國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

透過重新排序與近似最鄰近搜尋以改進基於殘差網路之人臉特徵擷取器

Improving ResNet-based Feature Extractor for Face Recognition via Re-ranking and Approximate Nearest Neighbor Search

蕭勝興

Sheng-Hsing Hsiao

指導教授：張智星 博士

Advisor: Jyh-Shing Roger Jang, Ph.D.

中華民國 108 年 6 月

June, 2019

口試委員審定書

致謝

在臺灣大學資訊工程研究所這兩年讓我學到很多，學到與大學中基礎理論不同的課程，許多課程都要實作出一個成果。也正是藉由這些實作的課程，我們接觸到很多不同的知識、不同領域的技巧，這些都是讓我們成長的養分。除了系上的課程外，感謝指導教授張智星老師，採取非常開明的態度，並不會強迫我們要往什麼方向走，而是想知道先走一步後會發生什麼事，想要知道我們是怎麼解決問題的。「先做看看再說！」是老師常常激勵我們的一句話，離開學校踏入職場後，我也會將這句話奉為圭臬並盡全力勇敢向前。

感謝我最強大的後盾，父母，在我學業上、生活上皆給予極大的支持與鼓勵，讓我能心無旁騖的在研究上，感謝父母給予的一切！另外，我也要感謝 MIR 實驗室一群敬業樂群的好夥伴，文澤、岳庭、宣伯、冠廷、庭宇、顥馨（依筆畫排序），不論在修課、計畫或最後的碩士論文都一起打拼，有大家的陪伴，讓我這兩年過得既充實又快樂。最後，也感謝實驗室的學長姐學弟，在我遇到瓶頸一起討論、給予幫助，讓問題迎刃而解；特別感謝范哲誠博士的諮詢，在論文遣詞用字上提供許多的意見。

摘要

深度殘差網絡 (ResNet) 是圖像分類和物件偵測中最先進的架構之一，當網路架構中最後一層被移除時，它可以當作一個良好的特徵向量抽取器。將人臉轉換成特徵向量後，一些任務例如人臉識別、人臉驗證，就可以使用一些距離測量方法來實現。我們提出了一個基於 ResNet 特徵提取器的人臉識別框架，並加上其他改善性能的步驟，包括人臉偵測，人臉對齊，人臉驗證/識別以及透過近似最近鄰搜索 (ANNS) 來重新排序。首先，我們在三個常見的人臉檢測資料集上評估兩種人臉偵測演算法，MTCNN 和 FaceBoxes，接著總結這兩種方法的最佳使用場景。其次，經過特定的預處理和後處理，我們的系統選擇基於 ResNet 的特徵提取器，並在 LFW 資料集中達到 99.33%的驗證準確度。第三，我們使用懲罰曲線來確定最佳配置並獲得良好的臉部驗證結果。最後，基於這篇論文提出的重新排序策略，我們的方法在大型類別間變異數據集 (在 CASIA-faceV5 資料集上提升 1.47%，在 CASIA-WebFace 資料及上提高 2.28%) 與大型類別內變異數據集 (在 FG-NET 資料集上提高 1.3%，在 CACD 資料集上提高 2.43%) 上都能使辨識率上升。

關鍵字： 深度殘差網絡，特徵向量抽取，人臉識別，人臉驗證，重新排序，近似最近鄰搜索。

Abstract

Deep residual network (ResNet) is one of the state-of-the-art architectures in image classification and object detection, which can serve as a robust feature extractor when the last layer is removed. Once the faces are embedded as feature vectors, tasks such as face recognition, verification and identification can be easily implemented using some distance measurements. This paper proposes a framework for face recognition based on feature extractor from ResNet, together with other steps for improving its performance, including face detection, face alignment, face verification/identification, and re-ranking via Approximate Nearest Neighbor Search (ANNS). First, we evaluate two face detection algorithms, MTCNN, and FaceBoxes on three common face detection benchmarks, and then summarize the best usage scenario for each approach. Second, with certain preprocessing and postprocessing, our system selects the ResNet-based feature extractor, which achieves 99.33% verification accuracy on LFW benchmark. Third, we use the penalty curve to determine the best configuration and obtain improved results of face verification. Lastly, based on the proposed re-ranking policy, our method not only boosts the accuracy in large inter-class variation datasets (1.47% and 2.28% improvement in rank-1 accuracy for CASIA-faceV5 and CASIA-WebFace respectively) but also in large intra-class variation datasets (1.3% and 2.43% improvement in rank-1 accuracy for FG-NET and CACD respectively).

Keywords: ResNet, feature extractor, face recognition, face verification, face identification, re-ranking, Approximate Nearest Neighbor Search.

Contents

口試委員審定書 i

致謝 ii

摘要 iii

Abstract iv

Contents v

List of Figures vii

List of Tables ix

1 Introduction 1

2 Related Work 5

2.1 Face Detection 5

2.2 Face Alignment 9

2.3 Face Identification and Verification 10

2.3.1 Face representation 11

2.3.2 Discriminative metric learning 12

2.4 Approximate Nearest Neighbor Search 16

2.5 Re-ranking for object retrieval 20

3 Method 21

3.1 Face Detection 22

3.1.1. MTCNN 22

3.1.2. FaceBoxes 23

3.2 Face Alignment 24

3.3 Face Feature Extractor 26

3.4 Approximate Nearest Neighbors Search 27

3.5 Re-ranking policy 29

4 Experiments 33

4.1 Evaluation of Face Detection 33

4.1.1 Benchmark evaluation 33

4.1.2 Face Detection Runtime Efficiency 36

4.1.3 Face Detection Conclusion 37

4.2 Evaluation of Face Verification 37

4.2.1 Performance on LFW 38

4.2.2 Performance on CASIA-FaceV5 and Helen dataset 39

4.2.3 Alignment and Average 41

4.3 Evaluation of Face Identification 43

4.3.1 Experiments on CASIA-FaceV5 43

4.3.2 Experiments on CASIA-WebFace 44

4.3.3 Experiments on FG-NET 45

4.3.4 Experiments on CACD 47

4.4 Parameter Analysis 49

5 Conclusions and discussions 51

Reference 52

List of Figures

Figure 1.1 Proposed Re-ranking framework (without average) 3

Figure 2.1 Triplet Loss 13

Figure 2.2 Geometry interpretation of A-Softmax loss 14

Figure 3.1 System flowchart 21

Figure 3.2 The pipeline of MTCNN that includes three-stage multi-task deep convolutional networks 23

Figure 3.3 Architecture of the FaceBoxes 24

Figure 3.4 Landmark estimates at different levels of the cascade of regressors ... 25

Figure 3.5 The process of rotating a face into canonical view 25

Figure 3.6 VGGFace2 template examples 27

Figure 3.7 Illustration of the Hierarchical NSW idea 28

Figure 3.8 An example of the re-ranking policy (without average) 32

Figure 3.9 An example of the re-ranking policy (with average) 32

Figure 4.1 Some examples of PASCAL Face dataset 34

Figure 4.2 Some examples of AFW dataset 34

Figure 4.3 Precision-recall curves on the PASCAL and AFW benchmarks 35

Figure 4.4 Evaluate on FDDB benchmark 35

Figure 4.5 Some examples of UTKFace 36

Figure 4.6 Some examples of LFW 38

Figure 4.7 ROC curves on LFW 39

Figure 4.8 Some examples of CASIA-FaceV5 and Helen dataset 40

Figure 4.9 Penalty curves of different False Accept penalties and different distance measurements 41

Figure 4.10 Performance comparison for different configurations 43

Figure 4.11 A visualization of size and estimated noise percentage of datasets 45

Figure 4.12 Some examples of FG-NET 46

Figure 4.13 The performance of mAP of our approach compared with SOTA algorithms on CACD 50

Figure 4.14 The impact of the parameter λ on rank-1 accuracy on large inter-class/ intra-class variation datasets 51

List of Tables

Table 4.1 Comparison of mAP and FPS on different approaches 37

Table 4.2 Performance evaluation on LFW dataset 39

Table 4.3 Comparison of different configurations on CASIA-faceV5 44

Table 4.4 Comparison of different configurations on CASIA-WebFace 46

Table 4.5 Comparison of different configurations on FG-NET 47

Table 4.6 Performance of different methods on FG-NET 47

Table 4.7 The Rank-1 accuracy of different methods on three subsets in CACD .50

Chapter 1

Introduction

Faces, a basic attribute that can distinguish one person to another. Face recognition is a popular area of research. There are many face recognition applications that have been deployed on many areas, such as face biometrics for payments, public security, authentication on devices, self-driving vehicles, etc.

Face recognition is a generic term that could imply either face verification and identification or both. Verification is the process of affirming that a claimed identity is correct by comparing the offered claims of identity with one or more previously enrolled templates. To put it another way, verification is trying to answer the question "*Is this person who they say they are?*". A synonym for verification is authentication, which checks whether or not the identity is in the database. Identification means the system need to determine the identity of the individual or tries to answer the question "*Who is this person?*". The facial information is collected and compared to all the identities in a database.

Both face verification identification and identification have three steps in general, 1) face detection 2) face alignment 3) face feature embedding. Face detection is a fundamental step for many facial applications. The essential properties of a face detector must be detecting faces with a high degree of variability in scale, pose, illumination and expression. Alignment is the module that localizes the facial landmarks of eyes, nose, lips, etc. These landmarks are used to align faces through rotation or scaling. Third, the face feature extractor encodes the facial information to a feature. These features are used to measure the degree of similarity between two faces. A good feature extractor should be robust to distinguish one from the other person.

Since 2012, AlexNet [47] has won the ImageNet competition by using some deep learning techniques and large data as input, Convolutional Neural Networks (CNNs) has become the mainstream for computer vision like image classification [92] and object detection [29]. Deep CNNs (DCNNs) are now widely used because they learn hierarchical levels of features that correspond to different levels of abstraction. Every level includes the different features, showing strong invariance to the face pose, scale, and lighting change. Therefore, increasingly large datasets are needed to provide a large

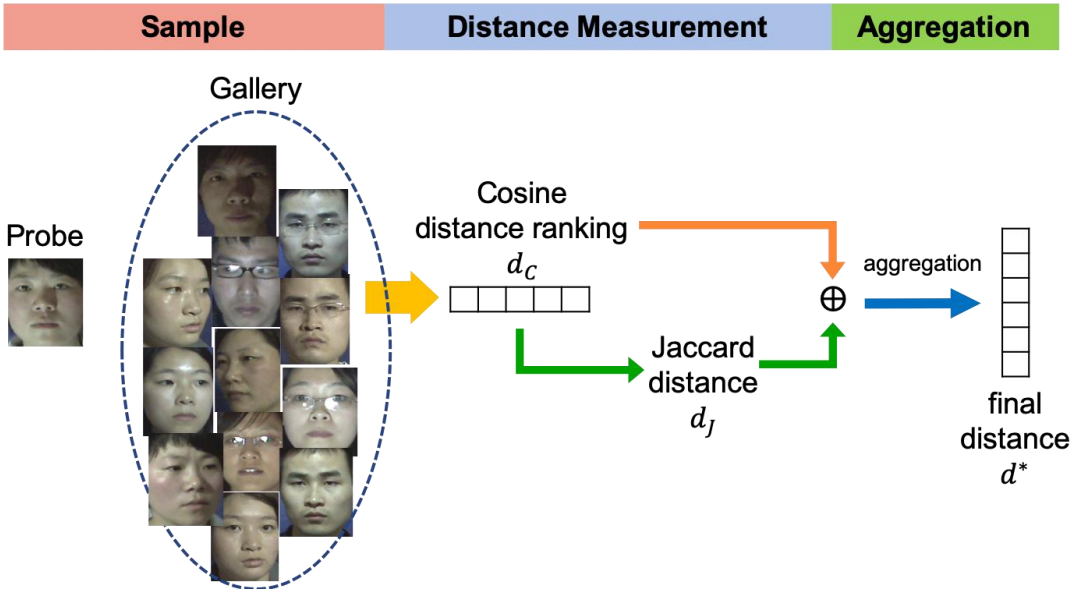


Figure 1.1: Proposed re-ranking framework (without average). Given the probe p and the gallery, the cosine distance d_C is applied to obtain the original ranking list, and then the corresponding Jaccard distance d_J can be computed from the k-nearest neighbors of each candidate in original ranking list. The final distance d^* is the combination of d_C and d_J, which is used to obtain the final ranking list.

number of faces featuring large variations. Widely used datasets include CASIA-WebFace [122], MegaFace [43], VGGFace [75], MSCeleb-1M [32], and WIDER Face [120].

