The Hong Kong Polytechnic University

Department
of
Electronic and Information Engineering

Machine Learning for Face Recognition

Author:
Zhong GUAN

Supervisor:
Prof. Kin-Man Lam

A thesis submitted for the degree of

BEng (HONS) Electronic and Information Engineering

April 15, 2019

Acknowledgements

I would like to convey my sincere gratitude to my supervisor, Prof. Kin-Man LAM, for his guidance and support. I thank him for the opportunity to work on these fascinating projects presented here, for teaching me the right way to do research, and for the kind advice on how to do things right in general. His teachings and academic research lead me to enter the field of machine learning and computer vision.

I would also like to thank Mr. Shun-Cheung LAI for his critical contributions to my projects, for providing me with required resources to perform experiments, and for always being able to enlighten me on programming and theoretical questions I struggle with. Special thanks to Mr. Jun XIAO for help and discussion on many deep learning algorithms across my research.

Abstract

Nowadays, deep convolutional neural networks (CNNs) have dominated the computer-vision research community, because they can achieve state-of-the-art performance in many different applications, in particular face recognition. In order to further improve the performance of deep face recognition, many researchers have put a lot of efforts into creating large training datasets, improving the neural network architecture, and designing more discriminative loss functions. Although some of the proposed advanced methods can achieve an unprecedented performance on some benchmarks, due to the challenging unconstrained conditions of face images, including different poses, expressions, age progression, varying illumination, and low resolution of images, the existing methods can hardly achieve robust performance in all the unconstrained conditions, in particular when the illumination condition is very poor.

In this project, we first investigate face recognition using conventional methods, which include the eigenface method, i.e. using Principal Component Analysis, and the local binary pattern (LBP) method. After that, we investigate deep-learning-based methods for face recognition. We conducted extensive experiments for deep face recognition on several important face recognition benchmarks, with different deep neural models and different loss functions to evaluate their performance in unconstrained conditions. These experiments were conducted under the open-set protocol, where the identities of the testing set are not included in the training set. Traditional face recognition methods, i.e. the LBP and eigenface methods, are compared with the deep face recognition methods. In addition to using the RGB channels of images as the input to CNNs, we also add the LBP feature as an additional input channel, which can improve deep face recognition when the lighting condition is poor.

In addition to this final year project, we also joined an undergraduate competition organized by the IEEE Signal Processing Society, the IEEE SPS Video and Image Processing (VIP) Cup, from June to September 2018. In this competition, we proposed a deep neural model to perform detection and segmentation of lung cancer tumor regions, from screening Computed Tomography (CT) scans. Our proposed network can achieve satisfactory results on the provided dataset based on our designed evaluation metrics. We have also published a paper on our proposed methods at the IWAIT 2019 conference.

Contents

List of Figures

List of Tables

1 Introduction

In this report, two projects completed by us will be discussed. Part I of this report states our proposed deep neural model for lung tumor segmentation, which was completed from June to September 2018. Our proposed method in this project was published as a paper at the IWAIT 2019 conference. In Part II of this report, the project of machine learning for face recognition will be discussed in detail.

1.1 Deep Neural Network for Lung Tumor Segmentation

Image-based medical diagnosis involves a huge amount of data. The interpretation of such a large amount of data has always relied on the experience of the radiologist and prior knowledge of the tumor location, and thus has made it a very time-consuming process. This outlines the fact that automated segmentation of tumor region is crucial. In this project, we have developed a deep neural network to detect and segment the lung tumor regions.

This project is partially based on our work in the 2018 Video and Image Processing (VIP) Cup competition [9], organized by the IEEE Signal Processing Society, that we joined from June to September 2018. We have studied efficient methods for the segmentation of lung cancer tumor regions via screening CT scans and proposed a deep neural network to perform detection and segmentation of lung cancer tumor regions from more than 400 patients. Our proposed method uses a deep neural network for segmentation, followed by a radiomic analysis [10] to remove false positives from the segmentation results. In order to achieve better performance, another deep neural network is employed for radimoic analysis.

The major part of this work was published as the following paper:

Runze Zhang, Zhong Guan, Shun-Cheung Lai, Jun Xiao, and Kin-Man Lam (2018). "Deep Neural Networks for Lung Cancer Tumor Region Segmentation".

In Part I of this paper, the methods we used in our tumor-segmentation method and the final experimental results will be presented. Particularly, Chapter 1 of Part I states the

methodologies that we used in the deep neural network for lung tumor segmentation, and the experimental analysis and results will be discussed in Chapter 2 of Part I.

1.2 Machine Learning for Face Recognition

Nowadays, the development of face recognition results in many successful real-world applications covering a broad range of areas, such as finance, military, and public security. According to [11], face recognition typically contains two central tasks: face identification and face verification, where face identification solves the problem of correctly classifying a face image to a specific identity while the face verification focuses on if two faces are the same person or not. The general face recognition task can be also classified into open-set and closed-set face recognition based on different testing protocol [1]. As illustrated in Fig.1, under the closed-set protocol, all identities of the testing set are included in the training set. However, a real system cannot adopt such closed-set face recognition due to the fact that the gallery has limited identities. Therefore, the open-set protocol of face recognition, where the identities of the testing set not exist in the training set, makes face recognition much more challenging.

Before the introduction of deep neural networks, even traditional face recognition algorithms based on low-level face features can achieve good performance on frontal images. As one of the holistic approaches, Eigenface [12] has made the study of face recognition popular, which derive the low-dimensional representation of images through linear subspace. Local Binary Pattern (LBP) [3], one of the local-feature-based face recognition methods, demonstrated outstanding illumination robustness through some invariant properties of non-linear local filtering.

Over the past few years, with the extraordinary development of deep convolutional neural networks (CNNs), deep-learning-based face recognition achieved unprecedented performance improvement on various benchmarks. With the use of a cascade of multiple layers of performing convolution and a well-designed loss function, deep learning can extract multiple levels of abstract features to recognize faces with unprecedented stability [13].

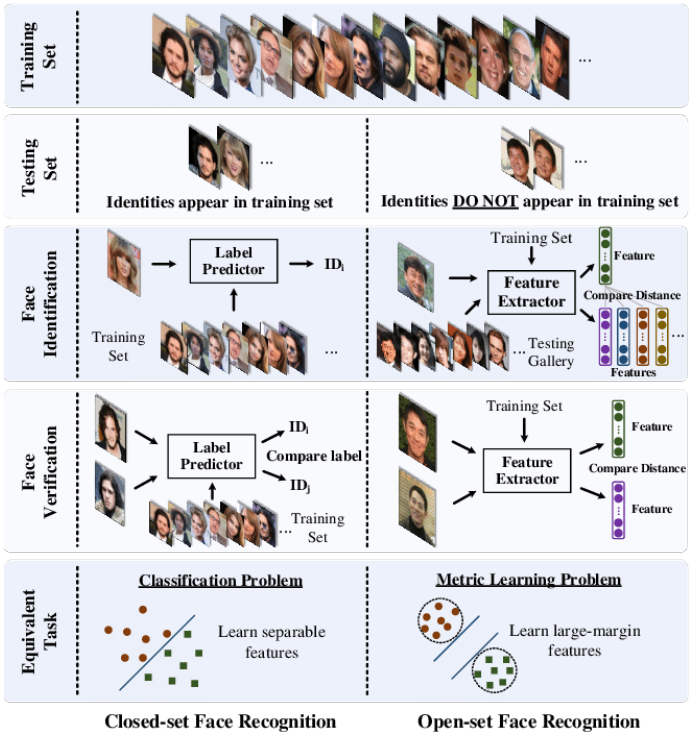


Figure 1: Comparison of open-set and closed-set face recognition [1]

In order to improve the performance of deep face recognition under the open-set protocol, many research works have done to improve neural networks so that they can better learn face features with larger inter-class variation and smaller intra-class distance. According to [11], the unconstrained conditions of face images, including different poses, age progression, varying illumination, and low resolution of images, make face recognition so difficult and many advanced methods cannot achieve satisfactory results in every condition. Therefore, developing a model that can learn discriminative feature representations in all unconstrained conditions has become the major topic of deep face recognition.

In Part II of this paper, eigenface [12], LBP [3], and deep face recognition [14] on open-set face recognition are evaluated as the baseline for algorithm comparison. For deep face reognition, ResNet [5] with different number of layers based on different loss functions, including Softmax loss, A-Softmax loss, and AM-softmax loss, are evaluated on many important face recognition benchmarks, i.e. Labeled Faces in the Wild (LFW) [15], QMUL-SurvFace [16], and Multi-PIE [17]. Chapter 1 of Part II states the related works of this project and Chapter 2 states the methodologies used in the projects. The details of the experimental setting will be shown in Chapter 3 and the results will be given in Chapter 4. In the end, the conclusion and possible future improvements will be given in Chapter 5.

Part I

Deep Neural Network for Lung Tumor Segmentation

1 Methodology

In this section, the techniques used in our published work on deep neural network for lung tumor region segmentation will be described. Section 1.1 is about the tumor cancer region segmentation done by U-Net and Section 1.2 is about using ResNet-18 to perform radimoic analysis for removing the false-positives generated by U-Net.

1.1 U-Net for Tumor Region Segmentation

In this work, U-Net [2] is used to perform end-to-end tumor detection and segmentation, which has shown outstanding performance on medical image segmentation. The structure used in our work is shown in Fig.2. Each blue block denotes two convolutional layers, with 3×3 filters, each is followed by a rectified linear unit (ReLU) layer. The white block represents the copied feature maps. Each downward arrow is a 2×2 max pooling layer with a stride of 2, where the number of feature channels is doubled. The upward arrow denotes a 2×2 upsampling convolution, where the number of feature maps will be halved. Finally, a 1×1 convolutional layer is used to normalize the results.

