Development of face analysis approximation methods by machine learning

Binod Bhattarai

Normandie Université

THESE

Pour obtenir le diplôme de doctorat

Informatique et applications

Préparée au sein de l'ENSICAEN et de l'UNICAEN

Développement de méthodes de rapprochement physionomique par apprentissage machine

Présentée et soutenue par
Binod BHATTARAI

Date de soutenance: 2016/12/16

Thèse soutenue publiquement le (date de soutenance) devant le jury composé de		
Mme. Elisa FROMONT	Maitre de conférences HDR, Université St Etienne	Rapporteur
M. Jakob VERBEEK	Charge de Recherche INRIA HDR, INRIA Grenoble	Rapporteur
M. Laurent HEUTTE	Professur des Universités, Université de Rouen	Examinateur
M. Philippe-Henri GOSSELIN	Professur des Universités, ENSEA CERGY-PONTOISE	Examinateur
M. Alexis LECHERVY	Maitre de conférences, Université de Caen	Examinateur
M. Frédéric JURIE	Professur des Universités, Université de Caen	Directeur de thèse

Thèse dirigée par Frédéric Jurie, laboratoire GREYC

UNICAEN ENSICAEN ÉCOLE NATIONALE SUPÉRIEURE D'INGÉNIEURS DE CAEN & CENTRE DE RECHERCHE ED 508 GREYC

Abstract

The work presented in this PhD thesis takes place in the general context of face matching. More precisely, our goal is to design and develop novel algorithms to learn compact, discriminative, domain invariant or de-identifying representations of faces.

Searching and indexing faces open the door to many interesting applications. However, this is made day after day more challenging due to the rapid growth of the volume of faces to analyse. Representing faces by compact and discriminative features is consequently essential to deal with such very large datasets. Moreover, this volume is increasing without any apparent limits; this is why it is also relevant to propose solutions to organise faces in meaningful ways, in order to reduce the search space and improve efficiency of the retrieval.

Although the volume of faces available on the internet is increasing, it is still difficult to find annotated examples to train models for each possible use cases e.g. for different races, sexes, *etc.* for every specific task. Learning a model with training examples from a group of people can fail to predict well in another group due to the uneven rate of changes of biometric dimensions e.g., ageing, among them. Similarly, a model learned from a type of feature can fail to make good predictions when tested with another type of feature. It would be ideal to have models producing face representations that would be invariant to these discrepancies. Learning common representations ultimately helps to reduce the domain specific parameters and, more importantly, allows to use training examples from domains well represented to other domains. Hence, there is a need for designing algorithms to map the features from different domains to a common subspace – bringing faces bearing same properties closer.

On the other hand, as automatic face matching tools are getting smarter and smarter, there is an increasing threat on privacy. The popularity in photo sharing on the social networks has exacerbated this risk. In such a context, altering the representations of faces so that the faces cannot be identified by automatic face matchers – while the faces look as similar as before – has become an interesting perspective toward privacy protection. It allows users to limit the risk of sharing their photos in social networks.

In all these scenarios, we explored how the use of Metric Learning methods as well as those of Deep Learning can help us to learn compact and discriminative representations of faces. We build on these tools, proposing compact, discriminative, domain invariant representations and de-identifying representations of faces.

We applied the proposed methods on a wide range of facial analysing applications. These applications include: large-scale face retrieval, age estimation, attribute predictions and identity de-identification. We have evaluated our algorithms on standard and challenging public datasets such as: LFW, CelebA, MORPH II *etc.* Moreover, we appended 1M faces

crawled from Flicker.com to LFW and generated a novel and more challenging dataset to evaluate our algorithms in large-scale. Our experiments show that the proposed methods are more accurate and more efficient than compared competitive baselines and existing state-of-art methods, and attain new state-of-art performance.

Keywords

Facial Analysis • Metric Learning • Deep Learning • Joint Learning • Multi-task Learning

Contents

Chapter 1

Introduction

Contents

1.1 Objectives and Motivation

Broadly speaking, the objective of this thesis is to design and develop algorithms for learning representations of face images for various facial analysis tasks. Such tasks require compact, discriminative, and domains invariant representations : most of the applications we are dealing with in this thesis are indeed large scale applications (up to $1\mathbf{M}$ faces). Our works can be divided into two categories with contradictory objectives. In the first category we find methods aiming at representing faces by compact representations embedding information such as identity, age, expressions, *etc.* There are many applications requiring these types of representations, such as video surveillance, personalized and age-specific advertisements, human-computer interactions *etc.* Methods in the second category aim at altering face images in such a way that automatic face matchers cannot match the faces while they can still be easily recognized by human beings. This is done by preventing the extraction of information such as the identity of the person. The main application of such type of representation is privacy protection.

The simplest way for representing faces[1] could be the concatenation of raw pixels intensities represented under the form of vectors. However, such representations would raise many concerns: the dimensions of the signatures would be high, they would not encode the dependencies and locality relationships between the different parts of the images, and, would not be robust to poses, illumination changes, *etc.* Hand-crafted features such as Local Binary Patterns (LBPs) [2, 23], Local Quantized Patterns (LQPs) [65], Local Higher-order Statistics (LHSs) [116], Scale Invariant Feature Transforms (SIFTs) [47], Fisher Vectors (FVs) [118], Histogram of Gradients (HOGs) [34] have been successful in several face identification and recognition tasks. Recently, Convolutional Neural Networks (CNNs) [130, 101, 131] features, which are learned end-to-end, have been proven to be successful for face verification and identification. One important issue with all these representations is that their dimensions are very high, so high that using them as such in large scale setups is not always feasible. The length of the dimensions of these representations is in the orders up to a few thousands. Recently, some of the deep CNNs like DeepIds [126, 129, 128, 125] and [154] learned representations of faces having dimensions up too few hundreds, not as compact as 32 (more in Sec. Experiments in Ch. 3). Thus, these representations require to be compressed before they can be used. Unsupervised dimensionality reduction techniques such as Principal Component Analysis (PCA), Whitened PCA (WPCA) are commonly used to reduce the dimensions. But these techniques are not specialized for the tasks considered, and thus can lose some information that would have been very useful for a particular task. In contrast, Metric learning (ML) approaches [52, 93] are quite successful to represent faces with compact and discriminative representations. ML is a principle allowing to compress efficiently high dimensional features into compact and meaningful representations adapted to a particular task.

