result reproduction and new dataset training procedures.
1.4 Outline of the Report

In the rest of this interim report, several key points are discussed. In Chapter 2, Literature review introduces the theoretical background with essential prerequisite of our project. Methodology part in Chapter 3 describes how the inspiration from previous work transforms to our key approach and their implementation details. Data construction details as well as the research on CASIA-VGG dataset will be illustrated in chapter 4. Detailed experiments, including result reproduction and convolutional neural network training process, will be discussed in chapter 5. Chapter 6 will conclude the limitations for the constructed dataset. Then chapter 7 will give a summary to this report and provide the future prospect.
Chapter 2

Literature review

In this chapter, several face recognition models will be reviewed in detail and their corresponding loss functions. Data construction overview will also be given and face recognition benchmarks will be introduced. Theoretical background presents the basic task definitions of Face Recognition and Metric Learning. Then methods related to dataset construction, face recognition models and facial benchmarks are elaborated and several existing works are reviewed in details.

2.1 Theoretical Background

Face Recognition

Essentially, face recognition is a specific verification problem, which requires model to evaluate the similarities between query image sets and gallery sets. If we define $I^d = I_1, I_2, ..., I_n$ to be the set of all persons in image set. Denote that the set of query images to be $Q^d = Q_1, Q_2, ..., Q_n$. Then the problem can be separated into two sub-problems, closed-set problem and open-set problem. Closed set face recognition is to find model $f(Q_x)$ mapping from testing image to existing identity, which is to find what identity this image belongs to.
A more difficult task called open-set face recognition, is to remove the constraints that every test image must belong to one of existing identities. Without prior knowledge about whether query image belongs to gallery image set or not, it needs to be determined that if the query image belongs to anyone in our dataset, i.e. that Whether the model has ”observed” the person before or not.

**Multi-class classification**

Closed set problem is a kind of multi-class classification problem. For each query image $Q_i$, the model should allocate $Q_i$ to the correct identity. This is a classical and prevalent problem in machine learning. Specifically, classification models optimize their parameters so that they can map image to the correct identities. Models ranging from SVM[30], random forest[18], basic perceptrons[27] to deep learning can all perform this task with different performance and training time.

**Metric Learning**

Open set problem is more challenging due to its nature which is demanding of general discriminative features. Metric learning shows its superior in this problem. Instead of directly mapping from image instances to identities, metric learning model firstly maps the input to a general feature $h$ which theoretically lies in a hyperspace manifold[17]. Then training a discriminator $D$ to accomplish the task

$$D(h_i, h_j) = \begin{cases} 
0, & \text{if } h_i, h_j \text{ do not belong to the same identity} \\
1, & \text{if } h_i, h_j \text{ belong to the same identity}
\end{cases}$$

The verification task involving a latent space empirically yields better performance as Parkhi, Vedaldi, Zisserman, et al.[24] shows, especially for deep metric learning which harnesses deep neural network to handle the latent feature extraction phase.
2.2 Dataset construction overview

Our project was motivated by [24], in which a general procedure of dataset construction is proposed. It provides a well-designed framework to build and evaluate a facial dataset as well as meaningful heuristics for image purification. The data construction part of our project borrows the framework but we try to utilize more advanced feature extraction models and explore filtering automatically. Also we will build new benchmark specifically for the evaluation of the automatic filtering performance which receive less consideration in the literature.

As Lapuschkin et al. [15] shows, the fisher vector used in [24] generally tends to rely more on contextual features rather than overall structural features which have been explored better using deep learning model. This observation shows the feasibility of replacing fisher vectors by deep neural networks. The loop of automatic filtering and human purifying process indicates the potential potency of a iterative process of handling filtering process. Apart from the iterative collaboration of computer and human effort, purification by iterative automatic filtering and model fine-tuning could be possibly useful.

2.3 Face Recognition Models

Triplet Loss model

Facenet [28] directly maps image instances into Euclidean space which provides convenience for applying direct clustering on the features of faces. It utilizes an innovative loss function called triplet loss, which calculates the difference using three samples: \( Q_i^a, Q_i^p \) and \( Q_i^n \). \( Q_i^a \) is the query image, which is also called anchor, from a random person. \( Q_i^p \) and \( Q_i^n \) are the image from the same person(positive) and image of another person(negative) respectively. [28] tries to maintain small distance metric among same classes and large distance metric
Figure 2.1: Triplet loss for classifying different classes

among different classes. As figure 2.1 shows, the learning procedure forms three-sample pairs
and generates loss value from their distances.

\[ L_{Triplet} = \sum_{i} \| f(Q^a_i) - f(Q^p_i) \|^2 - \| f(Q^a_i) - f(Q^n_i) \|^2 + \alpha \]

A more precise mathematical expression for triplet loss is given below. Aside from what
have been mentioned above, Triplet Loss tried to solve open-set problem by adding an extra
positive \( \alpha \) to its loss function. The aim is not only classifying identities but also enforcing a
margin between different identities.

**SphereFace**

Sphereface\[19\] derives an angular expression from traditional softmax loss function\[21\], which
leads to discriminative features. Softmax loss is one kind of decision criteria that gives
posterior probabilities for each sample. A general softmax loss posterior probability formula
and softmax loss function are given in formula below.

\[ P_i = \frac{exp(W^T_i x + b_i)}{\sum_j exp(W^T_j x + b_j)} \]

\[ L = \frac{1}{N} \sum_i -log\left( \frac{exp(W^T_i x + b_i)}{\sum_j exp(W^T_j x + b_j)} \right) \]
It means that the probability of the input data $x$ belonging to class type $i$ equals $P_i$. $W_i^T$ is the i-th column for the weight matrix $W$ in softmax layer and $b_i$ is the i-th entry for the bias vector. For instance, in binary case, the probability for input to be class 1 is

$$P_1 = \frac{e^{W_1^T x + b_1}}{e^{W_1^T x + b_1} + e^{W_2^T x + b_2}}$$

SphereFace further utilizes the property of angular expansion by decomposing $W_i^T x + b_1$ term into $||W_i|| \times ||x|| \times \cos(\theta_i) + b_i$. To simplify calculation, it normalizes weights ($||W_i^T|| = 1$) and initializes bias to zero ($b_i = 0$). Using the same margin setting technique, SphereFace adds another pre-defined parameter $m$ to its degree term $\theta_i$. $\theta_i$ controls the size of angular margin and the author derives a function $\phi(\theta_{yi,i}) = (-1)^k \cos(m\theta_{yi,i}) - 2k$ to make it optimizable. $\theta_{yi,i}$ is the angle between $W_{yi}$ and $x_i$ with $K \in [0, m - 1]$. Omitting some details here, we obtain the final expression of angular softmax loss (A-Softmax) as shown below. The 2-Dimension and 3-Dimension visualizations of SphereFace are shown in Figure 2.2.