The k-reciprocal nearest neighbor [78] is an effective solution to improve the accuracy on image retrieval. In this paper, inspired by k-reciprocal nearest neighbor, we develop a re-ranking method to calculate the final distance between two different similarity measurements. The framework of the proposed approach (without average) is illustrated in Fig. 1.1.

To summarize, the major contributions of this paper are as follows:

- By comparing two different face detectors, MTCNN [125] and FaceBoxes [126] on three major face detection benchmark [25, 38, 80], our work chooses the suitable detector in different scenarios.
- This work uses CASIA-FaceV5[1] dataset as the authorized users and Helen [49] dataset as the intruders to evaluate the performance of face verification in different configurations.
- We present an ANNS-based re-ranking method to improve the performance of face identification on large inter-class variation datasets (CASIA-FaceV5 and CASIA-WebFace) and large intra-class variation datasets (FG-NET[2] and CACD [13]).

The rest of the paper is organized as follows. First, we discuss recent researches in face detection (section 2.1), face alignment (section 2.2), face recognition (section 2.3), approximate nearest neighbors search (section 2.4) and re-ranking methods for object retrieval (section 2.5). Then, we present our face recognition system pipeline and re-ranking methods in section 3, and the experimental results are provided in section 4. Finally, the conclusion of this paper and discussion of future works are presented in section 5.

[1] http://biometrics.idealtest.org/
[2] https://yanweifu.github.io/FG_NET_data/index.html

Chapter 2

Related Work

In the face recognition pipeline, it contains face detection (section 2.1), face alignment (section 2.2) and face identification/verification (section 2.3). After that, we will discuss some approaches for Approximate Nearest Neighbors Search (ANNS) in section 2.4 and focus on some methods of re-ranking for object retrieval in section 2.5.

2.1 Face Detection

The first step in any face recognition/ verification system is face detection. Face detection algorithm will output all the locations of faces from an input image or a frame, most of the outputs are in the form of bounding boxes. A good face detection method should be robust to variations in pose, facial expression, rotation, illumination, scale, resolution, gender, age, ethnicity, occlusion, make-up, etc. Face detection methods can be divided into two different subcategories, one is based on 1) handcraft features and the other is 2) CNN-based face detector.

Handcraft-based methods are usually used for previous face detection systems. As the pioneering work, Viola-Jones face detector [104] utilized Haar-Like features and

AdaBoost learning to train cascade inference for face detection, which achieves real-time face detection. Many subsequent works are inspired by this detector, such as Liao *et al.* [55] and Yang *et al.* [117] proposed new local features, Pham *et al.* [76] and Brubaker *et al.* [11] designed new boosting algorithms to reduce the training time and Bourdev *et al.* [10] and Li, *et al.* [52] indicated that new cascade structures can speed up the detection process.

In addition to the cascade structure, some researches [67, 80, 115, 116] achieved remarkable performance in deformable part models for face detection tasks. These works need high computational expense and may usually require expensive annotation in the training stage to achieve better detection performance.

CNN-based methods can be traced back to 1994, Vaillant *et al.* [103] as the pioneer to train CNN in the manner of a sliding window to detect faces. Osadchy *et al.*[73] proposed a multi-tasks CNN system for simultaneous face detection and pose estimation. Recently, CNN achieves remarkable progress in many computer vision tasks and consequently, most of detection methods are replaced by CNN to learn deeper features. CNN-based face detectors are derived from object detection approaches and can be divided into twosub-classes: 1) region-based, and 2) sliding window-based.

1) Region-based methods first generate a set object-proposals and use CNN to classify each proposal as a face or not. R-CNN [29] obtained region proposals by selective search [102]. This approach has inspired some recent face detector such as multi-task learning algorithm HyperFace [84] and multi-purpose algorithm All-in-One Face [85]. Faster R-CNN [86], R-FCN [18] uses a Region Proposal Network (RPN) to generate region proposals. Besides, ROI-pooling [27] and position-sensitive RoI pooling [18] are applied to extract features from each region. Jiang *et al.* [40] use Faster R-CNN to detect faces and Li *et al.* [53] proposed a multi-task face detector based on Faster R-CNN framework. These improvements allow them to speed up by significantly reducing the number of face proposals and get better quality of face proposals.

2) Sliding window-based approaches directly output every face detection on multi-scale feature maps. Each detection is composed of the detection confidence and a bounding box. This approach does not need the separate region proposal step and thus is faster than region-based approaches. Ranjan *et al.* [83] created an image pyramid at multiple scales to detect faces and Farfade *et al.* [26] also fine-tuned CNN model for face / non-face classification tasks. Yang *et al.* [119] trained a series CNNs for facial attribute recognition to yield candidate windows of occluded faces. CascadeCNN [51] used a

cascade architecture built on CNNs for multiple resolutions. Qin *et al.* [79] proposed joint training to achieve end-to-end optimization for CNN cascade. MTCNN [125] adopted a cascaded CNNs that predict faces and landmark locations in a coarse-to-fine manner. The Single Shot Detector (SSD) [59] is also a multi-scale sliding window-based object detector, instead of using image pyramid, it utilizes the hierarchal structure of a single deep CNN. This one-stage face detection method has attracted more attention due to its higher inference efficiency and straightforward system deployment, like ScaleFace [121], S3FD [127], and Pyramidbox [100]. FaceBoxes [126] also inspired by the RPN in Faster R-CNN [86] and the multi-scale mechanism in SSD [59].

In addition to the development of improved face detection algorithms, the benchmark of face detection has been collected by large annotation datasets. Pascal person layout dataset, which is a subset from Pascal VOC [25]. It contains 1,335 faces from 851 images with large appearance variations. Annotated Faces in the Wild (AFW) [80], it has 205 images with 473 faces. FDDB [38] consists 5,171 faces in 2,845 images. MALF [118] dataset has a similar scale which has 5,250 images with 11,931 faces. WIDER Face [120] is a much larger dataset, it contains 32,203 images and labels 393,703 faces with a high degree of variability in scale, pose and occlusion. This dataset

has many tiny faces, some of face detectors mentioned before still struggle with recognizing small faces. Hu *et al.* [34] indicate that context is crucial and define templates that make use of massively large receptive fields.

2.2 Face Alignment

Face alignment is the process of transforming a face into some canonical view and usually done through the localization of facial keypoints [42]. Facial keypoints include the points around the eyes, nose, and mouth on a face. Bansal *et al.* [6] indicated it is important for face identification or verification.

There are two types of facial keypoints detection methods, model-based and regression-based. **Model-based** methods create a representation of shape during training and use model to fit facial landmarks during testing, include the classic Active Appearance Model (AMM)[68, 89] and Constrained Local Models (CLM) [3, 44].

Regression-based methods directly fit the image appearance with the target output. Kazemi *et al*. [42] showed an ensemble of regression trees can be used to estimate the face's landmark positions and achieve real-time performance with high quality. CFAN [124] used coarse-to-fine auto-encoder networks which cascades a few successive stacked auto-encoder networks. Kumar *et al*. [48] proposed an algorithm for extracting

key-point descriptors using CNN and also presented a face alignment algorithm base on these descriptors.

The capability of facial landmark detection often evaluates on the 300 Faces In-the-Wild database (300W) [87] which comprises of four sub-datasets: Annotated Faces in the Wild (AFW) [80], IBUG [87], LFPW [7], and Helen [49].

2.3 Face Identification and Verification

In this section, we will introduce some related works base on deep network architecture for face recognition. According to [50], there are two widely used paradigms: identification and verification.

In identification, the system has to learn the information from a specific set of identities. At the testing period, a new image or group of images is presented, and the target is to decide which of the gallery identities. By contrast, verification is to analyze two faces images and decide whether they are the same person or not.

There are two main parts of a face identification/ verification system: 1) Face representation; and 2) a classifier (for identification) or similarity measure (for verification).

2.3.1 Face representation

Deep networks have been shown to learn the difference between two different people. Huang *et al.* [35] proposed combining deep learning with traditional methods, such as Local Binary Patterns (LBP), achieving comparable results on the LFW [36] dataset.

In 2014, DeepFace [99] achieved state-of-the-art performance on the LFW benchmark. They used a proprietary face dataset consisting of four million faces belonging to more than 4,000 identities. Similarly, in 2015, Google's FaceNet [90] used over 200 million training data of 3 million people. It directly optimized the embedding itself by using triplets of roughly aligned matching / non-matching face patches. In the same year, VGGface [75] designed a procedure to collect a large-scale dataset from the Internet, and Parkhi *et al.* [75] trained the VGGNet [92] on this dataset and then fine-tuned the networks via a triplet loss function similar to FaceNet. This model achieved competitive results on both LFW and YTF [111] datasets.

The DeepID series models [93-95] used an ensemble of the smaller number of hidden neurons than DeepFace or FaceNet. There are four layers for feature extraction and their training data, CelebFaces+, is enlarged from CelebFaces [97] dataset, which contains 202,599 face images of 10,177 celebrities. The proposed features are extracted

from various face regions to form complementary and over-complete representations, which help it achieve almost as good as human performance on LFW dataset.

2.3.2 Discriminative metric learning

Learning a classifier or a similarity metric is the next step after obtaining face features. For verification, features of two faces from the same person should be similar while features from different persons should be dissimilar.

Derived from object classification networks such as AlexNet [47], cross-entropy based softmax-loss is widely used for feature learning. But in recent years, the softmax loss has been found to often bias in the sample distribution. Several modified studies have been proposed to explore discriminative loss functions for more robust face representation. Prior to 2017, Euclidean-distance-based loss was the mainstream approach, but some loss functions like Angular/cosine-margin-based loss were later designed to facilitate the training procedures.

Euclidean-distance-based Loss: Euclidean-distance-based loss is a metric learning method [109, 113] that maps images into Euclidean space to cluster the same person face representations while separate the different person face representations. The most intuitive approach is contrastive loss [96], using pairs of images to train a feature

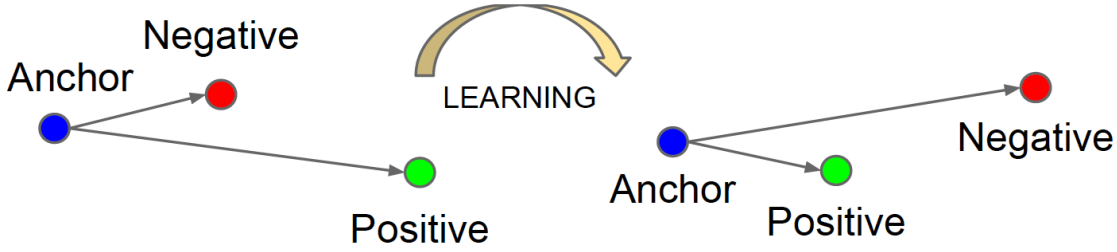


Figure 2.1: Triplet Loss. The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. [90]

embedding where positive pairs are closer and negative pairs are farther apart. But the margin parameters of the contrastive loss are difficult to choose.

The triplet loss, however, tries to enforce a margin between each pair of faces for one identity to all other faces (Fig. 2.1). FaceNet [90] from Google, Parkhi *et al.* [75], Swami *et al.* [88], Liu *et al.* [57] and Ding *et al.* [24] are also use triplet loss to embed features into a discriminative space and achieved improvements face verification.