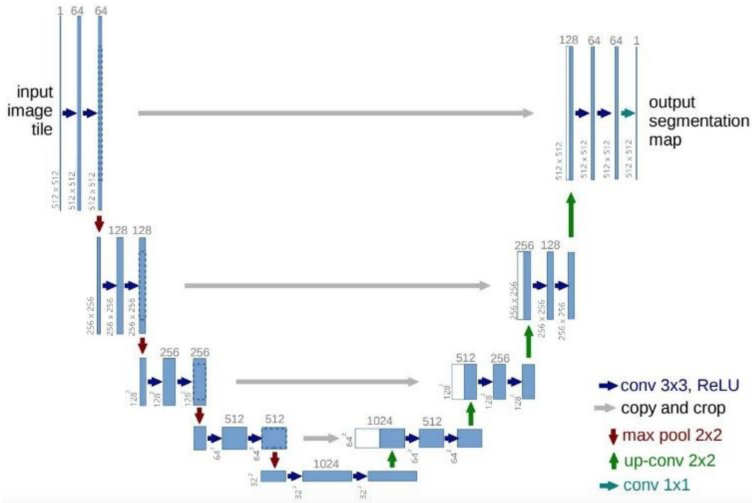


Figure 2: Structure of the U-Net model, modified from [2]

In [2], the cross-entropy loss is used as the training loss function for the deep neural network. However, because the size of the tumor is usually very small, the cross-entropy loss will most

likely result in a segmentation map only having the background. Therefore, we used the Dice loss [18] as our training loss function. The dice loss function is given as follows:

$$D = \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i + \sum_{i=1}^{N} g_i} \tag{1.1}$$

where N is the number of pixels in one CT slide, and p_i and g_i are the predicted binary segmentation result and the binary ground truth for pixel i in the CT slide.

At the training stage, we used data augmentation to enlarge the number of training data. We randomly cropped the boundary of the images with a random boundary size and then resized them to original image size. The images were also rotated and flipped. After data augmentation, we have totally 70,000 samples for training. We trained the network with 100 epochs and a batch size of 10. Overfitting was avoided to calculate the dice coefficient in the validation dataset subsequently. Adam optimizer [19] with a learning rate of 0.01 was adopted to improve the result.

1.2 ResNet-18 for Radimoic Analysis

In the field of medical study, radiomics [10] refers to the use of a huge volume of features extracted from medical images for comprehensive quantification of tumor phenotype. As there are a number of false-positives in the segmentation results generated by the U-Net deep learning model, a radiomic analysis is used to remove the false-positives and preserve the true-positives as many as possible. In our study, we selected the pre-trained ResNet-18 neural network model for the tumor classification.

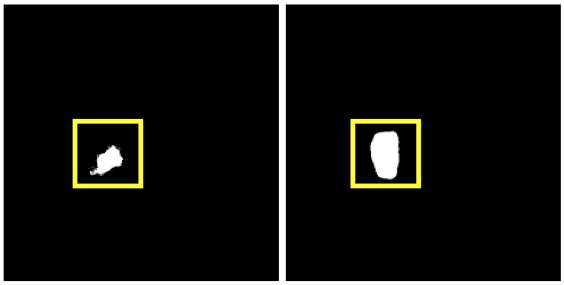


Figure 3: The region considered for feature extraction and classification

The classifier is the major component in a statistical pattern-recognition system. ResNet-18 [5] is used as deep features extractors and classifier for the regions segmented by U-Net to determine if the region is a true tumor or not. Fig.3 shows the segmentation results generated by U-Net. A 170×170 square window centered at each of the segmented regions is considered for feature extraction and classification, which is large enough to cover all the tumors. We modified the output of the last fully-connected layer of the original ResNet-18 from 1000 to 2 and used Adam [19] with a learning rate of 0.0001 for training.

2 Evaluation and Analysis

In order to report the performance of our proposed method, Six evaluation metrics were measured: the dice coefficient, mean surface distance, 95% Hausdorff distance, slice-wise missing rate, false-alarm rate, and CT-scan-based accuracy. Their definitions are given in the Appendix.

Since it is impossible to calculate the first three metrics on those slices without a tumor, we only focused on the slices with a tumor. The results, in terms of dice coefficient, mean surface distance, 95% Hausdorff distance, slice-wise missing rate, and false-alarm rate, based on U-Net only with the threshold of 0.0001, were 0.475, 27.014, 75.978, 22.2%, and 75.2%, respectively. By considering 3 consecutive slices to determine if the patient has a tumor or not, the accuracy is 95.0%, or the missed detection is reduced from 22.2% to 5%, which is called CT-scan-based accuracy.

As shown in Table 1, by applying the trained ResNet-18 to the segmentation results from U-Net, the overall results on the validation data set, in terms of dice coefficient, mean surface distance, 95% Hausdorff distance, slice-wise missing rate, false-alarm rate, and CT-scan-based accuracy, are 0.563, 10.896, 26.052, 23.3%, 24.6%, and 90.0%, respectively. Fig.4 shows some segmentation results on the validation data set. With ResNet-18, most of the false-positives can be removed, so the false-alarm rate can be reduced significantly while the overall accuracy can be maintained. Therefore, our proposed method can achieve satisfactory results on lung tumor segmentation.

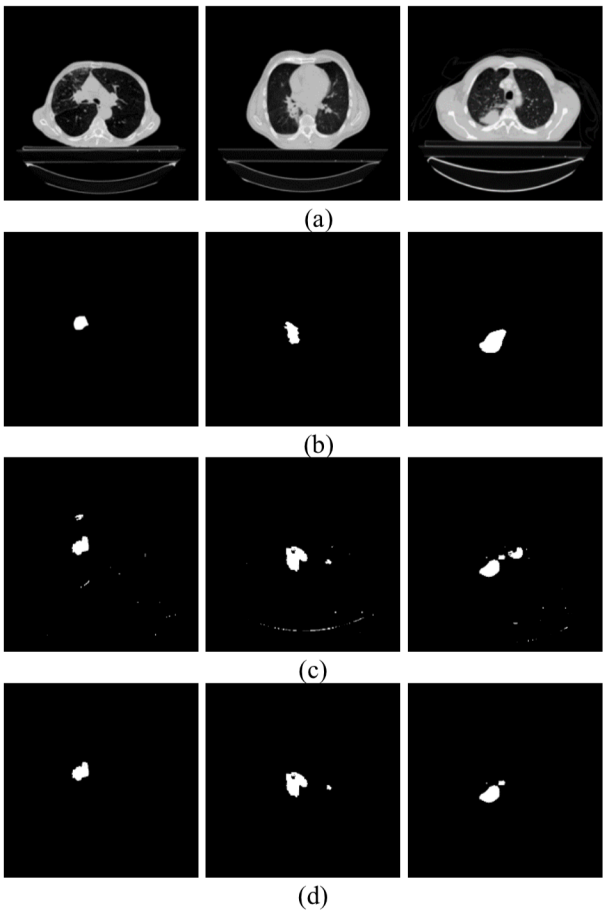


Figure 4: Segmentation results: (a) original images, (b) ground truth, (c) results generated by U-Net, and (d) results after radiomic analysis.

Metrics	U-Net Only	U-Net and ResNet-18
Dice coefficient	**0.475**	0.563
Mean surface distance	27.014	**10.896**
95% Hausdorff distance	75.978	**26.052**
Slice-wise missing rate	**22.2%**	23.3%
False-alarm rate	75.2%	**24.6%**
CT-scan-based Accuracy	**95.0%**	90%

Table 1: Performance of our proposed method with a threshold of U-Net = 0.0001

Part II

Machine Learning for Face Recognition

1 Related Work

In this chapter, the related works about this project will be discussed, which can be mainly divided into three parts. Firstly, the deep learning in Section 1.1 is about deep neural networks which can apply to large volumes of data. Secondly, the unconstrained face recognition in Section 1.2 is about the mainstream face recognition methods designed for unconstrained conditions. Finally, the recent work of deep face recognition will be discussed in Section 1.3.

1.1 Deep Learning

In recent years, large volumes of images have uploaded to the networks and other media because of the development of the internet and search engines. These images typically contain many objects in unconstrained condition with different resolution and illuminations, which are the perfect resources for researchers to conduct experiments on their recognition models.

Many powerful statistical models, such as Support Vector Machines [20], Principal Component Analysis [21] and Linear Discriminant Analysis [22], are widely improved because of the availability of huge volume of data and effective computational resources, which significantly improved the performance of computer vision in many applications. However, these models still have limited capacity to apply to a huge amount of data.

Instead, deep learning has exhibited great computational capacity and scaling properties. When deep networks are trained with a large number of training data on the resources with high computational power, such as hundreds of CPU cores [23] and/or GPU [13], they can achieve impressive results. In [13], Krizhevsky *et al.* propose a deep convolutional neural network which can achieve outstanding recognition accuracy on various objects when trained on a large dataset by utilizing standard backpropagation [24]. [14] develop a deep neural network with nine layers that can obtain face representation from a large labeled dataset, which achieved state-of-the-art face verification performance at that time.

1.2 Unconstrained Face Recognition

In some unconstrained domain, face recognition methods have achieved good results by combining a number of hand-crafted local descriptors together, such as SIFT [25], LBP [3], and Gabor representations [26]. In [27], Nguyen *et al.* combine pixel intensity, LBP, and Gabor representations to form the feature representation and use cosine similarity as distance metric on face verification. [28] proposed a Generative Adversarial Network (GAN) with a typical CNN for frontal face synthesis and large pose face recognition, which demonstrates compelling perceptual results on MultiPIE. Yin *et al.* [29] use pose information from MultiPIE to minimize intra-class distance in LFW which achieve state-of-the-art performance among the methods using labeled training data excluded from LFW.

1.3 Deep Face Recognition

With the use of a cascade of multiple convolutional layers, deep face recognition are able to extract the discriminative features of faces with a relatively simple classification architecture and a well-defined loss function. [14] uses CNNs with Softmax loss to address open-set face recognition problem. In [5], He *et al.* propose a residual neural network (ResNet) that can encourage the network to learn face features with deep layers. [30] propose a center loss to minimize the distance between the deep features of the same class and their center, which can efficiently minimize the intra-class distance of deep features. In [8], Liu *et al.* introduce an angular margin to the original Softmax function and formulate the A-softmax loss to project deep features onto a hypersphere, which pushes features of same class much closer together. Based on the A-softmax loss, [31] further improves the A-softmax loss by introducing an additive margin to the Softmax to encourage intra-class variance minimization, which achieves current state-of-the-art face recognition accuracy.