$$L_{ang} = \frac{1}{N} \sum_i -\log\left(\frac{\exp(||x_i|| \phi(\theta_{yi,i}))}{\exp(||x_i|| \phi(\theta_{yi,i})) + \sum_{j \neq y_i} \exp(||x_i|| \cos(\theta_{j,i}))}\right)$$

In [19], sphereface is trained using 64-layer residual-block structure and sets the margin parameter $m$ through trial and error. With batch size of 128 and 28K iterations, it achieves 99.42% accuracy in LFW dataset.

**Cosface**

Cosface[33] is another work by Wang et al., who are inspired by sphereface. They also utilize the angular expansion of softmax loss and add margin to get more separable features. They name the new loss function as large margin cosine loss (LMCL). Compared with angular-softmax loss used in sphereface, LMCL adds the margin parameter outside the co-
Figure 2.2: The sub-figure (a) shows 2D and 3D cases for angular softmax loss margin, where the orange color represents the class 1 and green color represents class 2, which are not only classified but also separated by a margin. The corresponding parameters are discussed in SphereFace paragraph. The right-upper graph illustrates the A-softmax margin effect when selecting $m = 4$ after projecting features into 3. The right-lower picture shows the distribution of angles under this setting.
CHAPTER 2. LITERATURE REVIEW

Figure 2.3: weight and feature normalization are both implemented in arcface model. This setting is compared by the author to have a better performance\cite{9} sine function which makes it easier to implement in code. The formulae for LMCL is given below:

\[ L_{lmc} = \frac{1}{N} \sum_i -\log \frac{\exp s(\cos(\theta_{y,i}) - m)}{\exp s(\cos(\theta_{y,i}) - m) + \sum_{j \neq y_i} \exp s(\cos(\theta_{j,i}))} \]

Since in the testing stage, the face recognition score of face pair is usually calculated according to cosine similarity of feature vectors. Cosface normalizes weight parameter and feature vectors to emphasize the training effect towards the angular difference. This is different with sphereface which only normalizes weight parameters. Figure 2.3 visualizes the normalization process.
Arcface

Arcface further extends the idea of angular-softmax loss expression and obtains new benchmark result in LFW, CFP-FP and AgeDB-30 dataset. This work mainly contributes to three aspects of face recognition including data preparation, network structure and loss function.

**Data Preparation** At first, data refinement is conducted to the MS-Celeb-1M dataset in automatic and manual way. They automatically remove images, of which feature vectors are too far away from the class center. Then images with distance greater than a threshold are manually reserved or discarded. The size of dataset shrinks from 100K identities with 10 million images to 85K identities with 3.8 million images.

**Efficient Network Structure** Secondly, they attempt different model structures and new residual unit settings for training face recognition model. A newly proposed residual unit structure and original basic residual block are compared in figure 2.4.

Batch Normalization and PReLU are added into the block through experience. Batch normalization was typically used to keep input feature satisfy the same distribution at each layer. With deeper network structures, training neural network becomes harder and the loss may be divergent. After applying batch normalization operation between convolutional layers, the network can be easier to converge and learning efficiency is improved. Parametric Retified Linear Unit was first mentioned in and was designed for deep convolutional neural network training process. As shown in figure 2.5, PReLU has another parameter to learn during the training process. Previous experiments show that parametric Retified Linear Unit made large-scale network easier to train and the author used this structure to beat ILSVRC 2014 winner by more than 25% improvement. In Arcface training process, 100-layer ResNet-like network structure was trained, which contains very deep layers and a large amount of parameters. To avoid divergence of loss and reduce training time, the newly proposed residual unit adopted both batch norm and PReLU, which is
CHAPTER 2. LITERATURE REVIEW

(a) Residual unit

Figure 2.4: Traditional residual block used ReLU as activation function without batch normalization. The newly proposed structure applied parametric ReLU and batch normalization to increase learning efficiency

reasonable. A detailed structure for newly proposed residual block unit is given in Table 2.1 and the explanations to each component are also analyzed. By adopting this new residual unit into network, the 100-layer ResNet network with Arcface loss achieved 99.83% as shown in [9].

Arcface Loss function Since the angular expansion of softmax proposed by Sphereface [19] has made improvement to the FR problem, more researchers have put their attention to this new kind of loss function. Compared with Cosface [33], Arcface seems to be a subsequent progress which moves margin parameter $m$ into cosine function. Explicit expression of Arcface loss function is shown in figure 2.6. A binary comparison among traditional softmax, Sphereface [19], Cosface [33] and Arcface [9] is also provided in figure 2.6. The decision bound-
Figure 2.5: Left image shows ReLU function and right image shows the PReLU. PReLU has another parameter $a_i$ to learn in back propagation process.

<table>
<thead>
<tr>
<th>Layer Details</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Input</td>
</tr>
<tr>
<td>Batch Normalization</td>
<td>Ensure i.i.d, Boost learning speed</td>
</tr>
<tr>
<td>3x3 Conv, stride=1</td>
<td>Normal convolutional layer</td>
</tr>
<tr>
<td>Batch Normalization</td>
<td>Ensure i.i.d</td>
</tr>
<tr>
<td>Parametric ReLU</td>
<td>Reduce learning time</td>
</tr>
<tr>
<td>3x3 Conv, stride=2</td>
<td>Normal convolutional layer</td>
</tr>
<tr>
<td>Batch Normalization + Input</td>
<td>Normal residual operation. Avoid degradation</td>
</tr>
</tbody>
</table>

Table 2.1: Newly proposed residual block. In Batch Normal layer explanation, i.i.d means independently identically distributed. This new residual block structure incorporates batchnorm, PReLU to reduce training time and avoid degradation.

