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

Final Report

Haicheng Wang
3035140108

COMP4801 Final Year Project
Project Code: 17007
Abstract

Current face recognition usually faces problems with the training dataset due to the insufficient size and potential manual labelling errors. The project introduces a dataset construction and filtering process to deal the problem with less cost. FaceNet\textsuperscript{[35]} and Sphereface\textsuperscript{[29]} are harnessed for the purpose of filtering the dataset scratched from Google. Results show the impressive effectiveness of automatic filtering and purity enhancement after filtering with considerable attention on labeling errors in the view of web search. Except exclusively self-constructed dataset, filtered and merged dataset from CASIA-WebFace\textsuperscript{[54]} and VGG Face \textsuperscript{[32]} were also tested and analyzed. Subsequent research and experiment can target at the further improvement of filtering process with lower false negative rate as well as getting rid of labeling errors due to web search. And those further improvements are expected to contribute more to the unsupervised learning in the general fine-grained object recognition.
Acknowledgments

I would like to express my great appreciation to Prof. Yu for his supervision and help in providing computational resources. Thanks to Weifeng Ge for his instructive guidance through the whole project period. I would also want to thanks all classmates that help us in our project.
Contents

Abstract ii

Acknowledgements iii

Contents iii

List of Figures v

List of Tables vi

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

2 Literature review 6
   2.1 Theoretical Background ....................................... 6
   2.2 Dataset Construction .......................................... 8
   2.3 Face Recognition Models ..................................... 9
   2.4 Benchmark for Face Recognition .............................. 10

3 Methodology and Approach 11
   3.1 Original Dataset Construction ......................... 11
   3.2 Dataset Filtering ........................................... 14
   3.3 Models as Automatic Filters ............................. 16
   3.4 Evaluation of Face Recognition Performance .......... 22
4 Experiments and Results
  4.1 HKUFace: exclusively self-constructed dataset ............... 25
  4.2 CASIA-WebFace, CASIA-VGG ............................... 37

5 Conclusion and Future Works

Bibliography
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Illustration of dataset without filtering (in one class).</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>Illustration of occupational distribution of celebrity in MS-Celeb-1M</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of different alignments.</td>
<td>14</td>
</tr>
<tr>
<td>3.3</td>
<td>Initial state and the learning process driven by triplet loss.</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Comparison between two models in MNIST, the left one is center loss feature space, the right one is the plotting for 1st and 2nd activation layer which shows the basic feature representation of figure instances.</td>
<td>19</td>
</tr>
<tr>
<td>3.5</td>
<td>The left column 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.</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Decision margins of angular loss functions under binary classification case</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>Illustration of clustering problem</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>Purity distribution of benchmark set using newly generated features</td>
<td>30</td>
</tr>
<tr>
<td>4.3</td>
<td>Illustration of Google search output for key &quot;Jordon, Phil&quot;, where GREEN boxes belongs to Jordon, while RED boxes belongs to Michael Jordan&quot;</td>
<td>36</td>
</tr>
</tbody>
</table>
List of Tables

1.1 Representatives of FR models ........................................ 2
1.2 Face recognition datasets ........................................... 3

2.1 VGG2 Facial dataset working process ............................. 9

3.1 Mathematical comparison between different angular distance in binary case ........................................ 21

4.1 Statistics of benchmark for auto filtering ......................... 27
4.2 Purity level of benchmark dataset .................................. 30
4.3 SphereFace structure in dataset evaluation ....................... 33
4.4 Evaluation for HKUFace with Loosely ACF alignment .......... 34
4.5 Evaluation for HKUFace with Loosely/Tightly MTCNN alignment ........................................ 34
4.6 Experiment on strictly cleaned subset ............................... 35
4.7 Sample pairs that confirmed to be the same in HKUFace .... 36
4.8 Reproduction of Sphereface on CASIA-WebFace ................ 39
4.9 Experiment of influence of similarity transformation ........ 39
4.10 Experiment of influence of image filtering ...................... 39
Chapter 1

Introduction

1.1 Background

Face recognition (FR) is one of the most heated and representative tasks of Artificial Intelligence for many years\cite{2, 25, 26, 28, 42, 44, 51}. From pure facial images recognition to more challenging faces in the wild, FR is becoming mature in terms of theoretical and practical views. Human beings have already been beaten in most academic benchmarks since more and more record breakers like Gaussian-Face\cite{30}, FaceNet\cite{35}, SphereFace\cite{29} and ArcFace\cite{5} came out. Face recognition has been applied in various areas like identity authentication in smart home, electronic devices unlocking and procedure recording, benefiting the industry world with its increasing accuracy and lower cost.
CHAPyER 1. INTRODUCTION

<table>
<thead>
<tr>
<th>Gaussian-Face 30</th>
<th>FaceNet 35</th>
<th>SphereFace 29</th>
<th>ArcFace 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA-WebFace 54</td>
<td>Google Private Dataset</td>
<td>CASIA-WebFace 54</td>
<td>VGG-Face2 3 &amp; MS-1M 6</td>
</tr>
<tr>
<td>LFW</td>
<td>0.9852</td>
<td>0.9963</td>
<td>0.9942</td>
</tr>
<tr>
<td>MegaFace</td>
<td>N/A</td>
<td>0.8647</td>
<td>0.9005</td>
</tr>
</tbody>
</table>

Table 1.1: Representatives of FR models

1.2 Problem Definition

Despite achievements of FR so far, this area still encounters great challenge. Firstly, academic models are pretty sensitive to variations like illumination, portraits of faces 36, and potential obfuscation so that they may not be able to perform well in front of a great amount of test cases and distractors. Secondly, the industrial companies have to tackle more critical issue of maliciously designed adversarial samples. As a research-oriented project, our focus is basically on the potential mitigation strategies for the former one, especially on automatic datasets construction process.

