Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Face Selection for Improving Set-to-Set Face Verification

Bachelor thesis

Andrii Yermakov

Study program: Software Engineering and Technology
Supervisor: Vojtech Franc, Ph.D.

This work was supported by the Czech Science Foundation
Project GAČR GA19-21198S

Prague, June 2020

Thesis Supervisor:
Vojtech Franc, Ph.D.
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo namesti 13
12135 Praha 2
Czech Republic

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Yermakov Andrii** Personal ID number: **478153**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Software Engineering and Technology**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Face Selection for Improving Set-to-Set Face Verification

Bachelor's thesis title in Czech:

Výběr tváří pro zlepšení přesnosti verifikace

Guidelines:

A set-to-set face verification aims at deciding whether two sets of faces capture the same identity or not. The accuracy of the face verification largely depends on quality of the analyzed faces. The goal of the project is to improve the verification accuracy by identifying and removing low-quality faces from the compared sets. The effect of the proposed face removal will be statistically evaluated on standard benchmarks and compared against common baseline methods.
Tasks:
1. Implement several baseline strategies to compute a face quality score.
2. Implement a method learning the face quality from failure examples of the face verification system.
3. Compare influence of the face removal on the verification accuracy using the baseline and the learned quality scores.

Bibliography / sources:

[1] Klare at al. Pushing the Frontiers of Unconstrained Face Detection and Recognition: {IARPA} Janus Benchmark A. In proc. of CVPR. 2015.
[2] Best-Rowden et al. Learning Face Image Quality from Human Assessments. IEEE Trans. on Information Forensics and Security. 2018.
[3] Abaza et al. Design and Evaluation of Photometric Image Quality Measures for Effective Face Recognition. IET Biometrics. 2014.

Name and workplace of bachelor's thesis supervisor:

Ing. Vojtěch Franc, Ph.D., Machine Learning, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **14.02.2020** Deadline for bachelor thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

Ing. Vojtěch Franc, Ph.D.	Head of department's signature	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature		Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Declaration

I hereby declare I have written this bachelor thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, June 2020

.....
Andrii Yermakov

Abstract

In this thesis we propose a statistical model which predicts performance of a pre-trained face verification system based on analysing quality of input images. A core part of the proposed model is a convolutional neural network, named CNN-FQ, which marks the input facial image as low-quality or high-quaity one. The concept of quality is not defined explicitly, but instead it is learned from mistakes the verification system makes when ranking triplets of faces. We applied the CNN-FQ in a set-based face verification to down-weight negative impact of low-quality faces when aggregating them to a template descriptor. It is shown on IJB-B 1:1 Face Verification benchmark that using CNN-FQ quality predictor for template aggregation leads to consistently higher recognition accuracy if compared to previously used face quality scores.

Keywords: Face verification, convolution neural networks, facial image quality prediction.

V této práci navrhujeme statistický model, který predikuje výkonnost natrénovaného systému pro verifikaci tváří na základě analýzy kvality vstupních obrázků. Základní část navrhovaného modelu je konvoluční neuronová síť s názvem CNN-FQ, která klasifikuje vstupní obrazky tváří na nízko kvalitní anebo vysoce kvalitní. Pojem kvality není definován explicitně, ale učí se z chyb, které verifikační systém dělá při vyhodnocení trojic obličejů. Naučenou CNN-FQ jsme použili pro verifikaci identit popsaných sadou obrázků, abychom snížili negativní dopad nízko kvalitních fotografií při jejich agregaci do deskriptoru šablony. Při 1:1 Verifikaci s použitím IJB-B protokolu se ukázalo, že použití predikce kvality z CNN-FQ při agregaci šablony vede k vyšší přesnosti rozpoznávání v porovnání s dříve používanými metodami odhadu kvality obrázků tváře.

Klíčová slova: Rozpoznávání tváří, konvoluční neuronové sítě, predikce kvality obrazku s obličejem

Acknowledgements

I would like to express my sincere appreciation to my supervisor Vojtěch Franc, who was always patient with me and motivated me to go on. He constantly provided me with useful knowledge and honest feedback. Without his help, work on this project would not have been possible.

Secondly, I also want to thank my mom who always encouraged me with a kind word and supported me in all endeavors. I am also grateful from the bottom of my heart to my soulmate Anna, who dispelled all my inner doubts and gave me the opportunity to express all my ideas.

I gratefully acknowledge that my work was supported by the Czech Science Foundation project GAČR GA19-21198S, CZ.02.1.01/0.0/0.0/16_019/0000765.

List of Abbreviations

CNN	Convolutional neural network
EM	Expectation–maximization
FAR	False Acceptance Rate
ROC	Receiver Operating Characteristic
SOTA	State-of-the-art
TAR	True Acceptance Rate

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Face verification belongs among the most fundamental face recognition tasks. Given two face images, the task is to decide whether both images captures the same or two different identities. During past few years research in face verification moved from using single images to setting when each identity is described by a set of images referred to as a template. This thesis is centered around set-based face verification.

Progress in face recognition has been always accelerated by challenging benchmarks like Labeled Faces in the Wild (LFW) [6] being one of the most distinguished examples. With the advent of deep learning the LFW benchmark and its predecessors have quickly become to easy. Nowadays, IARPA Janus Benchmark (IJB) [7, 4] is most frequently used in literature due to its large variation in poses, occlusions, arbitrary level of illumination and compression. We also use IJB-B protocol for experiments in this thesis.

In this thesis we propose a method for learning Convolution Neural Network based predictor of face image quality, which we termed CNN-FQ. Loosely speaking, the face image quality measures how informative the image is when used for face verification. The proposed learning algorithm does not require face examples explicitly annotated by image quality but instead it uses triplets of faces, two of them corresponding to the same identity and the third to a different one. Such triplets can be generated from existing databases containing faces annotated by identity.

We apply the learned face quality predictor in set-based face verification. Namely, the face image predictor is used to down-weight negative impact of low-quality images when aggregating face descriptors into a compact template descriptor.

The thesis is organized as following. Chapter 2 reviews most relevant methods for face verification as well as current state-of-the-art. Set-based face verification systems and protocols used for their evaluation are described in Chapter 3. In Chapter 4 we propose a method for learning CNN-FQ which predicts face image quality. Experiments are described in Chapter 5. Chapter 6 is dedicated to conclusions and future work.

Chapter 2

State-of-the-art

Early approach to set-based face verification was based on average similarity computed from similarities between all face pairs generated from two templates to be compared [13, 12]. The main disadvantage of this approach are time and memory demands associated with considering all pairs of faces.

Much more time and memory efficient approach is to represent image set by a compact fixed-length template descriptor. A common approach is to compute the template descriptor by aggregating descriptors extracted independently from each face of the template set by a deep convolution network. The simplest descriptor aggregation method boils down to averaging of all feature vectors [9, 8]. More advanced techniques use quality based fusion which takes into account quality scores when aggregating face descriptors. The quality scores correspond to informativeness of faces from which the scores were extracted. Existing strategies for quality based fusion involve weighted averaging, quality pooling [10] and low-quality face removal [1]. Multiple methods were proposed to extract the quality scores. For example, using L2-norm of face descriptor and confidence of a face detector as the quality score was proposed in [10]. Automatic selection of a subset of video frames based on their memorability was proposed in [5].

End-to-end approach to learn a compact template descriptor was proposed in [15]. Their method, termed GhostVLAD, is a neural network architecture that involves extraction of face descriptors from individual faces followed by layers for quality based fusion which output a compact template descriptor. Authors trained the network in an end-to-end manner, i.e. feature vectors extraction and aggregation is carried out jointly inside a single network. That means, they do not learn quality scores explicitly and then apply them to down-weight low quality images but instead let the network discover the optimal behaviour for template aggregation. Figure 2.1 shows schematic architecture of the network. This approach differs primarily from the common approach in template descriptor computation, see Figure 3.2, in that it is a single neural network. This architecture shows remarkable results in IJB-B benchmark and outperforms state-of-the-art approaches by a large margin.

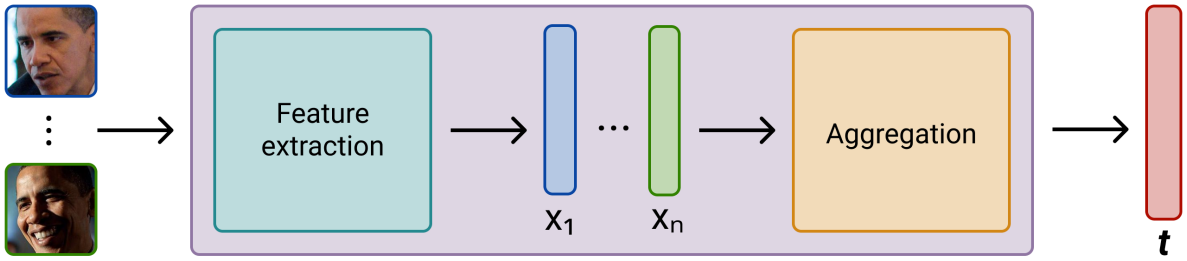


Figure 2.1: The violet box represents GhostVLAD network that takes a set of images, extracts feature vectors $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and aggregates them into a single vector $\boldsymbol{t}$.

The closest work to our approach is [1]. They propose to learn Support Vector Regression predictor of face image quality from annotated set of face images. Face labels are obtained by two different approaches. First, the labels are devised from human quality ranking of the face images. Second, the labels are computed from similarity scores obtained from a face verification system. Our approach is fundamentally different in that it does not require explicit annotation of the face image quality but, instead, the quality is a latent variable used to explain mistakes of a face verification system.