2 Methodology

In this chapter, the essential techniques and models utilized and referenced in this project will be discussed. One is the Eigenface [12] method described in Section 2.1. Another is the LBP [3] method stated in Section 2.2. Finally, deep face recognition, including the neural network architecture and loss function, will be discussed in Section 2.3.

2.1 Eigenface

The Eigenface approach was proposed in 1991 by Pentland et al. [12], which uses the eigenvectors of the set of faces for face recognition. The eigenfaces can be extracted out of original image data by Principal Component Analysis (PCA) [21], which simply projects face images onto a low-dimensional feature space. These eigenfaces can be viewed as characteristic features of the faces and each face can be reconstructed by combining the eigenfaces. According to [21], a certain weight is associated with each eigenface which represents to what extent the eigenface exist in the face image. Therefore, the face images with similar weights are likely to be of the same identity. To summarize, the algorithms of face recognition by using eigenfaces approach are as follows [12]:

Algorithm 1 Eigenface

1: Calculate the average face Ψ of the training dataset $\Gamma_1, \Gamma_2, ..., \Gamma_n$

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \quad \cdots\cdots (1)$$

2: Subtract each training face $\Gamma_1, \Gamma_2, ..., \Gamma_n$ by the average face Ψ

$$\Phi_i = \Gamma_i - \Psi \quad \cdots\cdots (2)$$

3: Calculate the eigenvalue μ_i and eigenvectors $\boldsymbol{v_i}$ for the matrix A = $[\Phi_1, \Phi_2, ...\Phi_n]$

$$AA^T\boldsymbol{v_i} = \mu_i\boldsymbol{v_i} \quad \cdots\cdots (3)$$

The eigenvectors for the matrix $C = AA^T$ can be solved by first calculating the eigenvectors of A^TA.

Multiply both sides of Eq.(3) by A^T:

$$A^TAA^T\boldsymbol{v_i} = \mu_iA^T\boldsymbol{v_i} \rightarrow A^TA\boldsymbol{u_i} = \mu_i\boldsymbol{u_i} \quad \cdots\cdots (4)$$

$$\boldsymbol{u_i} = A^T\boldsymbol{v_i} \rightarrow \boldsymbol{v_i} = A\boldsymbol{u_i} \quad \cdots\cdots (5)$$

4: Sort the eigenvectors $\boldsymbol{v_i}$ according to the value of the eigenvalue μ_i, then select the M most significant eigenvectors for face recognition

5: Transform new face images Γ into its eigenface components

$$\omega_k = \boldsymbol{v_k}^T(\Gamma - \Psi),\ \Omega = [\omega_1, \omega_2, ..., \omega_M] \quad \cdots\cdots (6)$$

6: Calculate the L2-norm distance between weight vector of two faces as the similarity score

$$\epsilon = ||\Omega_i - \Omega_j||^2 \quad \cdots\cdots (7)$$

7: **if** $\epsilon >$ threshold **then return** 1 (matching identity) **else return** 0 (nonmatching identity)

2.2 Local Binary Pattern

[3] proposes a Local Binary Pattern (LBP) operator to perform face classification. It is suitable for open-set face recognition due to its high discrimination power and computationally efficiency.

LBP summarizes local features of face images by considering the neighborhood of each pixel. According to [3], a face image is first divided into many local regions of the same size and each pixel in each local region is compared with its eight neighbors in a 3x3 window. The resulting positive values are encoded with 1 and the others are 0. Concatenating these 0 and 1 starting from the top-left pixel in a clockwise direction to form a binary number and its corresponding decimal value is referred to as the label [3]. The illustration of the basic LBP operator is shown as the Fig.5.

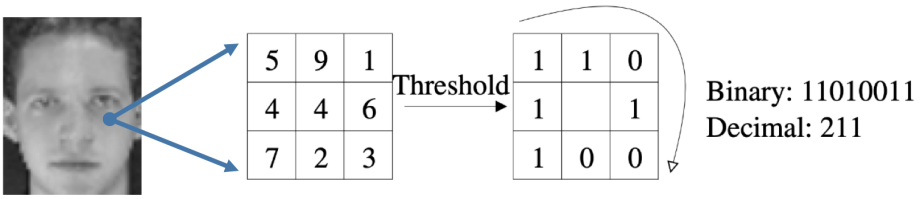


Figure 5: LBP operator illustration based on [3]

After the local binary patterns of the face image are extracted, the decimal value can be formed into a sequence of a histogram. For 8-bits binary code, there are 256 different decimal values. Therefore, each face image can be represented as a sequence of a histogram with 256 bins.

According to [3], Chi-square distance is used as the evaluation metric to calculate the similarity of two sequences of histograms of two faces. As illustrated in the Fig.6. Two faces can be decided if they are of the same identity or not by thresholding the score. The Chi-square distance metric is defined as:

$$\epsilon = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i} \tag{2.1}$$

where n is a number of bins, x_i is the i-th bin of the first face, y_i is the i-th bin of the second face.

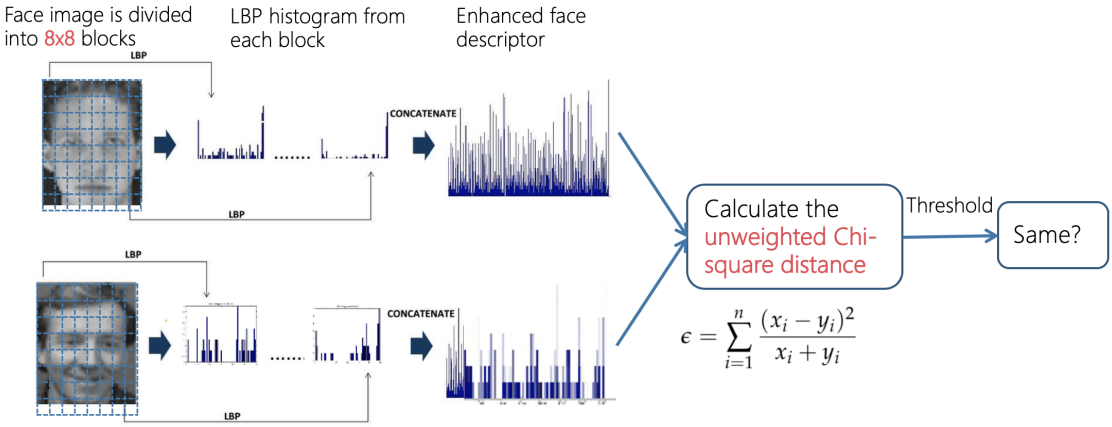


Figure 6: The working pipeline of LBP-based face recognition

2.3 Deep Face Recognition

The general deep face recognition pipeline is shown as Fig.7. There are training phase and testing phase for the deep CNNs to perfrom face recognition tasks. In the training phase, the deep CNN uses a huge labeled face dataset as input to update its weights and bias and the loss function can be optimized. At the test stage, the deep CNN is used as a feature extractor and a distance evaluation metric is used to compare face descriptors for the purpose of either face identification or face verification. The weights and bias of deep CNNs during the training phase and the testing phase are shared with each other. In this project, the ResNet [5] is used as the CNN backbone and different loss functions, including Softmax, A-Softmax, and AM-Softmax, are evaluated on many important face recognition benchmarks.

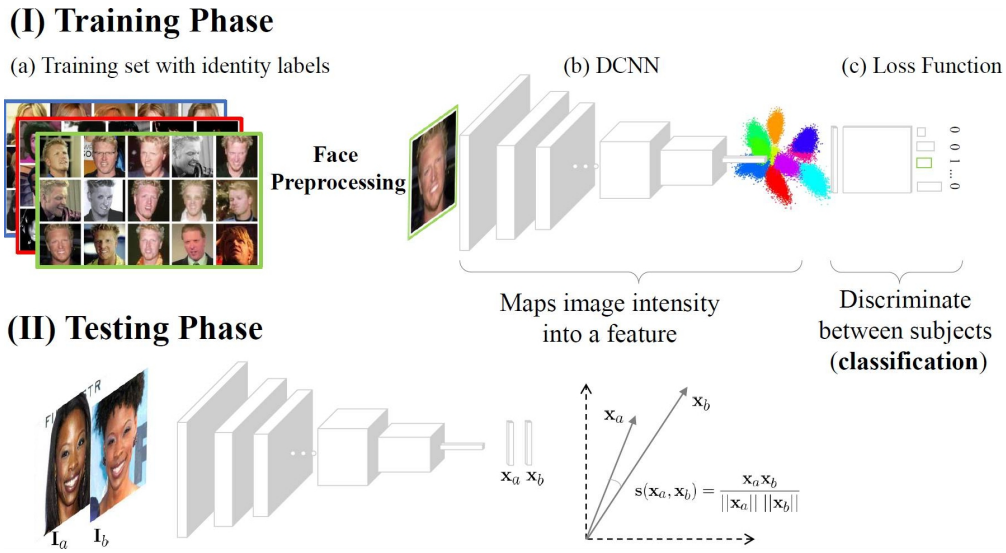


Figure 7: The working pipeline of Deep face recognition [4]

2.3.1 Residual Neural Network

Although deep neural networks can extract multiple levels of facial features and can achieve outstanding face recognition accuracy, it is very difficult to train, especially when the neural network becomes very deep. It is because that the gradient can gradually vanish along with the layers. He *et al.* [5] propose the residual neural network (ResNet) to solve the degradation problem of the traditional neural network during training. As shown in the Fig.8, the ResNet use "shortcut connections" to feedforward neural networks by reformulating the layers into $y = F(x) + x$ while the plain neural networks only use $y = F(x)$, where x is the input of the residual unit and $F(x)$ is the output of a number of convolutional layers [5]. Experiments show that the ResNet can enable deep rectified models to be trained directly from scratch.