The problem of sensitivity comes from the nature of deep leaning models. Almost all of the models are trained by supervised learning where the datasets are labeled manually. Two following features of dataset contribute to the issue of sensitivity: (1) relatively small size of datasets due to limited human effort; (2) accuracy problem due to human perceptual bias.

Parkhi, Vedaldi, Zisserman, et al. 32 discussed the issue on the small training dataset. It was described that giant companies hold private face database with larger size, while other research institution could only access public database for training such as CASIA-WebFace 54, which is much smaller. Considered the huge difference in size, smaller dataset acts like a barricade
to even higher performance (See the comparison in Table 1.1). However, even the largest private database is still small compared with the size of images on the Internet since it can be inferred as impossible to annotate faces in trillion size.

![Table 1.2: Face recognition datasets](image)

In [6], the second problem has been tackled in an indirect way by training fault-tolerance classification models. However, training dataset in MS-Celeb-1M[6] is only obtained from popular search engine without any manual filtering. As Figure 1 shows, these images should belong to one identity but apparently, men’s and women’s faces and even a non-face image are in one folder. The performance of the classifier is more likely to be higher trained on purified dataset.
CHAPTER 1. INTRODUCTION

1.3 Objectives

This project targets at dealing with the issue of sensitivity from the perspective of dataset construction. A general face dataset development procedure especially the dataset cleaning procedure is proposed, which can make the construction process of large-scaled face datasets efficient and with controllable noise. Retrained or fine-tuned on automatic built datasets, previous models are expected to achieve better performance. Moreover, similar procedures can be generalized into other types of dataset construction processes, which may yield benefit to other fine-grained object recognition tasks.

1.4 Outline of the Report

In the rest of this final report, several key aspects are presented. 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. The experiments and results are shown in Chapter 4 as well as the empirical evaluations. Conclusion and future works are described in the last chapter.
Chapter 2

Literature review

Key components connected to the project are discussed in this chapter. Theoretical background introduces the basic definitions of Face Recognition and Metric Learning. After that, methods about dataset construction, face recognition models and verification benchmarks are elaborated in details as well as some existing works.

2.1 Theoretical Background

**Face Recognition**  In essence, face recognition is a specific verification task targeting at the similarity between the query and the gallery set. Mathematically, consider gallery set as $G = \{g_0, g_1, \cdots, g_n\}$, and denote identity set $\hat{I} = \{I_1, I_2, \cdots, I_m\}$, where all elements form a partition of $G$. Closed set face Recognition problem is defined as to find a mapping $f(x)$ from testing set $T$ to identity set $\hat{I}$, in another word, to find the identity of testing queries. More generally and more challenging, an open-set face recognition task abandons the constraint that query set must only contain images in training set. It
involved the requirement of telling whether the model has "seen" identity before or not.

**Classification models** The close set problem is usually treated as classification problem which is a very basic but prevalent task in machine learning. Essentially, classification models optimize their parameters so that they can map any input instance to the correct predefined classes. All machine learning algorithms are able to perform the classification task, from basic perceptrons\cite{34} to deep learning approaches. More importantly, classification methods arcface\cite{5} and cosface\cite{43} are major solutions in the area of FR, since the datasets usually fail to reach the requirement of metric learning which will be described in the next paragraph.

**Metric learning** Open set problem is more demanding of the general discriminative features. Although feature learning guided by classification methods like softmax generates satisfactory performances, metric learning are believed to be more effective in this problem. Different from the direct mapping from faces to identities, metric learning firstly maps the input instance to a feature $h$ which theoretically lies in a hyperspace manifold\cite{24,11,41}. A discriminator $D$ is trained to serve the following task,

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

Traditional metric learning\cite{19,45,52,55} usually learns a matrix $A$ as distance measurement. Given two features vectors $x_1, x_2$, the distance of
them are defined as follows:

\[ \|x_1, x_2\|_A = \sqrt{(x_1 - x_2)^T A (x_1 - x_2)} \]

The verification task with a latent space empirically has better testing performance as Parkhi, Vedaldi, Zisserman, et al.\cite{32} shows, specifically for deep metric learning in the FR task. However, except FaceNet\cite{35}, few metric learning models are invented in the literature due to the dataset quality.

### 2.2 Dataset Construction

\cite{32} is quite heuristic, in which a general procedure of dataset construction with partial automatic process was firstly proposed. A well-designed framework was introduced to build and evaluate a facial dataset. This final year project borrowed the framework but utilized more advanced feature extraction models to conduct innovative filtering process. What’s more, a new benchmark specifically for the evaluation of the automatic filtering performance was built, as this area receives less consideration in the literature. After this paper and dataset called VGG-Face, VGG group also made a subsequent contribution to the construction of facial datasets, VGG2-Face\cite{3}. They paid more attention to enlarging variance inside one class, in terms of post and age. One particular point has been used in our reflection of dataset construction. As Table 2.1 shows, their major manual efforts were put into the very first stage. This inspires us to rethink of our initial dataset construction process of our exclusively self-constructed dataset.
CHAPTER 2. LITERATURE REVIEW

Table 2.1: VGG2[3] Facial dataset working process

<table>
<thead>
<tr>
<th>Stage</th>
<th>Aim</th>
<th>Type</th>
<th># of subject t</th>
<th>total # of images</th>
<th>Annotation effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name list selection</td>
<td>M</td>
<td>500K</td>
<td>50.00 million</td>
<td>3 months</td>
</tr>
<tr>
<td>2</td>
<td>Image downloading</td>
<td>A</td>
<td>9244</td>
<td>12.94 million</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>Face detection</td>
<td>A</td>
<td>9244</td>
<td>7.31 million</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>Automatic filtering by classification</td>
<td>A</td>
<td>9244</td>
<td>6.99 million</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>Near duplicate removal</td>
<td>A</td>
<td>9244</td>
<td>5.45 million</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>Final automatic and manual filtering</td>
<td>A / M</td>
<td>9131</td>
<td>3.31 million</td>
<td>21 days</td>
</tr>
</tbody>
</table>

MS-1M-Celeb[6] presents a systematic identities verification procedure. However, there is much less effort in the training set preparation as the introduction chapter shows.