With the steady increase of the size of face databases, searching for faces of a particular person or of a given age is becoming very challenging. As said before, the need for compact and discriminative face representations is becoming more and more critical. In addition, designing architectures which can organize the images bearing certain sets of common attributes is also important to reduce the search space.

Although, large volume of face images is available publicly on the web, it is often hard to find training data for specific applications. Furthermore, when available, these training data often does not contain enough training examples. Annotating large number of examples for each and every task is difficult, time consuming and expensive. Consequently, there is a need to design algorithms which can utilise training examples of related (but different) tasks to improve their performances or at least the performance of the main task. Moreover, the algorithms need to be scalable to high dimensional features and large scale datasets. Joint learning and multi-task learning methods [20, 28, 6] are quite successful in using training examples from related tasks and learning the parameters simultaneously. Joint learning methods optimise the parameters of more than one task at a

[1] *Faces* and *face images* are used interchangeably unless specified

time, and help in better generalizing the model. The key idea of multi-task learning is to utilise the training examples annotated for some related tasks to improve the performance of a main task for which there is not sufficient annotated data.

Recently, automatic age estimation from face images has become a popular research problem [60, 26, 121, 133, 21]. There are many important applications such as age-specific human-computer interaction [48], business intelligence [114], *etc.* Some of the works on estimating ages from faces have shown that the rate of ageing [56, 57, 58, 55] differs from a group of people to another group of people. These works categorized the people in different groups based on their sexes (Male, Female) and races (Black, White, Asian, *etc.*). Their experimental results show that an age estimating model learned from training examples of a group of people *e.g.* Black Male (BM), when used to predict ages in another group *e.g.* White Female (WF), failed to make a good estimation of ages. These kinds of problems are commonly known as a domain adaptation problem [75], where each of the groups is considered as a domain. Guo *et al.* [57] proposed a sequential method to solve this type of problem *i.e.* identifying the group of people, segregating them into separate groups and finally, training a separate model for every group. In a practical scenario, this is quite impossible due to difficulty in collecting enough amount of training examples for every group of people. To address such type of problems in image classification, [111, 80, 45, 49] proposed a sequential method. First, they align features from such groups into a common subspace and then learn a classifier for all. It made possible to learn a common set of parameters for all the domains and ultimately helped to reduce the number of parameters to learn, which increases with the number of domains. Most importantly, the domain which lacks sufficient amount of training examples, can utilise training examples from other domains and can learn a robust model. However, the problem with these existing methods is that they do not learn the parameters for their end tasks (*e.g.* classification) while aligning the features. Jointly optimising the parameters for both the common subspace and the end task can even improve the performance.

In addition to the above-mentioned domain adaptation problems related to face analysis, there is a similar concern in the combination of different image features. For example, a classifier trained with FVs computed from a set of training examples fails to make correct predictions when it is used to make inferences on test sets represented with feature others than FVs, even if both the train and the test set belongs to the same domain. LBPs, FVs, CNNs (three typical image features), have their own strengths from the point of efficiency and accuracy. We recommend the readers to refer the results on Labeled Faces in the Wild (LFW) [85, 86] – the most popular dataset in face analysis, due to various types of image features for face verification. Moreover, as these features are complementary in nature [115], combining them could improve the performance of the end task. Depending upon the requirements and availability of computing resources, a device can compute a type or multiple types of features of a face image at a time. For making evaluation in such scenario, we are obliged to train and place model for each specific

types of features and their all possible combinations. It is because, you never know what will be the features available for the system at train/test time. Thus, it increases the set of parameters exponentially with the increase in numbers of feature types. To address such problems, we need to design a system which aligns all the features in a common subspace and generates feature-type-invariant representations and fuse them. This kind of system will be ready to accept any kind and number of features and be optimal to them. Such a system can be easily deployed in client server architecture. Model parameters will be saved in the server and a client can send any type(s) of feature(s) to the server and server will send back the outcomes after evaluation. Moreover, such server-client architecture will be useful for privacy protection. Client computes the features of the image, sends to the server for evaluation but it is not obliged to share images with the server. Low-cost computing and mobile devices can benefit from such an architecture. These devices can compute any type(s) of feature(s) from face images and send it to the server.

Apart from the issues related to the tasks of recognizing the identity, age, expressions from faces, there are some issues with masking them, particularly identity. Every day millions of photos are shared on social network sites such as Facebook, Twitter, Instagram *etc.* A recent study on the content of photos shared on Instagram [63] shows that nearly half (46.6%) of the total shared photos on it comprises the *selfies* and *photos with friends*. Press statements of Instagram [67] show that more than $95\mathbf{M}$ of photos and videos are shared every day. From these statistics, we can clearly imagine that millions of face images are shared in the social network sites per day. Due to the increase in robustness of face matching algorithms, automatic tagging of photos is quite prevalent in social network sites such as Facebook. Because of this functionality, there are serious problems in the privacy of an individual. Automatically matching the profile picture of a person with the images available publicly in web and sharing the activities corresponding with the photos to his relatives, friends, groups *etc.* or sharing friends, members of groups, liked page information, *etc.* to the public creates serious privacy threat. To maintain the privacy, there is a need for designing a tool which alters the representations of the face in such a way that the automatic face matching algorithm fails to match the identities. This process is commonly known as *face de-identification*. Moreover, after the de-identification, unlike one of the previous works [41](see Fig. 1.4), if the aesthetics of the face image can be preserved as close as before, this motivates users to de-identify their photos before uploading in the social network. Consequently, the risk factor of privacy issues can decrease by a large factor.