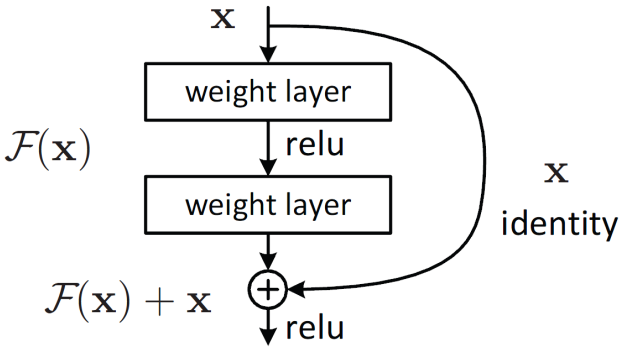


Figure 8: Residual learning: a building block [5]

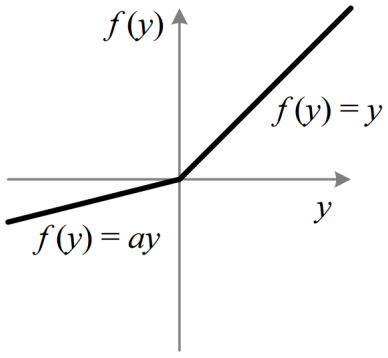


Figure 9: PReLu Activation Function [6]

Activation function plays an important role in the deep neural network, which performs a non-linear operation on the output of each layer. Without the non-linear activation function, the deep neural network is simply a linear regression model, which cannot represent non-linear complex mappings between input and output. In this project, the Parametric Rectified Linear Unit (PReLU) [6] is used as the activation function after each convolutional layer of ResNet instead of using the common ReLu. The property of PReLu is shown in the Fig.9. PReLU suggests that all the positive values can pass through the activation while there would be a parametric penalty for negative values to pass through, which can solve the 'vanishing gradient' problem stated in the [32].

2.3.2 Softmax

Nowadays, the softmax loss proposed by [8] is widely used to supervise the deep CNNs for face verification models. More specifically, the softmax loss used in the deep CNNs is the combination of softmax function and cross entropy function. Like many other activation functions, softmax function takes an N-dimensional vector of real numbers and normalizes it into a probability distribution in the range of (0,1) [8]. The cross-entropy function is then used as the loss function to indicate the distance between the true labels and the predicted labels. The softmax loss is defined as follows:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log(\frac{e^{f_{y_i}}}{\sum_i e^{f_j}}) \tag{2.2}$$

in which N is the number of training samples and f_j represents the j-th element of the class score vector $\boldsymbol{f}$. $f_j = \boldsymbol{W_j^T x_i} + b_j$ and $f_{y_i} = \boldsymbol{W_{y_i}^T x_i} + b_{y_i}$, where $\boldsymbol{x_i}, \boldsymbol{W_j}, \boldsymbol{W_{y_i}}$ are the i-th training sample, the j-th and y_i-th column of $\boldsymbol{W}$, respectively [14]. Therefore, the softmax loss L_i in Eq.2.2 can be reformulated as

$$L_i = -\log(\frac{e^{\boldsymbol{W_{y_i}^T x_i}+b_{y_i}}}{\sum_i e^{\boldsymbol{W_j^T x_i}+b_j}}) = -\log(\frac{e^{||\boldsymbol{W_{y_i}}||||\boldsymbol{x_i}||\cos(\theta_{y_i,i})+b_{y_i}}}{\sum_i e^{||\boldsymbol{W_j}||||\boldsymbol{x_i}||\cos(\theta_{j,i})+b_j}}) \tag{2.3}$$

where $\theta_{j,i}(0 \le \theta_{j,i} \le \pi)$ is the angle between weights $\boldsymbol{W_j}$ and feature $\boldsymbol{x_i}$.

2.3.3 A-Softmax

However, both [30] and [33] found that the softmax loss lacks the discriminating power for classification because it cannot make intra-class features compact. Therefore, [8] proposed A-Softmax loss to use an angular space to represent the Euclidean space of features. According to [8], unlike the original softmax loss, the weight vectors are normalized $||\boldsymbol{W}|| = 1$ and the biases are zero $b = 0$ in the A-Softmax loss, which predicts the results only considering the angles between the weight vector $\boldsymbol{W}$ and the feature $\boldsymbol{x}$. An angular margin m is multiplied to the target angle θ_{y_i} to learn an angular margin between different classes, which imposes discriminative power to the learned features by requiring $||\boldsymbol{W_1}||||\boldsymbol{x}||\cos(m\theta_1) > ||\boldsymbol{W_2}||||\boldsymbol{x}||\cos(\theta_2)$. Because the cosine function is non-monotonic, a piece-wise function is applied to guarantee the monotonicity for the A-Softmax [8]. Therefore, the formulations of the A-Softmax loss is proposed as

$$L_{A-Softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log(\frac{e^{||\boldsymbol{f_i}||\psi(\theta_{y_i})}}{e^{||\boldsymbol{f_i}||\psi(\theta_{y_i})} + \sum_{j=1,j\neq y_i}^{c} e^{||\boldsymbol{f_i}||\cos(\theta_j)}}) \tag{2.4}$$

where $\psi(\theta) = \frac{(-1)^k\cos(m\theta)-2k+\lambda\cos(\theta)}{1+\lambda}, \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$

2.3.4 AM-Softmax

In [31], A-Softmax is found to be difficult to converge due to the angular margin is generated by the multiplication of angle and m. Therefore, [31] and [7] both proposed the AM-Softmax loss $\cos(\theta) - m$ to formulate an additive angular margin, which is much easier to implement and converge. Meanwhile, both feature normalization and weight normalization are applied to build a cosine layer as well as a hyper-parameter s is then used to scale the cosine values [34]. In [31], the AM-Softmax loss is formulated as

$$L_{AM-Softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{s\cdot(\cos\theta_{y_i}-m)}}{e^{s\cdot(\cos\theta_{y_i}-m)}+\sum_{j=1,j\neq y_i}^{c}e^{s\cdot\cos\theta_j}}\right) \tag{2.5}$$

2.3.5 Decision Boundaries

In this subsection, the decision margin of the above-mentioned loss functions: Softmax, A-Softmax, and AM-Softmax are compared as illustrated in Fig.10. Meanwhile, the decision boundary equations of these three losses under the binary classification case are given in Table 2.

From the Fig.10, we can know that the angular margin-based loss functions have a larger decision boundary and smaller intra-class variance than the normal Softmax loss function. It indicates that both A-Softmax and AM-Softmax can result in discriminative feature representations by introducing margin constraints in the Softmax.

Loss Functions	Decision Boundaries
Softmax	$(\boldsymbol{W_1}-\boldsymbol{W_2})\boldsymbol{x}+b_1-b_2=0$
ASoftmax	$\|\|\boldsymbol{x}\|\|(\cos m\theta_1-\cos\theta_2)=0$
AM-Softmax	$\hat{\boldsymbol{x}}(\cos\theta_1-m-\cos\theta_2)=0$

Table 2: Decision boundaries of loss functions under the binary-class case for class 1, where $\hat{x}$ is the normalized feature

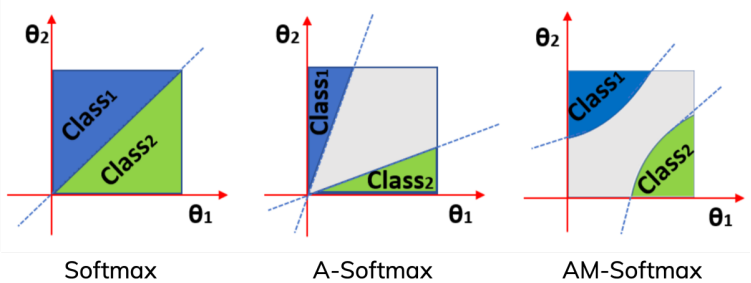


Figure 10: Decision margins under a binary classification case for loss functions [7]

3 Experimental Setting

In this chapter, the experimental settings are described in details, which can be divided into six parts. Training datasets are stated in Section 3.1 and benchmark datasets are introduced in Section 3.2. In Section 3.3, some techniques used to preprocess the datasets are described. The experimental setup of Eigenface, LBP, and deep face recognition are described in Section 3.4, Section 3.5, and Section 3.6, respectively.

3.1 Training Dataset

The publicly available CASIA-WebFace [35] is used as the training dataset, which is one of the largest scale image datasets for face recognition. The CASIA-WebFace dataset contains 494,414 face images out of 10,575 identities, which are all collected from the web [35]. The reason to choose WebFace dataset as training data is because the faces are all captured and annotated from the website under uncontrolled condition. Therefore, the model trained on the WebFace can be more easily to generalize to new environments in practical applications.

3.2 Benchmark Dataset

In the evaluation phase, we selected three different datasets to evaluate the implemented methods. Labeled Faces in the Wild (LFW) [15] comprises 13,323 web photos belonging to 5,749 celebrities which are divided into 6,000 face pairs in 10 splits, including 3,000 matched face pairs and 3,000 mismatched face pairs, for face verification. The SurvFace [16] is one of the largest low-resolution face datasets, which contains 463,507 face images belonging to 15,573 identities captured in real-world surveillance scenes across wide space and time. The Multi-PIE [17] is a large dataset which contains more than 750,000 face images with huge changes in pose, illumination, and expression, which is the best dataset to evaluate the performance of the neural network in a condition with a large variance in luminance. We use images with neutral expression under 20 different illumination conditions and 8 face angles within $\pm 90°$as the probe images, which has totally 126,819 images of 249 identities. The frontal faces with the best illumination of each identity are used as the gallery images.

We evaluate 1:N face recognition on MultiPIE dataset and evaluate face verification on both LFW and SurvFace by strictly following the unrestricted with labeled outside data protocol [15]. The statistics of used benchmark datasets are summarised in Table 3.