2.4 Benchmark for Face Recognition

In the literature, Labeled Face in the Wild (LFW)\cite{5,6} is a classical dataset for evaluation of face recognition performance. LFW contains 5,000 identities and only 13,000 images. The small size of its scale does not allow it to become an effective training dataset. Cur-
CHAPTER 2. LITERATURE REVIEW

<table>
<thead>
<tr>
<th>Loss Functions</th>
<th>Decision Boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td>((W_1 - W_2)x + b_1 - b_2 = 0)</td>
</tr>
<tr>
<td>W-Norm Softmax</td>
<td>(|x| (\cos \theta_1 - \cos \theta_2) = 0)</td>
</tr>
<tr>
<td>SphereFace</td>
<td>(|x| (\cos m\theta_1 - \cos \theta_2) = 0)</td>
</tr>
<tr>
<td>F-Norm SphereFace</td>
<td>(s(\cos m\theta_1 - \cos \theta_2) = 0)</td>
</tr>
<tr>
<td>CosineFace</td>
<td>(s(\cos \theta_1 - m - \cos \theta_2) = 0)</td>
</tr>
<tr>
<td>ArcFace</td>
<td>(s(\cos(\theta_1 + m) - \cos \theta_2) = 0)</td>
</tr>
</tbody>
</table>

Figure 2.6: Decision boundary comparisons for softmax, sphereface, cosface and arcface. For softmax and Weight normalized softmax, only classification task is learned without enforcing a margin. The loss function does not have a good performance in solving open-set problem. The four rest loss functions all utilize parameter m to enforce a margin, which makes the learned features more distinctive and separable.

Figure 2.7: Visualization for class margins of softmax, sphereface, cosineface and arcface. As we can observe, adding parameter m can enforce a margin between two classes.

Recent research trend usually regards it as a benchmark for model evaluation. Mega-face\cite{13} is a generally new and more challenging benchmark because of more variations and distractors. The training set contains 672,057 identities and 4,753,520 images. Though the dataset size is very large, the dataset purity is not high and also very challenging to train. Recently MS-Celeb-1M\cite{2} has been proposed with more authenticated testing set assisted with knowledge-based annotation.
Chapter 3

Methodology and Approach

Four phases are included in the project. The first phase is a pipeline of the construction of the original dataset. The second one is the key filtering process which involves automatic methods and strategies to remove the noise in the original dataset. Another face dataset merged from CASIA\[10\] and VGG\[23\], is also experimented in our project to increase its quality. With the filtered dataset and improved CASIA-VGG dataset, the third phase includes model training and testing as the performance evaluation for the dataset. Four phases are elaborated below.

3.1 Original Dataset Construction

Data Crawling

Original image data should be collected from Google image according to a newly designed celebrity list. The celebrity list is expected to contain about 50K identities with more variance. For example, multiple occupations of celebrities are key metrics of identity variation. The MS-Celeb-1M (Figure 3.1) highlights its superior quality in occupation variation to whose celebrity list only comes from movies actors and actresses. Therefore, the celebrity
list of our dataset should have sufficient variation in occupations. Moreover, we will explore to extend the diversity of nationality of the celebrities if time and resource allow.

### Dataset Preprocessing

After getting the raw data from Google image search, one step face preprocessing should be done in order to fit into the domain of face recognition. There are many existing models like ACW [39] and MTCNN [42]. ACW is chosen for its wide application as well as its comparative stability.

According to the celebrity list, top 100 images per identity in Google Image are scratched. Giving credence to search engine ranking is a general technique across research and industrial institutions [6, 5, 2]. During the process of image scratching, the issue of anti-robot program and potential network limitation should be taken into consideration.

### 3.2 Dataset Filtering

**Model Selection**  Based on current trends on face recognition described in the last chapter, it is empirically reasonable to choose FaceNet [28] and SphereFace [19] for the model selection. Both of them are trained models which bring convenience for model evaluation and direct
CHAPTER 3. METHODOLOGY AND APPROACH

application. At the same time, more models such the ones driven by center loss [35] will be explored for more variations of filters. Using ensemble learning which is basically a fusion of different models, it is more likely to achieve better filtering performance with more distinctive model filters. There are two possible strategies for model fusions. Firstly, features extracted from different models can be concatenated and then classified by a new model. Secondly, a voting system could be constructed and remove the particular item when the majority of models prefer to remove it.

Evaluation of filtering performance Before conducting the real filtering of the whole dataset, a small but useful benchmark dataset will be built to evaluate the filtering performance of selected models. Concretely, 100 identities are selected randomly from the whole dataset. Then, all images in each identity folder will be labeled into either positive or negative group based on the human judgment on whether they should be filtered out or not. The labeling process should receive enough iteration of examination and cross-checking such that inter-class variations are reserved and ambiguities are reduced to lowest. Apart from the intention of evaluating existing models filtering performance, another target can be achieved, which is to find appropriate clustering parameters for the best filtering potency of the model. This parameter tuning process can be easily automated by the grid search algorithm [16].

Filtering process The real filtering process contains two parts: (1) Feature (or embedding) extraction from the original dataset using models selected from last step and (2) Clustering image instances according to the corresponding algorithm with carefully chosen parameters. Feature extractions can be done by the built-in function in Caffe [8] to extract data from any specific layer in the model. The clustering process adopts the method proposed in [26], which is a modified version from K-means [20], by tuning two key metric delta and density [26].
CHAPTER 3. METHODOLOGY AND APPROACH

Quality control of filtered dataset  The evaluation of filtering is the quality control for the iterative filtering process. After one iteration of filtering, a set of randomly selected data from the output is evaluated according to its remaining noise proportion. A reasonable threshold should be determined for the final pass of the filtering process with empirically rigorous exploration.