1.2 Tasks

As mentioned at the beginning of this chapter, the main objective of this thesis is to design and develop novel algorithms to learn the representations of faces which are discrimina-

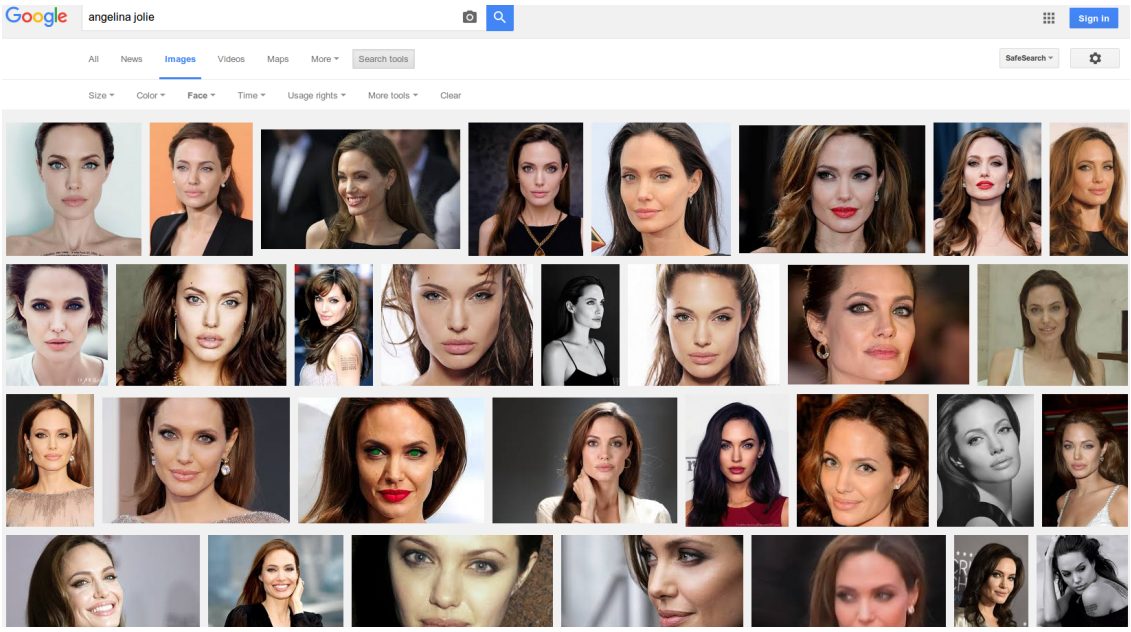


Figure 1.1: Face Retrieval: Top-ranked Face images of Angelina Jolie by Google image search engine.

tive, compact, domain invariant and de-identifying. We apply the methods proposed in this thesis for following different tasks to analyse faces from various perspectives.

Face Retrieval. Face retrieval involves the task of comparing a query with all the faces in the face database and returns the top-k faces. More precisely, the task is to retrieve faces similar to a query, according to the given criteria (e.g. identity) and rank them using their distances to the query. Fig. 1.1 shows the top ranked faces of Angelina Jolie by an image search engine of Google.

Age Estimation. Age estimation from face images is an interesting and challenging problem. It involves predicting the age of a person from his/her face image. Fig. 1.2 shows the predicted age of Actress Angelina Jolie by How-Old.net [2].

Face Attributes Predictions. Representing faces with higher levels of features such as: *wearing a hat, pointed nose, smiling, young, woman, etc.* have been quite successful in multiple applications such as face verification [13, 82, 120], identification [120], *etc.* Face attributes prediction involves the task of predicting the presence of such higher level of features for a given face image. Fig. 1.3 shows a face image of Albert Einstein and the different attributes to describe his face. The value corresponding to each of the attributes is the probability of an attribute to be present in the image.

Face De-identification. Face de-identification involves task of altering the representation of faces in such a way that automatic face matching algorithms fail to match photos from the same identities. Fig. 1.4 shows the face de-identified by the methods proposed in [41].