Dataset	# photos	# subjects	Metrics
LFW [15]	13,323	5,749	1:1: Acc, TAR vs. FAR (ROC)
SurvFace [16]	463,507	15,573	1:1: Acc, TAR vs. FAR (ROC)
MultiPIE [17]	126,819	249	1:N: Rank-N (CMC)

Table 3: Statistics of benchmark datasets

3.3 Data Preprocessing

We followed the standard data preprocessing procedures in [8]. The overlapped identities between training data and benchmark dataset are removed to make sure open-set face recognition can be performed. Because the face images in SurvFace have extremely low resolution, it is difficult to detect the facial landmarks by using the existing methods. Therefore, we did not perform face alignment for the SurvFace. Other than SurvFace, facial landmarks for the entire training and testing images are detected by MTCNN [36]. Then, each face image is aligned by performing the similarity transformation upon the detected 5 facial points and resized to be 112 x 96. Some WebFace faces before and after alignment are shown in Fig.11. Fig.12 shows some LFW example face pairs and their aligned faces. Fig.13 shows some face images in SurvFace dataset and Fig.14 shows face images and their corresponding aligned faces in MultiPIE dataset.

For the LBP experiments, the RGB images are transformed into grayscale images, from which the LBP features would be extracted. For the deep CNNs experiment, each RGB image is normalized by subtracting each pixel 127.5 and then dividing by 128. In order to augment the dataset, each face image is horizontally flipped to concatenate to the original face image.

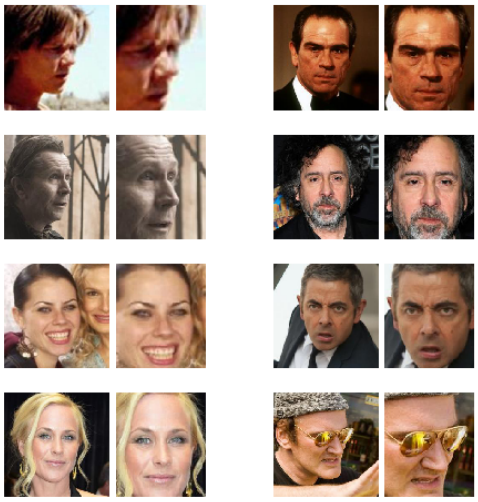


Figure 11: Example faces in CASIA-WebFace dataset. Left are the original faces and right are the aligned and resized faces

Figure 12: Example face pairs in LFW dataset. Left are the original 250 x 250 face pairs and right are aligned and resized face pairs.

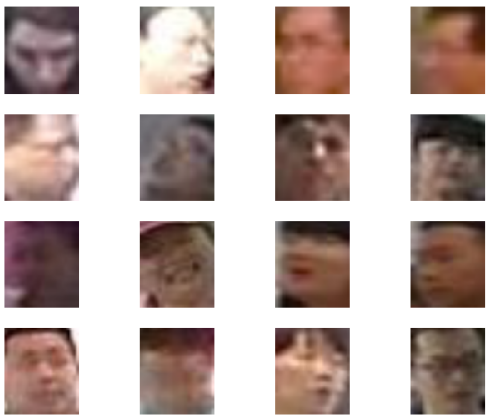


Figure 13: Example faces in SurvFace dataset

Figure 14: Example faces in MultiPIE dataset and their aligned faces

3.4 Eigenface Setup

During the EigenFace experiment, the average face of CASIA-WebFace is calculated as shown in Fig.15 and the most 50 significant principal components are selected. Some eigenfaces of the calculated from the CASIA-WebFace are shown in Fig.16. Then, the 50 principal components are used to transform the testing dataset into their corresponding eigenface components. L2-norm distance is calculated between the weights of two faces and the optimal threshold is found between the minimum and maximum of the distance for the EigenFace to achieve the best accuracy.

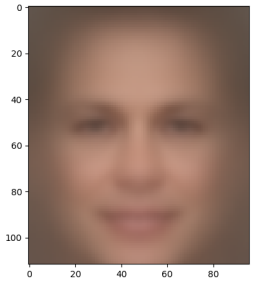


Figure 15: The average face of training dataset, with a size of 112 x 96

Figure 16: Twelve of the most significant eigenfaces calculated from the CASIA-WebFace

3.5 LBP Setup

Because there is no training step for LBP experiment, LBP is directly performed on the pair of testing data for face verification. Each 112 x 96 testing face image is firstly divided into 168 local regions with a size of 8 x 8. Then, histogram sequences of uniform LBP patterns with 3x3 sizes are extracted from each local region. Unweighted Chi-square distance is calculated between the histograms of two faces and the optimal threshold is found between the minimum and maximum of the distance for the LBP to achieve the best accuracy.

In addition to using RGB images as the input of the deep neural networks, the LBP feature of each face are used as the additional input channel to the CNNs. The network architecture is shown in the Fig.17. In addition, we also tried to only use the LBP features as the input without the RGB channel of images. Their results compared with the traditional deep face recognition are shown in Chapter 4.

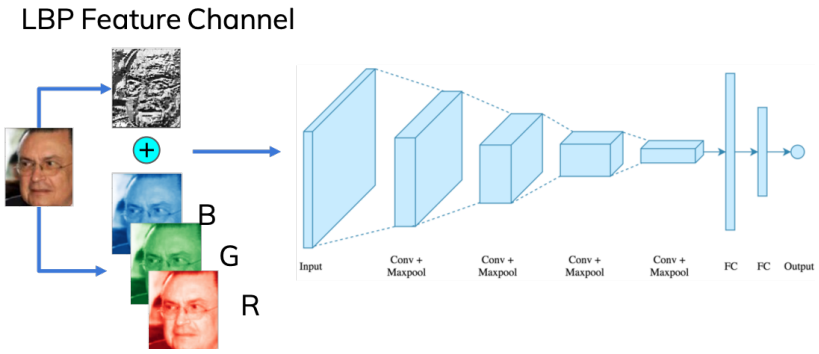


Figure 17: Using LBP features as the additional input feature channel of CNNs

3.6 CNNs Setup

Pytorch is used to implement the architecture and loss function in the experiment. We followed the entire framework shown in Fig.18 to train and test the deep CNNs for face recognition. The architecture of ResNet-20 used in the experiment is shown in Fig.19, which has 20 convolutional layers. Each blue block represents a 3x3 convolutional layer with a padding of 1, separately followed by a PReLU [6] layer. The loop with an alongside count denotes the set of residual connections [5] which contains two convolutional layers with a stride of 1. The convolutional layers outside the residual blocks have a stride of 2. Finally, the results are normalized by a fully-connected layer followed by a Softmax layer. All the CNN architectures with different convolutional layers that we used are shown in Table 4.

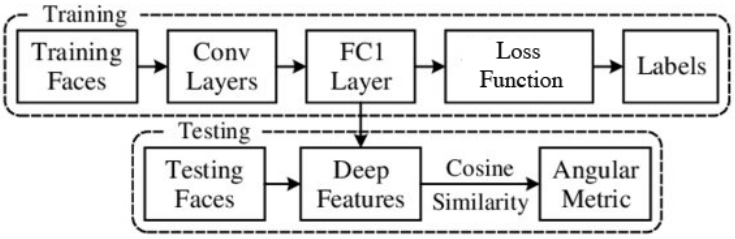


Figure 18: Training and extracting deep features for open-set face recognition. [1]

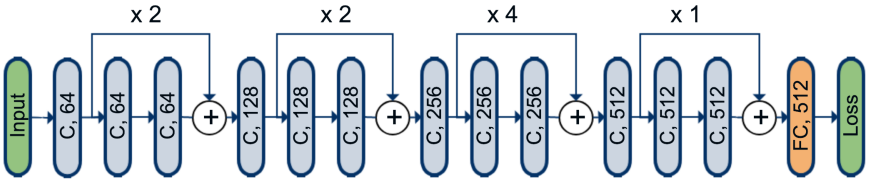


Figure 19: The ResNet-20 architecture used for the experiments. ***C*** denotes Convolution Layer followed by PReLU and the loop denotes residual unit

Layer	10-Layer CNN	20-Layer CNN	64-Layer CNN
Conv1.x	$[3\times3, 64] \times 1, S2$	$[3 \times 3, 64] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 1$	$[3 \times 3, 64] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
Conv2.x	$[3 \times 3, 128] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 1$	$[3 \times 3, 128] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$[3 \times 3, 128] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 8$
Conv3.x	$[3 \times 3, 256] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$[3 \times 3, 256] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 4$	$[3 \times 3, 256] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 16$
Conv4.x	$[3\times3, 512] \times 1, S2$	$[3 \times 3, 512] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 1$	$[3 \times 3, 512] \times 1, S2$ $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
FC1	512	512	512

Table 4: CNN architectures with different convolutional layers used in the project, modified from [8]. Double-column brackets represent the residual units

Training. The parameters settings during training we used are the same as the setting in [31]. During the training, the learning rate is set to be 0.1 initially and then divided by 10 at the 16K, 24K, 28K iterations, and the training is completed at 30K iterations. The batch size is set to be 256 and the weight decay is 5e-4. The scaling parameter s used to scale the cosine values for all three loss functions is fixed to be 20. For the A-Softmax, the multiplicative angular margin m is 4. The additive angular margin m is set to 0.35 for the AM-Softmax.

Testing. At the testing stage, features of both flipped face images and frontal face images are extracted by the CNN from the first fully-connected layer. The final representation of the face image is composed by concatenating the two features together. In order to compare two faces, the cosine similarity is computed as the measurement and a threshold is used to determine the results.

In this Chapter, the experimental setting for eigenface, LBP, and deep face recognition was discussed in detail. We have also discussed the training and benchmark dataset we used in our project. In the next Chapter, the evaluation results on the benchmark datasets and important findings will be discussed.

4 Evaluation and Analysis

In this project, LBP and eigenface are evaluated on the LFW benchmark and ResNet based on different loss functions, including Softmax, A-Softmax, and AM-Softmax loss functions, are evaluated on LFW, MultiPIE, and SurvFace benchmarks by strictly following the unrestricted with labeled outside data protocol [15]. We also evaluated the effects of the number of convolutional layers on the deep face recognition. In addition, we use the LBP features as the input of the ResNet and their performances are also reported.

4.1 Evaluation on LFW

LFW benchmark contains 10 batches, where each batch has 300 matched pairs and 300 mismatched pairs. Therefore, 10-folds cross-validation shown in Fig.20 is used to report the face recognition accuracy on LFW benchmark.