Chapter 3

Problem definition

3.1 IJB protocols

In this thesis, we use two different protocols defined in IJB-A [7] and IJB-B [4]. The first one, 1:1 Face Verification protocol, tests if verification system can distinguish between two sets of faces and verify whether or not they belong to the same person. The second one, 1:1 Covariate Verification protocol, is designed to test a face recognition algorithm's robustness on different covariates.

3.1.1 1:1 Face Verification

The face verification task emerges in an access control or in re-identification type of applications. The human subjects to be recognized are described by templates. A template is a set of facial images extracted from still photos and/or videos of a subject. Facial images forming a template are aggregated by a face verification algorithm into a compact (usually a vector) representation suitable for matching.

In 1:1 Face Verification protocol, the algorithm under evaluation is presented with pairs of templates and it has to decided which of them correspond to the same subject and which to different ones. The test pairs defined by the protocol are manually annotated as matching (capturing the same subject) or non-matching (capturing two different subjects). A typical face verification algorithm computes a real-valued similarity score between input templates and compares it with a decision threshold. When the similarity is above the threshold, the algorithm marks input pair as matching otherwise as non-matching. The algorithm is evaluated on the test set for different settings of the decision threshold. The algorithm's predictions are summarized by two evaluation metrics: false acceptance rate (probability that non-matching pair will be marked as matching) and true acceptance rate (probability that matching pair will be correctly recognized). A precise definition of the TAR and FAR metrics is given in Section 3.3. TAR and FAR values computed for different decision thresholds are presented in form of a Receiver Operation Characteristic (ROC) curve.

3.1.2 1:1 Covariate Verification

It is interesting to study how is performance of a face recognition system influenced by conditions, so called covariates, under which input facial images are captured. Similarly to 1:1 Verification protocol, in a 1:1 Covariate Verification protocol, the algorithm makes a

binary decision whether templates belong to the same subject or a different one. However, this time each template consists of a single facial image endowed with a set of manually annotated covariates. The annotated covariates include: forehead visible (yes = 1 / no = 0), nose/mouth visible (yes = 1 / no = 0), gender (male = 1 / female = 0), capture environment (indoor = 1 / outdoor = 0), facial hair (none = 0 / mustache = 1 / goatee = 2 / beard = 3), age group (0-19 = 1 / 20-34 = 2 / 35-49 = 3 / 50-64 = 4 / 65+ = 5), and skin tone (from Light Pink = 1 to Dark Brown = 6). There are also covariates labeled with GOTS pose-estimation algorithm [4]: roll (from $-53°$ to $61°$) and yaw (from $-87°$ to $78°$).

In this thesis we develop a neural network predicting a quality of facial images. We use this protocol to see how predicted qualities correlate with the verification accuracy within each covariate.

3.2 Building blocks of face verification system

In this section we outline building blocks of current face verification systems. We concentrate on the methods used to extract vector representation of templates because this is the part we attempt to improve in this thesis.

3.2.1 Face localization and pre-processing

At the beginning, all images should be pre-processed to have an appropriate dimension that matches input of neural networks used for recognition. In particular, the network architectures (SE-ResNet-50 and CNN-FQ) used in this thesis require 224x224px input images. The faces are localized by a face detector the search space of which is constrained by a rectangular area defined in IJB-X protocol. The bounding box found by the detector is extended by a certain scale factor the optimal setting of which is tuned on training examples (c.f. Section 5.2). Then, an image is cropped around the extended bounding box and resized to 256x256px. Finally, a central area of size 224x224px is is cropped and used as input to following neural network. An example of pre-processing steps is given in Figure 3.1.

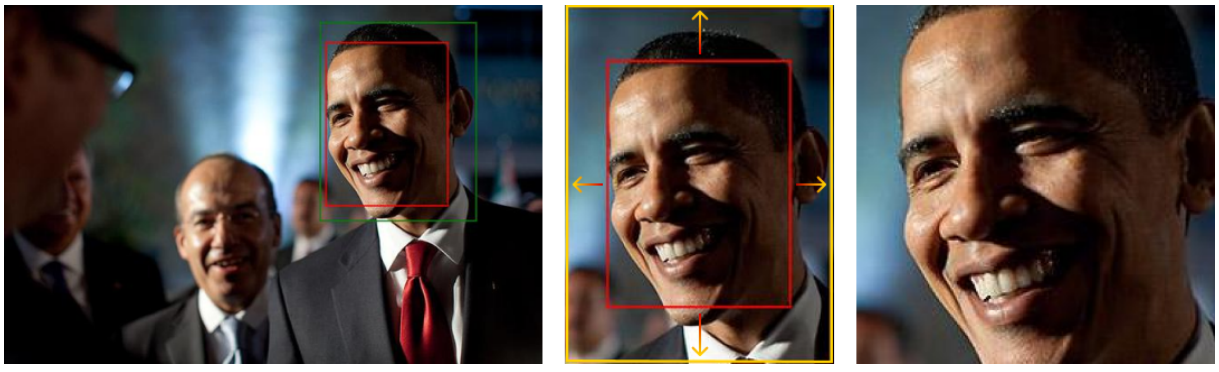


Figure 3.1: Left image is an input for a face detector, where green bounding box is the ground truth defined in IJB-A and red one is obtained with RetinaFace detector. The next step is a rescaling of the bounding box by a scale of 0.5. Right image is obtained by resizing to 256x256 px and central cropping of size of 244x244 px.

3.2.2 Decision strategy of a verification system

Having faces of a template localized and size normalized, the next step is to represent the faces by a single vector describing the template. Let $A = (a_1, \ldots, a_n)$ be a template represented by a tuple of n normalized faces, and let $\boldsymbol{t} = \mu(A) \in \mathbb{R}^d$ denote a template descriptor. Computation of the template descriptor involves extraction of feature vectors $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ from each face in A by a CNN and subsequent l2 normalization, after that the set of normalized feature vectors is aggregated into a single vector $\boldsymbol{t}$. Methods to aggregate the feature vectors are discussed in the next section. Decision about templates $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_m)$ is made based on a cosine similarity of their descriptors computed by

$$d(A, B) = \frac{\langle \mu(A), \mu(B) \rangle}{\|\mu(A)\| \, \|\mu(B)\|} \,.$$

The decision function $h(A, B; \Theta)$ predicts that the templates are matching, $h(A, B; \Theta) = 1$, or non-matching, $h(A, B; \Theta) = 0$, based on comparing the cosine similarity with a decision threshold Θ, that is,

$$h(A, B; \Theta) = \begin{cases} 1 & \text{if} \quad d(A, B) \geq \Theta \,, \\ 0 & \text{if} \quad d(A, B) < \Theta \,. \end{cases} \tag{3.1}$$

Computation of the template descriptor and the process of decision making are visualized in Figure 3.2 and Figure 3.3, respectively.

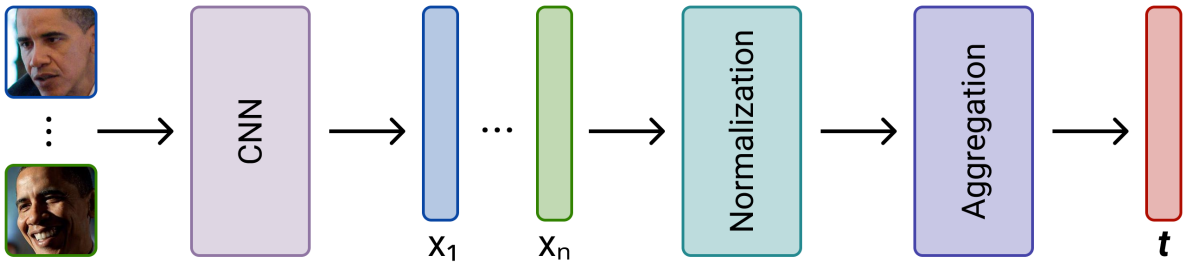


Figure 3.2: A template descriptor $\boldsymbol{t}$ is computed by aggregation of feature vectors $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ extracted with a CNN from each facial image of the template and normalized to length 1.

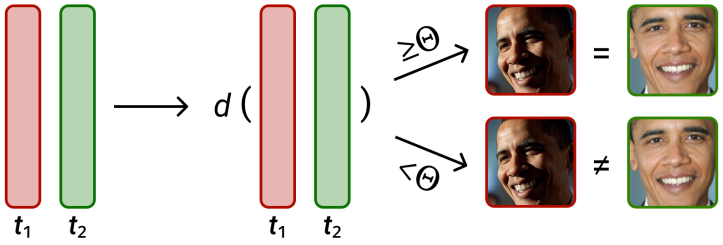


Figure 3.3: Verification system makes a decision if two template descriptors belong to the same person by comparing their cosine similarity with a decision threshold.