[2] http://www.how-old.net/#

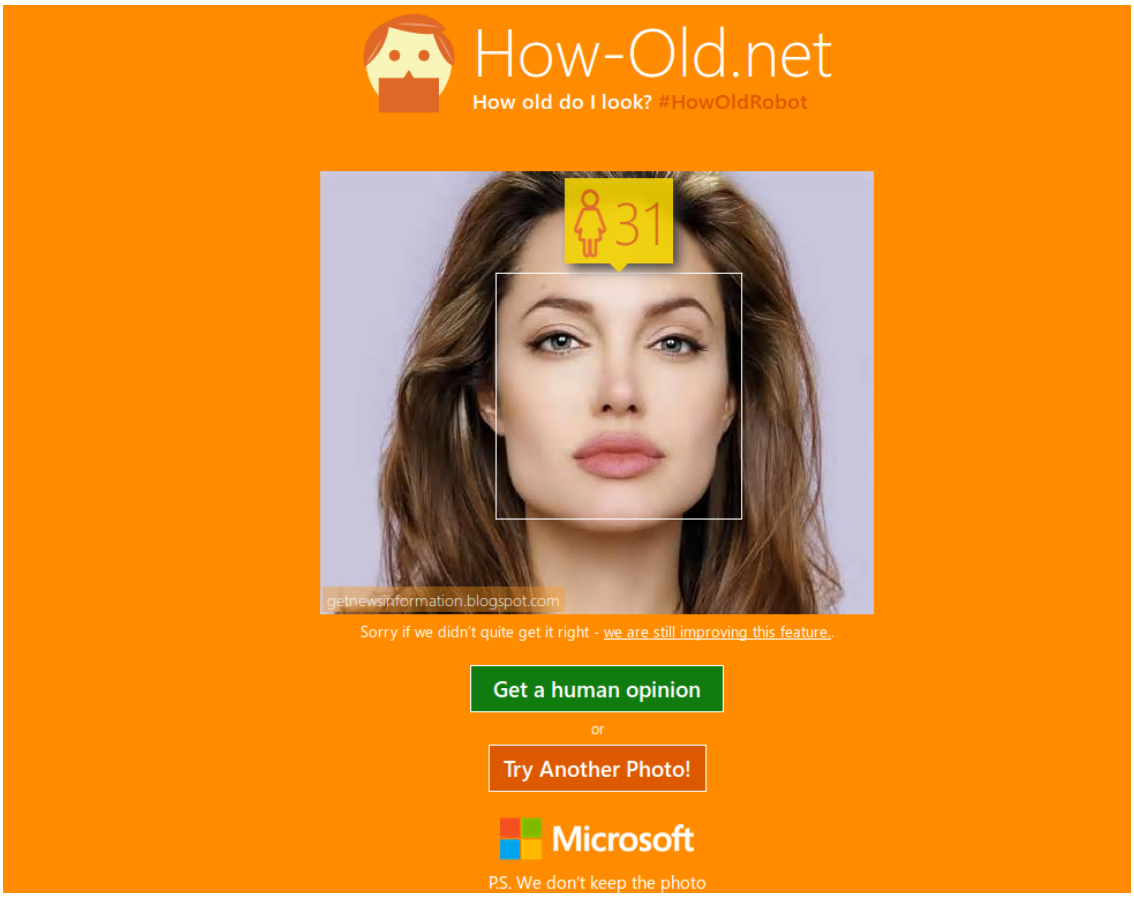


Figure 1.2: Age Estimation: Prediction of the age of Angelina Jolie from her face image. The result is predicted by How-Old.net, an age predicting tool from Microsoft.

1.3 Datasets

To evaluate the performance of our proposed methods for the tasks mentioned above, we used several standard public and challenging datasets. We introduce all the datasets we used briefly in the following section. We explain in detail how we use them for the different experiment purposes in the coming chapters.

Labeled Faces in the Wild (LFW) [64] [3] is a standard benchmark for faces, with more than $13,000$ images and around $5,000$ identities. This dataset is specifically designed for identity-based face analysis tasks such as face verification, identification, retrieval, and others. To know more about the recent works evaluated with LFW, we recommend the readers to refer [86]. Fig. 1.5 shows some of the randomly sampled images from LFW.

CASIA Web [154] [4] dataset consists of $494,414$ images annotated with identities for $10,575$ people. The images are annotated in weakly supervised manner. As we mention before, CNN features are getting successful in the different computer vision applications.

[3] http://vis-www.cs.umass.edu/lfw/
[4] http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html

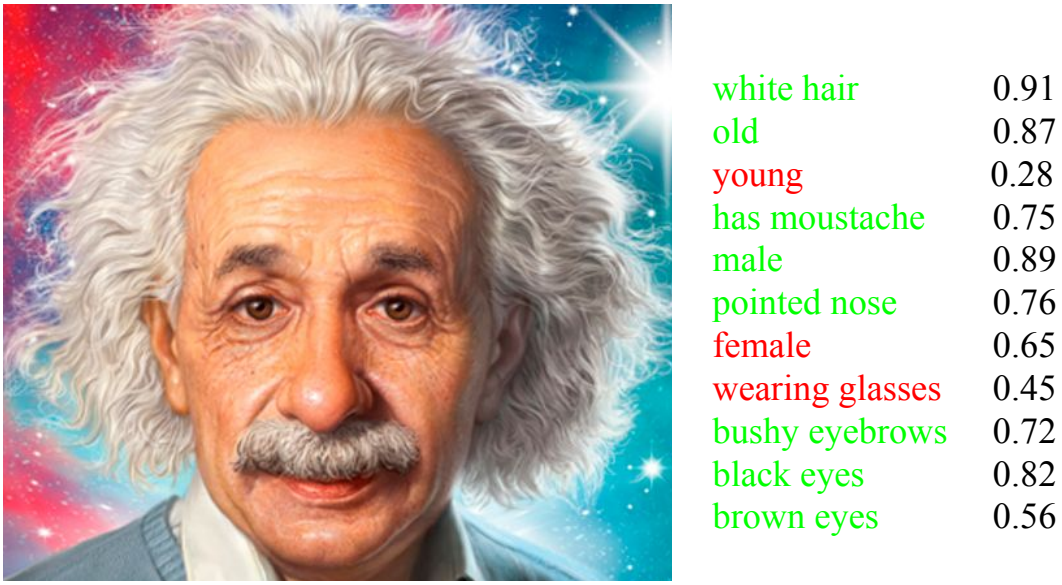


Figure 1.3: Attribute Prediction: Prediction of different facial attributes of Albert Einstein from his face image. In the figure, each new line represents an attribute and the value corresponding to it represents the probability that the attribute is present in the given image. Green colored attributes are true attributes while red colored attributes are false attributes.

Figure 1.4: Face De-identification: De-identified faces from [41]. From left to right: (a) original image, (b) pixelized with block size 16*, (c) Gaussian blurred image with standard deviation* 8*, (d) scrambled by random sign inversions and (e) scrambled by random permutations.*

But, to learn the CNN parameters, it requires a significant number of annotated examples. The main objective behind to release this dataset from the authors is to train a Convolutional Neural Network for generating CNN features of faces.

MORPH-II [109] [5] is a benchmark dataset for age estimation. It has around $55,000$ images annotated with both age and identity. There are around $13,000$ identities, with an average of 4 images per person, each at different ages. In addition to it, each of the face images is annotated with *sexes* (Male and Female) and *races* (White, Black, Asian, *etc.*) of the people. Fig. 1.6 shows some of the randomly sampled images from this dataset and their annotations.