Efforts on purifying MS-Celeb-1M are kept going on after the release of the original database. People from ArcFace[5] achieve state-of-the-art FR performance in MegaFace[17] by training on their own purified version of Microsoft face database. Almost 60% of images and 15% identities are removed for sake of purity enhancement.

2.3 Face Recognition Models

As Lapuschkin et al.[21] shows the feasibility of replacing fisher vectors by deep neural networks. The fisher vector (a representative hand-made features) used in [32] basically tends to focus more on local contextual features instead of general structural features. However, those critical features are explored better by deep learning model. That explain why our project focus on deep learning approaches.

Facenet[35] directly maps image instances into Euclidean space which provides convenience for applying direct clustering on the features of faces. Sphereface[29] interprets the embedding extracted in the view of angular distance with
CHAPTER 2. LITERATURE REVIEW

effective margins. Furthermore, the general margins in embedding space are elaborated and analyzed by [50]. CosFace [43] and ArcFace [5] are new modified version of angular losses which are the current state-of-the-art models. More details are elaborated in section 3.3.

2.4 Benchmark for Face Recognition

In the literature, LFW [13, 14] is a classical and prevalent dataset for face recognition performance, and Mega-face [17] is a comparatively new but more challenging benchmark due to more variations and distractors. Recently MS-Celeb-1M [6] has been proposed with more authenticated testing set assisted with knowledge-based annotation. A more comprehensive comparison is elaborated in section 3.4.
Chapter 3

Methodology and Approach

Three phases and one additional reflection phases are included in the whole project. The first one is a pipe-lining process of constructing 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. After getting the purified dataset, model evaluation revealed several problems in the previous dataset construction process. Therefore we added one more phases as the reflection phase to give detailed analysis.

3.1 Original Dataset Construction

Data Crawling

Image data were scratched from Google Image relied on a newly collected celebrity list. The celebrity list contains about 50K identities where large variances of attributes can be found. For instance, occupations of celebrities are significant metrics of identity variation according to various dataset.
The MS-Celeb-1M (Figure 3.1) strengthens its superioress in this area to whose celebrities are only from movie industry (actors, actresses and producers). In this scope, the celebrity list of our dataset is expected to have sufficient variation. However, it was the web search algorithms used by Google affect the labeling Independence of our dataset, which stimulated us to change our exclusively self-made database to database augmentation to existed dataset. Details are covered in the Chapter 4.

Based on the designed celebrity list, top 100 in Google Image are retrieved from original websites. Relying on search engine ranking is a prevalent method across research and industrial institutions. During the process of image crawling, the influence of anti-robot program and potential network limitation should be paid enough attention to.
Dataset Preprocessing

When all of the raw data are retrieved from Google, a critical step of face preprocessing should be done to fit all training data into the same domain of distribution. Existing models like ACF\cite{53} and MTCNN\cite{56} can be utilized in this task. ACF is traditional techniques with wide application as well as its comparative stability. And the main stream of face alignment uses MTCNN\cite{56} to get five points as landmarks and then unified all images distribution to target landmarks position.

Concretely, MTCNN\cite{56} utilizes a group of three deep neural networks called Proposal Network (P-Net), Refine Network (R-Net) and Output Network (O-Net) to get final cropping bounding box as well as five crucial landmarks. Our final alignment adopted five landmarks and then use similarity transformation\cite{8} to transform those landmarks to target position by affine transformation. Mathematically, for a specific crucial landmark $[x, y]^T$ and target position after transformation $[u, v]^T$, the similarity transformation is represented as follow,

$$
\begin{bmatrix}
    u \\
v
\end{bmatrix} = \begin{bmatrix}
    m_1 & -m_2 \\
m_2 & m_1
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix} + \begin{bmatrix}
t_x \\
t_y
\end{bmatrix}
$$

where $[t_x, t_y]^T$ is the translation parameter vector, $m_1 = s \cos \theta$ and $m_2 = s \sin \theta$, and $\theta$ and $s$ are rotation degree and scaling parameter, respectively. Essentially, similarity transformation is is an affine transformation by least square estimate of the transformation parameters.
14

From the comparison above, better positional consistence can be found on similarity transformation than other methods.

3.2 Dataset Filtering

Model Selection  Based on current performances on face recognition described in the Chapter 2, it is empirically reliable to select FaceNet [35] and SphereFace [29]
as model filters. Both of them are trained models having convenience for model evaluation and direct application. What’s more, models such the ones driven by center loss\cite{46} can also be harnessed for increasing the variations of filters. Ensemble learning is also a good choice. It is more likely to achieve better filtering performance with more distinctive model filters. Two possible strategies for model fusions are available: 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 critical benchmark dataset is to be built to evaluate and boost the filtering performance of selected models. Specifically, 100 identities are selected randomly from original dataset. After that, images in each identity folder are labeled with either positive or negative according to the human judgment on whether they should be filtered out or not. The labeling process is to receive enough rounds of examination and cross-checking to reserve inter-class variances and reduce ambiguities to the lowest. Aside from the intention of evaluating existing models filtering performance, we can reach another target, which is to find proper parameters for the clustered-based filtering of the model. The parameter tuning process can be easily automated by the grid search\cite{22}.

**Filtering process** The filtering phase involves two parts: (1) Feature (or embedding) extraction by models with parameters chosen from last step and
(2) Clustering image instances with respect to the corresponding parametric algorithm. Feature extractions can be conducted with a built-in program in Caffe\cite{16} framework. The build-in program is able to extract data from any specific layer in the model at any iteration. The clustering process utilizes the method proposed in \cite{33}, which is a advanced modified version from K-means\cite{31} clustering. By tuning two key metrics \textit{delta} and \textit{density}\cite{33}, it can find the density peak which is highly likely to be the feature center of the target identity.