3.2.3 Template descriptor

Existing strategies obtain a template descriptor $\boldsymbol{t}$ by computing a weighted sum of the corresponding feature vectors $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, that is,

$$\boldsymbol{t} = \sum_{i=1}^{n} w_i \, \boldsymbol{x}_i \, . \tag{3.2}$$

Individual methods described below differ in the way how they define the weights $w = (w_1, \ldots, w_n)$. Most methods deduce the weights based on quality scores $q = (q_1, \ldots, q_n)$ extracted from individual faces of the template $A = (a_1, \ldots, a_n)$. The quality score can be, for example, confidence of the face detector used, L2-norm of the feature vector (i.e. $q_i = \|\boldsymbol{x}_i\|$ as proposed in [10]) or the quality score can be estimated by a CNN learned for this purpose as we propose in this thesis, in Chapter 4. We tested all methods described below on IJB-B 1:1 Face Verification protocol, results are described in Chapter 5

Averaging The simplest strategy is to calculate the template descriptor by averaging the features which corresponds to using (3.2) with uniform weights.

$$w_i = \frac{1}{n} \, .$$

Weighted Averaging In this case we use weighted arithmetic mean to force feature vectors to contribute differently based on quality scores. We also ensure that quality scores q are normalized to [0, 1] range. The final template descriptor is calculated using (3.2) with weights

$$w_i = \frac{q_i}{\sum_{j=1}^{n} q_j} \tag{3.3}$$

Quality Pooling This approach was proposed and applied in [10]. As quality score q the authors used confidence of a face detector, in particular, the Single Shot Detector [10]. The weights w_i in this case are computed in the following way

$$w_i = \frac{e^{\lambda l_i}}{\sum_{j=1}^{n} e^{\lambda l_i}} \tag{3.4}$$

$$l_i = \min(\frac{1}{2} \log \frac{q_i}{1 - q_i}, 7)$$

where λ is a hyperparameter, and l_i is the logit corresponding to the quality q_i. We tested this method with three different quality scores. While the scores returned by CNN-FQ and RetinaFace detector have ranges $q_i \in [0, 1]$, the L2-norm of the feature vector ($q_i = \|\boldsymbol{x}_i\|$) was normalized to $[0, 1]$ interval.

Low-quality face removal We found as interesting to try to improve performance by completely discarding low quality faces from the templates. To verify the idea, we compute the template descriptor from a subset of faces with quality score above a given

threshold τ as was proposed in [1]. Having τ, we can calculate weights that are later used in aggregation of the template descriptor (3.2) as follows

$$w_i = \begin{cases} \dfrac{1}{\sum_{j=1}^{n} [\![q_j \geq \tau]\!]} & \text{if } q_i \geq \tau \,, \\ 0 & \text{if } q_i < \tau \,. \end{cases} \tag{3.5}$$

To avoid discarding all faces in case when all quality scores are below the threshold τ, we set $w_{i^*} = 1, (w_i = 0, i \neq i^*)$, where $i^* = \arg\max_{i=[1\,..\,n]} q_i$ is the index of the highest quality face in a template. The optimal setting of the threshold τ is tuned on a training set. We generate a training set that consists of N matching and M non-matching template pairs as well as K thresholds τ_i, $i \in \{1, \ldots, K\}$. Finally, we chose τ for which the area under the ROC curve computed from the training set was maximal.

3.3 Evaluation metric

IJB-B protocol defines test pairs of labeled templates for evaluation of a verification system. Formally, the test set $\mathcal{T} = \{(A_1, B_1, y_1), \ldots, (A_l, B_l, y_l)\}$ contains pairs of templates (A_i, B_i) each assigned a label binary $y_i \in \{0, 1\}$ which equals 1 for matching and 0 for non-matching pairs. Let $\mathcal{I}_1 = \{i \mid y_i = 1\}$ denote set of indices of matching pairs, and $\mathcal{I}_0 = \{i \mid y_i = 0\}$ a set of non-matching pairs. The verification system is represented by a decision rule $h(A, B; \Theta)$ defined by (3.1).

Performance of a verification system is measured in terms of two metrics: True Acceptance Rate (TAR) and False Acceptance Rate (FAR). TAR (also known as sensitivity, recall or probability of detection) corresponds to the probability that the system correctly accepts an authorised person. TAR is estimated by computing a fraction of matching pairs whose similarity score correctly exceeds the threshold Θ, that is,

$$\mathrm{TAR}(\Theta) = \frac{1}{|\mathcal{I}_1|} \sum_{i \in \mathcal{I}_1} h(A_i, B_i; \Theta) \,. \tag{3.6}$$

FAR (also known as fall-out or the probability of false alarm) corresponds to the probability that the system incorrectly accepts a non-authorised person. FAR is estimated by computing a fraction of non-matching pairs whose similarity incorrectly exceeds the threshold Θ, that is,

$$\mathrm{FAR}(\Theta) = \frac{1}{|\mathcal{I}_0|} \sum_{i \in \mathcal{I}_0} h(A_i, B_i; \Theta) \,. \tag{3.7}$$

The values of $\mathrm{TAR}(\Theta)$ and $\mathrm{FAR}(\Theta)$ are evaluated for different settings of the decision threshold $\Theta \in \{\Theta_1, \ldots, \Theta_K\}$. The obtained values $\{(\mathrm{FAR}(\Theta_i), \mathrm{TAR}(\Theta_i) \mid i = \{1, \ldots, K\}\}$ are visualized as a curve in 2D, referred to as the Receiver Operating Characteristic (ROC) curve. In addition, TAR is reported for operating points at which FAR equals to $\{10^{-x} \mid x \in \{0, \ldots, 6\}\}$. Finally, the area under the ROC curve (AUC) is also reported as a single number characterizing performance of the system in the whole range of operating points.

IJB-A protocol provides 10 splits of the data into testing and training part. The training part is dedicated for tuning the verification algorithm and test part for evaluation. The process is repeated 10 times and reported are averages of the evaluation metrics

computed on the test parts. IJB-B protocol provides test data only, however, it us ensured that the test data do not overlap with CASIA Webface dataset [4] which we use for evaluation and training CNN-FQ respectively. At the same time, VggFace2 dataset which is used for training SE-ResNet-50 and IJB benchmarks are disjoint [9].

Chapter 4

Proposed method

Current datasets of facial images contain hundreds of thousands or even millions of photos. Thus, manual annotation of such datasets with good statistical confidence would require a huge amount of resources. The state-of-the-art approach presented in [15] employs artificial data degradation (blurring and compression) and learns how to down-weight influence of photos that have degraded qualities.

In our method, instead of performing data degradation to learn concepts of "good" and "bad" features of quality, the model will learn image quality from mistakes it makes when ranking triplets of faces. Hence our method can also exploit face datasets without any additional annotation of image quality.

Our goal is to exploit existing databases containing face images annotated only by identity of captured subjects. We want to avoid collecting additional annotation which would be costly. To this end, we use triplets of faces, where two of them belong to the same identity, and the third one belongs to a different one. If all faces in such a triplet are of good qualities then similarity between faces representing the same identity should be higher than similarities between faces representing different identities. If we find triples violating this condition we know that at least one of the faces does not carry enough information for recognition, i.e. it is of a low quality. The challenge is to pick the low quality faces from the erroneous triples which we solve by an instance of the Expectation-Maximization algorithm [11] described in the next section. The outcome of the proposed algorithm is a CNN predicting face quality, hence denoted as CNN-FQ. We use the learned CNN-FQ to extract quality scores for computation of the template descriptors as described in Section 3.2.3. One can envision other applications of the face quality predictor. For example, it might be used for building databases with high quality photos, however, this use case is not a topic of this thesis.

4.1 Model of triplet ranking errors

Let $(A, B, C) \in I^3$ be a triplet of facial images such that A and B captures the same identity while identity of C is different. Let $d\colon I \times I \to \mathbb{R}_+$ be a similarity score of a pre-trained face verification system. Then, the verification system ranks a triplet (A, B, C) correctly if $d(A, B) > \max(d(A, C), d(B, C))$, i.e. similarity between the same identities is higher than similarity between different ones. We introduce label $y \in \{0, 1\}$, defined as,

$$y = [\![d(A, B) > \max(d(A, C), d(B, C))]\!] , \tag{4.1}$$

which is 1 if the triplet (A, B, C) is ranked correctly and 0 for erroneous triplets. Here $[\![S]\!]$ denotes the Iverson bracket that evaluates to 1 if S is true and to 0 otherwise.

Assuming that triplets of faces (A, B, C) are generated from a random process, the corresponding labels defined by (4.1) are also realizations of random variables. We propose to model the distribution of label y by

$$p_\theta(y|A,B,C) = \sum_{(a,b,c)\in\{0,1\}^3} p_\theta(y|a,b,c)\, p_\theta(a|A)\, p_\theta(b|B)\, p_\theta(c|C) \tag{4.2}$$

where $(a, b, c) \in \{0, 1\}^3$ are latent variables each corresponding to one image in the triplet. The value $p_\theta(y = 1|A, B, C)$ is then the probability that triplet of faces (A, B, C) will be correctly ranked by the verification system. It will be shown later, that the state of the latent variables can be interpreted as indicator of the image quality. Hence we will call them face quality labels. By a clever initialisation of the learning algorithm described in the next section, we enforce the latent variables to be 1 for high quality faces and 0 for low quality faces. The function $p_\theta(y|a, b, c)$ describes a distribution of triplet label y conditioned on latent quality labels (a, b, c). The distribution of a latent label $x \in \{a, b, c\}$ conditioned on an image $X \in \{A, B, C\}$ is governed by distribution $p_\theta(x|X)$. We model $p_\theta(x|X)$ by the Logistic distribution defined on features extracted from face X by a CNN, which we will call CNN-FQ. That is,

$$p_\theta(x = 1|X) = \frac{1}{1 + \exp(-\langle \boldsymbol{\phi}(X), \boldsymbol{u}\rangle)} \quad \text{and} \quad p_\theta(x = 0|X) = 1 - p_\theta(x = 1|X)\,,$$

where $\boldsymbol{u}$ denotes weights and $\boldsymbol{\phi}(X)$ an output of the last and the penultimate layer of CNN-FQ, respectively. Let $\boldsymbol{\theta} \in \mathbb{R}^d$ denote a concatenation of all parameters of the distributions $p_\theta(y|a, b, c)$ and $p_\theta(x|X)$ that determine the model. In particular, the distribution $p_\theta(y|a, b, c)$ is given by $2^3 = 8$ real numbers and $p_\theta(x|X)$ by weights $\boldsymbol{u}$ and convolution filters of CNN-FQ defining $\boldsymbol{\phi}(X)$. Learning of the parameters $\boldsymbol{\theta}$ from data is discussed in the next section.