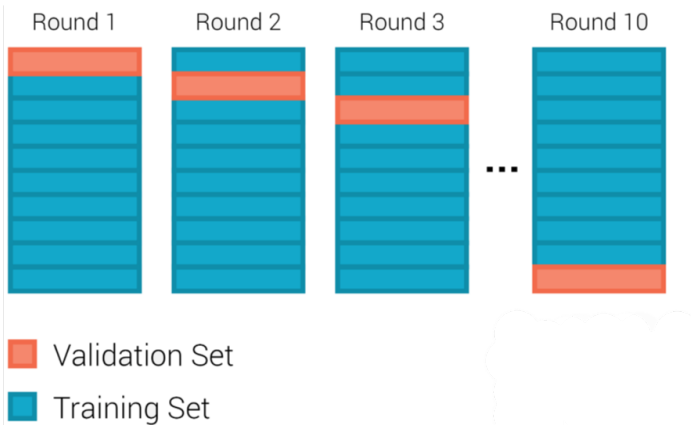


Figure 20: 10-folds cross-validation on LFW

For each round, the best threshold is found in 9 training sets and the accuracy in the validation set is calculated by using this threshold. The reported accuracy is the average value of the 10 accuracies:

$$\hat{\mu} = \frac{\sum_{i=1}^{10} p_i}{10} \tag{4.1}$$

The reported standard deviation is given by:

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{10} (p_i - \hat{\mu})^2}{9}} \tag{4.2}$$

4.1.1 Experimental Results

With the use of 50 principal components trained on the CASIA-WebFace dataset, the EigenFace method achieves 60.85% accuracy on LFW dataset while the unweighted LBP can achieve 68.47%.

For the deep face recognition, when the ResNet-10 is used as the backbone, softmax loss, A-Softmax, and AM-Softmax can achieve 96.00%, 98.78%, and 98.58% on LFW, respectively. With the use of ResNet-20 as the backbone, which has 20 convolutional layers, softmax loss, A-Softmax loss, and AM-Softmax loss can achieve 96.28%, 98.92%, and 99.05% accuracy on LFW, respectively. When the ResNet-64 is used as the backbone, which has 64 convolutional layers, softmax loss, A-Softmax and AM-Softmax can have 97.27%, 99.25%, and 99.18% accuracy, respectively. The results are shown in Table 5.

Backbone	Loss	Accuracy (%)
ResNet-10	Softmax Loss	96.00 ± 0.65
	A-Softmax Loss	**98.78 ± 0.51**
	AM-Softmax Loss	98.58 ± 0.64
ResNet-20	Softmax Loss	96.28 ± 1.17
	A-Softmax Loss	98.92 ± 0.52
	AM-Softmax Loss	**99.05 ± 0.43**
ResNet-64	Softmax Loss	97.27 ± 0.82
	A-Softmax Loss	**99.25 ± 0.36**
	AM-Softmax Loss	99.18 ± 0.38

Table 5: Accuracy (%) of deep-learning-based face recognition on LFW

Then we investigated the effects of the input data, especially the LBP features, on the performance of deep face recognition. As shown in the Fig.21, when we add LBP features to be the additional channel to the ResNe-20, the softmax, A-Softmax, and AM-Softmax can achieve 96.63%, 98.95%, and 98.5%, respectively. If we use the LBP features as the only one input channel of the CNNs, the results for softmax, A-Softmax, and AM-Softmax are, 95.40%, 98.52%, and 98.28%, respectively.

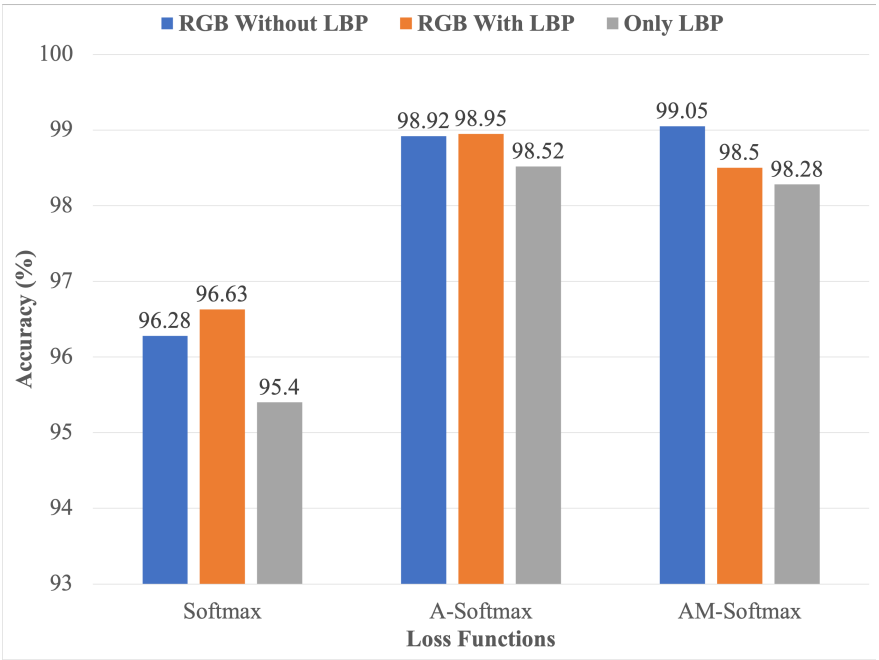


Figure 21: Accuracy (%) on LFW with different input data for ResNet-20

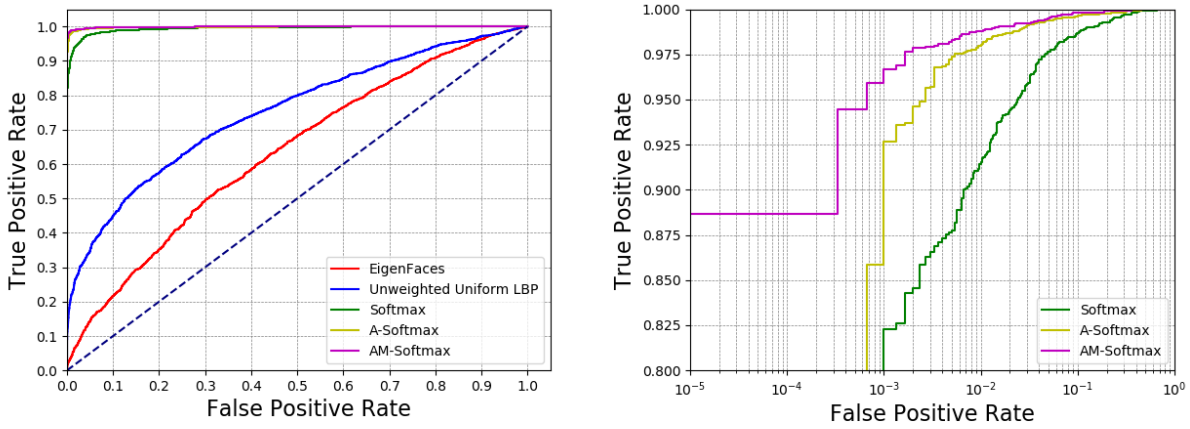
4.1.2 Findings

From Table 5, one can observe that the larger number of convolutional layers can result in a better performance on the LFW benchmark. Another point can be observed is that A-Softmax and AM-Softmax consistently outperform the softmax loss. However, both AM-Softmax and A-Softmax can perform similarly on the LFW benchmarks, which indicates that their studies on the LFW have already become saturated.

In Fig.22a, the Receiver Operating Characteristic (ROC) curves are drawn to better evaluate the difference of Eigenface, LBP, and deep-learning-based method on the open-set face verification, where ResNet-20 is used as the backbone for comparison. From this figure, it can be clearly seen that the deep face recognition performs much better than EigenFace and LBP. However, the three loss functions: Softmax loss, A-Softmax loss, and AM-Softmax loss, cannot display any noticeable difference on the LFW benchmark. Therefore, the Fig.22b is drawn to only display the ROC curves of deep face recognition, which can clearly show the classification performance of AM-Softmax is better than the Softmax and A-Softmax on the LFW when ResNet-20 is used as the backbone.

From Fig.21, one can observe that adding LBP features as one additional channel to the CNNs based on Softmax and A-Softmax can result in almost the same results as the CNNs using RGB channels as input on the LFW. However, adding additional LBP features can lead to worse performance for AM-Softmax-based CNNs. It can be also noticed that the CNNs can still perform well when the input data is LBP features only without RGB channels, indicating that local descriptors can also help CNNs to learn the features and do the classifications to

some extent. Therefore, more possible ways can be figured out to leverage other image descriptors and improve the performance of deep face recognition by using those descriptors as the input channel of the neural network.



(a) All approaches

(b) Deep face recognition only

Figure 22: ROC curves of different approaches on LFW benchmark (ResNet-20 is used for deep face recognition)

4.2 Evaluation on MultiPIE

MultiPIE dataset has a huge difference across the face angle, illumination, and expression, which becomes one of the best datasets to evaluate the performance of deep CNNs in unconstrained condition with a large variance in illumination and face angle. Therefore, we evaluated the ResNet-20 with different configurations on the MultiPIE in order to see how these configurations can affect the CNNs on the face recognition in different illuminations and pose angles.

We performed 1:249 face identification on the MultiPIE dataset, where 249 denotes the number of gallery images. As mentioned in Chapter 3, we only use images with neutral expression across 8 poses and 20 illuminations. We calculated the cosine similarities between probe images and gallery images and reported Rank-1 recognition accuracy.

4.2.1 Experimental Results

Table 6 shows the Rank-1 recognition rate of the ResNet-20 based on different loss functions on the MultiPIE dataset across different pose angles, including the Softmax loss, A-Softmax loss, and AM-Softmax loss. All these models are trained by the WebFace.

Method	±90°	±75°	±60°	±45°	±30°	±15°
ResNet-20 + Softmax	22.19	52.70	82.33	97.32	99.31	99.78
ResNet-20 + A-Softmax	25.74	59.25	89.33	99.00	99.85	99.98
ResNet-20 + AM-Softmax	**48.86**	**76.39**	**93.39**	**99.19**	**99.87**	**99.99**

Table 6: Rank-1 recognition rate (%) of ResNet-20 on different poses

Table 7 shows Rank-1 recognition rate of ResNet-20 with different input data, including the normal RGB channels, the RGB channels plus the additional LBP feature channel, as well as the only one LBP feature channel, on different poses. The scores in bold denote the best results of ResNet-20 for each loss function with different input and the underlined scores are the best recognition results among all the approaches.