[5] `http://people.uncw.edu/vetterr/MORPH-NonCommercial-Stats.pdf`

(a) Zhu Rongji (b) Jose Mourinho (c) Dona Ralston (d) Paul Wilson

Figure 1.5: Sample face images from the database LFW. This dataset is annotated with identities.

(a) Male, Black, 21 (b) Female, Black, 30 (c) Female, White, 36 (d) Male, White, 46

Figure 1.6: Some of the images from the database MORPH-II. This dataset is annotated with sex, age and race.

FACES [43] [6] is a dataset of facial expressions with 2052 images of 171 identities. Each identity has 6 different expressions (*neutral, happy, angry, in fear, disgusted,* and *sad*) with 2 images of each. This dataset is used for expression matching tasks. In the Fig. 1.7, we can see some of the sample images from this database annotated with different expressions.

SECULAR [16] is a dataset having one million face images extracted from Flickr[7] by the INRIA-TEXMEX group. These are randomly crawled images and these images are not biased to any of the tasks or datasets mentioned above. It is because, the images of this dataset are from ordinary individuals unlike LFW who contains celebrities. The purpose of this dataset is to use it as distractors in large scale identity-based face retrieval and make the task even more challenging.

CelebA [89] [8] the largest publicly available dataset annotated with facial attributes. There are more than $200,000$ face images annotated with 40 facial attributes. Some of the annotated attributes on the face images are *bangs, wearing glasses, wearing hat, young, etc.* In Fig. 1.9, we can see some of the sampled images and their corresponding attributes

[6] http://faces.mpib-berlin.mpg.de/album/escidoc:57488
[7] http://flickr.com
[8] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

(a) Sad (b) Neutral (c) Happy (d) Fear

Figure 1.7: Sample face images from the database FACES. This dataset is annotated with different expressions. Each column represents images from an expression.

from this dataset.

1.4 Evaluation Metrics

To compare the performance of our proposed methods with baselines and existing state-of-the-art methods, we followed standard evaluation procedures. To evaluate every proposed method, we randomly split the datasets into three disjoint sets *i.e.* train, val and test sets (if not available beforehand). Train and val set are used for learning the hyper-parameters and model parameters. However, we report the performances on the test set. We describe more in detail about this in the forthcoming chapters. Since we are dealing with wide ranges of facial analysis tasks, we used different task-specific evaluation metrics to compare the performances with that of the existing works.

1-call@K One of the major tasks we are interested in this thesis is large scale identity based face-retrieval. In our case, there is a human operator and we want to give the operator a reasonable amount of images to look at. In the ideal case, the top ranked retrieved face would be of the same person, but it would make a practical system if the correct face is ranked in the top n images. For a small value of n, the images can be manually verified by an operator easily. Hence, we propose to use k-call@n [24] (with $k = 1$) for our evaluation purpose. For a query, the retireved attempt is considered as

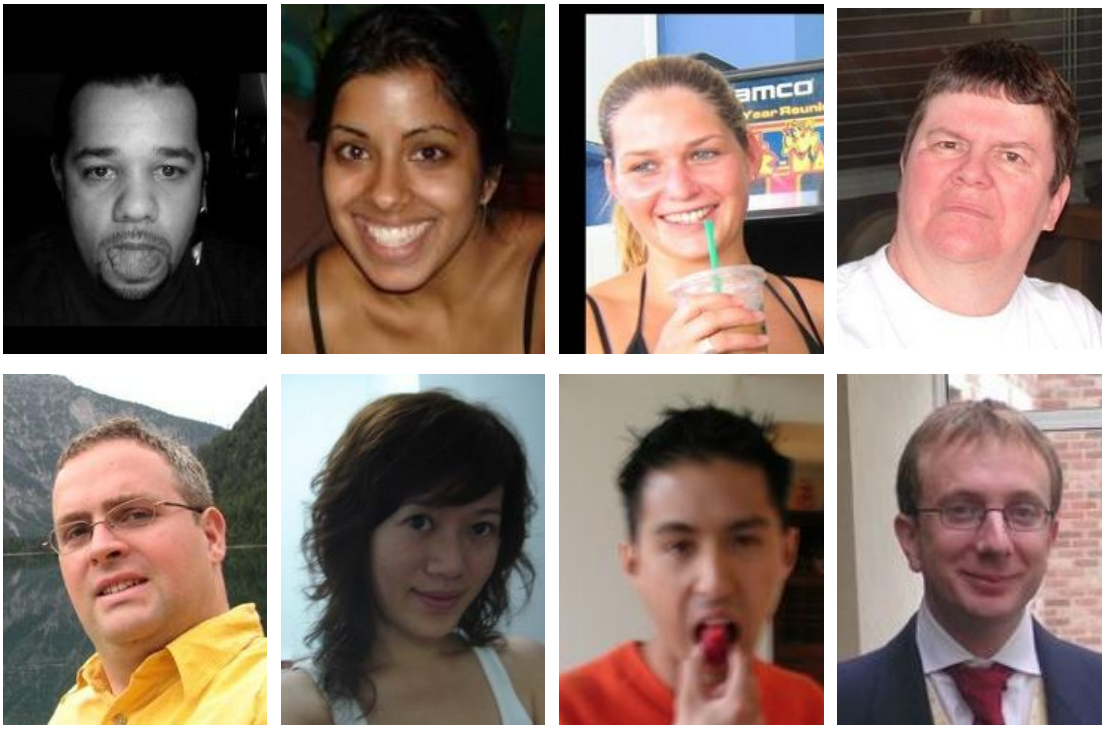


Figure 1.8: Sample face images from the database SECULAR. This dataset is created from the images crawled from Flickr.com. This set comprises images from ordinary people uploaded in Flickr.

success if at least k of the top n retrieved results are relevant. We average this over our queries and report the mean 1-call@n.