Nevertheless, both contrastive loss and triplet loss require considerable time to achieve converge due to the ineffective sampling policies. Therefore, Wen *et al.* [112] introduced the Center loss, which provides a learned center for each class and penalized the distances between the deep features and their corresponding class centers. After center loss was proposed, many of its variations are also presented. Range loss [128] is used to handle long-tailed data and reduce overall intrapersonal variations while enlarging

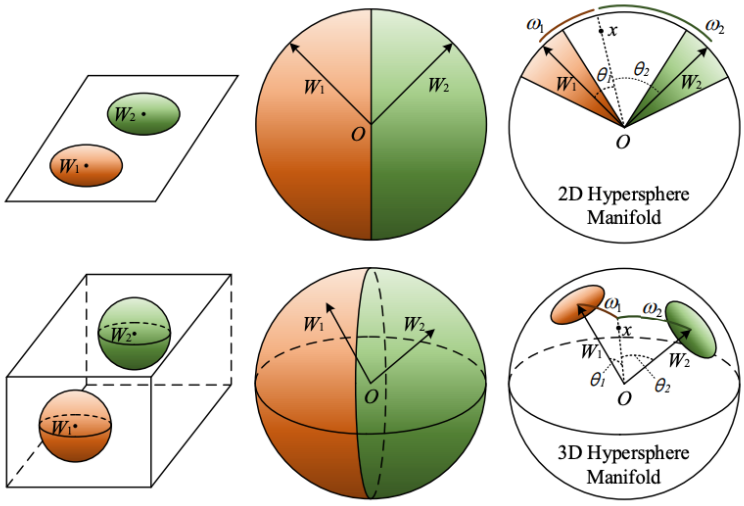


Figure 2.2: Geometry interpretation of A-Softmax loss. [60]

interpersonal differences simultaneously. Wu *et al.* [112] proposed a center invariant loss that penalizes the different between each center of classes. Deng *et al.* [23] proposed marginal loss which simultaneously minimizes the intra-class variances as well as maximizes the inter-class distances by focusing on the marginal samples.

Angular/cosine-margin-based Loss: In 2016, Liu, *et al.*[61] proposed the large-margin softmax (L-Softmax) loss, reformulated from original softmax loss. L-softmax makes the classification more rigorous to produce a decisive margin. The following year, SphereFace [60] proposed angular softmax (A-Softmax) loss to further normalize the weight by its L2 norm such that the normalized vector will lie on a hypersphere, and then the discriminative face features can be learned on a hypersphere manifold with an angular margin (Fig. 2.2). To solve the difficulty of optimizing L-Softmax and A-Softmax,

ArcFace [22] and CosFace [108] both added the angular margin, $cos(\theta + m)$ while Wang *et al.* [106] added cosine margin, $cos\theta - m$. They are easy to implement and can converge without softmax supervision.

In addition to the two losses mentioned above, there are many other approaches to modify the original softmax loss. Normface [107] explained the necessity of feature and weight normalization. Ranjan *et al.* [82] regularized the softmax loss with a scaled L2-norm constraint (L2-softmax loss) and achieved the SOTA on IJB-A [46]. COCO loss [62] optimized the cosine similarity among data and Ring loss [132] presented a simple soft normalization, where it gradually learns to constrain the norm rather than directly enforcing through a hard normalization operation. Crystal loss [81] restricts the features to lie on a hypersphere of a fixed radius and overcomes the limitations of the regular softmax loss.

2.4 Approximate Nearest Neighbor Search

Rapid and constant increases in data volumes significantly increases computation time and complexity. K-Nearest Neighbor Search (K-NNS) is a common approach for information retrieval, such as image feature matching in the large dataset [63] and semantic document retrieval [21]. A naïve brute force approach is to compute the distance between the query and every element in the dataset. But the complexity of the approach scales linearly with the number of elements in storage. There are several solutions to speed up the retrieval time when the dimension is small, such as Voronoi diagrams [77], kd-trees [8], metric trees [101], ball-trees [72]. However, many real-world applications are facing the high dimensional data and also demanded to achieve sub-linear efficiency, for example in computer vision.

Approximate Nearest Neighbors Search (ANNS) was proposed to overcome the "*curse of dimensionality*" [37], by providing a good approximation rather than the exact nearest element. ANNS can be performed efficiently and sufficiently useful for many practical problems and therefore attracting many studies. There are two ANNS benchmarks, [70] and ANN-benchmarks [4]. In addition, ANNS algorithms can be divided into three classes: **Hashing-based**, **Partition-based** and **Graph-based**.

Hashing-based: The approaches belonging this category project data to low-dimensional representation. Thus, each element could be encoded as a hash code. Locality sensitive hashing (LSH) is a basic hashing-based approach, referring to a family of functions (known as LSH families) to hash similar data (distance$< r$) to the same bucket with high probability, while dissimilar data points (distance$> cr$) are likely to be in different buckets. It is vital for the LSH-based method to design a good locality sensitive hash function. In Euclidian distance measurement, many hash functions have been proposed, such as [20], [1] and [2]. Random linear projections [28, 74] are the most commonly used hash functions to generate hash code, and the parameters are chosen from Gaussian distribution.

If we apply many hash functions, the hash table can be constructed and the collision probability from dissimilar points will decrease. However, it also reduces the collision probability of nearby points, the common solution is creating multiple hash tables but causing time-consuming query time and excessive memory usage. Some methods [64] , [41] , are proposed to search more hash buckets that may contain the nearby neighbors of the query point so that can improve the quality of query and decrease the number of hash tables.

Recently, Product Quantization (PQ) [39] methods become popular for ANNS, which decomposes the original high-dimensional space into the Cartesian product of a finite number of low-dimensional subspaces that are then quantized separately. There are many approaches base on PQ, one of them is Optimized Product Quantization (OPQ) [27], which uses pre-rotation to further minimize the quantization distortion.

Partition-based: Methods in this category divide the high dimensional space into several disjointed regions hierarchically. If query q is located in a region r_q, then its nearby neighbors should be in region r_q or near r_q.

According to the way to partition, there are three subcategories, pivoting, hyperplane and compact partitioning schemes. Pivoting approaches partition the vector space relying on the distance from the data point to pivots, such as VP-Tree [123] , ball-tree [58] , M-Tree [17] ,etc. Hyperplane partitioning methods recursively divide the space by the hyperplane with random direction, such as Annoy[3], Random-Projection Tree [19] , FLANN [69] . Lastly, compact partitioning methods either divide the data points into clusters or create possibly approximate Voronoi partitions to use its locality [71] , [9].

[3] https://github.com/spotify/annoy

Graph-based: The most common type of graphs used for complex data retrieval using a similarity search is the Proximity Graphs, which define as $G = (V, E)$. A proximity graph is a graph in which each pair of vertices $(u, v) \in V \times V$ is connected by an edge $e = (u, v)$, $e \in E$, if and only if u and v satisfy the neighbor-relationship. The core idea of graph-based methods is "*a neighbor's neighbor also be a neighbor*", and thus they can explore neighbor's neighbor efficiently by following the edges.

Recently, new types of graphs have been proposed for ANNS. A representative of this type of graph is the Navigable Small World graph (NSW) [65]. The NSW is based on an approximation of the Delaunay graph and there are two types of edges in NSW: *short-range links* for greedy search, and *long-range links*, which define the small-world navigation property. The construction of NSW iteratively inserts new vertices to the graph. For each new vertex, we locate the position and then search its k nearest neighbors in the current graph. The edges connected the new vertex and its k nearest neighbors are defined as *short-range links*. At the i iteration, the *short-range links* of $i - 1$ iteration become *long-range links*. HNSW [66] is an extension of NSW and can be seen as a multi-layer structure consisting of a hierarchical set of proximity graphs for nested subsets of the

stored elements. As far as we know, HNSW is one of the state-of-the-art ANNS algorithms so far.

2.5 Re-ranking for object retrieval

The face recognition can be seen as a retrieval process, the input image is the probe image and query the database or gallery. Nowadays, Re-ranking methods have been successfully studied to improve object retrieval accuracy [130]. There are many approaches utilize the k-nearest neighbors to explore similarity relationships to address the re-ranking problem. Chum *et al.* [16] propose the average query expansion (AQE) method, which averages the vectors in the top-k returned results. Shen In 2016, Sparse contextual activation (SCA) [5] encodes the local distribution of an image and indicates samples similarity by generalized Jaccard distance. Qin *et al.* [78] present the k-reciprocal nearest neighbors, which are considered as highly relevant candidates. In this paper, we inspired by k-reciprocal nearest neighbors and calculated the final distance between two different similarity measurements to overcome the "*curse of dimensionality*".

Chapter 3

Methods

This section presents the pipeline of our system, including face detection, face alignment, face encoding, and re-ranking stage. An overview of the pipeline is displayed in Fig. 3.1. First, the two different face detectors are introduced in section 3.1. The use of 2D face alignment to obtain canonical representations of faces is described in section 3.2. Section 3.3 describes the ResNet-based face feature extroctor which pre-trained on MS-Celeb-1M [32] and fine-tuned on VGGFace2 [12] . Lastly, for the purpose of accelerating

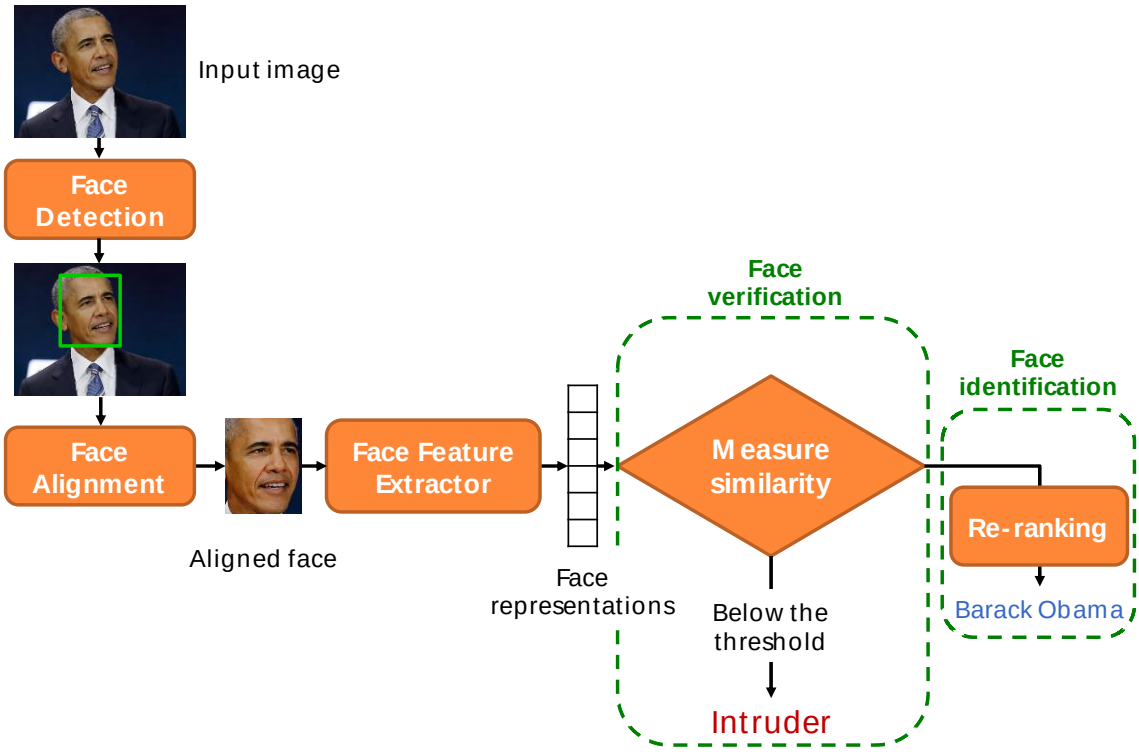


Figure 3.1: System flowchart

and re-ranking the retrieval of face recognition, we apply ANNS and explain our proposed re-ranking method in section 3.4.

3.1 Face Detection

This section briefly describes the two face detection methods we applied, MTCNN and FaceBoxes.