\textbf{Post Quality control} As a quality control insurance, the evaluation of filtering should be done at the end of iterative filtering process. A set of randomly selected data from the output is evaluated based on its remaining noises. A threshold should be set as the final pass of the filtering process with carefully examinations.

\section{3.3 Models as Automatic Filters}

As Deng, Guo, and Zafeiriou\cite{5} summarizes, three key elements are explored in current literature on FR task. The first one is dataset, which is also the focus of our final year project; the second one is the deep architecture (usually CNN-based architectures\cite{20, 37, 38}) which can be generalized to many other tasks for feature extraction; the last one is the loss function which drives the total model to the its peak performance. The main difference between the models we explored is the loss function and more details are elaborated below.
CHAPTER 3. METHODOLOGY AND APPROACH

FaceNet (Triplet Loss)

FaceNet[35] adopts a loss function called "triplet loss". Three instances forms as a unit of calculation. The anchor point is firstly selected, then an instance with same class with the anchor and an instance with different class to the anchor are chosen to form the triplet. As figure 3.3 shows, the loss forces smaller inner-class distance and larger inter-class distance at the same time.

Figure 3.3: Initial state and the learning process driven by triplet loss.

Concretely, the mathematical expression for triplet loss is as follows,

$$\mathcal{L}_{\text{triplet}} = \sum_{i} [||f(Q^a_i) - f(Q^p_i)||_2^2 - ||f(Q^a_i) - f(Q^n_i)||_2^2 + \alpha]_+$$

Center Loss

Wen et al.[46] proposed an innovative but intuitive design of loss of metric learning. In stead of learning global manifold by the direction of local optimal solution (such as triplet loss[35, 27]), they make a center for every class and optimize them by enlarging the distances between class centers as well as reducing in-class data points’ distances to their class centers.
Concretely, the center loss is defined as follow[46]:

Concretely, the mathematical expression for center loss is as follows,
\[
\mathcal{L}_{center} = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2,
\]
where \(c_{y_i} \in \mathbb{R}^d\) denotes the \(y_i\)-th class feature center, and the update of the center follows the expression from iteration \(t\) to \(t+1\),

\[
c_{j}^{t+1} = c_{j}^{t} - \alpha \Delta c_{j}^{t} = c_{j}^{t} - \frac{\alpha \sum_{i=1}^{m} \delta(y_i = j) \cdot (c_{j} - x_{i})}{1 + \sum_{i=1}^{m} \delta(y_i = j)}
\]

The final loss is the linear combination of softmax loss and center loss, which is

\[
\mathcal{L} = \mathcal{L}_{softmax} + \lambda \mathcal{L}_{center}
\]

Centerloss [46] improves the softmax with a even constrained decision boundary as the toy MNIST [23] example shown in the Figure 3.4.

**SphereFace (Angular Loss)**

SphereFace [29] firstly proposes an angular expression from traditional softmax loss function which empirically leads to better depicted features. Softmax loss is the most traditional loss that gives posterior probabilities for each class. The mathematical expression is in the following,

\[
P_i = \frac{e^{W_i^T x + b_i}}{\sum_j e^{W_j^T x + b_j}}
\]

\[
\mathcal{L}_{softmax} = \frac{1}{N} \sum_i - \log\left(\frac{e^{W_i^T x + b_i}}{\sum_j e^{W_j^T x + b_j}}\right)
\]
Figure 3.4: Comparison between two models in MNIST\cite{23}, the left one is center loss feature space, the right one is the plotting for 1st and 2nd activation layer which shows the basic feature representation of figure instances.

By normalizing parameters $W$, SphereFace successfully changes the euclidean distance on a sphere to angular distance intuitively. Mathematically,

$$W^T_ix + b_i \Rightarrow ||W_1|| * ||x|| * \cos(\theta_i) + b_i$$

SphereFace adds another parameter $m$ as the margin to its degree term $\theta_i$. Further more, the special modified function $\phi(\theta_{y_i,i}) = (-1)^k \cos(m\theta_{y_i,i}) - 2k$ replaces the naive cosine distance expression to make it monotonically decreasing. The 2-Dimension and 3-Dimension visualizations of SphereFace are shown in Figure 3.4.

$$\mathcal{L}_{\text{angular}} = \frac{1}{N} \sum_i \left[-\log\left(\frac{e^{||x_i||\phi(\theta_{y_i,i})}}{e^{||x_i||\phi(\theta_{y_i,i})} + \sum_{j \neq y_i} e^{||x_i||\cos(\theta_{j,i})}}\right)\right]$$
CHAPTER 3. METHODOLOGY AND APPROACH

Figure 3.5: The left column 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.

**CosFace and ArcFace**

After SphereFace\(^{29}\) showed the potency of angular distance, more research attention are put into this new kind of loss set. CosFace\(^{43}\) from Tencent AI Lab gives a another variation of angular loss, moving the margin term from the multiplicative position to additive position. ArcFace as a subsequent loss type move the addictive margin inside the cosine function. Mathematical
Table 3.1: Mathematical comparison between different angular distance in binary case

<table>
<thead>
<tr>
<th>Loss Functions</th>
<th>Decision Boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softmax</td>
<td>((W_1^T - W_2^T)x + b_1 - b_2)</td>
</tr>
<tr>
<td>W-Norm Softmax</td>
<td>(|x| (\cos(\theta_{W_1,x}) - \cos(\theta_{W_2,x})))</td>
</tr>
<tr>
<td>SphereFace</td>
<td>(|x| (\cos(m\theta_{W_1,x}) - \cos(m\theta_{W_2,x})))</td>
</tr>
<tr>
<td>F-Norm SphereFace</td>
<td>(s (\cos (m\theta_{W_1,x}) - \cos (\theta_{W_2,x})))</td>
</tr>
<tr>
<td>CosineFace</td>
<td>(s (\cos (\theta_{W_1,x}) - \cos (\theta_{W_2,x}) - m))</td>
</tr>
<tr>
<td>ArcFace</td>
<td>(s (\cos (\theta_{W_1,x} - m) - \cos (\theta_{W_2,x})))</td>
</tr>
</tbody>
</table>

Figure 3.6: Decision margins of angular loss functions under binary classification case[5].

expressions are shown in Table 3.1 and graphical demonstrations can be viewed in the Figure 3.6.