Mean Average Error (MAE) Predicting an age from a face image is another important task we are addressing in this thesis. We are reporting the performance in Mean Average Error. For a face image, if the annotated is y and the predicted age is $\hat{y}$, then absolute error is computed as the modulus of their difference. We average the error on all the face images of the test set to compute Mean Average Error (MAE).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(|y_i - \hat{y}_i|) \tag{1.1}$$

Average Precision (AP) We proposed facial attributes prediction tasks as multi-label prediction problems. For this multi-labels prediction problem, we computed average precision from the prediction scores of classifier and compared the performance with existing state-of-the-art methods. For each relevant label, average precision computes the proportion of relevant labels that are ranked before it, and finally averages over all relevant labels. This corresponds to the area under precision *vs.* recall curve. The higher the value of average precision, the better the performance and average precision = 1 means the perfect performance

Figure 1.9: Sample images and their attributes from CelebA dataset.

Accuracy Face verification involves the matching two different images from the same person. To evaluate the performance on face verification, we compute accuracy. We compute the accuracy as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1.2}$$

In the Eqn. 1.2, TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative pairs.

In addition to these above mentioned empirical evaluation metrics, we evaluated our methods with qualitative visualisations.

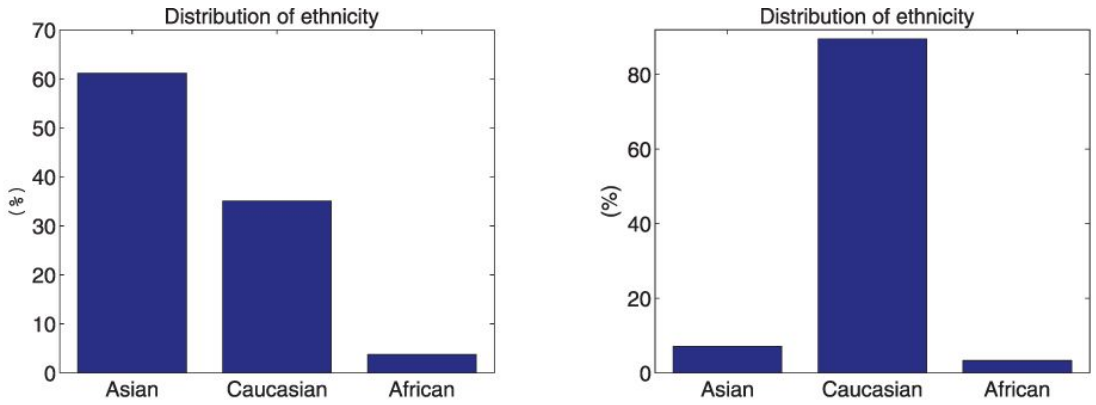
1.5 Challenges

As we are mainly focused on analysing faces, we present here some of the challenges we encountered while dealing with face images.

Variations in Appearances. Facial appearance of a person is determined by various factors such as age, expressions, illuminations, poses, and partial occlusions *etc.* To design a model which is robust to these variations is a challenging problem. Fig. 1.10 shows the faces of persons at their different expressions, age *etc.*

Figure 1.10: Variations in appearances of persons due to wide ranges of expressions, age, etc. *The first row shows the photo of President Hollande in his different expressions. Similarly, in the second row, the photos are of Actor Tom Cruise at his different ages. At the end row, the photos are of Singer Brayan Adams with various levels of expression.*

Domain Discrepancy and Lack of Annotated Data. As it is well known that the major obstacles with supervised learning methods is to collect task-specific training examples. A large number of training examples are required to design a robust model. Recent success stories of CNNs in the field of faces such as verification [101, 130], require millions of annotated examples. As we all know, collecting annotated training examples is tough, expensive, needs experts, and is time consuming. In the domain of facial image analysis, one of the largest publicly available annotated datasets [154] consists of 500K images labeled with identities. But the datasets annotated with other attributes of faces such as ages, expressions, emotions are small and hard to find. The largest publicly available dataset annotated with age is: MORPH II [109], consisting of around 55×10^3 face images. Recently, Cross-Age Celebrity Dataset (CACD) [22] is released annotated with ages, but this dataset is weakly annotated and cannot be directly used for supervised learning algorithms. If we further look into MORPH-II, the distribution of people from different races, sexes are different and skewed in nature. For instance, it has about 77% Black faces, 19% White, and 4% other races, *e.g.* , Hispanic, Asian, and Indian. The number of males is also higher than that of females. Note that the previous studies [56, 57, 58, 55] have shown



(a) KFW-I (b) KFW-II

Figure 1.11: Distribution of face images on the basis of ethnicity in KFW-I and KFW-II [90].

that the rate of ageing is different from a group of people to another group of people. Fig.1.11 shows a distribution of faces on the basis of races in KinFaceW-I (KFW-I) and KinFaceW-II (KFW-II) [90]. In this figure, we can see that there is an uneven distribution of faces between the different ethnicity, and some of the ethnicity of people (in Fig.1.11, *African*) really lack of annotated examples. Solving these problems of shift in features from a domain to another domain and learning robust model for the domains lacking enough training examples is also a major challenge.

Multiple Feature Types Adaptation. As we mentioned before, an image can be encoded with multiple types of features (LBPS, FVs, CNNs, LHSs *etc.*), and they have their limitations on speed and accuracy. The use and applicability of these types of features depend from system to system and is determined by the available computing resources and time frame. It allows us to remind you that previous studies [115] shows that these features are complementary in nature. A system may compute single type of features or more than a type of features depending on its need. Separately training a model for each and every combination of features to make predictions is quite unrealistic, as the number of parameters to learn grows exponentially with the number of types of features. Thus designing a unified system which can generate feature-type-invariant representations, optimal to every kind of features and their any possible combinations is also a challenge.