3.1.1 MTCNN

Multi-task cascaded CNN (MTCNN) [125] includes three-stage coarse-to-fine CNNs trained on WIDER Face [120]. As can be seen in Fig. 3.2, MTCNN build a coarse-to-fine detection approach. Before passing an image to the first CNN stage, MTCNN initially resizes it to different scales to build an image pyramid. **Stage 1**: The first stage is the Proposal Network (P-Net), to obtain the candidate windows. After that, we use the estimated bounding box regression vectors to calibrate the candidates, then use non-maximum suppression (NMS) to eliminate highly overlapped windows. **Stage 2**: This is a Refine Network (R-Net) to reject false candidates and performs bounding box calibration and NMS. **Stage 3**: Output Network (O-Net) is similar to R-Net but aims to locate the positions of five facial landmarks. This cascading CNN architecture achieves

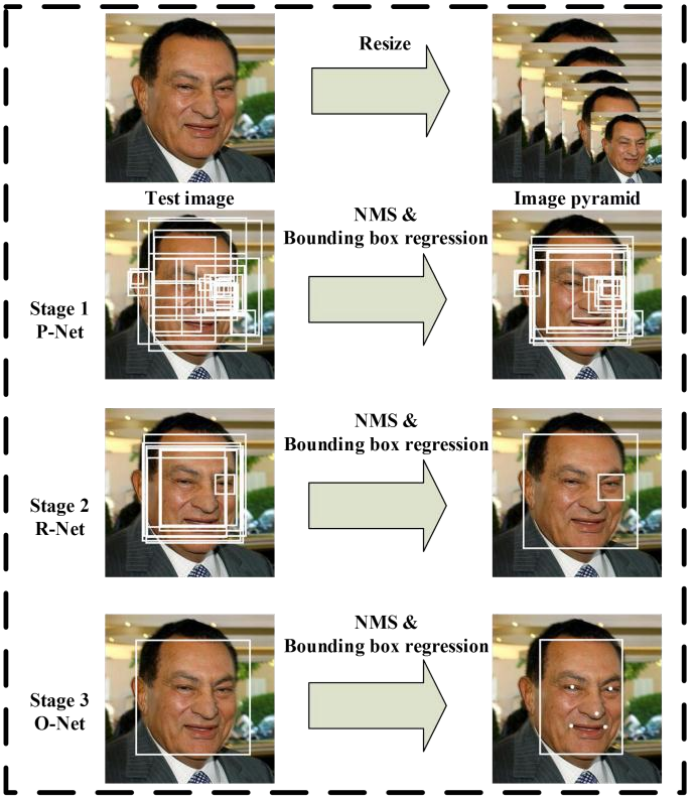
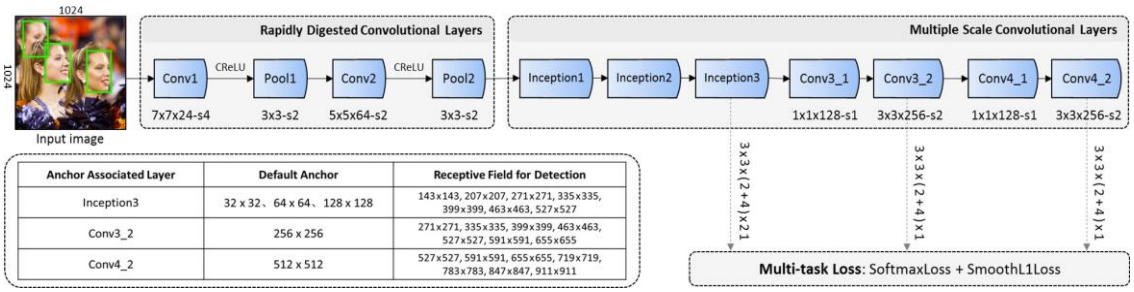


Figure 3.2: The pipeline of MTCNN that includes three-stage multi-task deep convolutional networks. [125]

superior accuracy on the several challenging benchmarks [38] while keeps real-time performance (99fps) on GPU.

3.1.2 FaceBoxes

As illustrated in Fig. 3.3 , FaceBoxes [126] consists of the Rapidly Digested Convolutional Layers (RDCL), the Multiple Scale Convolutional Layers (MSCL) and the anchor densification strategy. **RDCL** is designed to rapidly shrink the input spatial size by choosing a suitable kernel size. Furthermore, adoption of the CReLU [91] activation function can double the number of output channels by simply concatenating negated



Anchor Associated Layer	Default Anchor	Receptive Field for Detection
Inception3	32 x 32、64 x 64、128 x 128	143x143, 207x207, 271x271, 335x335, 399x399, 463x463, 527x527
Conv3_2	256 x 256	271x271, 335x335, 399x399, 463x463, 527x527, 591x591, 655x655
Conv4_2	512 x 512	527x527, 591x591, 655x655, 719x719, 783x783, 847x847, 911x911

Figure 3.3: Architecture of the FaceBoxes. [126]

outputs while significantly increasing the processing speed with a negligible decline in accuracy. **MSCL** is similar to SSD [59] , consisting of several layers, which decrease in size progressively and form the multi-scale feature maps in order that can naturally handle faces of various sizes. Also, the Inception modules [98] are engaged in not only to learn visual patterns for different scales of faces but enrich the receptive fields. **Anchor densification strategy:** the author proposes a new anchor densification strategy to eliminate the imbalance of the density of anchor. This strategy guarantees that different scales of anchor have the same density (i.e., 4) on the image. so that various scales of faces can match almost the same number of anchors. As a consequence, FaceBoxes runs at 20 FPS on a single CPU core and 125 FPS using a GPU for VGA-resolution images.

3.2 Face Alignment

This section we describe the intuitive 2D face alignment procedure of our system. The Ensemble of Regression Trees [42] is used to find 68 facial landmarks, and

implementation of the algorithm is available in dlib [45] toolkit. The core of the algorithm is based on the gradient boosting tree for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labeled data. Fig. 3.4 shows the Landmark estimates at different levels of the cascade initialized with the mean shape centered at the output of a basic Viola & Jones [104] face detector. After the first level of the cascade, the error is already greatly reduced.

After predicting the facial landmarks, we can use these landmarks to rotate the image such that the eyes lie on a horizontal line. As seen in Fig. 3.5, we first use the eyes region landmarks to compute the center of each eye and compute the angle between two eyes. After that, calculate the midpoint between two eyes, which will serve as the coordinates

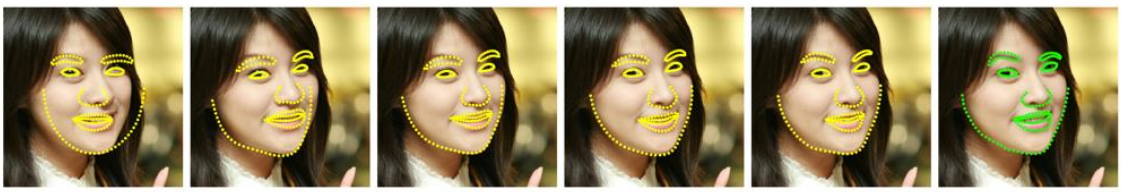


Figure 3.4: Landmark estimates at different levels of the cascade of regressors. [42]

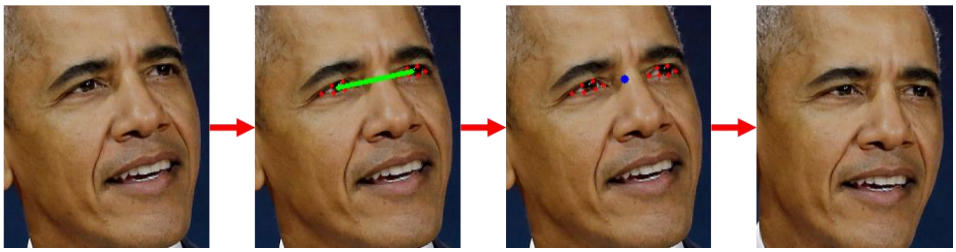


Figure 3.5: The process of rotating a face into canonical view.

in which we rotate the face. Finally, we get the canonical view of a face after rotating the image.

3.3 Face Feature Extractor

In this section, we discuss the detail of the face feature extractor in our system for face verification and identification. We employ VGGFace2 [12] as the face descriptor, which is a ResNet-50 [33] architecture pre-trained on MS-Celeb-1M [32] and fine-tuned on VGGFace2 dataset.

VGGFace [75] is the previous version, different from VGGFace2, VGGFace initially trains a VGG16 [92] based N-ways classifier on the VGGFace dataset. After that, remove the classifier layer to fine-tune the score vector in Euclidean space using the triplet-loss training scheme. The performance of them is illustrated in section 4.

The VGGFace2 dataset contains 3.31 million images from 9131 celebrities (8631 for training, 500 for evaluation) spanning a wide range of different ethnicities, accents, professions, and ages. The Images were downloaded from Google Image Search and show large variations in pose, age, illumination, and background. Face distribution for different identities is varied, from 87 to 843, with an average of 362 images for each subject. Examples are shown in Fig. 3.6.

Figure 3.6: VGGFace2 template examples. **Left**: examples of three different viewpoints (arranged by row) – frontal, three-quarter, profile. **Right**: examples of two subjects for young and mature ages (arranged by row). [12]

As discussed above, the training scheme of VGGFace2 removes the fully connected layer from a pre-trained model trained on MS-Celeb-1M, and then fine-tuned on the VGGFace2 dataset as an 8631-classes classifier. Finally, we removed the classifier of the model as the face feature extractor. Note that all the networks are using softmax loss function. During training and testing, the preprocessing of the images is subtracting the mean value of each channel for each pixel.

3.4 Approximate Nearest Neighbors Search

Approximate Nearest Neighbors Search (ANNS) is a popular and fundamental method in various applications, such as database, multimedia, and computer vision. As

far as we know, Hierarchical Navigable Small World (HNSW) [66] algorithm is one of the most efficient ANNS algorithms; hence we apply HNSW to our system.

HNSW can be seen as a coarse-to-fine hierarchical NSW, where the ground layer has all data points and the higher layer contains fewer points. Similar to NSW, HNSW is constructed by inserting new data points one-by-one. For each inserting, an integer maximum layer l is randomly selected and there are two phases of the insertion process. The first step starts from the top layer to $l+1$ by greedily traversing the graph to find the closest neighbor to the inserted data point in the layer, which is used as the enter-point to continue the search in the next layer. The second step adds the new data point to all layers from layer l to layer ground. M nearest neighbors are found and are connected with the new point. The number of enter-points ef also controls the searching quality of HNSW. The search can be seen as a $l=0$ insertion, starting from the upper layer which has longer links and greedily traverse the upper layer until a local minimum is reached. After that, the search switch to the lower layer (which has shorter links), restarting the traversal to the local minimum (see Fig. 3.7 for illustration).

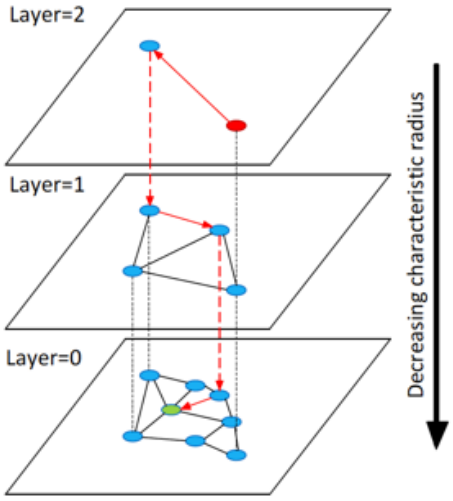


Figure 3.7: Illustration of the Hierarchical NSW idea. The search starts from an element from the top layer (shown red). Red arrows show direction of the greedy algorithm from the entry point to the query (shown green). [66]

3.5 Re-ranking policy

In this section, we introduce our re-ranking approach in two different configurations, Without and With Average. **Without Average** means we compare the probe image to every image in gallery and **With Average** is comparing every user vector which is averaging all images of each identity in the gallery.

Without Average: Define our feature extractor $f(x)$, which returns face representation. Given a probe image p and compared with whole gallery $\mathcal{G}$ which contains N images $\mathcal{G} = \{ g_i | \ i = 1,2, \dots N\}$. The distance between p and g_i is measured by Cosine distance,

$$d_C(p, g_i) = d_{g_i}^p = 1 - cosine \ \ similarity = 1 - \frac{\sum_1^n \bar{V}_p V_{g_i}}{\sqrt{\sum_1^n \bar{V}_p^2} \ \sqrt{\sum_1^n V_{g_i}^2}}$$

where the vector $\bar{V}_p = \frac{1}{2}(f(p) + f(p'))$ is the average of normalized representation of p and its mirror image p' and $V_{g_i} = f(g_i)$ represents the face vector of the gallery image g_i. The $\bar{V}_p$ is a query expansion from the original probe image p, it may share similar features and bring more comprehensive information of this person.