This set of loss functions demonstrates the effectiveness of angular distance in FR problems. Essentially, angular loss is the transformation of euclidean distance under normalized weight parameters with more reasonable distinctive boundaries.
3.4 Evaluation of Face Recognition Performance

Training existing models on the constructed dataset

FaceNet\cite{Facenet} 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.

To make the evidence more well-rounded and convincing, tests should be conducted on more models including but not limited to contrastive loss\cite{ContrastiveLoss}, triplet loss\cite{TripletLoss} and angular distance\cite{AngularDistance, AngularDistance2, AngularDistance3}.

There can be several explanation for the problem of low evaluation result. The purity is the first concern because it cause the distortion of the inner class feature space. Moreover, labeling error in identity level can be another cause of it, since mapping two different identities to the same label or mapping one identity to different labels can affect the feature matching between classes.

Designing and training new models

New models would be explored and proposed based on deeper understanding of current metric learning models, which vary on network architectures and loss designs. Incorporating new feature extraction architectures such as ResNet\cite{ResNet1, ResNet2, ResNet3}, Inception-ResNet\cite{InceptionResNet1, InceptionResNet2}, DPN\cite{DPN} or DenseNet\cite{DenseNet} could bring features with even higher quality in terms of more crucial abstractions.
and higher sensitiveness. Trying innovative loss design would lead to better target function and appropriate updating directions.

**Testing with existing benchmarks**

After training with filtered data, models should be tested and evaluated by existing face recognition benchmarks. LFW [13, 14] provides a direct reference to most of the models in the literature. MegaFace [17] will evaluate model performances in a challenging environment. MS-Celeb-1M [6] helps to test the models potency on face data in the accurate real world. The following are the detailed description of these three prevalent benchmarks on Face Recognition and Face Verification.

**Labeled Faces in the Wild** LFW [14, 13] was firstly proposed as a facial database with unconstrained surroundings and backgrounds. It was the first benchmark that was pretty close to real situation of FR and was generally adopted as unified testing benchmark until even today. The dataset has 13,323 images and 5,749 identities, which was a great amount at the time it published. Although limited in today’s view of dataset size and quality, it provided sufficient variance for testing any model in a basic level, as surpass the bar of 99% verification performance should be a must for today’s FR models. One point that was unrecognized by our team is that LFW testing set is strictly cleaned by research teams in FR, so that the direct usage of the LFW dataset from its official website may not be able to achieve stunning performances in the literature. The core issue resorts in the detection and
alignment section, without which FR models cannot perform to a basic level in wild background and various distractors. Alignments result should be strictly examined manually with information searched assistance. Multiple aligned faces will be proposed by current algorithms like MTCNN\cite{56} and only valid identification can be used in the testing set.

**MegaFace**  MegaFace\cite{17} by group in University of Washington challenged the LFW dataset in terms of size and variation in pose, illuminance and age. Previously satisfactory models on LFW got much lower results on MegaFace. From the Table 1.1, it is easy to spot the difficulty of MegaFace compared with LFW. The core contribution of MegaFace is that it provided concepts of ”distractors”, introduce more unlabelled noises from popular image websites. These images from a large identification set provide even larger variance than LFW thus involve more difficulties but useful generalization effect.

**MS-Celeb-1M**  The benchmark from Microsoft\cite{6} is treated as the most accurate and comprehensive benchmark due to its systematic, rigorous benchmark construction process with intensive manual checking. All the identifications have been reviewed and matched with information retrieval techniques so that their images are ensured to match with their identities. What’s more, the diversity of identities across countries, races, occupations etc. can be treated as the highest among all existing benchmarks from the particular section in their paper (Also shown as Figure 3.1).
Chapter 4

Experiments and Results

The results chapter is divided into two separate sections. The first one is the dataset building process without any data from other existed dataset like CASIA-WebFace\cite{54} and VGG\cite{32}. After a rigorous evaluation, another dataset is provided based on the reflection on the first one. Section 2 will introduce the second dataset which is augmented by the data from existing datasets.

4.1 HKUFace: exclusively self-constructed dataset

Data Collection and Preprocessing

It took about one month to finish the data scratching task which was divided into two subsections, celebrity selection and image scratching. Then a Bash script drove the face data preprocessing automatically. The detection and alignment took about a month.
40K identities from the Wiki celebrity page\cite{48} were scratched by Python scripts with for its advantage in web programming. Those lists and meta-data will be available after the publication of the project. After that, the image crawling followed by the method in Chapter 3 was conducted by the parallel work of four computers (i5, 4G RAM) in the lab 311 in Haking Wong Building, HKU. The parallel process reduced the cost of this task by around 75%.

One innovative point that can be mentioned is that the crawler program solved the issue of anti-robot counterpart from Google and network limitation by employing strategic slowing down. By stopping around 0.1 second during large requests sent by the browser simulator, we successfully got around the network control in the CS Department and Google reCAPTCHA (an anti-robot system). On the one hand, CS Department did not hinder a lower volume by a specific threshold, on the other hand, random sleeping made the requests more like a natural human in the view of Google anti-robot system. Detailed code are available on Github\footnote{https://github.com/whcacademy/imageDownloader}.