Speed and Scalability. Due to the exponential growth in the size of digital data shared on the Internet, the volume of data that needs to be processed for different applications such as searching a person, persons of an age *etc.* is enormous. The volume is still growing without any limits. Searching relevant faces from a large face database requires a lot of time as it requires to compare a query with all the faces present in the database. In addition to this, the dimensions of the state-of-the-art features such as CNNS, FVs, High dimension LBPs, *etc.* are quite high. Thus, designing an algorithm or a system for searching faces efficiently and accurately from a large database is another challenge.

Privacy Protection. Nowadays, staying out of the virtual social network is almost impos-

sible. Facebook alone has around $1.65\mathbf{B}$ [138] monthly active users. Due to the gaining popularity of social networks, people are sharing their photos every day and in every occasion, mostly they are frontal faces. Automatic face matching algorithms are also getting more mature than before. As we briefly mention before in this chapter, due to the increase in the robustness of the face matching algorithms and trend of huge sharing of photos on social networks, the security threat is alarming. To protect from such threats, previous works such as [41] propose to blurring, pixelizing, scrambling, *etc.* techniques to de-identify the faces. Fig. 1.4 shows the de-identified images from this method. We can observe that the beauty of photos is completely lost and we, humans cannot identify them. For the sake of privacy, people will not be interested in sharing such photos in the social network as their profile or any other photos. Thus, alternating the representation of photos to maintain the privacy and keeping the aesthetics of photos as close as before is another big challenge.

1.6 Overview of existing face representations

As we mentioned in the beginning of this chapter that the main goal of this thesis is to generate compact, discriminative, domain invariant and anonymized representations of faces. The works carried out in this thesis are not aimed to propose novel representations of faces, rather investigating the limitations of the existing features and overcoming the above-mentioned challenges due to existing features. In this part of this thesis, we review some of the representative face representations from the beginning to these days and identify some of their limitations.

Face analysis has been an active research topic since last few decades. Researchers from all over the world are continuously proposing several types of features to make them robust to the change in illumination, occlusion, variations in ages, expressions etc. The existing features can be grouped into three main categories viz. linear function of pixel intensities, local non-linear hand-crafted representations and deep non-linear representations. We review some of the main works from each of these categories below.

Linear function of pixel intensties This category of features is the earliest representations of faces. Some of the peculiar representations are: eigen faces [135], fisherfaces [11], laplacian-faces [61] *etc.* These methods take the pixel intensities from whole the region of face and apply linear functions: Principal Component Analysis, Linear Discriminant Analysis and Locality Preserving Projections respectively. These representation are fully linear but the (inter-) intra-personal facial relations are highly non-linear. Since this category of features applies linear functions on pixel intensities, these features are not robust to change in illuminations, translations, pose *etc.*

Local non-linear hand-crafted representations To address the limitations of the linear

function of pixel intensities, researchers proposed locally non-linear hand-crafted features such as Local Binary Patters (LBPs) [2], Scale Invariant Feature Transforms (SIFTs) [47], Local Phase Quantisation [3] *etc.* These types of features are quite successful in controlled setting [117] but their performances degrade when applied in uncontrolled setting [86]. But, when Mahalanobis-like metric learning is used [53, 93] on these types of features, the performance is improved by a large margin in uncontrolled setting too. The Mahalanobis-like distance metric learning involves learning of transformation matrix to better satisfy the imposed constraints such as must-link and must-not-link. We introduce and explain about metric learning in more details in the upcomming chapters.

Several arts and techniques have been applied to compute these types of features. Guillaumin *et al.* [53] computed SIFTs on key points of the faces [44] rather than taking the whole region of face into considerations. Similarly, Chen *et al.* [23] proposed to crop the images on key points, rescale the cropped regions by multiple factors and compute the dense LBPs. Their technique resulted face representations with very high dimensions (dimensions up to 1000K). However, some of the other successful representations such as LQPs [65], LHSs [116], FVs [118] *etc.* use whole region of faces to compute the representations. LQPs are generalized version of local patterns such as LBPs [2], LTP [132] and are more robust than their local counterparts. Similarly, LHSs and FVs use the similar encoding technique to summerise densely computed local features LBPs and SIFTs respectivly. To summerise such patterns, these methods uses soft paritioning of feature space using parameteric mixture model (Gaussian Mixture Model (GMM)) followed by encoding the derivatives of the log-likelihood of the model with respect to its parameters [68].

As this category of features need to use metric learning to improve their performance, without loss of generality we could say that this type of features are generic. These features are important to transform and make task specific and discriminative.

Non-linear deep representations Recently, Taigman *et al.* [130] proposed to train Convolutional Neural Network (CNN) to induce deep representations of faces for face verification. After this work, deep learning techniques are successfully used in several facial analysis tasks such as face identification [131], face verification [101], facial attribute predictions [89] *etc.* Deep architecture has multiples of different non-linear layers *i.e.* convolution, pooling and fully connected. All the parameters of this network is configurable except for the image pixel intensities (input to the CNN). And, the activations of penultimate layers of the network are taken as representations of faces. To learn the parameters of such networks, large volume of annotated data is required. [130] used 4**M** of faces annotated with identities to train their network. Similarly, FaceNet [113] uses 200**M** images to learn the parameters of their network.

Some of the other important deep learning methods such as DeepID [128], DeepID2 [125], DeepID2+ [129], DeepID3 [126] propose to learn ensemble of CNNs. The peculiar difference between DeepId and other version of DeepIDs is, the former uses

only identity-classification loss but the later one uses additional identity matching (verification) loss. From their experiments, it is shown that additional identity matching loss is useful to improve the performance in face verification and retrieval. Thus, deep representations are also benefited by metric learning approach *i.e.* learning parameters to push dis-similar identities apart and bringing the images from same identity closer.