The initial ranking list $L(p, G) = \{g_1^0, g_2^0, \dots g_M^0 \}$ can be obtained by the original Cosine distance between p and g_i, where $d_C(p, g_i^0) < d_C(p, g_{i+1}^0)$. Our goal is to update the original $L(p, G)$, so that the more positive samples move to the top of the list. We define $N(p, k)$ as the k-nearest neighbors of $\bar{V}_p$,

$$N(p, k) = \{g_1^0, g_2^0, \dots g_k^0 \}, |N(p, k)| = k$$

where $|\cdot|$ denotes the number of the candidates. To find the further relationship between probe p and g_i, we define $N(g_i, k)$ as the k-nearest neighbors of each g_i in $N(p, k)$. And the Jaccard distance of g_i can be define as:

$$d_J(p, g_i) = 1 - \frac{|N(p, k) \cap N(g_i, k)|}{k}$$

After that, we jointly aggregate the original Cosine distance and the corresponding Jaccard distance to final distance d^*:

$$d^*(p, g_i) = \lambda d_C(p, g_i) + (1 - \lambda) d_J(p, g_i)$$

where $\lambda \in [0,1]$ denotes the weight between the distances. When $\lambda = 0$, only the Jaccard distance is considered. By contrast, when $\lambda = 1$, only the Cosine distance is considered. The effect of λ is discussed in section 4.4. We can obtain the final ranking list by ascending sort of the d^* and the re-ranking process is illustrated in Fig. 3.8.

With Average: In this configuration, the original gallery which contains M images in N identities can also be defined as $\mathcal{G} = \{ g_i^j | \ i = 1,2, \dots N; \ j = 1,2 \dots n_i \}$, where each identity has n_i images in the gallery. We update the original gallery to $\bar{\mathcal{G}}$ by averaging every normalized feature of the images belonging to the identity in the dataset (*i.e.* $\bar{\mathcal{G}} = \{ \bar{V}_i | \ i = 1,2, \dots N\}$, where $\bar{V}_i = \frac{\sum_{j=1}^{n_i} f(g_i^j)}{n_i}$). Given a probe image p, The Cosine distance between p and $\bar{V}_i$,

$$d_C(p, \bar{V}_i) = d_{\bar{V}_i}^p = 1 - cosine \ \ similarity = 1 - \frac{\sum_1^n V_p \bar{V}_i}{\sqrt{\sum_1^n V_p^2} \ \sqrt{\sum_1^n \bar{V}_i^2}}$$

where $V_p = f(p)$ which is the representation of probe p.

As discussed above, two ranking lists, $N(p,k)$ and the ranking list of the flipped image p', $N(p',k)$ can be obtained in cosine distance by HNSW algorithm. After that, we can get the final ranking list by ascending sort of the averaged distance of each identity who is in $N(p,k)$ and $N(p',k)$. The process of re-ranking in this configuration is shown in Fig 3.9.

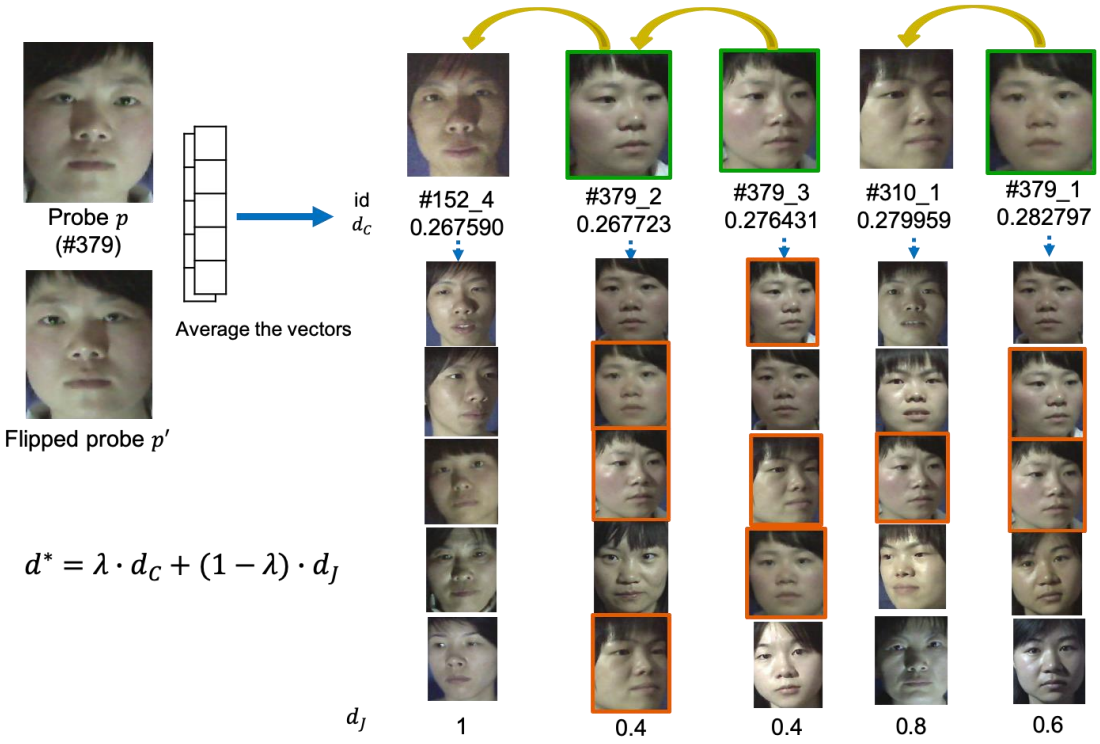


Figure 3.8: An example of the re-ranking policy (without average). The query vector shown in the top-left is the average of the face representations of probe p and its flipped p'. The original top-5 ranking list and its Cosine distance d_C are shown in the top-right, where the green box denotes the image is positive. Each column in the bottom shows the 5-nearest neighbors of the corresponding candidate in the original list. The image with orange box represents that the face is also appearing in the original ranking list so that Jaccard distance d_J can be calculated for each candidate. We can observe that the positive candidates can move forward in the appropriate λ to obtain a better ranking list.

Figure 3.9: An example of the re-ranking policy (with average). We calculate the average of the cosine distances from those closest neighbors of probe p and p' to obtain the final ranking list.

Chapter 4

Experiments

In section 4.1, we firstly compare the performance and the runtime efficiency of two different face detection algorithms, MTCNN and FaceBoxes. After analyzing face detection, we evaluate the performance of face verification in section 4.2 and face identification in section 4.3. Finally, the parameters of our method are analyzed in section 4.4. In the following sections, we describe the evaluation datasets and our experimental protocols. We also describe the changes to the system we made if there are any. All the experiments were conducted on Ubuntu 16.04 LTS operation system using GeForce GTX 1080 Ti and cuDNN v5 with Intel Xeon E5-2630v4 @2.20GHz and 128GB RAM.

4.1 Evaluation of Face Detection

In this section, we firstly evaluate MTCNN and FaceBoxes on the common face detection benchmarks, then compare the runtime efficiency between them.

Figure 4.1: Some examples of PASCAL Face dataset. [25]

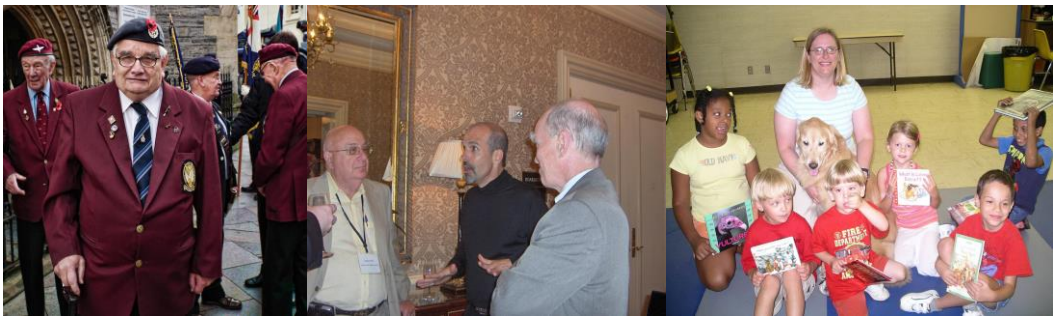


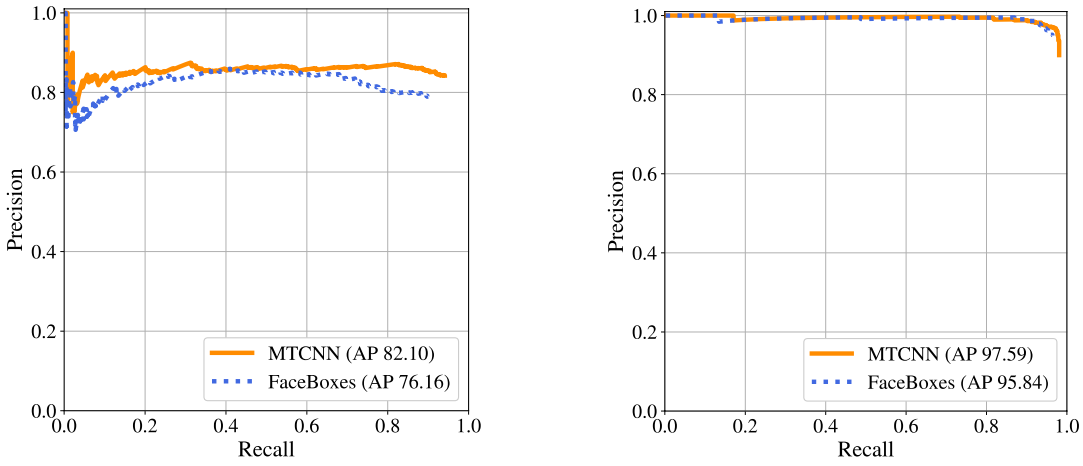
Figure 4.2: Some examples of AFW dataset. [80]

4.1.1 Benchmark evaluation

We evaluate the FaceBoxes [126] and MTCNN [125] on three common face detection benchmark datasets, including PASCAL Face [25], Annotated Faces in the Wild (AFW) [80] and Face Detection Data Set and Benchmark (FDDB) [38].

PASCAL Face dataset is collected from the test set of PASCAL person layout dataset, consisting of 1335 faces with large face appearance and pose variations from 851 images. Some examples are shown in Fig. 4.1. Fig. 4.3 (a) shows the precision-recall curves on this dataset, where MTCNN significantly outperforms FaceBoxes.

AFW dataset is built using Flickr images. It has 205 images with 473 faces. Some sample images are shown in Fig. 4.2. For each face, annotations include a rectangular



(a) PASCAL Face dataset (b) AFW dataset

Figure 4.3: Precision-recall curves on the PASCAL and AFW benchmarks.

bounding box, 6 landmarks, and the pose angles. As can be seen the precision-recall curves in Fig. 4.3 (b), no significant differences are found between MTCNN and FaceBoxes on this dataset.

FDDB dataset contains 5,171 faces in 2,845 images taken from news articles websites. FDDB provides the bounding ellipse, and thus our data preprocessing converts

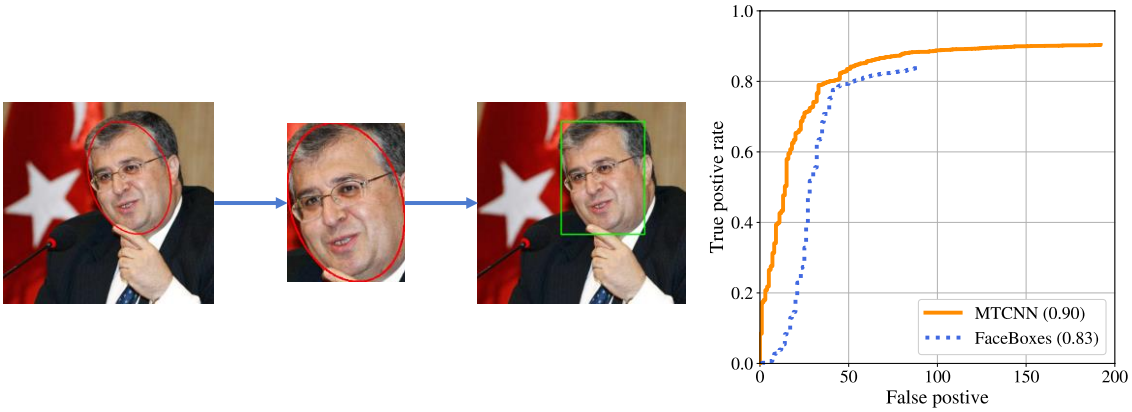


Figure 4.4: Evaluate on FDDB benchmark. **Left**: The processes of converting bounding ellipse to bounding box. **Right**: Discontinuous ROC curves on the FDDB dataset.

the ellipses to minimum bounding rectangles. The process and the results are shown in Fig. 4.4. MTCNN outperforms FaceBoxes a large margin on discontinuous ROC curve.