3.3 Models in Automatic Filtering

FaceNet

Triplet loss function was discussed in section 2.3. Two network structures are experimented in their research team as described in the paper\cite{28}, named NN1 and NN2. The detailed network structure and number of parameters are shown in figure 3.2. NN1 is a 22-layer deep network structure and uses 1x1 kernel-sized convolutions as mentioned in \cite{41}. NN2 utilizes GoogLeNet style inception model\cite{31}, which contains fewer parameters. In our project, we refered to the implementation in [https://github.com/davidsandberg/facenet](https://github.com/davidsandberg/facenet) and used the pretrained Inception-ResNet-v1 model to do feature extraction.

SphereFace

Sphereface is also attempted in our project to extract embedding features for raw dataset. Sphereface trained four network structures to test their 1) performance in validation dataset (LFW, YTF). 2) time consumption in training process. In training phase, it trains network using training dataset with angular loss in back-propagation. Then the trained model is used to generate 512-dimension deep features for face recognition. Details of network structure and complete training-testing procedures are given in Figure 3.2 and Figure 3.3. We refered to the Pytorch\cite{25} implementation of sphereface in [https://github.com/clcarwin/sphereface\_pytorch](https://github.com/clcarwin/sphereface\_pytorch).
CHAPTER 3. METHODOLOGY AND APPROACH

20

Figure 3.2: Five network structures experimented in sphereface. They are all ResNet based with weight normalization. The return is 512 hyper dimensional features that will be used for face recognition.

<table>
<thead>
<tr>
<th>Layer</th>
<th>4-layer CNN</th>
<th>10-layer CNN</th>
<th>20-layer CNN</th>
<th>36-layer CNN</th>
<th>64-layer CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1.x</td>
<td>[3×3, 64]×1, S2</td>
<td>[3×3, 64]×1, S2</td>
<td>[3×3, 64]×1, S2</td>
<td>[3×3, 64]×1, S2</td>
<td>[3×3, 64]×1, S2</td>
</tr>
<tr>
<td></td>
<td>3×3, 64</td>
<td>3×3, 64</td>
<td>3×3, 64</td>
<td>3×3, 64</td>
<td>3×3, 64</td>
</tr>
<tr>
<td></td>
<td>3×3, 64</td>
<td>3×3, 64</td>
<td>3×3, 64</td>
<td>3×3, 64</td>
<td>3×3, 64</td>
</tr>
<tr>
<td></td>
<td>3×3, 128</td>
<td>3×3, 128</td>
<td>3×3, 128</td>
<td>3×3, 128</td>
<td>3×3, 128</td>
</tr>
<tr>
<td></td>
<td>3×3, 128</td>
<td>3×3, 128</td>
<td>3×3, 128</td>
<td>3×3, 128</td>
<td>3×3, 128</td>
</tr>
<tr>
<td></td>
<td>3×3, 256</td>
<td>3×3, 256</td>
<td>3×3, 256</td>
<td>3×3, 256</td>
<td>3×3, 256</td>
</tr>
<tr>
<td></td>
<td>3×3, 256</td>
<td>3×3, 256</td>
<td>3×3, 256</td>
<td>3×3, 256</td>
<td>3×3, 256</td>
</tr>
<tr>
<td></td>
<td>3×3, 512</td>
<td>3×3, 512</td>
<td>3×3, 512</td>
<td>3×3, 512</td>
<td>3×3, 512</td>
</tr>
<tr>
<td></td>
<td>3×3, 512</td>
<td>3×3, 512</td>
<td>3×3, 512</td>
<td>3×3, 512</td>
<td>3×3, 512</td>
</tr>
<tr>
<td>FC1</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
</tbody>
</table>

Figure 3.3: Complete training-testing procedures in sphereface. In the training phase, features from Fully connected layer will be used in Angular-softmax loss function for updating network weights in backpropagation. In test phase, test images will be passed into the network and deep features will be extracted for verification or recognition purpose.

We adopted the 36-layer CNN structure for limited training time.

3.4 CASIA-VGG dataset

CASIA is a large scale face dataset first mentioned in [40]. CASIA contains 10575 different identities and 494,414 images. The average number of images for each identity approximates
47, which is far larger than small scale dataset like LFW. VGG\(^{23}\) is another large scale face dataset, which contains 2622 identities and around 2.6 million images. With the motivation to obtain a larger size dataset, some pioneering work to combine CASIA and VGG dataset was finished by our colleagues. After some human-filtering effort, CASIA dataset and VGG dataset are merged to own 10991 identities and 739338 images. However, with deficiencies in image preprocessing procedures, the training performance using CASIA-VGG did not reach our expectation. Possible reasons have been researched in this project to boost the model training performance. Potential problems include noisy samples, duplicated images, difference in gray-color scale image and variation of alignment methods etc. An example of noisy images, duplicated images and gray-color images is given in figure 3.4-3.6.
3.5 Evaluation of Face Recognition Performance

Training existing models on the constructed dataset  Sphereface\cite{19} is a standard model to be retrained or fine-tuned on the newly constructed and filtered dataset. Retraining step basically helps evaluate the quality of the dataset and the improvement of the model performance will act as the evidence of the new datasets effectiveness.

Testing with existing benchmarks  After training with filtered data, models should be tested and evaluated by existing face recognition benchmarks. LFW\cite{5,6} provides a direct reference to most of the models in the literature. It is widely adopted as a benchmark set for examining model performance. In our work, LFW is mainly used as the validation set.
Chapter 4

Dataset Construction

In this section, data collection, purification, and pre-processing of scratched dataset will be elaborated. Adjusted preprocessing procedures of CASIA-VGG dataset are also discussed in this chapter.