The parallel method was also utilized in the face alignment and detection process. Multiprocessing played the key role in this manner because all preprocessing processes were conducted in one server. Since the detection software required a special environment, computers in the lab 311 is impossible to perform this task. The results show an unexpected improvement on the purity of dataset. It was found to contribute the filtering process as 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.
Dataset filtering

As the key of this project previously, each step of dataset filtering received great attention. The project was in this phase for about 5 months (from Oct. 2017 to Feb. 2018) and several achievements made are described in the following.

Benchmark Construction

The benchmark, which acts as the indicator of the filtering performance, was built by the joint work of the team of two. 100 identities, 6,000 images (around 40% of images were filtered out by preprocessing step) were randomly extracted, labeled and cross-checked to compose the benchmark. After the judgment and labeling by human being, 80 of 100 identities were selected as effective and valid ones. The rest were filtered because either not have enough images or contain too much noise. To facilitate the filtering measurement process, we deleted these invalid identities. 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
Model Evaluation

Three iterations of filtering were done utilizing FaceNet model. After every iteration, the purity scores were calculated on the benchmark. FaceNet pre-trained model is declared to have 99.63% verification accuracy on LFW dataset, which was a proper (previous) state-of-the-art model filter. The model was harnessed to extract 128-dimensional features from instances in benchmark dataset. Then an unsupervised clustering algorithm was applied on these features to remove noisy images.

1. **First Iteration: Cluster-based filtering** As the first iteration of the filtering experiments, the strategy was to remove all other clusters classified by the density peak clustering algorithm [33]. A major cluster with majority instances in it should be reserved. An example of the clustering output is shown in Figure 4.1. Nonetheless, those red points, which should be preserved by the filtering process range from low to high values, were discarded in this iteration. It revealed that this choice of filtering rule may not be appropriate.

2. **Second Iteration: Density-based filtering**

   In first iteration, problem were founded that (1) to filter cluster out was not a proper way for de-noise process; (2) some positive images with high density values were unexpectedly discarded since they belonged to other clusters and (3) noisy images happened frequently to blend into the major cluster and were reserved, which may be hazard to training performance. To tackle these issues, a density-based reference method
Figure 4.1: Illustration of clustering problem

was brought out in the second iteration of filtering. Features remained unchanged in this iteration. The density threshold was chosen to be $8$ and images with lower density were removed. This method improves the mean purity of benchmark set, increasing around 5 percent, from 58% to 63%. Since the target purity of our dataset is at least above 90%, new filtering approach needed to be proposed.

3. Third Iteration: Uniformly sized images

To increase the purity of our dataset and find out the key problem, we put attention to the feature extraction procedures. After detection and alignment, extracted parts of the faces were utilized as refined images for feature extraction. However, the random cropping were applied due to varied size of images. If the image size is quite large, chances are higher that the cropped images contain partial faces or even no faces.

The solution to this problem was to firstly unify the image size. By preserving the central $224 \times 224$ square of aligned images, the facial
parts of images are mostly reserved well. Therefore, the quality of features were better without the affect of the random cropping. The third iteration of filtering made a pretty satisfactory performance, which is 90%. The distribution of filtering outcome is visualized by box-plot in Figure 4.2. More than three quantiles of identities have above 90% purity value and over two quantiles of identities achieved 100%. Some data points in the graph have low values because their raw datasets are originally jumbled with few correct images. Table 4.2 shows the summary of three purity after filtering.

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

Table 4.2: Purity level of benchmark dataset

We wrapped up our cleaning process until the third round filtering. On the
one hand, purity boosting is getting even harder if we try more hand-crafted
features or more features extracted by deep learning models.

Dataset Evaluation

The evaluation of exclusively self-crawled HKUFace followed by the filtering
in the last section. Unexpectedly, several evaluation results with further
improvement and analysis still make the dataset sub-optimal compare with
existed dataset. Below are the evaluation results as well as reflection analysis.

Evaluation Configurations

To show the dataset evaluation results in a more clear and convenient way,
dataset configurations are firstly defined as follows:

- Option **HKUF-LA-full**: Exclusively self-crawled HKUFace, using loosely
  ACF\[53\], all data.

- Option **HKUF-LA-1k**: Exclusively self-crawled HKUFace, using loosely
  ACF, random sampled 1k identities’ data.

- Option **HKUF-LA-5k**: Exclusively self-crawled HKUFace, using loosely
  ACF, random sampled 5k identities’ data.

- Option **HKUF-LM-1k**: Exclusively self-crawled HKUFace, using loosely
  MTCNN\[56\], random sampled 1k identities’ data.

- Option **HKUF-LM-5k**: Exclusively self-crawled HKUFace, using loosely
  MTCNN, random sampled 5k identities’ data.
• Option **HKUF-TM-1k**: Exclusively self-crawled HKUFace, using tightly MTCNN, random sampled 1k identities’ data.

• Option **HKUF-TM-5k**: Exclusively self-crawled HKUFace, using tightly MTCNN, random sampled 5k identities’ data.

• Option **HKUF-TM-1k-RGray**: Exclusively self-crawled HKUFace, using tightly MTCNN, random sampled 1k identities’ data, randomly change to gray-scale from RGB setting.

• Option **HKUF-TM-5k-SS**: Exclusively self-crawled HKUFace, using tightly MTCNN, random sampled 5k identities’ data with 30 images per identity.

• Option **HKUF-TM-782**: Exclusively self-crawled HKUFace, using tightly MTCNN, random sampled 782 identities’ data.

• Option **HKUF-TM-782-MF**: Exclusively self-crawled HKUFace, using tightly MTCNN, same 782 identities’ with **HKUF-TM-782** but with one more round manual strict filtering.