4.1.2 Face Detection Runtime Efficiency

We measure the speed on the UTKFace [129] dataset. During inference, we filter the boxes by a confidence threshold of 0.5 before applying NMS, then we perform NMS with IOU of 0.5.

UTKFace is a large-scale face dataset of over 20,000 face images, with only a single face in each image. The detectors only return a bounding-box with the highest confidence score. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. Some samples are displayed in Fig. 4.5.

Figure 4.5: Some examples of UTKFace. [129]

Table 4.1: Comparison of mAP and FPS on different approaches.

Approach	mAP(%)	FPS (on GPU)	FPS (on CPU)
MTCNN	**95.1**	8.86	6.25
FaceBoxes	89.3	**42.91**	**20.07**

The result is listed in Tab. 4.1, where MTCNN gets better performance (95%) on mean average precision (mAP). However, FaceBoxes achieves real-time face detection at 42FPS on the GPU and 20FPS on the CPU with adequate mean average precision.

4.1.3 Face Detection Conclusion

Face detection in CNN-based methods have achieved remarkable progress but always been accused of its runtime efficiency. In different scenarios, we can choose corresponding approach to meet the requirements. Considering the high accuracy in the preprocessing stage of the face feature extraction, we use MTCNN to detect as many faces as possible. In addition, FaceBoxes is fast enough to satisfy many practical applications, thus we use it on the user interface to smoothly detect user. Note that, in the following sections, if not other specified, we applied MTCNN as the face detection algorithm.

4.2 Evaluation of Face Verification

In section 4.2.1, we evaluate two different backbone feature extractors, VGG16 and ResNet-50 on LFW [36] dataset and select the better architecture. Then, we simulate a scenario with the CASIA-FaceV5 [14] dataset as the authorized user dataset while Helen [49] dataset as the intruder dataset. Lastly, we present the performance for different preprocessing and postprocessing, including the integration of user features and face alignment.

4.2.1 Performance on LFW

Labeled Faces in the Wild (LFW) contains 13,233 images with 5,749 identities and is the standard benchmark for automatic face verification. We follow the standard protocol for *unrestricted, labeled outside data*. LFW provides 10 sets., each set has 300 matched pairs and 300 mismatched pairs, some samples are shown in Fig. 4.6. Nine sets are used to select the cosine similarity threshold. Verification (same or different) is then performed on the tenth set. The selected optimal thresholds for VGG16 and ResNet-50 are respectively 0.694 and 0.463.

Fig. 4.7 shows the mean ROC curve on LFW, it can be observed that ResNet-50 architecture is significantly outperformed VGG16. Table 4.2 shows the accuracy and time

Match pairs  Mismatch pairs 

Figure 4.6: Some examples of LFW [36] .

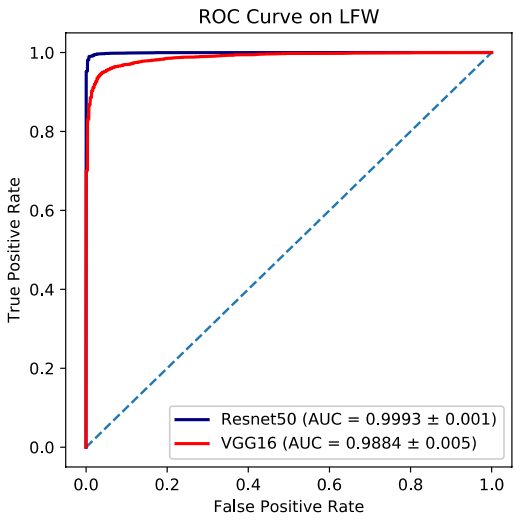


Figure 4.7: ROC curves on LFW

Table 4.2: Performance evaluation on LFW dataset

Backbone	Accuracy (%)	Time (s)
VGG16	95.33	**129**
ResNet-50	**99.33**	139

consumption. ResNet-50 can achieve verification accuracy of 99.33%; thus, we choose ResNet-50 as our face descriptor backbone.

4.2.2 Performance on CASIA-FaceV5 and Helen dataset

After choosing the face feature extractor, we use the CASIA-FaceV5 dataset [14] as the authorized user dataset and the Helen [49] dataset as the intruder dataset to evaluate performance in distinguishing between authorized users and intruders..

CASIA-FaceV5 (CASIA Face Image Database Version 5.0) contains 2,500 color facial images of 500 subjects. That is, there are 5 images for each identity, we randomly

Figure 4.8: Some examples of CASIA-FaceV5 and Helen dataset. **Left**: CASIA-FaceV5 [14] dataset. **Right**: Helen [49] dataset.

chose one image as the probe image and the others as the gallery images. All face images are 16-bit color BMP files and the resolution is 640*480, Fig. 4.8 shows some examples.

Helen dataset is a high-resolution dataset originally built for facial feature localization, and it contains a broad range of appearance variation, including pose, lighting, expression, occlusion, and individual differences. This dataset consists of 2000 training and 330 test images, we use training images as the intruder dataset. If there are multiple faces in the picture, we select the one closest the image center as the test face. Some samples are shown in Fig. 4.8.

Inheriting from the confusion matrix, we define ground Truth $g(x) = 1$ for authorized user and $g(x) = 0$ for intruder. **False Accept (FA)** means our system $f(x)$ will incorrectly accept an access attempt by an unauthorized user. **False Reject (FR)**

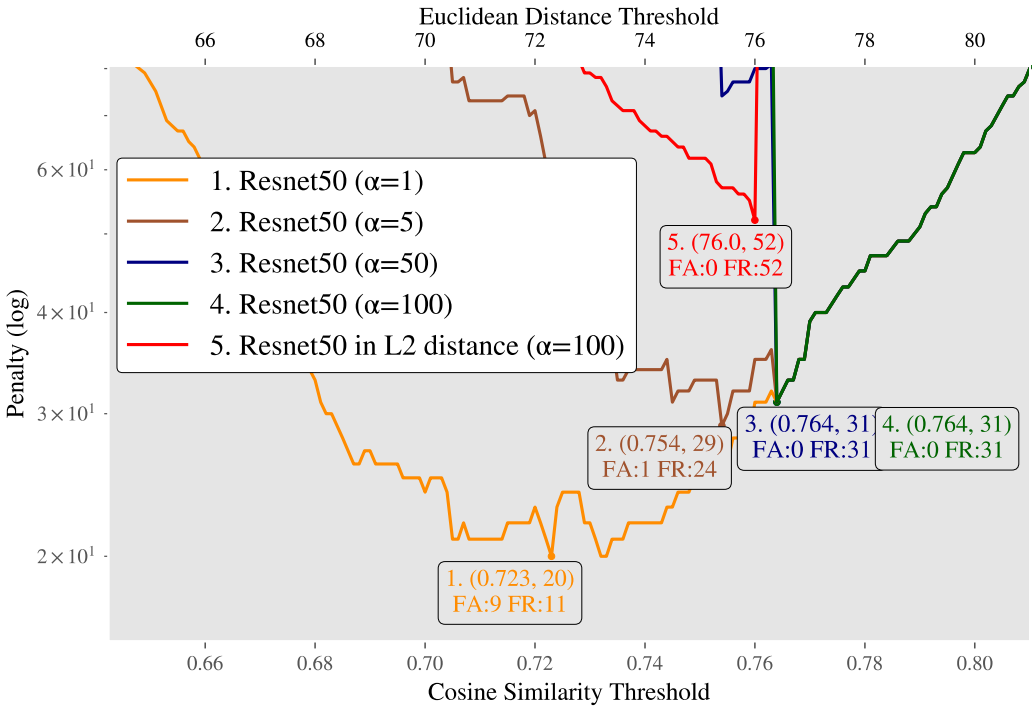


Figure 4.9: Penalty curves of different False Accept penalties and different distance measurements.

means failing to recognize an authorized person and rejecting that person as an intruder.

We defined the face verification penalty as:

$$\boldsymbol{Face\ Verification\ Penalty\ (FVP)} = FA \cdot \alpha + FR \cdot \beta$$

where α and β are respectively the penalty for FA and FR. A good face verification system is more concerned about the penalty of FA; thus, we fix $\beta = 1$ and alter the FA penalty α. Fig 4.9 shows the minimum FVP in different α, we observed that the penalty increases with α. It also shows that Cosine similarity obtains a lower penalty than Euclidean distance for the same α.

4.2.3 Alignment and Average

This section presents the results of different preprocessing for face verification. Note that the penalty of FA α is fixed at 100 using the same dataset as mentioned in section 4.2.2.

Alignment: According to the component analysis of [75], we compared the performance of performing 2D alignment in the process of building the database and also aligned the test images.

Average: (i) **Without average**: compares every image of the authorized user in the dataset, i.e. our system returns the identity name which is most similar to the probe image. (ii) **With average**: for each authorized user, we averaged the face vectors of their gallery images which were embedded by the ResNet-50.

Fig. 4.10 shows the penalty curves of different configurations, as can be seen, the performance of averaging each user's face representations is better than comparing all gallery images (orange and blue line). The probable reason is that averaging the vectors may obtain more robust and general facial features, thus avoiding the effects of angle, hairstyle or glasses. Furthermore, performing 2D face alignment on both training and test images gives a boost in performance (solid and dotted line), comes to the same conclusion

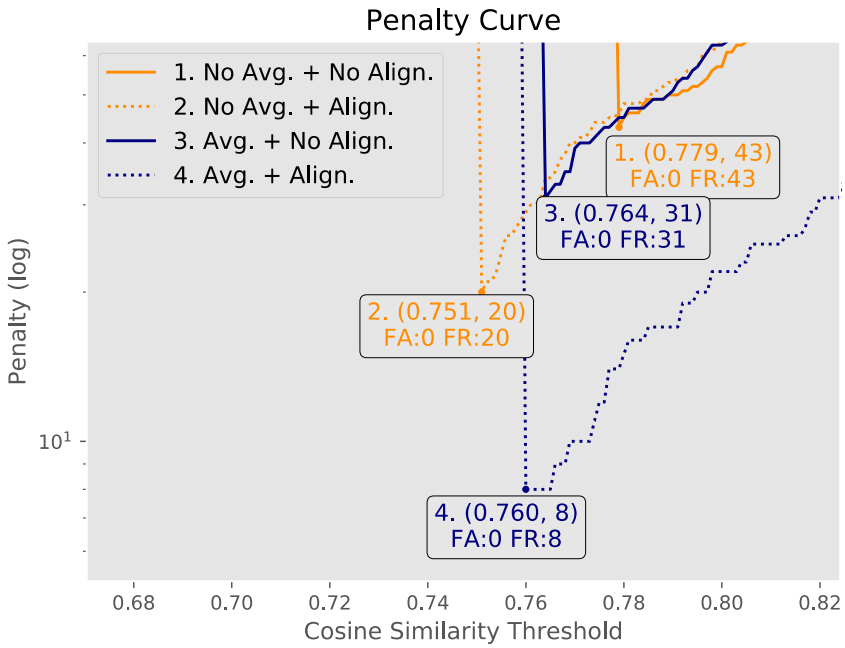


Figure 4.10: Performance comparison for different configurations.

as [6]. In conclusion, the best configuration of face verification is the average of user face representations and performing alignment on all images.