4.1 Construction of scratched face dataset

Data Collection and Preprocessing

The data collection, identity selection and image scratching tasks were conducted within one month. Then initial trials in face data preprocessing, face detection and alignment were experimented following that. Original image data were crawled from Google image according to celebrity list crawled from Wikipedia. The celebrity list contains about 40K identities from around 10 occupations and most of the identities are movie and TV actors (actresses) and sports people. The distribution of our identities is similar to the Microsoft 1M-FACE (See Figure 2), which includes more variation than [24] whose celebrity source only comes from movies actors and actresses. Python scripts were harnessed to extract identities from Wiki celebrity page[36] for its advantage in web programming. Then the image crawling followed
by the proposed method in Chapter 3 was conducted subsequently by the joint work of 4 local computers in the instruction lab 311 in Haking Wong Building, HKU. The parallel work reduced the time cost of this task by around 75%. Particularly, the crawler program solved the issue of anti-robot counterpart and network limitation by employing strategic slowing down. By sleeping around 0.1 second during iterative request sending driven by the browser simulator, we successfully get around the network control in the CS Department and Google reCAPTCHA (an anti-robot system). Detailed code are available on Github\footnote{https://github.com/whcacademy/imageDownloader}. The parallel method was also used in the face alignment and detection process but it only involved in one server computer with multiprocessing. Because the detection software required a special environmental setting, local computers in the lab mentioned above is costly to perform this task. The preprocessing process was found to contribute the filtering process because for some extreme cases like no face or too many faces in one image could be dropped by the failure of single face alignment and detection. Around 30% of images are filtered out in the detection and alignment process.

**Benchmark Construction**

The small benchmark, which is the filtering performance indicator, was built by joint work of team members. 100 identities with about 6,000 images (40% of images were filtered out by preprocessing step) were randomly selected, labeled and cross-checked before being used by the model evaluation later. After manual labeling, around 80 of 100 identities are effective and valid. The rest of the identities do not have enough images or only contain too much noise. To guarantee an efficient filtering accuracy measurement process, we deleted these invalid identities. Because of its human-manageable size, labeling process was not affiliated by labeling application and directly scripted by hand, small content maintaining scripts are
used to help to simplify the label file so team members only needed to fill one type of tag and leave another type empty to be auto-filled. The detailed statistics of the benchmark is listed in the following table.

<table>
<thead>
<tr>
<th>Image size</th>
<th>Identity size</th>
<th>Average Image per Identity</th>
<th>initial purity level</th>
</tr>
</thead>
<tbody>
<tr>
<td>5200</td>
<td>82</td>
<td>63.4</td>
<td>58%</td>
</tr>
</tbody>
</table>

Table 4.1: Statistics of benchmark for auto filtering

**Dataset filtering & Model evaluation**

As the key of this project, each step of dataset filtering should receive great attention. The project spent 5 months (from Oct. 2017 to Feb. 2018) in filtering dataset. Several contributions made are described in the following.

Three rounds of filtering process have been conducted using FaceNet pre-trained model, Sphereface pre-trained model and Sphereface self-trained model in CASIA-Web Face dataset. The purity scores have been evaluated on our small benchmark. FaceNet model achieved 99.63% accuracy on LFW dataset, which was previously state-of-the-art score. The model was used in extracting 128-dimensional features from images in benchmark dataset. Following that, an unsupervised clustering algorithm was adopted in these features in order to filter out noisy images. Sphereface pre-trained model in the online accessible pytorch version reaches 99.22% accuracy in LFW dataset. And we rigorously followed the instruction to train the model in CASIA-Webface dataset as shown in paper. Our self-trained model of Sphereface scored 99.20% accuracy in LFW dataset. With the same feature extraction and clustering procedures as FaceNet, both the pre-trained and self-trained model are used to filter HKU-Face dataset.

[https://github.com/clcarwin/sphereface_pytorch/tree/master/model](https://github.com/clcarwin/sphereface_pytorch/tree/master/model)
1. **First Round: Cluster-based filtering** At first trials of the experiments, our strategy was to delete irrelevant clusters created by the density peak clustering algorithm\cite{26}. One main cluster was chosen based on number of images and other clusters were discarded. A sample output of executing this clustering algorithm is given in Figure 4.1. After observation, it is obvious that density of red points, which should be preserved by the filtering process, range from low to high values. This represents our current choice of filtering rule does not satisfy the purification requirement.

2. **Second Round: Density-based filtering** With the lesson learnt from initial experiments, we realized that deleting images based on cluster was not correct. Many images deleted from dataset also have high density values, which means they should be reserved. A more serious problem is that many noisy images with low density blend into the major cluster, which did harm to the purity of dataset. To solve this problem, our second round adopted a density-based filtering method to purify our dataset. The features put into clustering algorithm were identical to the first round. After our experiments, we chose the density threshold to be 8 and images with density smaller than this value were removed from dataset. This method improves the mean purity of benchmark set from 58% to 63%. Since the target purity of our dataset is at least
above 90%, new filtering approach needs to be proposed.

3. **Third Round: Uniformly sized images**  The improvement ratio of purity for our dataset is not high by modifying clustering strategy. Then we move our attention to feature extraction procedures to further increase purity level. After face detection and alignment, facial parts were detected and extracted from images in raw dataset. These parts were utilized as refined images and put into FaceNet model for feature extraction. However, these images do not have uniform sizes when feeding into neural network. Our previous pipeline randomly resized or cropped the images before using it for training purpose. We realized that the quality of generated features was effected since facial part of images may be removed after random cropping. The solution we adopted to this problem was to manually unify the image size. By preserving the central $224 \times 224$ square of aligned images, the facial part is reserved. Quality of features is better without losing useful information by random cropping. In our experiments, using the density-based clustering algorithm with newly generated features, the purity value on benchmark set can achieve around 90%, which is an evident improvement. The distribution of filtering outcome is visualized by boxplot in Figure 4.2. We can see that more than three quantile of identities have above 90% purity value. In fact, most of these identities have completely purified dataset. Some samples in the graph have low values because their raw datasets are originally noisy with few correct images. Table 4.2 shows the summary of purity levels during our filtering steps.

<table>
<thead>
<tr>
<th></th>
<th>Original Purity</th>
<th>After first round</th>
<th>After second round</th>
<th>After third round</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>58%</td>
<td>58%</td>
<td>63%</td>
<td>90%</td>
</tr>
</tbody>
</table>

Table 4.2: Purity level of benchmark dataset

4. **Forth Round: Attempting Sphereface network structure** After obtaining im-
Figure 4.2: Purity distribution of benchmark set using Facenet generated features.