4.2 Learning model parameters by EM algorithm

Let $\mathcal{T} = \{(A_i, B_i, C_i, y_i) \in I^3 \times \{0, 1\} \mid i \in \{1, \ldots, n\}\}$ be a training set consisting of n triplets of faces and corresponding labels calculated by equation (4.1). We learn the model parameters $\boldsymbol{\theta}$ by maximizing the conditional log-likelihood of the training set $\mathcal{T}$ defined as

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_\theta(y_i|A_i, B_i, C_i) = \sum_{i=1}^{n} \log \left[\sum_{(a,b,c)\in\{0,1\}^3} p_\theta(y|a,b,c)\, p_\theta(a|A)\, p_\theta(b|B)\, p_\theta(c|C) \right]. \tag{4.3}$$

We use the Expectation-Maximization (EM) algorithm [2] which decomposes maximization of (4.3) into a sequence of simpler maximization problems. The EM algorithm replaces maximization of $L(\boldsymbol{\theta})$ by maximization of an auxiliary function

$$F(\boldsymbol{\theta}, q) = \sum_{i=1}^{n} \sum_{(a,b,c)\in\{0,1\}^3} q_i(a, b, c) \log \frac{p_\theta(y_i|A_i, B_i, C_i)}{q_i(a, b, c)} \tag{4.4}$$

where $q_i(a, b, c)$, $i \in \{1, \ldots, n\}$, are auxiliary variables defining distribution over the latent labels (a, b, c). This replacement is justified by the fact that $F(\boldsymbol{\theta}, q)$ is a tight lower bound of $L(\boldsymbol{\theta})$, that is, $L(\boldsymbol{\theta}) = \max_q F(\boldsymbol{\theta}, q), \forall \boldsymbol{\theta}$, and $L(\boldsymbol{\theta}) \geq F(\boldsymbol{\theta}, q)$, $\forall \boldsymbol{\theta}$, q.

The EM algorithm 1 maximizes $F(\boldsymbol{\theta}, q)$ by alternating maximization w.r.t. θ in M-step and w.r.t. q in E-step which breaks calculations to several simple maximization tasks. Namely, the E-step problem $\max_q F(\boldsymbol{\theta}^t, q)$ has a closed form solution

$$q_i^t(a,b,c) = \frac{p_{\theta^t}(y_i|a,b,c)\, p_{\theta^t}(a|A_i)\, p_{\theta^t}(b|B_i)\, p_{\theta^t}(c|C_i)}{\sum\limits_{(a,b,c)\in\{0,1\}^3} p_{\theta^t}(y_i|a,b,c) p_{\theta^t}(a|A_i) p_{\theta^t}(b|B_i) p_{\theta^t}(c|C_i)}\,, \quad i \in \{1,\ldots,n\}. \tag{4.5}$$

The M-step problem $\max_{\boldsymbol{\theta}} F(\boldsymbol{\theta}, q^{t-1})$ decomposes into two independent sub-problems. The first sub-problem involves maximization w.r.t. parameters defining $p_{\boldsymbol{\theta}}(y|a,b,c)$ which has also a closed for solution

$$p_\theta^t(y|a,b,c) = \frac{1}{\sum_{i=1}^n q_i^{t-1}(a,b,c)} \sum_{i=1}^n [\![y_i = y]\!] q_i^{t-1}(a,b,c)\,. \tag{4.6}$$

The second sub-problem requires maximization w.r.t. weights of CNN-FQ defining the distribution $p_{\boldsymbol{\theta}}(x|X)$ which boils down to maximization of

$$Q(\boldsymbol{\theta}) = \sum_{i=1}^n \Big(\sum_{a\in\{0,1\}} \alpha_i(a) \log p_\theta(a|A_i) + \sum_b \beta_i(b) \log p_\theta(b|B_i) + \sum_c \gamma_i(c) \log p_\theta(c|C_i) \Big) \tag{4.7}$$

where $\alpha_i(a) = \sum_{b,c} q_i^{t-1}(a,b,c)$, $\beta_i(b) = \sum_{a,c} q_i^{t-1}(a,b,c)$ and $\gamma_i(c) = \sum_{a,b} q_i^{t-1}(a,b,c)$ are constants computed from q^{t-1}. Note that maximization of (4.7) corresponds to learning CNN with the standard cross-entropy loss defined over soft-labels. Hence we solve this problem by Adam algorithm.

Algorithm 1 EM algorithm

```
init q^0
t ← 0
while converge do
    t ← t + 1
    θ^t ← arg max_θ F(θ, q^{t−1}) // M-step
    q^t ← arg max_q F(θ^t, q)    // E-step
end while
```

To initialize auxiliary distribution q^0 used by the EM algorithm, we use the following approach

$$q_i^0(a,b,c) = \begin{cases} \dfrac{a+b+c+\epsilon}{\sum\limits_{(a,b,c)\in\{0,1\}^3} (a+b+c+\epsilon)} & \text{if } y_i = 1 \\ \dfrac{3-a+b+c+\epsilon}{\sum\limits_{(a,b,c)\in\{0,1\}^3} (3-a+b+c+\epsilon)} & \text{if } y_i = 0 \end{cases} \tag{4.8}$$

where $\epsilon = 0.1$ prevents zero probabilities from which the EM cannot recover. Note that $q_i^0(a,b,c) = p_\theta(a,b,c \mid y_i, A, B, C)$, i.e., it is the probability that i-th triplet is ranked correctly if $y_i = 1$ or incorrectly if $y_i = 0$. Hence this initialization forces the model to associate the probability that a triplet is ranked correctly with the value of corresponding

latent label equal to 1. That is, higher the number latent labels equal to 1 the higher the probability of correctly ranked triplet. Because the EM algorithm is a local optimization method, the initially enforced semantics of the labels is usually preserved during the course of training. Thanks to this initialization one could interpret the latent variables as qualities of the corresponding facial images. We verify this hypothesis empirically in Chapter 5.

Chapter 5

Experiments

In our experiments we use SE-ResNet-50 trained on VGGFace2 as the base network for feature extraction. The same architecture is used for CNN-FQ trained on CASIA with sigmoid in the last layer.

In this chapter we are primarily interested in two different questions. At first, we want to compare four different methods for template descriptor aggregation described in Section 3.2.3 and learn what influence they have on discriminative abilities of template descriptors. As the baseline approach, we use Averaging strategy in which template descriptor is calculated by averaging feature vectors with uniform weights. Three others: Low-quality face removal, Weighted Averaging and Quality Pooling utilize quality scores from CNN-FQ, RetinaFace detector and L2-norms of feature vectors. Lastly, we are also interested in how the various covariates are reflected in the quality scores predicted with the proposed CNN-FQ.

In Section 5.1 we describe how we use datasets in experiments and show their summary. Section 5.2 explains how we tune the scale factor of bounding box and show results for comparison of two tested face detectors: MTCNN and RetinaFace. The following Section 5.3 shows a detailed learning process of CNN-FQ, which is precisely described in Section 4.2. Evaluation of different approaches in aggregation of template descriptors is presented in Section 5.4. In the same section we compare results with state-of-the-art method described in Chapter 2. The last Section 5.5 explores the correlation between covariates defined in IJB-B protocol, performance of SE-ResNet-50-256D and the distribution of quality scores predicted by CNN-FQ.

5.1 Datasets summary

All datasets used in our experiments provide photos of faces in unconstrained environments, that means that photos have large variations in quality (distorted / blurred), poses (different angles of head rotation), age (even the same person can have photos taken in different periods of his/her life), illumination (light/dark), ethnicity, environment (indoor/outdoor), facial hair (none/beard/mustache/goatee) and skin color. Summary of used datasets is shown in Table 5.1

We use these datasets for various purposes: training, testing and validation. We make sure that testing data never overlap with the training data. IJB-A and IJB-B are used to test accuracy of face verification employing different methods for aggregation of template descriptor with quality scores obtained with CNN-FQ. IJB-A training set is also used for optimizing of bounding box and hyperparameter τ used in LQFR method of template

Dataset	Subjects	Images	Img/Subj	Used for
IJB-A	500	5,712	11.4	evaluation; bounding box tuning
IJB-B	1,845	21,798	6.37	evaluation only
CASIA	10,575	494,414	46.75	training CNN-FQ
VGGFace2	9,131	3.31 M	362.6	training SE-ResNet-50-256D

Table 5.1: Summary of datasets used in this thesis. The last column summarizes usage of the datasets.

computation. CASIA is used for training and validation of CNN-FQ. VGGFace2 was used for pre-trained SE-ResNet-50-256D [1].

5.2 Face Detector Tuning

As the first step we need to pre-process data to obtain the right image size to fit into CNN's first layer. This stage involves running a face detector and consequent geometrical normalization of the found faces as described in Section 3.2.1. We tested two face detectors to find bounding boxes around faces of interest. First, MTCNN [14] is a face-detector widely used in scientific papers [2]. Second, RetinaFace [3] was shown to outperfor other state-of-the-art methods in the current most challenging benchmarks for face detection [3].