4.3 Evaluation of Face Identification

This section reports experimental results for face identification on different datasets. Section 4.3.1 and 4.3.2 discuss the performance on large inter-class variation datasets, which are CASIA-FaceV5 and CASIA-WebFace [122] respectively. Section 4.3.3 and section 4.3.4 show the result for two large intra-class age-invariant datasets, FG-NET and Cross-Age Celebrity Dataset (CACD) [13]. Unless otherwise specified, we set the number of neighbors k to 10 and use HNSW to speed up the 1: N retrieval time and present the Rank-1 accuracy.

4.3.1 Experiments on CASIA-FaceV5

As mentioned previously, we randomly select one image for each identity as the probe image and do 5-fold validation. In without average configuration, we set λ to 0.8. Table 4.3 compares the performance of different components of face identification. As can be seen, in spite of taking double time, our re-ranking method does improve the accuracy (rows 1 and 2, rows 5 and 6). In addition, Table 4.3 shows averaging the user images has the capability of dealing with the factors of poses, illumination, and hairstyle (rows 1 and 3, rows 5 and 7). It also shows the performance can be improved 1.47% in rank-1 accuracy via our method (rows 1 and 8).

Table 4.3: Comparison of different configurations on CASIA-FaceV5.

No.	Align.	Average	Re-rank	Avg. Time (s)	Mean Acc.
1	×	×	×	27	0.9840 ± 0.008
2	×	×	✓	65	0.9880 ± 0.002
3	×	✓	×	22	0.9900 ± 0.006
4	×	✓	✓	48	0.9980 ± 0.002
5	✓	×	×	43	0.9847 ± 0.011
6	✓	×	✓	87	0.9853 ± 0.008
7	✓	✓	×	50	0.9920 ± 0.005
8	✓	✓	✓	92	**0.9987 ± 0.001**

4.3.2 Experiments on CASIA-WebFace

This section discusses the performance on a larger inter-class variation dataset, CASIA-WebFace, which consists of 0.5M images of 10K celebrities, collected from the IMDb website. The authors use a semi-automatically clustering step to construct the dataset, Wang *et al.* [105] analyzed the noise of the dataset, shown in Fig. 4.11. Note that we use the washed CASIA-WebFace[4], which removes 27,703 wrong images and select the identities those images count is more than 5.

We follow the 5-fold in which one image is used as the probe image, and the remaining are used as gallery images. We set λ to 0.8 for without average setting in this dataset. As observed in Table 4.4, it can be seen that the best setting is with average and re-ranking, which increases the rank-1 accuracy by 2.28%. To put it another way, this

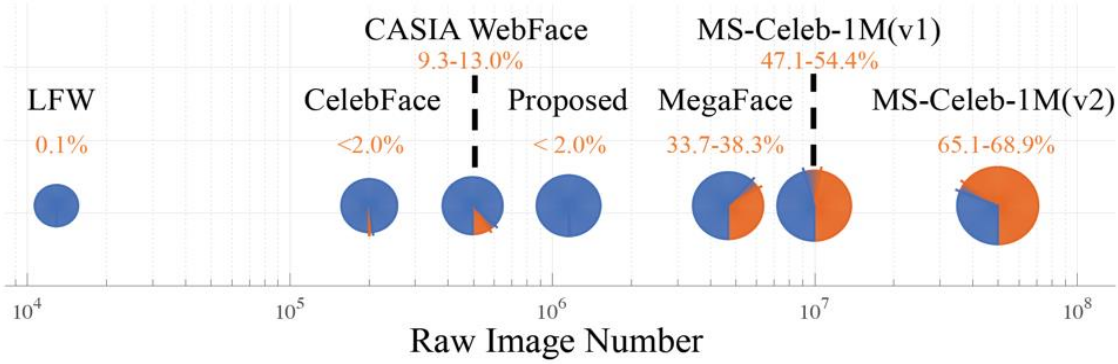


Figure 4.11: A visualization of size and estimated noise percentage of datasets. [105]

[4] https://github.com/cmusatyalab/openface/issues/119

Table 4.4: Comparison of different configurations on CASIA-WebFace.

Component	**Configuration**							
Alignment				✓	✓	✓		✓
Average			✓			✓	✓	✓
Re-rank		✓			✓		✓	✓
Mean Acc. (%)	84.47	85.84	86.48	82.97	84.98	85.26	**86.75**	86.14

configuration is robust to view changes and illumination variations. It is also worth noting that alignment is not conducive for this dataset.

4.3.3 Experiments on FG-NET

FG-NET database contains 1002 color or grayscale face images of 82 subjects, with ages from 0 to 69. Fig. 4.12 shows some examples.

Following the testing scheme in [54] , the leave-one-out strategy is used and we fixed the λ at 0.75. Table 4.5 shows our result on FG-NET, indicating the optimal setting on this dataset is contrary to previous. Without average and re-ranking can obtain better performance, possibly because single image comparison can handle significant intra-

Figure 4.12: Some examples of FG-NET.

personal variation and find the image most closely reflecting the subject's age. Combining all gallery image vectors of a person might be lead to representation bias toward the subject's middle-aged years.

We also compared our result with some state-of-the-art approaches [30, 31, 110, 114], finding that, on this dataset, the gap between our proposed method and specialized age-invariant face recognition methods is not significant. The comparative results are reported in Table 4.6.

Table 4.5: Comparison of different configurations on FG-NET.

Component	**Configuration**							
Alignment				✓	✓	✓		✓
Average			✓			✓	✓	✓
Re-rank		✓			✓		✓	✓
Mean Acc. (%)	86.62	87.22	83.43	87.32	**87.92**	84.33	83.63	84.53

Table 4.6: Performance of different methods on FG-NET.

Method	**Rank-1 Acc. (%)**
HFA [32]	69.0
MEFA [33]	76.2
CAN [115]	86.5
LF-CNNs [111]	88.1
Ours	**87.92**

4.3.4 Experiments on CACD

Cross-Age Celebrity Dataset (CACD) [13] is a dataset for age-invariant face recognition, containing 163,446 images from 2,000 celebrities with ages ranging from 16 to 62. The images in this dataset have varied illumination, different poses, different makeup and better simulate the practical scenario.

We followed the experimental setting in [13] , choose 120 celebrities with rank 3-5 as test sets where images taken at 2013 are used as query images. The remaining images are split into three subsets respectively taken in 2004-2006, 2007-2009 and 2010-2012 as database images.

In our experiment on CACD, we fix the λ at 0.75 and use mean average precision (MAP) as evaluation metrics. Cosine distance is used to compute the similarity of two images. Specifically, let $q_i \in Q$ is the query image and Q is the query dataset. For each q_i, the positive images are expressed as $Y_1, Y_2, Y_3, \dots, Y_m$ and we define E_{ic} as the retrieval results of q_i in descending order from the top to Y_c. Therefore, the average precision (AP) of q_i can be computed as below:

$$AP(q_i) = \frac{1}{m_i} \sum_{c=1}^{m_i} Precision(E_{ic}),$$

where $Precision(E_{ic})$ means the ratio of relevant images in retrieval results E_{ic}. After that, the Mean Average Precision (MAP) of all query images can be denoted as below:

$$MAP(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|} AP(q_i).$$

Using this evaluation metric, Fig. 4.13 reports the result that we compare our method with some state-of-the-art Cross-Age face recognition algorithms including hidden factor analysis (HFA) [30] , cross-age reference coding (CARC) [14] , generalized similarity model [56] (GSM-2 use more training data), coupled auto-encoder networks (CAN) [114] and age estimation guided convolutional neural network (AE-CNN) [131]. The result shows that our approach still yields good performance compared with age-invariant face recognition algorithms. We also show the rank-1 recognition accuracy of our approach and other methods [14, 15] in Table 4.7. Note that the baseline method is only considering cosine distance and without re-ranking. It can be seen that our approach outperforms the baseline and gains nearly 2.5% improvement in rank-1 accuracy.

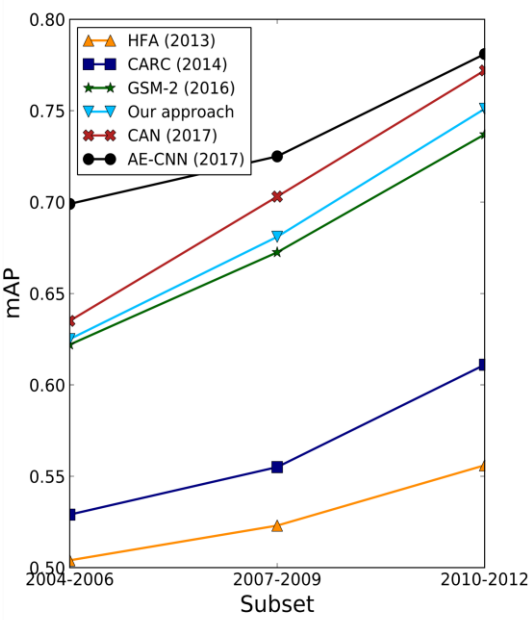


Figure 4.13: The performance of mAP of our approach compared with SOTA algorithms on CACD.

Table 4.7: The Rank-1 accuracy of different methods on three subsets in CACD.

Methods	Rank-1 accuracy (%)		
	04-06	07-09	10-12
LBP [14, 15]	78.0	80.3	85.5
CARC [14]	88.8	88.5	92.2
Baseline	81.5	84.1	85.4
Baseline + Ours	**84.7**	**86.2**	**87.4**

4.4 Parameters Analysis

The parameters of the re-ranking method are analyzed in this section. In Fig. 4.14, we evaluate the influence of λ on rank-1 accuracy on large inter-class and intra-class variation datasets. Note that, we fix the number of nearest neighbors k at 10 since assigning a too large value of k may lead to more false matches included. When λ is set to 0, we only consider the Jaccard distance; By contrast, when λ is set to 1, the baseline result is the original ranking list without re-ranking method which is only sorting by Cosine distance. It can be observed that when simultaneously considering both the Cosine distance and the Jaccard distance, the performance obtains about 1% of

improvement and shows that the proposed re-ranking method is beneficial for face identification. Overall, these results imply that the optimal value of λ for the large inter-class variation dataset is around 0.8 and 0.75 for the large intra-class variation dataset.

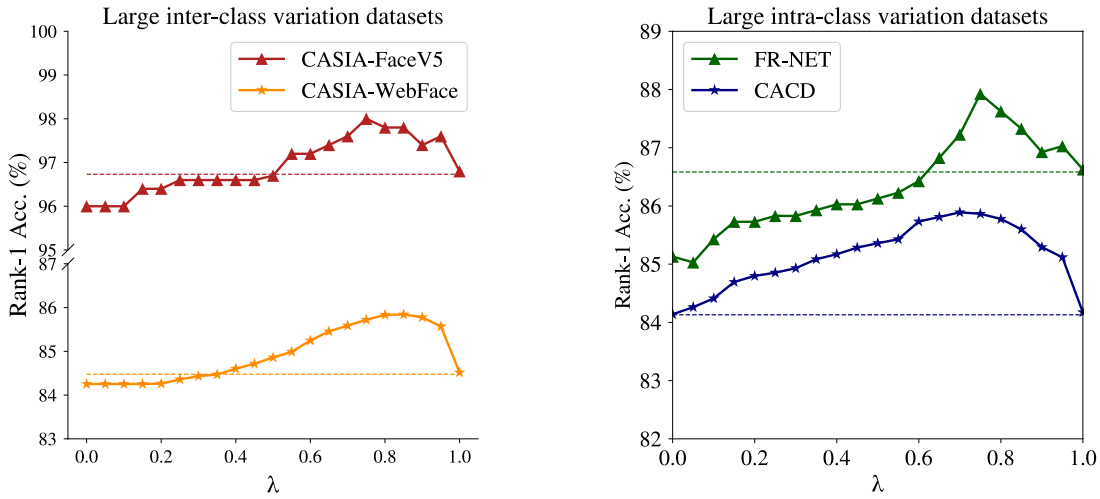


Figure 4.14: The impact of the parameter λ on rank-1 accuracy on large inter-class/intra-class variation datasets. We utilize the without average configuration on each dataset and fix the k at 10.