The first direct evaluation is conducted by training SphereFace \cite{29} on HKUFace dataset with above settings. **Table 4.3** shows the setting of the SphereFace, where B means batch size and Convolution layer settings are denoted as ”Conv-(number of filters)-(size of filters)-(stride of filters)-(padding)”. As for residual configurations, the block inside the residual is described and the outer pairwise plus is ignored. Moreover, batch normalization\cite{15} and activation layers are ignored in this table.
Table 4.3: SphereFace structure in dataset evaluation

<table>
<thead>
<tr>
<th>Layer configurations</th>
<th>output dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>input 112, 96, 3</td>
<td>B<em>3</em>112*96</td>
</tr>
<tr>
<td>Conv-64-3-2-1</td>
<td>B<em>64</em>56*48</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-64-3-1-1</td>
<td>B<em>64</em>56*48</td>
</tr>
<tr>
<td>Conv-128-3-2-1</td>
<td>B<em>128</em>28*24</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-128-3-1-1</td>
<td>B<em>128</em>28*24</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-128-3-1-1</td>
<td>B<em>128</em>28*24</td>
</tr>
<tr>
<td>Conv-256-3-2-1</td>
<td>B<em>256</em>14*12</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-256-3-1-1</td>
<td>B<em>256</em>14*12</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-256-3-1-1</td>
<td>B<em>256</em>14*12</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-256-3-1-1</td>
<td>B<em>256</em>14*12</td>
</tr>
<tr>
<td>Conv-256-3-1-1</td>
<td>B<em>256</em>14*12</td>
</tr>
<tr>
<td>Conv-512-3-2-1</td>
<td>B<em>512</em>7*6</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-512-3-1-1</td>
<td>B<em>512</em>7*6</td>
</tr>
<tr>
<td>Residual Block</td>
<td></td>
</tr>
<tr>
<td>Conv-512-3-1-1</td>
<td>B<em>512</em>7*6</td>
</tr>
<tr>
<td>Fully Connected 512</td>
<td>B*512</td>
</tr>
<tr>
<td>AngularLoss</td>
<td>B<em>512</em>2</td>
</tr>
</tbody>
</table>

Results and Ablation Study

The first trial of direct evaluation resulted in almost no loss reduction. Since it may caused by the general hardness of training with classes over 10k which Microsoft group also mentioned in [6]. Then similar to what Guo et al. [6] did, we extracted 1k and 5k identities randomly. Models were converged in these two settings but the performance are 20% lower than state-of-the-art, which showed problems in the initial HKUFace dataset. More details are shown in the Table 4.4.

Then one problem of loose alignment was founded by comparison of loose
aligned and tight aligned dataset by MTCNN, which is shown in the Table 4.5. From that table, we can finally get 10% performance enhancement compared to original dataset aligned by ACF loosely. However, there was still around 12% difference between our trained model and the state-of-the-art.

Table 4.5: Evaluation for HKUFace with Loosely/Tightly MTCNN alignment

<table>
<thead>
<tr>
<th>Dataset Option</th>
<th>LFW Verification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKUF-LM-1k</td>
<td>82%</td>
</tr>
<tr>
<td>HKUF-LM-5k</td>
<td>84%</td>
</tr>
<tr>
<td>HKUF-TM-1k</td>
<td>85%</td>
</tr>
<tr>
<td>HKUF-TM-5k</td>
<td>87%</td>
</tr>
</tbody>
</table>

After the exploration of the alignment, we moved our attention on the real purity influence of our dataset. The basic checking showed the highly pure inside most of the folders. However, we still conducted a second-round manual filtering process with zero tolerance filtering, i.e. any images suspected to be another id will be removed and only images fully confirmed will remain. 782 identities were extracted to do the most strict filtering and SphereFace was trained on them. The comparison in the Table 4.6 shows that the most strict manual filtering had no performance boost.

Another two factors were treated as no significance after corresponding control experiments: the gray-scaled images & color images, and same size per class
Table 4.6: Experiment on strictly cleaned subset

<table>
<thead>
<tr>
<th>Dataset Option</th>
<th>LFW Verification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKUF-TM-782</td>
<td>82%</td>
</tr>
<tr>
<td>HKUF-TM-782-MF</td>
<td>82%</td>
</tr>
</tbody>
</table>

or different size per class.

Another insight was found during the manual filtering of the subset of HKUFace as the paragraph above. The images in one id folder may not belong to the id itself. As Figure 4.3 shows, the filtered output of id "Jordon Phil" is actually "Michael Jordan" because of the majority of top 100 from Google Image Search are Mr. Jordan because his popularity and high search frequency. Google’s ranking is not purely based on the query matching. The advanced information retrieval and recommendation system will consider popularity and previous search history to rank the output. This problem essentially caused great labelling errors of folders. For example, there maybe more identities whose last name is "Jordan" or similar one to it. Also another case was found that in the folder called "Hilton, Justin", images remained were all from "Justin, Bieber".

After that observation, a set of cleaning methods were applied to HKUFace. For example, features extracted from every folder were calculate pair-wise distances in order to find the nearest classes and strict removal were applied on both classes. Table 4.7 shows Representative pairs that confirmed to be the same. However, the major cause of them are the repetition errors from Wikipedia which were not concerned and removed during the initial identity collection phase.
CHAPTER 4. EXPERIMENTS AND RESULTS

Figure 4.3: Illustration of Google search output for key ”Jordon, Phil”, where GREEN boxes belongs to Jordon, while RED boxes belongs to Michael Jordan”

Table 4.7: Sample pairs that confirmed to be the same in HKUFace

<table>
<thead>
<tr>
<th>Sanjana Gandhi (now Pooja Gandhi)</th>
<th>Pooja Gandhi (born Sanjana Gandhi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rogers, Roy</td>
<td>Roy Rogers</td>
</tr>
<tr>
<td>Carr, Greg</td>
<td>Carr, Gregg</td>
</tr>
</tbody>
</table>

Also, the ArcFace\textsuperscript{5} paper gave hint as Table 2.1 shows. VGG group used more than three months to select the original identities and there must be a reason for such intensive manual process. Although there is no information about how they select, reasonable estimation can be in three aspects: judgment (1)whether the majority of the search result matches the identity overall; (2) whether there is enough images in the target search engine and (3) whether the identity is duplicated with previous identities.
4.2 CASIA-WebFace, CASIA-VGG

Although several cleaning step were applied, the models trained on HKUFace remained around 87% without further improvements. One reason could be the labelling errors that were not cleaned totally, or other potential factors remained existed and have not been found out. Then the main research direction was changed to reviewed CASIA-WebFace as well as CASIA-VGG by our PhD leader, Weifeng. The review process included the validation of the CASIA-VGG and comparison between two dataset.