Bounding boxes found by the detectors are too tight to be readily used as input for face recognition, especially when face have large yaw rotation angle. We solve this by extending bounding with different scale factors and used the one which gives the best result in 1:1 Face Verification evaluated on a training set of IJB-A. The performance of face verification system for different scales of the bounding box is shown in Figure 5.1 and summarized in Table 5.2. It is seen that the used face detector has nonnegligible impact on the verification results. Based on the results obtained we use in all following experiments the scale factor 0.5 and the RetinaFace detector. In Figure 5.2 we show examples of original bounding returned by both detectors without scaling.

	MTCNN scale				RetinaFace scale			
T@F	0.4	0.5	0.6	0.7	0.4	0.5	0.6	0.7
1E-4	0.632	0.645	0.652	0.650	0.639	**0.668**	0.662	0.680
1E-3	0.786	0.801	0.805	0.802	0.810	**0.817**	0.817	0.813
1E-2	0.895	0.902	0.906	0.906	0.916	0.922	**0.923**	0.922
1E-1	0.958	0.961	0.963	0.962	0.974	0.977	**0.980**	0.980

Table 5.2: TAR@FAR for MTCNN and RetinaFace detectors for various scaling factors of bounding boxes on IJB-A training set.

[1] Pre-trained SE-ResNet-50-256D is avaliable on GitHub `https://github.com/ox-vgg/vgg_face2`

[2] MTCNN is freely available on GitHub `https://github.com/ipazc/mtcnn`.

[3] RetinaFace is available on GitHub `https://github.com/biubug6/Pytorch_Retinaface`.

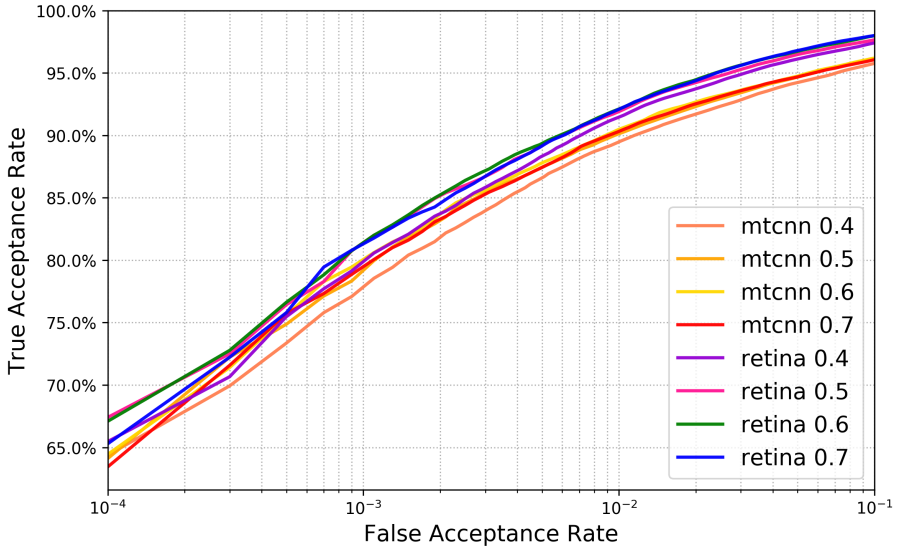


Figure 5.1: ROC curves for different bounding box scale factors of MTCNN and RefinaFace detectors using averaging method for template descriptor calculation, evaluated on IJB-A training set.

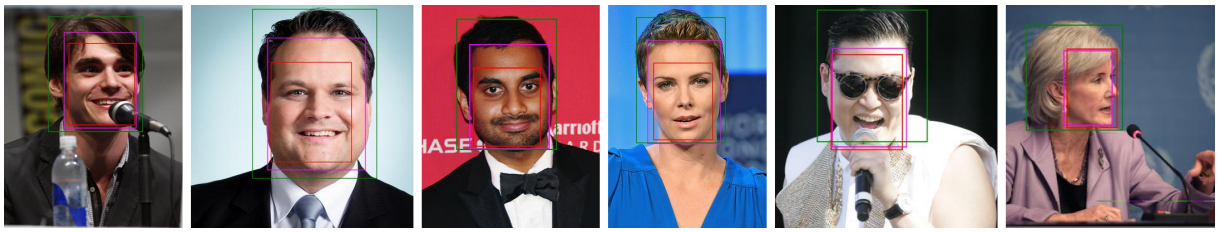


Figure 5.2: Exemplar bounding boxes for a sample of faces from the IJB-A dataset. Green box is the ground truth box determined by IJB-A protocol, magenta box is found by MTCNN and red color box by RetinaFace face detector, respectively.

5.3 CNN-FQ Training

We generate training set $\mathcal{T} = \{(A_i, B_i, C_i, y_i) \mid i \in \{1, \ldots, n\},\ y_i \in \{0, 1\}\}$ from CASIA WebFace database composed from $499k$ photos in unconstrained environments. For each identity we select 35 pairs (or less, depending on the number of images for that person) with the lowest and 35 pairs with the highest cosine similarity. Each pair is then augmented by one randomly sampled image of different identity. Obtained triplets of images are labeled as correct (label 1) or erroneous (label 0) using equation (4.1). We split triplets to training and validation set and ensure that identities in these two sets do not intersect. Finally, we have training set that consists of 19,153 erroneous triplets and 22,127 correct triplets, and validation set with 4,178 erroneous and 4,277 correct triplets. We also noticed that data augmentation applied during training, namely random horizontal flip, have a huge benefit for validation and training accuracy.

We train parameters $\boldsymbol{\theta}$ of model (4.2) by EM algorithm described in Section 4.2. In each EM iteration we ran 5 epochs of ADAM optimizer updating weights of CNN-FQ (M-step) and then recalculate distribution q with equation (4.5) (E-step). After each M-step, we compute use the neural network to predict the triplet ranking error for every triplet in the training set. The triplet is ranked correctly if the probability of y calculated

by (4.2) is greater than 0.5. Error is obtained by comparing classification results with ground truth labels. We also monitor log-likelihood $L(\boldsymbol{\theta})$ as well as the EM objective $F(\boldsymbol{\theta}, q)$ over epochs in order to verify that i) the likelihood monotonically grows and ii) $F(\boldsymbol{\theta}, q) \leq L(\boldsymbol{\theta})$, $\forall \boldsymbol{\theta}, q$. Development of the triplet classification error, log-likelihood and the EM objective is shown in Figure 5.3. One can notice that after epochs that are multiple of five when recalculation of q in E-step is made, we get a significant improvement in accuracy as well as a high leap in likelihood estimation.

Figure 5.4 shows sample images sorted in descending order by quality scores predicted with pre-trained CNN-FQ. It is seen that the images with a high quality score are mostly frontal and look towards the direction of the camera, have no blurring, are not overexposed or occluded. With the score going down the image quality starts to degrade.

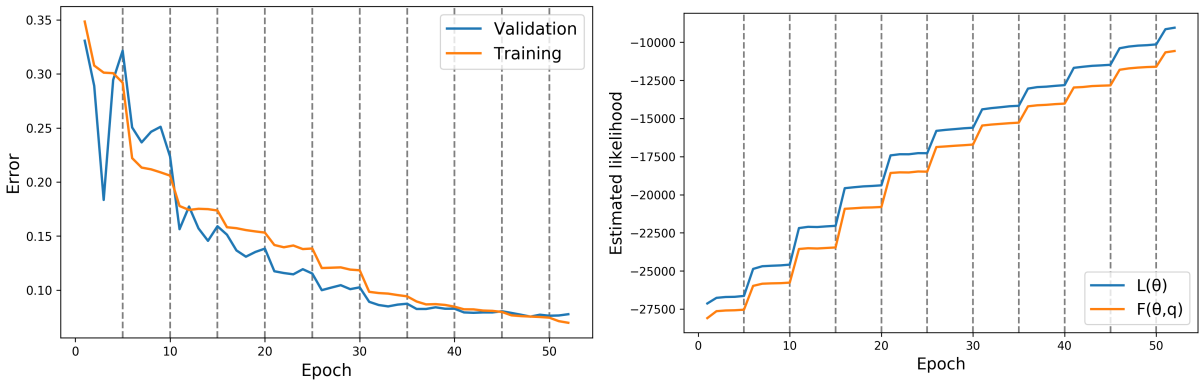


Figure 5.3: The left figure shows development of the triplet classification error computed from from predictions of the trained CNN-FQ on training and validation data. The right figure shows development of the log-likelihood and the EM objective function. The epoch refer to the epochs of the Adam solver updating the CNN-FQ. The vertical dashed lines represent times at which the E-step of the EM algorithm was executed.

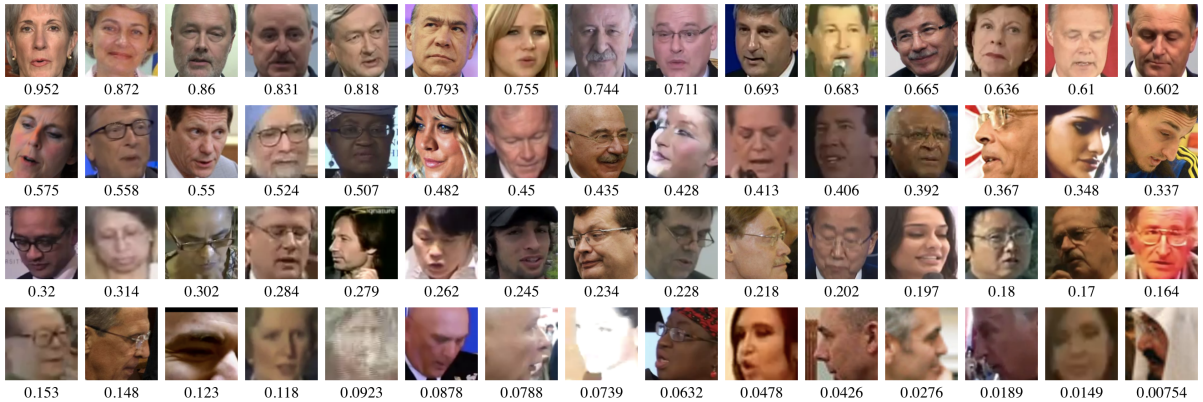


Figure 5.4: Sample images from IJB-A dataset sorted in descending order by quality scores predicted with CNN-FQ.