Improvement using uniformly sized images for feature extraction, we further repeated the same procedures by using Sphereface pre-trained model and self-trained model for dataset filtering. Sphereface resized 224x224 images into 112x96 in its implementation. For the pre-trained model, only forward-inference was performed to extract features of face images. Then the 512-Dimensional features were put into density-peak clustering system for dataset filtering. However, the purity of dataset did not have an improvement by using new features. Table 4.3 shows the filtering result using Facenet pre-trained model, Sphereface pre-trained model and Sphereface self-trained model. The clustering approach is prudentially identical for all of the three models. Since Facenet used a large-scale private dataset to train, only pre-trained model is used to extract the embedding features.
CHAPTER 4. DATASET CONSTRUCTION

<table>
<thead>
<tr>
<th>Network Structure</th>
<th>lfw-acc</th>
<th>feature Dimension</th>
<th>Filtering Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaceNet [14]</td>
<td>99.63%</td>
<td>128</td>
<td>89.97%</td>
</tr>
<tr>
<td>Sphereface-pre-trained [19]</td>
<td>99.22%</td>
<td>512</td>
<td>87.69%</td>
</tr>
<tr>
<td>Sphereface-self-trained [19]</td>
<td>99.20%</td>
<td>512</td>
<td>85%</td>
</tr>
</tbody>
</table>

Table 4.3: Facenet pre-trained model, Sphereface pre-trained model and self-trained model comparison in image filtering performance

4.2 Refinement in preprocessing CASIA-VGG dataset

The original filtered CASIA-VGG dataset was supplied to us by Weifei
[3]. Both the aligned version and raw version were offered to us. The face alignment method applied to obtain the aligned version is ACF
[39]. During our experiments, we first used the aligned version to train Sphereface network structure. The training setting is similar as described in
[4], we achieved around 95% accuracy in LFW validation set. This result did not reach our expectation since training on CASIA-Webface alone can obtain better accuracy score. Then we started to analyze the possible reasons to cause the problem.

Different face detection method

The original sphereface adopted mtcnn
[42] to detect faces in images. The detected face images of CASIA-Webface are then used to train residual-based neural network with angular-softmax loss. To verify the influence given by different face detection algorithm, We align the raw dataset of CASIA-VGG dataset with mtcnn
[42] and use it to train sphereface. The results got improved in using different face detection and alignment method.

[3] https://dblp.uni-trier.de/pers/hd/g/Ge:Weifeng
[42] https://github.com/clearwin/sphereface_pytorch
CHAPTER 4. DATASET CONSTRUCTION

Figure 4.3: Image in CASIA-VGG mtcnn aligned.

Figure 4.4: Image in CASIA-Webface mtcnn aligned after cropping.

Margin size matters

A further look into the difference between CASIA-Webface and CASIA-VGG dataset was taken after applying mtcnn. An obvious difference exists that the margin size between the two dataset varies significantly. An example in figure 4.3, 4.4 illustrates the difference.

Large margin brings much noisy background information and the training performance would be influenced. The cropped images in CASIA-Webface were then used to train sphereface and the performance in LFW dataset improved. The detailed experiments and improving result will be elaborated in next section.

Similarity transformation

In [9, 19, 33], similarity transformation is adopted to preprocess those images. The idea is to transform the facial part of images into the center and then generate an image with canonical pose.
CHAPTER 4. DATASET CONSTRUCTION

The \([u, v, 1]^T\) is the homogeneous coordinate of target position after transformation and \([x, y, 1]^T\) is the corresponding homogeneous points in raw images. The first step is to extract the five landmarks (left eye, right eye, nose, left and right corner of mouth) using mtcnn. Then five fixed points are defined in target image, which correspond to the five landmarks. Transformation matrix can be calculated by least square approximation. The raw image was multiplied by the similarity transformation matrix to obtain the target training image. An example is given in figure 4.5,4.6 to compare the raw image and the target training image after similarity transformation, cropping and resizing. The pre-processed face images were used to train network structure. The evaluation score on LFW verification task shows a salient improvement of network training performance without adjusting network parameters. Details of the experiment result will be discussed in next chapter.
Chapter 5

Face Recognition Experiments

In this chapter, training performance by using HKU-Face dataset and CASIA-VGG dataset are described. Sphereface\cite{19}, Arcface\cite{9} are two models mainly attempted in this project. Network structure and loss function of CosFace\cite{33} model is also researched and under experiments.

5.1 Result reproduction

Reproduced experiments for Sphereface were conducted using CASIA-Webface dataset. As shown in chapter 3 figure 3.2, Five residual-based networks are used for training purpose, which include 4-layer, 10-layer, 20-layer, 36-layer and 64-layer structures. 36-layer Sphereface implementation was used along our all experiments as shown in figure 5.1 below. The network implements weight normalization but not feature normalization as mentioned in chapter 3. Since the complete structure is hard to visualize here, We only show the abbreviated version here. Compared with 64-layer structure, which was experimented in the paper, 36-layer network takes less time to train and more suitable to our project. At first, CASIA-Webface dataset was preprocessed using MTCNN\cite{42} for face detection. Then
CHAPTER 5. FACE RECOGNITION EXPERIMENTS

Figure 5.1: The 36-layer residual based network used in our experiments. For instance, [3x3, 64] means one convolutional layer with 64 filters and 3x3 kernel size. S2 means stride equal to 2 and the 512 in bottom means the embedding feature size.

similarity transformation was applied to ensure the five landmark points(two in eyes, one in nose, two for mouth) locating in center part of image. The experiment sets epoch to be 20, initial learning rate to be 0.1, and declining to 0.1 times in 10th, 15th and 18th epoch. The batch size is 512, momentum is set to be 0.9 and weight-decay rate is set to be 0.0005. After some parameter tuning procedures, we obtained 99.20% accuracy on LFW validation dataset, which approaches the 99.22% accuracy given in online source. However, this performance cannot match the score given in paper, which achieved 99.42% in LFW. The potential reason is the difference in network structure and training time of neural network. As the paper uses 64-layer structure to train, the generalization capability is better.