In addition, in order to know the effect of input LBP features on the deep face recognition, especially to see whether it can help deep face recognition when the lighting condition is poor, we evaluated the performance of Softmax-based CNNs with different input data on the MultiPIE across 20 illuminations. Because the recognition rate under one illumination is the averaged result of all the poses. Therefore, we only look at the results when the pose angle is 60 °. The results are shown in Table 8.

Loss Function	Input Data	±90°	±75°	±60°	±45°	±30°	±15°
Softmax	RGB image	22.19	52.70	82.33	**97.32**	99.31	99.78
	RGB plus LBP	**23.35**	**53.81**	**84.56**	97.28	**99.55**	**99.85**
	Only LBP	12.04	37.56	72.62	93.45	98.45	99.67
A-Softmax	RGB image	**25.74**	**59.25**	**89.33**	99.00	99.85	**99.98**
	RGB plus LBP	23.91	56.21	88.29	**99.21**	**99.88**	**99.98**
	Only LBP	5.61	25.26	68.80	95.58	99.22	99.63
AM-Softmax	RGB image	**48.86**	**76.39**	**93.39**	**99.19**	**99.87**	99.99
	RGB plus LBP	42.72	72.07	91.86	98.78	99.83	**100.00**
	Only LBP	30.34	65.71	89.56	98.43	99.86	99.94

Table 7: Rank-1 recognition rate (%) of ResNet-20 with different input on different poses

Loss Function	Input Data	 00	 01	 02	 03	 04	 05
Softmax	RGB image	74.07	78.63	73.39	67.21	64.49	**86.89**
	RGB plus LBP	**79.84**	**81.45**	**75.00**	**67.61**	**69.8**	84.62
	Input Data	 06	 07	 08	 09	 10	 11
	RGB image	95.97	96.76	95.97	93.55	90.65	86.23
	RGB plus LBP	**96.37**	**98.38**	**96.77**	**94.35**	**94.31**	**89.47**
	Input Data	 12	 13	 14	 15	 16	 17
	RGB image	81.85	**73.98**	70.97	85.94	92.37	91.16
	RGB plus LBP	**85.08**	72.36	**75.40**	**89.96**	**96.79**	**95.18**
	Input Data	 18	 19	Avg			
	RGB image	91.16	77.79	83.45			
	RGB plus LBP	**95.58**	**82.30**	**86.03**			

Table 8: Rank-1 recognition rate (%) of ResNet-20 with different input on different illuminations at pose angle = 60°

4.2.2 Findings

As shown in Table 6, the AM-Softmax-based ResNet-20 trained on the WebFace can achieve the best performance across all the pose angles on the MultiPIE data, especially for the pose angle equals to ±75°and ±90°. Fig.23 is drawn in order to know the effects of different pose angles on face recognition. It can be noticed from the figure that the performance of all the neural networks can gradually decrease as the pose angle becomes larger and the accuracy can significantly drop when the pose angle reaches ±75°.

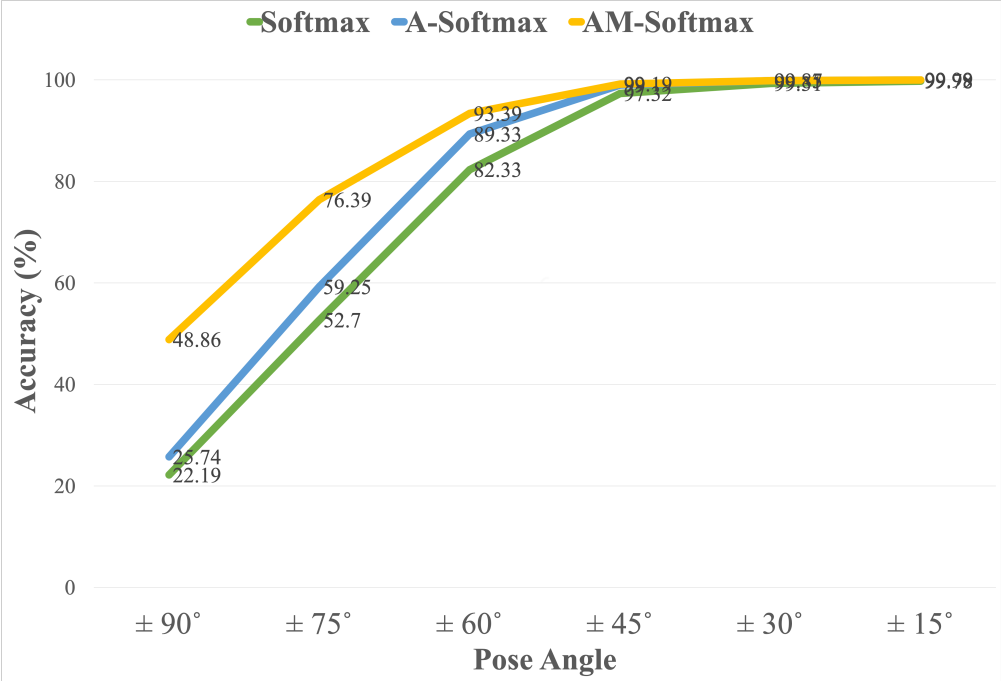


Figure 23: Rank-1 recognition rate (%) across different pose angles

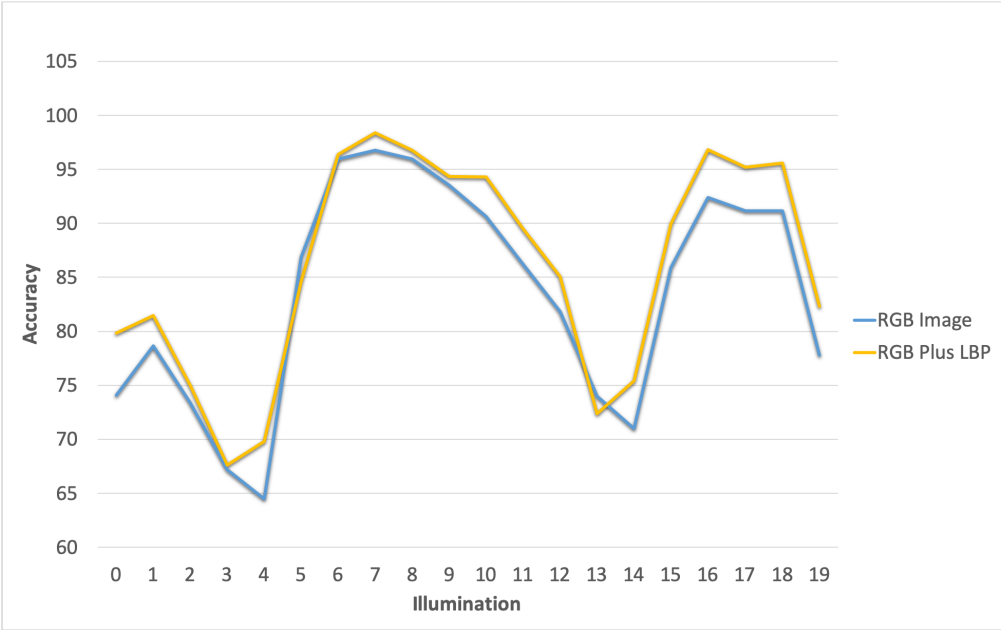


Figure 24: Rank-1 recognition rate (%) across different illuminations at angle = 60°

From Table 7, we can know that using the LBP features as the only one input channel cannot improve the deep face recognition no matter what loss functions are used for CNNs. However, the most important finding is that the performance of softmax-based neural network on pose-variant dataset can be improved by using LBP features as one additional channel, indicating that LBP features can help Softmax-based CNNs to achieve better performance when the face angle is large. According to Table 8, adding the LBP features can indeed help the CNNs to achieve better performance than using normal RGB channels as input when the lighting condition is poor. From Fig. 24, it can be clearly noticed that the overall performance of

adding the additional LBP features is better than only uing RGB channels as the input. More experiments would be performed to further verify this important finding.

However, things are totally different for both A-Softmax and AM-Softmax. From the Table 7, it can be noticed that adding the LBP feature cannot improve the performance of the ResNet-20 based on either A-Softmax loss or AM-Softmax loss. Although there are slight improvements for the A-Softmax-based CNNs for some face angles, i.e. $\pm 45°$and $\pm 30°$, when adding the extra LBP features channel, we cannot make sure if adding LBP features can indeed help the A-Softmax-based CNNs until more experiments can be done.

4.3 Evaluation on SurvFace

In order to evaluate the capability of low-resolution face recognition of deep face recognition, we performed face verification on the SurvFace dataset, which is one of the largest scale surveillance image datasets. To report the performance, the False Accept Rate (FAR) and True Accept Rate (TAR) are calculated. By varying the threshold, the ROC curve can be generated by TAR/FAR and the AUC overall model performance can be calculated.

The True Accept Rate and the False Accept Rate are given by:

$$\begin{aligned} \mathrm{FAR}(t) &= \frac{|\{s \geq t, \text{ where } s \in U\}|}{|U|} \\ \mathrm{FRR}(t) &= \frac{|\{s < t, \text{ where } s \in M\}|}{|M|} \\ \mathrm{TAR}(t) &= 1 - \mathrm{FRR}(t) \end{aligned} \tag{4.3}$$

where t is the threshold, U represents the set of unmatched face pairs and M represents the set of matched face pairs.

4.3.1 Experimental Results

We evaluated the performance of the ResNet with different numbers of convolutional layers as well as different loss functions, which are all trained on the WebFace, on the SurvFace. As shown in Fig.25, when trained on the WebFace, the ResNet-10 with Softmax, A-Softmax, and AM-Softmax can achieve 75.07%, 69.96%, and 63.85%, respectively. For the ResNet-20, the accuracy is 73.56%, 72.31%, and 63.95% for Softmax, A-Softmax, and AM-Softmax, respectively. When the number of convolutional layers becomes 64, Softmax, A-Softmax, and AM-Softmax can achieve 71.63%, 70.81%, and 66.61%, respectively.