5.4 Evaluation of Template Descriptors

In this section we provide results for different approaches in template calculation described in Section 3.2.3.

5.4.1 Low-quality face removal approach

We had a theory that discarding low-quality images, i.e. images that have quality below pre-defined threshold could improve accuracy of face verification system. We call this method Low-quality face removal (LQFR) which is defined in Section 3.2.3. As quality scores we use L2-norm of feature vector, confidence of the RetinaFace detector and scores estimated by the CNN-FQ.

For experiments with LQFR, we used IJB-A protocol which provides both training and testing data. This method is looking for the threshold τ for each split, which maximizes the area under the ROC curve on a training set and then employ τ in 1:1 Face Verification protocol. Finding the best τ requires a computation of ROC curves for each threshold value, it can be a computationally expensive task for the large number of thresholds. Threshold τ also depends largely on a training set. The results showed that complete removal of photos from templates with quality scores below a pre-defined threshold does not lead to a big improvement in recognition accuracy, see Figure 5.5.

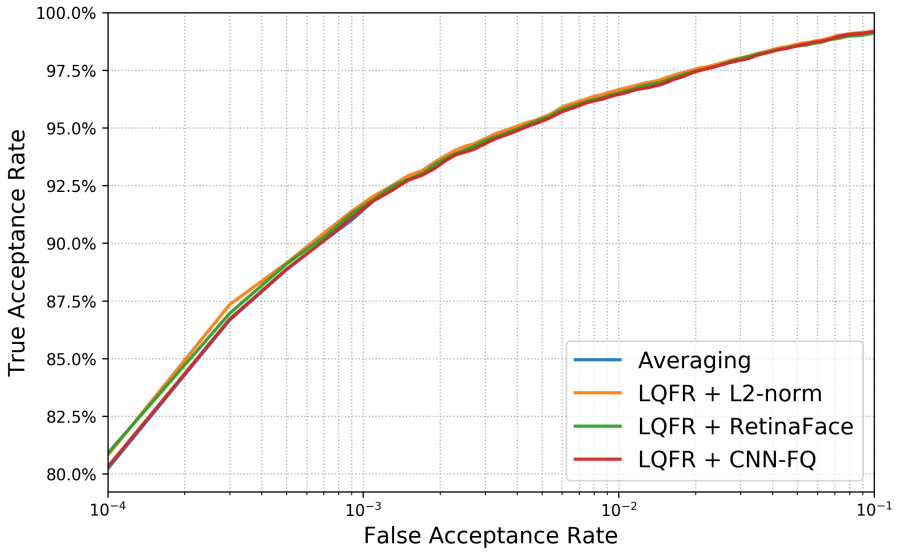


Figure 5.5: Results on IJB-A 1:1 Face Verification protocol. ROC curves of different quality scores where template descriptors are calculated with LQFR method. Averaging represents the ROC curve in which template descriptors are calculated with uniform weights, see 3.2.3.

We were interested in why removal of photos with low quality, that is, below pre-defined threshold, does not necessarily lead to an improvement in face recognition accuracy. We conducted the experiment in which we remove faces one by one from the template descriptor calculation, minimizing the cosine distance between two templates which represent the same identity. Cosine distance is calculated as one minus the cosine similarity, $1-d(A,B)$. The template descriptor is calculated with Averaging method. Finally, we are left with the set of photos of size n in which removal of any does not lead to a lower cosine distance. Secondly, we remove the same number of photos by CNN-FQ scores and leave the template with n photos having the highest qualities. Figure 5.6 reveals the results of removing photos with two different approaches. We also show how the cosine distance changes during the gradual deletion of photos using these two methods.

The experiment showed that the complete rejection of low-quality photos, in this case those that have a large yaw angle, does not lead to a decrease in the cosine distance and,

therefore, to better face recognition accuracy. Thus, that implies that the LQFR method is not a good candidate for a strategy of template aggregation.

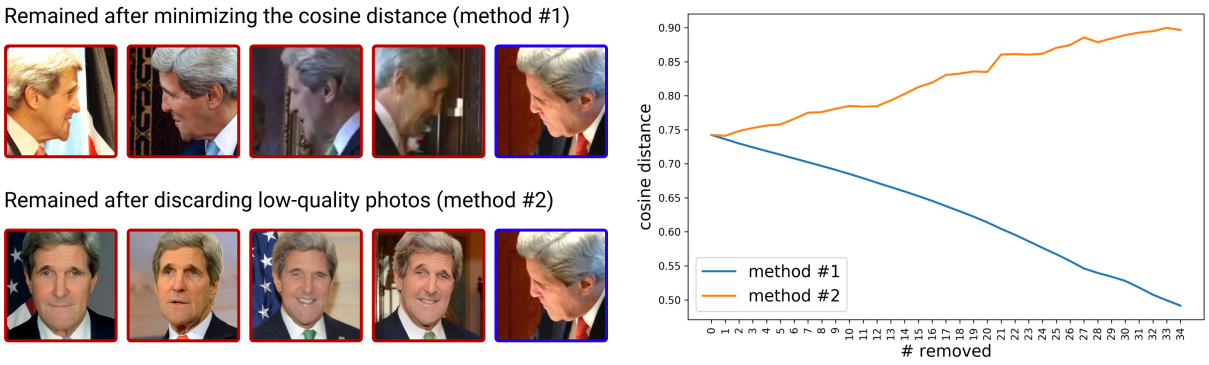
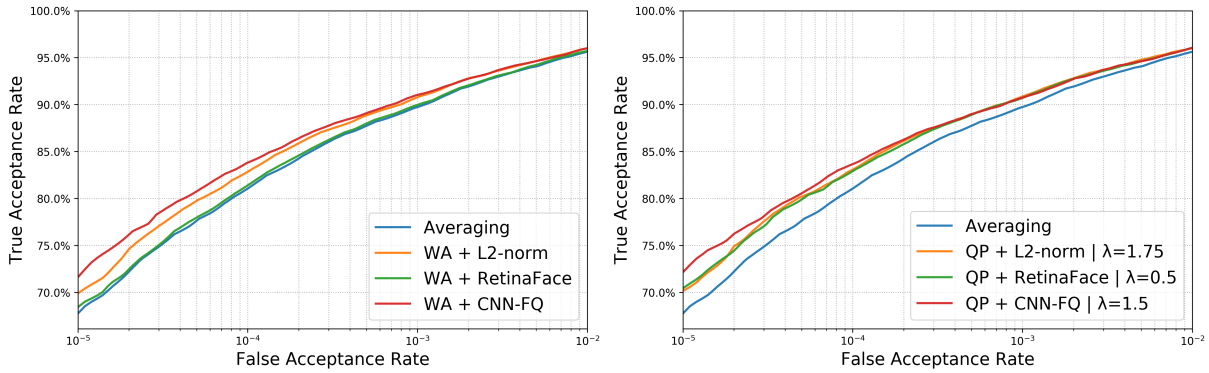


Figure 5.6: At the left there are photos that are remained after applying two methods: minimizing of cosine distance (method 1) and discarding low-quality photos (method 2). Red and blue bounding boxes signify different templates. At the right we plot the change of cosine distance when removing photos with these two methods.

5.4.2 Weighted Averaging and Quality Pooling

The following experiments use IJB-B 1:1 Face Verification protocol. Here we try two approaches for template computation: Weighted Averaging and Quality Pooling defined in Section 3.2.3 and compare different quality metrics: L2-norm, CNN-FQ and RetinaFace scores. As the baseline, we compute template descriptors using Averaging method, i.e. with uniform weights.

Using quality scores extracted with CNN-FQ, we can archive an additional 5% of accuracy in TAR@FAR $= 10^{-5}$ in comparison to Averaging method for both approaches. Quality Pooling utilizes a hyperparameter λ, the influence of different values λ for each quality metric are shown in Table 5.3. Results show that Quality Pooling can increase the area under the ROC for L2-norm and RetinaFace scores but the method does not show any noticeable influence using CNN-FQ scores in comparison to Weighted Averaging method.



(a) Weighted Averaging (b) Quality pooling

Figure 5.7: ROC curves with different quality scores for Weighted Averaging (a) and Quality Pooling (b) in 1:1 Face Verification on IJB-B. Averaging represents the ROC curve in which template descriptors are calculated with uniform weights, see 3.2.3.