Experiments to reproduce Arcface were also conducted using filtered MS1M-VGG2 dataset as mentioned in paper. Reproduction experiments towards Arcface takes longer
time due to the deeper network structure and larger scale training dataset. Following the experiment settings, modified ResNet-100 model was used as backbone. Margin parameter was set to be 0.5, Parametric Rectified Linear Unit [4] was used to replace ReLU, momentum equal to 0.9 and weight decay equal to 0.0005. The default experiment set initial learning rate to be 0.1 and batch size to be 512. Compared with default setting, the batch size was defined to 64 and initial learning rate was set to be 0.05 for limited GPU memory and more robust experiments. With around 20-hour training, the LFW accuracy in our experiments reaches 99.40%. Potential insufficiency of our training model may result from the difference in batch size and lack of enough training time. Smaller batch size led to higher deviation between batch samples and overall distribution, which could be possible reason for not achieving the same validation scores as in paper. The reproduction results for Sphereface and Arcface are given in table 5.1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Loss</th>
<th>LFW-accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA-Webface</td>
<td>36-resNet</td>
<td>Sphereface(Our)</td>
<td>99.20%</td>
</tr>
<tr>
<td>CASIA-Webface</td>
<td>64-resNet</td>
<td>Sphereface(paper[19])</td>
<td>99.22%</td>
</tr>
<tr>
<td>MS1M-VGG2</td>
<td>100-IresNet</td>
<td>Arcface(Our)</td>
<td>99.40%</td>
</tr>
<tr>
<td>MS1M-VGG2</td>
<td>100-IresNet</td>
<td>Arcface(paper[9])</td>
<td>99.83%</td>
</tr>
</tbody>
</table>

Table 5.1: Compare the reproduction experiment results of Sphereface and Arcface with results in paper.

5.2 HKU-Face experiments

After purification, HKU-Face dataset contains around 0.5 million images, 16601 identities with 30 images per identity in average. Number of images in HKU-Face is similar to CASIA-Webface but number of identities is around 1.5 times of it. At first, all identities and images were used as training data for Sphereface training procedures. But the training
suspended after a few epochs with loss divergence. With parameter tuning and learning rate adjustment, this problem was not solved. To examine whether it is caused by inadequate purity of our dataset, my partner and I have manually examined and adjusted all images of 1000 identities. After purification and identity deletion, the 1000 identities were purified into 782 clean identities with 30 images per identity in average. Then the 782 identities were used to train sphereface. With 20 Epoch training time, the model achieved 82% accuracy in LFW dataset. During the manual purification process, another problem was found for the deteriorated quality of HKU-Face dataset. Though some identities have different names, the images may belong to the same person. This is a problem from image extraction stage. We found this problem at a later schedule in this project and it is challenging to solve. Details for this mislabeling problem will be discussed in next chapter. To verify whether this problem results in a bad performance in training network, we randomly chose 782, 1000, 5000 identities from this dataset without human deletion and adjustment. Then these three sub-datasets are used to train sphereface and evaluate LFW verification accuracy. With the same network setting and alignment steps, the performance of Sphereface trained in three dataset is given in Table 5.2.

<table>
<thead>
<tr>
<th>Number of ID</th>
<th>Alignment method</th>
<th>LFW-accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>782-id</td>
<td>MTCNN[42]</td>
<td>79%</td>
</tr>
<tr>
<td>1k-id</td>
<td>same</td>
<td>82%</td>
</tr>
<tr>
<td>5k-id</td>
<td>same</td>
<td>84%</td>
</tr>
<tr>
<td>782-id-Purified</td>
<td>same</td>
<td>82%</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of model performance trained using randomly picked 1k, 5k id and manually purified 1k id dataset.

Though the manually labeled 782 identities have a better training result than randomly picked 782 identities (from 79% to 82%), the improvement ratio is not salient, which means that the dataset quality is still low. To further verify our thought, three sub-datasets con-
taining 782, 1k and 5k randomly picked identities have been chosen from CASIA-Webface. With corresponding purified 782, 1k and 5k id dataset in HKU-Face, The six datasets were used to train Sphereface. In preprocessing steps, we applied margin cropping method as discussed in section 4.2. After using the same network setting procedures, evaluation results in LFW have been compared at table 5.3. With 1k-id, CASIA-Webface dataset can train Sphereface to have 90.4% accuracy in LFW dataset, which means their dataset quality is better than ours.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of ID</th>
<th>LFW-accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKU-Face</td>
<td>1k-id</td>
<td>85.2%</td>
</tr>
<tr>
<td>HKU-Face</td>
<td>5k-id</td>
<td>86.1%</td>
</tr>
<tr>
<td>HKU-Face</td>
<td>782-id-Purified</td>
<td>83.4%</td>
</tr>
<tr>
<td>CASIA-Webface</td>
<td>782-id</td>
<td>89.3%</td>
</tr>
<tr>
<td>CASIA-Webface</td>
<td>1k-id</td>
<td>90.4%</td>
</tr>
<tr>
<td>CASIA-Webface</td>
<td>5k-id</td>
<td>94.6%</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of model performance being trained using randomly picked 782, 1k, 5k, manually purified 782 id from HKU-Face and 782, 1k, 5k randomly picked id from of CASIA-VGG.

5.3 CASIA-VGG experiments

Preprocessing Stage

CASIA-VGG dataset contains 739,338 images and 10991 identities in total after filtering. CASIA-VGG dataset was used to train both Sphereface and Arcface model. Experiments using different face detection approach, margin setting and similarity transformation were conducted as mentioned in section 4. In the baseline experiment, CASIA-VGG dataset was preprocessed by ACF detection method without margin processing and similarity transformation. Then this preprocessed dataset was used to train Sphereface with identical
network setting as before. By decreasing the learning rate and tuning parameters, the model can be trained to reach 90.5% accuracy in LFW dataset. The training performance did not reach our expectation, which had obvious gaps with online pre-trained model. As analyzed in chapter 4, we utilized MTCNN approach in detecting faces. The performance got obvious improvement to reach 95% accuracy in LFW dataset. Figure 5.2 and figure 5.3 illustrate the raw image and landmarks returned by MTCNN.