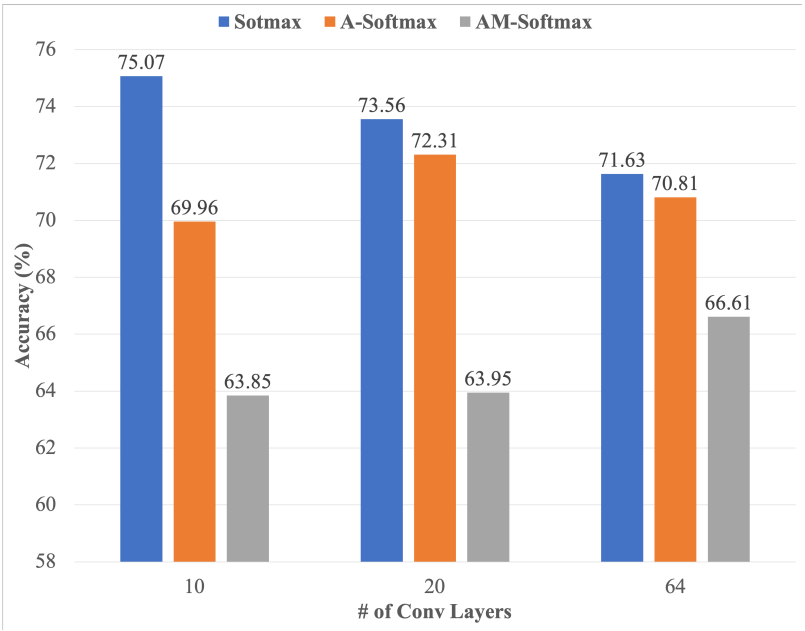


Figure 25: Accuracy (%) on SurvFace with different number of convolutional layers

4.3.2 Findings

From Fig.25, one interesting point can be noted is that the softmax consistently outperforms A-Softmax and AM-Softmax, when they are all trained on the WebFace. It can be noticed that A-Softmax is better than AM-Softmax on the SurvFace benchmark. The reason is that there is a huge domain difference between WebFace and SurvFace while WebFace is very similar to the LFW. A neural network performs well in one domain, then it would normally perform badly in another totally different domain. Therefore, A-Softmax and AM-Softmax can outperform Softmax on LFW but cannot perform better than Softmax on SurvFace.

In order to know their performance on the low-resolution face recognition, these loss functions should be trained on the low-resolution image dataset and tested on another dataset with low-resolution images under open-set protocol. Although these results in Fig.25 cannot reflect on the real performance of these different loss functions on the low-resolution face recognition, they can indeed indicate that using domain adaptation methods to solve the unmatched domain problems between training data and testing data would be the next important research direction.

5 Conclusion and Future Work

In this report, the deep convolutional neural networks (CNNs) with different configurations on open-set face recognition are discussed and their performances are fairly compared on many canonical public available datasets including LFW, MultiPIE, and SurvFace. Extensive experiments were performed to evaluate the effects of different loss functions, including the Softmax loss, A-Softmax loss, and AM-Softmax, as well as the number of convolutional layers on the deep face recognition. Evaluation results on the LFW show that increasing the number of convolutional layers of the CNNs can help to improve the overall performance of face recognition.

We also compared the EigenFace and LBP with deep CNNs on the LFW. We found that LBP can achieve slightly better performance than EigenFace while deep CNNs far surpasses both EigenFace and LBP. Although Softmax, A-Softmax, and AM-Softmax may achieve similar performance on the LFW, the AM-Softmax still demonstrates the best performance over the softmax and A-Softmax. All these results shown in this report are consistent with the state-of-the-art public official results. More importantly, as the evaluation results on the LFW suggest, a 99% accuracy can be achieved by deep face recognition on this benchmark, which means that the learning of deep CNNs has already become saturated on the LFW. However, these results may be too optimistic and the model may not have a good generalization capability. Therefore, these methods are also evaluated on the MultiPIE and SurvFace to see their performance in more complicated unconstrained conditions.

The evaluation results from the MultiPIE indicate that the AM-Softmax-based deep CNNs can achieve the best performance across different illuminations and pose angles. However, all these neural networks still cannot achieve satisfactory results when the pose angle becomes larger, especially when the angle is larger than $\pm 60°$. We also found out that adding the LBP features as one additional input channel of the deep CNNs can improve the face recognition when the illumination is bad and face angle is large.

We also explored the unmatched domain problems between training data and testing data when we evaluated the deep face recognition on the SurvFace. The results on SurvFace indicates that, by using the typical CNNs architecture and normal training process, the

deep CNNs are unable to perform face classification on testing data when there is a huge difference in resolution between the training data and testing data. Therefore, domain adaptation methods can be further investigated to solve this domain problem.

There are some possible future works can be done for this project. Firstly, more challenging benchmarks can be explored to evaluate the performance of the deep CNNs based on Softmax, A-Softmax, and AM-Softmax in unconstrained conditions. For example, the dataset with large expression and age variance can be used to evaluate the performance of these neural networks, then further performance improvement on these datasets may be possible.

Secondly, domain adaptation methods can be combined into the neural network to solve the unmatched domain problems between training data and testing data. Current work only focuses on the deep face recognition that is both trained and evaluated on the dataset shared with the same characteristics, i.e. same resolutions and same illuminations. However, advanced methods to make the deep CNNs adaptive to the testing data no matter what data they are trained on can be considered.

Thirdly, more experiments can be done to verify the performance improvement of deep face recognition in a poor lighting condition when using the LBP features as the additional input channel together with the normal RGB channels. Specifically, more datasets with a large variance in illumination conditions can be used. In addition, other local image descriptors, such as Gabor features, can be used as the input channel as well, and the performance improvement may be possible.

References

[1] M. Wang and W. Deng, "Deep face recognition: A survey," *CoRR*, vol. abs/1804.06655, 2018.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[3] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 2037–2041, Dec. 2006.

[4] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep face recognition: A survey," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, October 2018.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015.

[7] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *CVPR*, 2018.

[8] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," *CoRR*, vol. abs/1704.08063, 2017.

[9] "Video & image processing cup," Feb 2019.

[10] H. J. Aerts, E. R. Velazquez, R. T. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, *et al.*, "Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach," *Nature communications*, vol. 5, p. 4006, 2014.

[11] A. Ross and A. K. Jain, "Multimodal biometrics: An overview," in *2004 12th European Signal Processing Conference*, pp. 1221–1224, Sep. 2004.

[12] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, pp. 71–86, Jan. 1991.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, May 2017.

[14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, June 2014.

[15] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[16] Z. Cheng, X. Zhu, and S. Gong, "Surveillance face recognition challenge," *CoRR*, vol. abs/1804.09691, 2018.

[17] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," in *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pp. 1–8, Sep. 2008.

[18] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, IEEE, 2016.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[21] J. Shlens, "A tutorial on principal component analysis," *CoRR*, vol. abs/1404.1100, 2014.

[22] K. Fukunaga, *Introduction to statistical pattern recognition*. Elsevier, 2013.

[23] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, *et al.*, "Large scale distributed deep networks," in *Advances in neural information processing systems*, pp. 1223–1231, 2012.

[24] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

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

[26] H. G. Feichtinger and T. Strohmer, *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media, 2012.

[27] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*, pp. 709–720, Springer, 2010.

[28] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2439–2448, 2017.

[29] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," 2009.

[30] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016.

[31] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *CoRR*, vol. abs/1801.05599, 2018.

[32] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, 2013.

[33] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proceedings of The 33rd International Conference on Machine Learning* (M. F. Balcan and K. Q. Weinberger, eds.), vol. 48 of *Proceedings of Machine Learning Research*, (New York, New York, USA), pp. 507–516, PMLR, 20–22 Jun 2016.

[34] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L_2 hypersphere embedding for face verification," *CoRR*, vol. abs/1704.06369, 2017.

[35] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014.

[36] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," *CoRR*, vol. abs/1604.02878, 2016.

Appendix

There are six different metrics used to evaluate the deep neural network for tumor segmentation: Dice Score, Mean Surface Distance, 95% Hausdorff Distance, Slice-wise Missing Rate, False-Alarm Rate, and CT-scan-based Accuracy.

A. Dice Coefficient:

$$D = \frac{2|X| \cap |Y|}{|X| + |Y|} \tag{5.1}$$

where X is the ground truth region and Y is the predicted tumor region.

B. Mean Surface Distance and 95% Hausdorff Distance:

The mean surface distance is calculated by using the following two equations:

$$\begin{aligned} \vec{d}_{H,avg}(X,Y) &= \tfrac{1}{|X|} \textstyle\sum_{x \in X} \min_{y \in Y} \min d(x,y) \\ d_{H,avg}(X,Y) &= \tfrac{\vec{d}_{H,avg}(X,Y) + \vec{d}_{H,avg}(Y,X)}{2} \end{aligned} \tag{5.2}$$

The 95% Hausdorff Distance is calculated by using the following two equations:

$$\begin{aligned} \vec{d}_{H,r}(X,Y) &= K_r \left(\min_{y \in Y} d(x,y) \right) \forall x \in X \\ d_{H,r}(X,Y) &= \frac{\vec{d}_{H,r}(X,Y) + \vec{d}_{H,r}(Y,X)}{2} \end{aligned} \tag{5.3}$$

C. Slice-wise Missing Rate (MR)and False-Alarm Rate (FAR):

$$\text{MR} = \frac{\text{number of false negative masks in a CT scan}}{\text{number of masks with a tumor in the CT scan}} \tag{5.4}$$

$$\text{FAR} = \frac{\text{No. of false positive masks in a CT scan}}{\text{No. of ground truth without tumor in a CT scan}} \tag{5.5}$$

D. CT-scan-based Accuracy:

$$\text{Accuracy} = \frac{\text{number of patients with tumor detected}}{\text{number of patients with a tumor}} \tag{5.6}$$