Quality scores	λ	TAR@FAR 1E-5	1E-4	1E-3	1E-2	1E-1
L2-norm	0.25	0.686	0.822	0.902	0.958	0.986
	0.5	0.692	0.827	0.904	0.959	0.987
	0.75	0.695	0.830	0.907	0.959	**0.987**
	1	0.698	0.832	0.909	0.960	0.987
	1.25	0.700	0.833	0.908	0.960	0.987
	1.5	0.701	0.835	0.909	0.960	0.986
	1.75	**0.701**	**0.835**	**0.909**	**0.960**	0.986
RetinaFace	0.25	0.697	0.829	0.905	0.959	**0.987**
	0.5	**0.704**	0.833	**0.907**	**0.960**	0.986
	0.75	0.704	**0.834**	0.907	0.959	0.986
	1	0.701	0.833	0.906	0.959	0.986
	1.25	0.694	0.830	0.904	0.958	0.985
	1.5	0.685	0.824	0.902	0.957	0.985
	1.75	0.679	0.817	0.898	0.956	0.984
CNN-FQ	0.25	0.690	0.825	0.902	0.958	0.986
	0.5	0.698	0.831	0.904	0.960	**0.986**
	0.75	0.707	0.833	0.907	0.960	0.986
	1	0.713	0.837	0.908	0.960	0.986
	1.25	0.716	0.839	**0.909**	0.960	0.986
	1.5	0.721	0.840	0.908	**0.961**	0.985
	1.75	**0.724**	**0.840**	0.907	0.960	0.985

Table 5.3: Effect of Quality Pooling in 1:1 Face Verification on IJB-B dataset for different quality scores.

Aggregation method	Quality scores	TAR@FAR 1E-5	1E-4	1E-3	1E-2
Averaging	Uniform weights	0.677	0.816	0.898	0.956
Weighted Averaging	CNN-FQ	**0.716**	**0.841**	**0.910**	0.960
	RetinaFace	0.684	0.819	0.900	0.957
	L2-norm	0.697	0.832	0.907	**0.960**
Quality Pooling	CNN-FQ $\lambda = 1.5$	**0.721**	**0.840**	0.908	**0.961**
	RetinaFace $\lambda = 0.5$	0.704	0.833	0.907	0.960
	L2-norm $\lambda = 1.5$	0.701	0.835	**0.909**	0.960
GhostVLAD		**0.762**	**0.863**	**0.926**	**0.963**

Table 5.4: Accuracy comparison for different methods of template aggregation described in Section 3.2.3 as well as state-of-the-art GhostVLAD method.

5.5 Impact of covariates on quality

In this section we use IJB-B 1:1 Covariate Verification protocol to investigate correlation between individual covariates, performance of the verification system and the face quality predictor score extracted by the proposed CNN-FQ. The protocol defines pairs of matching and non-matching faces (that is, templates of unit size). Each test pair is assigned manual annotation of covariates which we use to cluster the test pairs into groups with similar properties like e.g. faces with a low resolution, large yaw angle and so on. For a group of faces with particular value of a covariate we plot the ROC curve of the verification system and a histogram of quality scores extracted by the CNN-FQ. The verification system uses the cosine similarity computed from feature vectors extracted from the pairs of faces by SE-ResNet-50-256D. The ROC curve tells us how covariate influences the performance of the verification system. The histogram of quality scores reveals the correlation between the score and the covariate.

Influence of forehead visibility

Visibility of forehead has a great influence in face recognition as we see in Figure 5.8. It is also seen that CNN-FQ quality score is mildly correlated with the forehead visibility, namely, faces with visible forehead tend to have slightly higher quality scores.

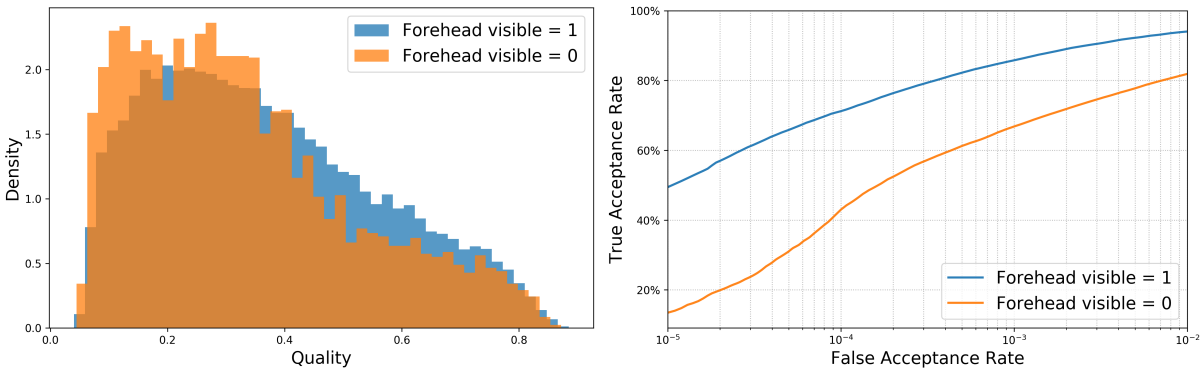


Figure 5.8: Histograms show dependency of extracted qualities with CNN-FQ on forehead visibility covariate (1 = visible / 0 = not visible). ROC curves show the performance of verification for the same covariate conducted with IJB-B 1:1 Covariate Verification protocol.

Influence of nose/mouth visibility

Visibility of nose/mouth has a great influence in face recognition as we see in Figure 5.9. CNN-FQ quality score is also mildly correlated with nose/mouth visibility.

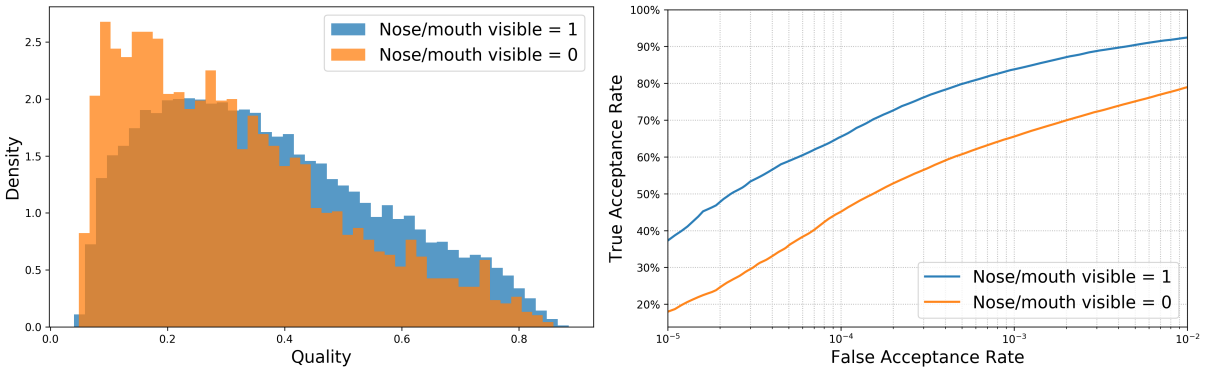


Figure 5.9: Histograms show dependency of extracted qualities with CNN-FQ on nose/mouth visible covariate (1 = visible / 0 = not visible). ROC curves show the performance of verification for the same covariate conducted with IJB-B 1:1 Covariate Verification protocol.

Influence of roll angle

The influence of the roll angle is shown in Figure 5.10. SE-ResNet-50 seems to be robust to changes in roll angle. Our experiments showed that the roll angle within $[0°, 30°]$ range does not have a noticeable influence on recognition performance. We further split the test data into two groups according to the roll angle, namely, faces with roll angle from 0° to 15° and from 15° to 65° so that each group has enough faces to conduct fair comparison. It is seen that angles larger than 15° can indeed have a negative influence on the performance. CNN-FQ quality score is clearly correlated with the roll angle.

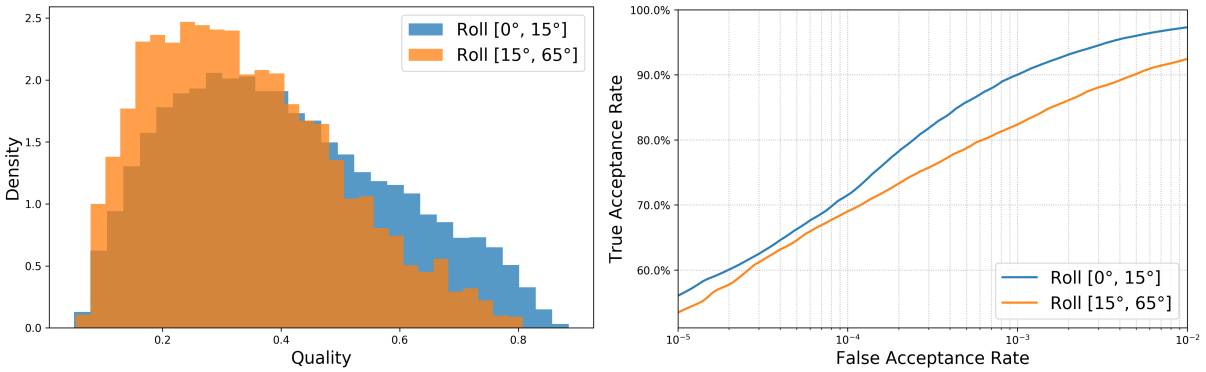


Figure 5.10: Histograms show dependency of extracted qualities with CNN-FQ on roll angle divided into two classes: [0°, 15°] and [15°, 65°]. ROC curves show the performance of verification for the same covariate conducted with IJB-B 1:1 Covariate Verification protocol.

Influence of yaw angle

Results are summarized in Figure 5.11. We separated examples according to the yaw angle into four different groups: $[0°, 15°]$, $[15°, 30°]$, $[30°, 45°]$ and $[45°, 90°]$ and restricted each group to have approximately similar number of faces to conduct fair comparison. It is seen that increase in the yaw angle is associated with decrease of both the face recognition performance and the value of CNN-FQ quality score. Out of other other covariates, the yaw angle seems to have the highest correlation with the CNN-FQ quality score.