Next, there still exist differences between CASIA-VGG dataset and the dataset used in original paper. The size of face bounding box of our dataset is larger, which brings noisy information into training. Margin cropping and similarity transformation were applied to images following face detection and the accuracy increased to 98.93%. After these pre-processing stages, we used the dataset to train both Sphereface and Arcface. The complete training results are summarized into table 5.4.
### Table 5.4

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>Loss</th>
<th>Number of ID</th>
<th>Alignment method</th>
<th>LFW-accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA-VGG</td>
<td>36-resNet</td>
<td>Sphereface</td>
<td>10988</td>
<td>MTCNN-ST+no margin</td>
<td>98.93%</td>
</tr>
<tr>
<td>CASIA-VGG</td>
<td>36-resNet</td>
<td>Sphereface</td>
<td>10988</td>
<td>MTCNN-no margin</td>
<td>98.68%</td>
</tr>
<tr>
<td>CASIA-VGG</td>
<td>36-resNet</td>
<td>Sphereface</td>
<td>10988</td>
<td>MTCNN-small margin</td>
<td>95.1%</td>
</tr>
<tr>
<td>CASIA-VGG</td>
<td>100-IresNet</td>
<td>Arcface</td>
<td>10988</td>
<td>MTCNN-ST+no margin</td>
<td>96.26%</td>
</tr>
<tr>
<td>MS1M-VGG2</td>
<td>100-IresNet</td>
<td>Arcface</td>
<td>85K</td>
<td>MTCNN-ST+no margin</td>
<td>99.40%</td>
</tr>
</tbody>
</table>

Table 5.4: Networks are trained in dataset with different preprocessing settings. MTCNN is used for face detection and alignment. Small margin, no margin and similarity transformation(ST) settings are set as experiment target. Sphereface used 36-layer residual network and Arcface used 100-layer improved version of resNet as discussed in **chapter 2**.
Chapter 6

Limitations & Difficulties

6.1 Limitations of HKU-Face dataset

There are some limitations existed for the constructed HKU-Face dataset.

First of all, an obvious limitation is the deficiency in carefully designed celebrity list. This project was initially designed to crawl celebrity images from Wikipedia celebrity list [36]. This approach suffers from a fatal problem that there exist name-image mismatching identities in the list. Some less famous celebrities’ name may be similar to another more well-known celebrity’s name. And their images from Google search may belong to the more famous identity. Several examples of this error are found in the constructed dataset, which led to the bad performance in training neural network. Though the purity within an identity folder could be high, the mistaken labels across identities can not be avoided.

The second problem is very similar. There exist celebrities with same name in the wiki list, which increased the chance to make searching mistakes. With the same name, the probability to make mismatching mistake was greatly increased. For instance, figure below shows two celebrity pair with identical names.

In order to solve this challenging problem, we are inspired by the data processing proce-
Figure 6.1: Two persons in the left image are both called Steve McQueen and the identities in the right image are both called Michael Douglas. These two identities are in the wiki celebrity list.

Their experiment faced a replicated identity removal problem between MegaFace\cite{13} and FaceScrub\cite{22} dataset. They first calculated the feature vector center within identity. Then deletion or reserve of an identity is based on cross-identity cosine similarity score. Our problem can also be abstracted as an replicated identity removal problem, for which this approach should be attempted. Since I and my partner found this potential solution recently, we will keep working on this project and try to solve this problem.

6.2 Limitations on CASIA-VGG dataset

Thanks to Weifeng’s previous work, we have a relatively cleaned dataset incorporating CASIA-Webface dataset and VGG dataset. After researching on the preprocessing methodologies like face detection and similarity transformation, we successfully improve the training efficiency using CASIA-VGG. Before our preprocessing procedures, the initial training result using CASIA-VGG is around 80%. After our research and contribution, Sphereface model can be trained to have 98.93% accuracy in LFW dataset using CASIA-VGG and Arcface model can have 96.26% accuracy in LFW dataset. However, the result does not reach the
current state-of-the-art score and still have improvement space.

One potential problem may come from duplicated images of same identity. CASIA-VGG dataset contains around 0.74 million images compared to 0.4 million images of CASIA-Webface. But the number of identities of the two dataset is nearly the same. More additional images added to the same identity has a higher possibility to add more duplicated images within the same person. This may cause overfitting problem with smaller loss in training phase but higher loss in testing phase.
Chapter 7

Summary

In this final year project report, detailed information towards our project has been discussed in detail.

At first, literature review has been recalled in chapter 2. Current trend of face recognition focus on the construction of larger scale dataset with high purification level, selection of effective backbone network structure and development of more distinctive loss functions. Our project mainly focus on dataset construction and loss function analysis.

We introduced our methodologies in chapter 3, including how we extracted images using wiki celebrity list and dataset filtering steps. Both HKU-Face dataset and CASIA-VGG data are processed and purified in our project, which were used to train face recognition models.

In chapter 4, details to construct our scratched HKU-Face dataset and CASIA-VGG dataset were elaborated. For HKU-Face dataset, we used facenet pre-trained model to extract hyper-dimensional feature vectors and executed clustering algorithm for removing noisy images. For CASIA-VGG dataset, preprocessing steps were researched for improving its effectiveness in network training. With adjusted face detection, alignment and margin cropping procedures, CASIA-VGG dataset can train Sphereface network structure to achieve
98.63\% accuracy in LFW and 96.26\% for Arcface. This is an obvious improvement compared with 80\% accuracy achieved before. This is one of our contribution to the face recognition research in HKU.

**Chapter 5** discussed the experiments we have conducted and model setting details. We first tried to reproduce the result mentioned in literature and then applied our own dataset to train these face recognition models. **Chapter 6** introduced the limitations on HKU-Face dataset and CASIA-VGG dataset such as image-identity mismatching problem and image duplication problem.

This project mainly focuses on the general dataset building procedure and it is inspirational to find that current models can be utilized as the filters to purify the original dataset with noise. Future work of this project includes solving the image-identity mismatch problem as discussed **Chapter 6**. Further study on the accuracy improvement of the filter and loss analysis can be significant for the development of face recognition.
Bibliography