Evaluation Configurations

Similar to the last section, the configuration of exploration of existed dataset are described first.

- Option **CASIA-UF-RF**: Unfiltered CASIA-WebFace\textsuperscript{[54]}, using similarity transformation of five points from the Github repo, all data.

- Option **CASIA-UF-ST**: Unfiltered CASIA-WebFace\textsuperscript{[54]}, using similarity transformation of five points detected by MTCNN\textsuperscript{[56]}, all data.

- Option **CASIA-F-ST**: Filtered CASIA-WebFace, using similarity transformation of five points detected by MTCNN.

- Option **CASIA-UF-ST-1k**: Unfiltered CASIA-WebFace, using similarity transformation of five points from the Github repo, random sample 1k identities.
• Option **CASIA-UF-ST-5k**: Unfiltered CASIA-WebFace, using similarity transformation of five points from the Github repo, random sample 5k identities.

• Option **CASIA-VGG-LA**: Filtered and merged CASIA-WebFace and VGG, using loosely ACF, all data.

• Option **CASIA-VGG-LM**: Filtered and merged CASIA-WebFace and VGG, using loosely MTCNN, all data.

• Option **CASIA-VGG-TM**: Filtered and merged CASIA-WebFace and VGG, using tightly MTCNN, all data.

• Option **CASIA-VGG-ST**: Filtered and merged CASIA-WebFace and VGG, using similarity transformation of five points detected by MTCNN, all data.

**Evaluation Results**

Firstly, the reproduction of declared performance was done first. Moreover, sub-sampling of training configurations were explored and Table 4.8 shows the results. It can be found that only 1k classification problem can make the test performance pretty stunning, even larger identities can only improve the performance of model less.

Th evaluation of benchmark in code on Github\(^2\) revealed one missing point about previous alignment. The missing point was that main stream in FR

\(^2\)https://github.com/clcarwin/sphereface_pytorch
Table 4.8: Reproduction of Sphereface on CASIA-WebFace

<table>
<thead>
<tr>
<th>Dataset Option</th>
<th>LFW Verification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA-UF-RF</td>
<td>99.2%</td>
</tr>
<tr>
<td>CASIA-UF-RF-1k</td>
<td>90.4%</td>
</tr>
<tr>
<td>CASIA-UF-RF-5k</td>
<td>94.6%</td>
</tr>
</tbody>
</table>

uses MTCNN\cite{56} to get five facial points rather than direct alignment. Then those landmarks transformed into five predefined reference points in the target image by similarity transformation\cite{8}. Then several experiments were conducted as Table 4.9 shows.

Table 4.9: Experiment of influence of similarity transformation

<table>
<thead>
<tr>
<th>Dataset Option</th>
<th>LFW Verification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA-VGG-LA</td>
<td>91%</td>
</tr>
<tr>
<td>CASIA-VGG-LM</td>
<td>95%</td>
</tr>
<tr>
<td>CASIA-VGG-TM</td>
<td>98.62%</td>
</tr>
<tr>
<td>CASIA-VGG-ST</td>
<td>\textbf{98.93%}</td>
</tr>
</tbody>
</table>

The comparison was also conducted between unfiltered and filtered CASIA-WebFace, the verification error reduced by 18% as Table 4.10 shows.

Table 4.10: Experiment of influence of image filtering

<table>
<thead>
<tr>
<th>Dataset Option</th>
<th>LFW Verification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASIA-F-ST</td>
<td>\textbf{98.68%}</td>
</tr>
<tr>
<td>CASIA-UF-ST</td>
<td>98.40%</td>
</tr>
<tr>
<td>CASIA-UF-RF</td>
<td>99.20%</td>
</tr>
</tbody>
</table>

From this testing set, the influence of the purity enhancement of the dataset to the model is positive, but a concern raised that whether the person published the code on Github used CASIA-WebFace labelled by manual checking. Because
during the alignment of the benchmark, misalignments were found on images with multiple faces. With no information of identity, it is almost impossible for an alignment model to choose the right faces.
Chapter 5

Conclusion and Future Works

In this final report, detailed information about the project and results are presented.
Current research trends in the face recognition show the importance of the dataset in terms of its size and variation, which acts as the main motivation of the project. Literature review chapter which covers related works gives hints on the general procedure of constructing a dataset with filtering opportunity. By a careful composition with developed models, the filtering process has shown its first purity enhancement on the benchmark dataset. However, unexpected labelling errors contribute to the disturbance of decision manifold of trained models as examples shown in the Results. Then the current dataset and new filtered and merged dataset called CASIA-VGG was explored and tested. Although the performance is around 0.3% - 0.8% lower than the state-of-the-art models, models trained on CASIA-VGG still performed well with a minor multiple-face alignment issue. Overall, the HKUFace and CASIA-VGG give hints and heuristics about how to construct a facial dataset in an automatic manner.
Future works can be done in the following two aspects:

- Automatic id selection can be developed using several heuristics, such as the popularity estimation from web search results and duplicate removal by contextual similarity models driven by Natural Language Processing.

- Generalize automatic dataset construction from facial dataset to other dataset at least for fine-grained classification tasks.
Bibliography


[34] Frank Rosenblatt. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.