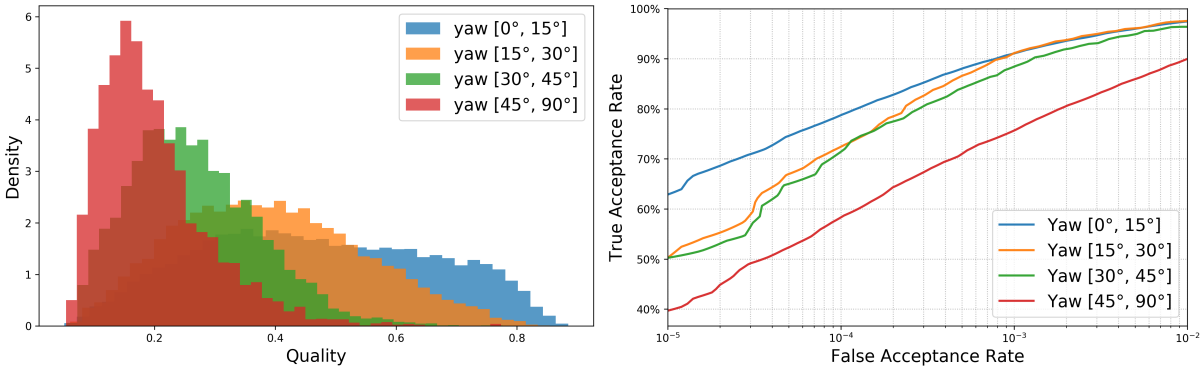


Figure 5.11: Histograms show dependency of extracted quality scores with CNN-FQ on faces split according to the yaw angle into four groups: [0°, 15°], [15°, 30°], [30°, 45°] and [45°, 90°]. ROC curves show the performance of verification for the same covariate conducted with IJB-B 1:1 Covariate Verification protocol.

Influence of face size

Influence of the input face size on the recognition performance and CNN-FQ quality score is shown in Figure 5.12. The face size is measured by the area of bounding box obtained from the RetinaFace detector. The examples are divided according to face size into four groups: less than 3k px, from 3k to 6k px, from 6k to 40k px and greater than 40k px. Experiment revealed that face size has enormous influence on recognition accuracy, and that CNN-FQ quality score is strongly correlated with this covariate.

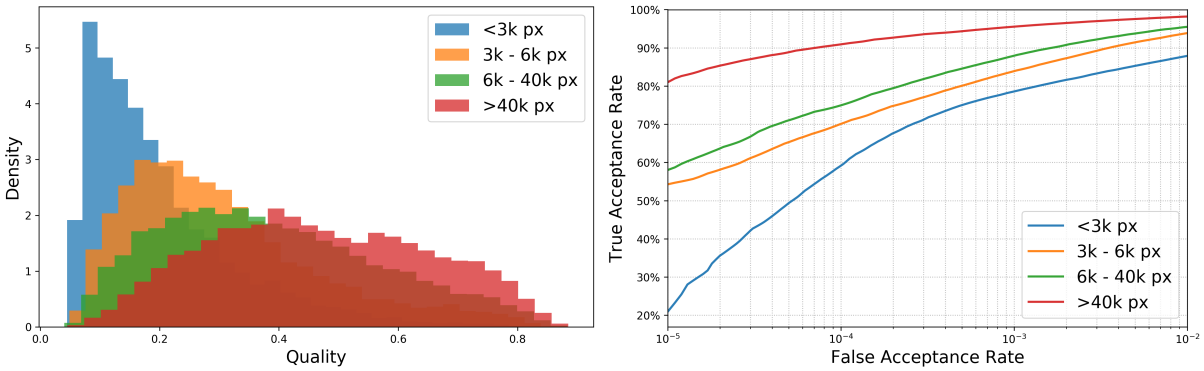


Figure 5.12: Histograms show dependency of extracted qualities with CNN-FQ on size of faces determined by RetinaFace detector bounding box. ROC curves show the performance of verification for the same covariate.

Influence of facial hair

Figure 5.13 shows influence of the facial hair on the recognition performance and the CNN-FQ quality score. Faces are split according the facial hair into four classes (none = 0 / mustache = 1 / goatee = 2 / beard = 3). The results show that people who have goatee are recognizable much better. We also see a peak in the quality score histogram for the goatee class but it seems to be shifted to the left which means that CNN-FQ recognizes this feature as a defect in quality. A similar, though not so pronounced, pattern is seen for mustaches. The reason for this discrepancy seems to be a low amount of training samples with people who have mustache or goatee in CASIA WebFace on which the CNN-FQ was trained. Otherwise, we do not see that facial hair has any significant influence on CNN-FQ quality scores.

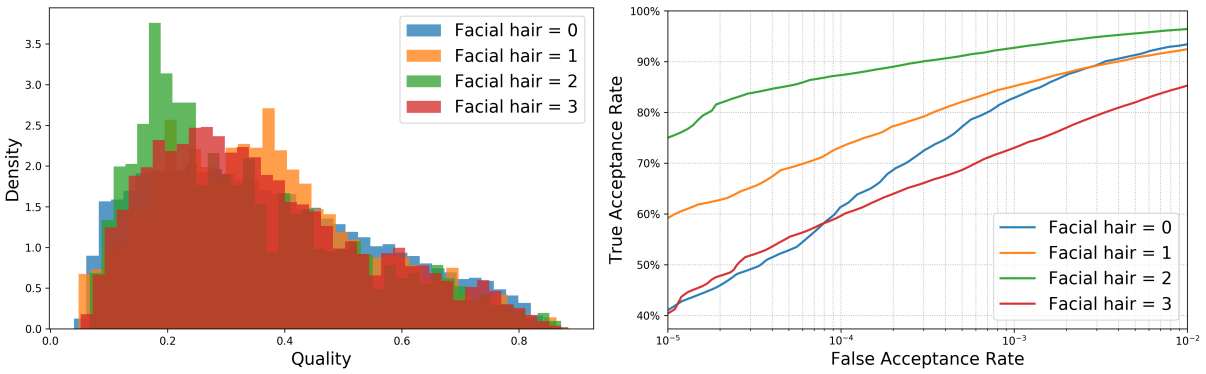


Figure 5.13: Histograms show dependency of extracted qualities with CNN-FQ on facial hair covariate with four classes (none = 0 / mustache = 1 / goatee = 2 / beard = 3). ROC curves show the performance of verification for the same covariate conducted with IJB-B 1:1 Covariate Verification protocol.

Chapter 6

Conclusions

We proposed method for learning a neural network termed CNN-FQ that predicts face image quality. Learning of CNN-FQ does not require face examples annotated by image quality. The concept of quality is learned from mistakes a pre-trained verification system makes when ranking triplets of faces. The training triplets can be constructed from faces labeled by identity databases of which are abundant. We have shown that quality scores predicted by the proposed CNN-FQ can be used as weights in quality based aggregation of face image set to a compact template descriptor. Such template descriptors appear to provide better accuracy in set-based 1:1 Face Verification evaluated on IJB-B protocol if compared to descriptors that use previously proposed quality scores, namely, L2-norms of feature vectors and a face detector confidence.

We experimented with different methods for quality based vector aggregation, namely, Weighted Averaging, Quality Pooling and Low-quality face removal (LQFR). Experiments showed that LQFR is not a good strategy to compute template descriptors, especially when dealing with templates that consist of bad quality photos. On the other hand both, Weighted Averaging and Quality Pooling show a noticeable increase in face recognition accuracy if compared to a Weighted Averaging.

Experiments on IJB-B covariates showed that quality scores predicted by CNN-FQ are mainly correlated with yaw and roll angle of the face, as well as the face size all being good indicators of face image quality. Less pronounced was correlation with forehead and nose/mouth visibility both having impact on recognition accuracy.

We believe that there is a place for a much better approach in quality based aggregation. For example, it may be beneficial to learn to re-scale the quality scores to increase discriminability of the template descriptors.

Bibliography

[1] Lacey Best-Rowden and Anil K. Jain. Learning face image quality from human assessments. *IEEE Transactions on Information Forensics and Security*, 13:3064–3077, 2018.

[2] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[3] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *CoRR*, abs/1905.00641, 2019.

[4] C. Whitelam et al. Iarpa janus benchmark-b face dataset. *IEEE*, 2017.

[5] G. Goswami, R. Bhardwaj, R. Singh, and M. Vatsa. Mdlface: Memorability augmented deep learning for video face recognition. In *IEEE International Joint Conference on Biometrics*, pages 1–7, 2014.

[6] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[7] Klein B. Taborsky E. Blanton A. Cheney J. Allen K. Grother P. Mah A. Jain A.K. Klare, B.F. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. *IEEE*, Jun 2015.

[8] O.M. Parkhi, A. Vedaldi, and Z. Zisserman. Deep face recognition. In *BMVC*, 2015.

[9] Weidi Xie Omkar M Parkhi Andrew Zisserman. Qiong Cao, Li Shen. Vggface2: A dataset for recognising faces across pose and age. *arXiv:1710.08092*, May 2018.

[10] Rajeev Ranjan, Ankan Bansal, Hongyu Xu, Swami Sankaranarayanan, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa. Crystal loss and quality pooling for unconstrained face verification and recognition. *CoRR*, abs/1804.01159, 2018.

[11] M.I. Schlesinger. A connection between learning and self-learning in the pattern recognition (in Russian). *Kibernetika*, 2:81–88, 1968.

[12] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embeding for face recognition and clustering. In *CVPR*, 2015.

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deep-face: Closing the gap to humun-level performance in face verification. In *CVPR*, 2014.

[14] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878, 2016.

[15] Arandjelović R. Zisserman A. Zhong, Y. Ghostvlad for set-based face recognition. *arXiv:1810.09951*, Oct 2018.