Chapter 5

Conclusions and discussions

In this paper, we provided a ResNet-based face recognition system employing the postprocessing method. We also presented the result of two different face detectors so that we can apply the most suitable method in different requirements. Moreover, a specific preprocessing is provided to obtain the best performance on identity authentication. According to the experimental results of face identification, our proposed re-ranking method has a positive effect on performance. Not only achieves comparable performance on the dataset with large inter-class variation but also the dataset with large intra-class variation. The configuration depends on the type of the dataset, which is applying without average setting for intra-class variation datasets while using average strategy can get more robust representations on inter-class variation dataset.

Future work will focus on improving the re-ranking policy and developing more efficient models on edge devices.

Reference

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Communications of the ACM,* vol. 51, no. 1, p. 117, 2008.

[2] A. Andoni and I. Razenshteyn, "Optimal data-dependent hashing for approximate near neighbors," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015: ACM, pp. 793-801.

[3] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3444-3451.

[4] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms," *Information Systems,* 2019.

[5] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *IEEE Transactions on Image Processing,* vol. 25, no. 3, pp. 1056-1069, 2016.

[6] A. Bansal, C. Castillo, R. Ranjan, and R. Chellappa, "The do's and don'ts for cnn-based face verification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2545-2554.

[7] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2930-2940, 2013.

[8] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM,* vol. 18, no. 9, pp. 509-517, 1975.

[9] A. Beygelzimer, S. Kakade, and J. Langford, "Cover trees for nearest neighbor," in *Proceedings of the 23rd international conference on Machine learning*, 2006: ACM, pp. 97-104.

[10] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 2: IEEE, pp. 236-243.

[11] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, and J. M. Rehg, "On the design of cascades of boosted ensembles for face detection," *International Journal of Computer Vision,* vol. 77, no. 1-3, pp. 65-86, 2008.

[12] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018: IEEE, pp. 67-74.

[13] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Cross-age reference coding for age-invariant face recognition and retrieval," in *European Conference on Computer Vision*, 2014: Springer, pp. 768-783.

[14] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset," *IEEE Transactions on Multimedia,* vol. 17, no. 6, pp. 804-815, 2015.

[15] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3025-3032.

[16] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *2007 IEEE 11th International Conference on Computer Vision*, 2007: IEEE, pp. 1-8.

[17] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An E cient Access Method for Similarity Search in Metric Spaces," in *Proceedings of the 23rd VLDB conference, Athens, Greece*, 1997: Citeseer, pp. 426-435.

[18] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379-387.

[19] S. Dasgupta and Y. Freund, "Random projection trees and low dimensional manifolds," in *STOC*, 2008, vol. 8: Citeseer, pp. 537-546.

[20] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004: ACM, pp. 253-262.

[21] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391-407, 1990.

[22] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, 2018.

[23] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 60-68.

[24] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049-2058, 2015.

[25] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303-338, 2010.

[26] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015: ACM, pp. 643-650.

[27] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization for approximate nearest neighbor search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2946-2953.

[28] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Vldb*, 1999, vol. 99, no. 6, pp. 518-529.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[30] D. Gong, Z. Li, D. Lin, J. Liu, and X. Tang, "Hidden factor analysis for age invariant face recognition," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2872-2879.

[31] D. Gong, Z. Li, D. Tao, J. Liu, and X. Li, "A maximum entropy feature descriptor for age invariant face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5289-5297.

[32] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European Conference on Computer Vision*, 2016: Springer, pp. 87-102.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[34] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 951-959.

[35] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012: IEEE, pp. 2518-2525.

[36] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[37] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998: ACM, pp. 604-613.

[38] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," UMass Amherst Technical Report, 2010.

[39] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence,* vol. 33, no. 1, pp. 117-128, 2010.

[40] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 2017: IEEE, pp. 650-657.

[41] A. Joly and O. Buisson, "A posteriori multi-probe locality sensitive hashing," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008: ACM, pp. 209-218.

[42] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867-1874.

[43] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4873-4882.

[44] J. Kim, C. Liu, F. Sha, and K. Grauman, "Deformable spatial pyramid matching for fast dense correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2307-2314.

[45] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research,* vol. 10, no. Jul, pp. 1755-1758, 2009.

[46] B. F. Klare *et al.*, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1931-1939.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[48] A. Kumar, R. Ranjan, V. Patel, and R. Chellappa, "Face alignment by local deep descriptor regression," *arXiv preprint arXiv:1601.07950*, 2016.

[49] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *European conference on computer vision*, 2012: Springer, pp. 679-692.

[50] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in face detection and facial image analysis*: Springer, 2016, pp. 189-248.

[51] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5325-5334.

[52] S. Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *European Conference on Computer Vision*, 2002: Springer, pp. 67-81.

[53] Y. Li, B. Sun, T. Wu, and Y. Wang, "Face detection with end-to-end integration of a convnet and a 3d model," in *European Conference on Computer Vision*, 2016: Springer, pp. 420-436.

[54] Z. Li, U. Park, and A. K. Jain, "A discriminative model for age invariant face recognition," *IEEE transactions on information forensics and security*, vol. 6, no. 3, pp. 1028-1037, 2011.

[55] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 211-223, 2015.

[56] L. Lin, G. Wang, W. Zuo, X. Feng, and L. Zhang, "Cross-domain visual matching via generalized similarity measure and feature learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1089-1102, 2016.

[57] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[58] T. Liu, A. W. Moore, and A. Gray, "New algorithms for efficient high-dimensional nonparametric classification," *Journal of Machine Learning Research*, vol. 7, no. Jun, pp. 1135-1158, 2006.

[59] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016: Springer, pp. 21-37.

[60] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212-220.

[61] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, 2016, vol. 2, no. 3, p. 7.

[62] Y. Liu, H. Li, and X. Wang, "Rethinking feature discrimination and polymerization for large-scale recognition," *arXiv preprint arXiv:1710.00870*, 2017.

[63] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91-110, 2004.

[64] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: efficient indexing for high-dimensional similarity search," in *Proceedings of the 33rd international conference on Very large data bases*, 2007: VLDB Endowment, pp. 950-961.

[65] Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Approximate nearest neighbor algorithm based on navigable small world graphs," *Information Systems*, vol. 45, pp. 61-68, 2014.

[66] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[67] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *European conference on computer vision*, 2014: Springer, pp. 720-735.

[68] I. Matthews and S. Baker, "Active appearance models revisited," *International journal of computer vision*, vol. 60, no. 2, pp. 135-164, 2004.

[69] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2227-2240, 2014.

[70] B. Naidan, L. Boytsov, and E. Nyberg, "Permutation search methods are efficient, yet faster search is possible," *Proceedings of the VLDB Endowment,* vol. 8, no. 12, pp. 1618-1629, 2015.

[71] G. Navarro, "Searching in metric spaces by spatial approximation," *The VLDB Journal,* vol. 11, no. 1, pp. 28-46, 2002.

[72] S. M. Omohundro, "Efficient algorithms with neural network behavior," *Complex Systems,* vol. 1, no. 2, pp. 273-347, 1987.

[73] M. Osadchy, Y. L. Cun, and M. L. Miller, "Synergistic face detection and pose estimation with energy-based models," *Journal of Machine Learning Research,* vol. 8, no. May, pp. 1197-1215, 2007.

[74] R. Panigrahy, "Entropy based nearest neighbor search in high dimensions," in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, 2006: Society for Industrial and Applied Mathematics, pp. 1186-1195.

[75] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *bmvc*, 2015, vol. 1, no. 3, p. 6.

[76] M.-T. Pham and T.-J. Cham, "Fast training and selection of haar features using statistics in boosting-based face detection," in *2007 IEEE 11th International Conference on Computer Vision*, 2007: IEEE, pp. 1-7.

[77] F. P. Preparata and M. I. Shamos, *Computational geometry: an introduction*. Springer Science & Business Media, 2012.

[78] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *CVPR 2011*, 2011: IEEE, pp. 777-784.

[79] H. Qin, J. Yan, X. Li, and X. Hu, "Joint training of cascaded CNN for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3456-3465.

[80] D. Ramanan and X. Zhu, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012: Citeseer, pp. 2879-2886.

[81] R. Ranjan *et al.*, "Crystal loss and quality pooling for unconstrained face verification and recognition," *arXiv preprint arXiv:1804.01159*, 2018.

[82] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[83] R. Ranjan, V. M. Patel, and R. Chellappa, "A deep pyramid deformable part model for face detection," in *2015 IEEE 7th international conference on biometrics theory, applications and systems (BTAS)*, 2015: IEEE, pp. 1-8.

[84] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121-135, 2019.

[85] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 2017: IEEE, pp. 17-24.

[86] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[87] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397-403.

[88] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," in *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)*, 2016: IEEE, pp. 1-8.

[89] J. Saragih and R. Goecke, "A nonlinear discriminative approach to AAM fitting," in *2007 IEEE 11th International Conference on Computer Vision*, 2007: IEEE, pp. 1-8.

[90] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815-823.

[91] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *international conference on machine learning*, 2016, pp. 2217-2225.

[92] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[93] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988-1996.

[94] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[95] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1891-1898.

[96] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2892-2900.

[97] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1489-1496.

[98] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[99] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701-1708.

[100] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 797-813.

[101] J. K. Uhlmann, "Satisfying general proximity/similarity queries with metric trees," *Information processing letters,* vol. 40, no. 4, pp. 175-179, 1991.

[102] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision,* vol. 104, no. 2, pp. 154-171, 2013.

[103] R. Vaillant, C. Monrocq, and Y. Le Cun, "Original approach for the localisation of objects in images," *IEE Proceedings-Vision, Image and Signal Processing,* vol. 141, no. 4, pp. 245-250, 1994.

[104] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision,* vol. 57, no. 2, pp. 137-154, 2004.

[105] F. Wang *et al.,* "The devil of face recognition is in the noise," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 765-780.

[106] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters,* vol. 25, no. 7, pp. 926-930, 2018.

[107] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: l 2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017: ACM, pp. 1041-1049.

[108] H. Wang *et al.,* "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265-5274.

[109] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research,* vol. 10, no. Feb, pp. 207-244, 2009.

[110] Y. Wen, Z. Li, and Y. Qiao, "Latent factor guided convolutional neural networks for age-invariant face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4893-4901.

[111] L. Wolf, T. Hassner, and I. Maoz, *Face recognition in unconstrained videos with matched background similarity*. IEEE, 2011.

[112] Y. Wu, H. Liu, J. Li, and Y. Fu, "Deep face recognition with center invariant loss," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017: ACM, pp. 408-414.

[113] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, 2003, pp. 521-528.

[114] C. Xu, Q. Liu, and M. Ye, "Age invariant face recognition and retrieval by coupled auto-encoder networks," *Neurocomputing*, vol. 222, pp. 62-71, 2017.

[115] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2497-2504.

[116] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, no. 10, pp. 790-799, 2014.

[117] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *IEEE international joint conference on biometrics*, 2014: IEEE, pp. 1-8.

[118] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Fine-grained evaluation on face detection in the wild," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015, vol. 1: IEEE, pp. 1-7.

[119] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3676-3684.

[120] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5525-5533.

[121] S. Yang, Y. Xiong, C. C. Loy, and X. Tang, "Face detection through scale-friendly deep convolutional networks," *arXiv preprint arXiv:1706.02863*, 2017.

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

[123] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *SODA*, 1993, vol. 93, no. 194, pp. 311-21.

[124] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European conference on computer vision*, 2014: Springer, pp. 1-16.

[125] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.

[126] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "Faceboxes: A CPU real-time face detector with high accuracy," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, 2017: IEEE, pp. 1-9.

[127] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3fd: Single shot scale-invariant face detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 192-201.

[128] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5409-5418.

[129] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5810-5818.

[130] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence,* vol. 40, no. 5, pp. 1224-1244, 2017.

[131] T. Zheng, W. Deng, and J. Hu, "Age estimation guided convolutional neural network for age-invariant face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1-9.

[132] Y. Zheng, D. K. Pal, and M. Savvides, "Ring loss: Convex feature normalization for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5089-5097.