Although these representations are quite successful in several facial analysis tasks, the major bottleneck to learn this kind of representations is, it requires large volume of annotated examples. In addition to this, this type of features are less generic in nature than hand-crafted feature but still need to optimise and make task specific.

From this brief review of the existing feature representations, we can see that metric learning has been quite successful to improve the performance of both hand-crafted and deep learning features. Moreover, deep learning is out-performing existing handcrafted representations. In this thesis, we use metric learning and deep learning as tools and techniques to tackle the challenges faced by existing representations of faces for the above-mentioned face analysis tasks.

1.7 Contributions

Here we briefly introduce the contributions that we made in this thesis. We present our contributions and their related works in detail in the forthcoming chapters.

Hierarchical Metric Learning. To address the problem of the linear increase of search time with the size of the face database and the inability of single linear projection matrix to capture all the non-linear facial relations, we propose a novel semi-supervised learning method for automatic hierarchical organization of face database for efficient and accurate face retrieval. Grouping faces based on their common characteristics and looking for the relevant images in the most likely group of faces reduces the search space, which ultimately reduces the search time. Guillaumin *et al.* [53] and Mignon and Jurie [93] proposed to learn a single projection matrix and project all the faces into the same subspace. Their approach forces us to compare query with all the faces in the database to find the relevant faces. We propose to learn multiple matrices in an inverted binary tree fashion. The learning of each projection matrix is followed by an unsupervised k-Means clustering to split the data into groups until we reach to the leaves of the tree. Our method groups the faces based on attributes from coarse ones such as *sex* to fine-grained ones such as *wearing glasses, bald, bangs, etc.* while going from the root node to the leaves of the tree. As our method groups the faces by their common intrinsic characteristics, to search relevant faces, queries need not be compared with all the groups but only with the most likely group of faces. Our approach reduces the face retrieval time complexity by a factor up to $10\times$ and performs better than the compared baseline metric learning

method [93]. In Chapter 2, we discuss in detail this contribution and its related works.

Multitask Metric Learning. Parameswaran *et al.* [100] proposed mt-LMNN, a multitask metric learning framework based on Large Margin Nearest Neighbour (LMNN) [144] – a distance metric learning method. It is one of the earliest and remains one of the most successful multitask metric learning methods. Their approach learns Mahalanobis-like distance of related tasks simultaneously. The major drawback of their approach is the inability to scale with the increase in dimensions of features. The size of parameters to learn increases quadratically in proportion to the dimensions of features. Hence, to reduce the size of parameters and to generate compact representations, data needs to be compressed by large margins. This reducing dimensionality of data beforehand, causes information loss and ultimately the performance drops. To address these shortcomings, we proposed a novel Coupled Projection multitask Metric Learning (CP-mtML) method which is highly scalable and can work up to thousands of feature dimensions quickly. Like our previous contribution, here, we also learn multiple matrices: a task specialized matrix for each task and a common matrix for all of them. Unlike our previous contribution, the matrices are learnt in flat label not in hierarchy. In comparison to existing method [100] which used multi-label dataset, we utilised heterogeneous datasets to learn the parameters. It is, in general, more challenging to use different datasets specialized for different tasks than a multi-label dataset for multi-task learning. In comparison to existing method, we impose sparse pairwise similarity and dissimilarity constraints instead of triplets. It is comparatively easier to collect pairwise similarity and dissimilarity annotations than triplets. We evaluated our method for Identity and Age based Face Retrieval in large scale setting. The proposed method outperforms the existing state-of-the-art methods and competitive baselines and attains state-of-the-art performance. We detail our contribution in Chapter 3 along with its related works.

Cross-Domain Age Estimation. To address the problem of domain adaptation in age estimation from faces, we proposed a novel joint projection matrix and regressor learning objectives. Some of the previous works such as [111, 80, 45, 49] proposed to learn a common subspace to align the features and then train a classifier on the subspace. The problem with their approach is that the values of their projection matrices are not optimised for the end task *e.g.* classification of images. We propose to learn the parameters of projection matrix and the regressors together. Projection matrix allows us to align the features from the constituent domains into a common subspace while regressor learns to predict the ages. Since we are learning both sets of parameters jointly, we expect that the projection matrix is better specialized for the end task *i.e.* regressor for us. We compared our methods with the existing state-of-the-art [59] and several strong baseline methods. Our experimental results show that our method outperforms compared method and attains new state-of-the-art performance. In Chapter 4 we explain more about this contribution and its related works.

Deep Multi-feature Fusion. We propose a unified hybrid deep neural network for aligning multiple types of features and fusing them for attribute predictions. In our previous work on domain adaptation for age estimation, we aligned the same type of features from different domains. In this work, we align various types of features from the same domain. Moreover, in this work, we took deep learning as the principal tool whereas our previous work was based on metric learning approach. Recently hybrid deep neural network [104] has been successful in generating features that are compact and more discriminative than the input features (*e.g.* FVs) for image classification. Its performance is comparable to Alexnet [78] – a deep CNN. We propose multi-input hybrid deep neural network to generate compact, discriminative and feature-type-invariant representations. Our network can accept from single to many different types of features (handcrafted e.g. LBPs, FVs and CNNs) at a time. We proposed an objective to align all the features in the same subspace. Moreover, the proposed network is optimal not only to input features; it is also optimal to any combinations of these input features. In comparison to most recent work [95] our proposed method is easy to train and can adapt new feature types easily. Empirical results show that our method outperforms the competitive baselines, existing best performing method and attains state-of-the-art performance. We will discuss this contribution and its related works in Chapter 5.

De-identification of Faces. All our contributions, that we briefly described before, concentrate on generating the representations of faces which are more discriminative, compact and domain invariant for the different applications which infer the information such as identities, ages, expressions, *etc.* Now, we will briefly explain here our application on faces which is one of the most important applications in the domain of face analysis, but has received the least attention so far in comparison to other applications. The objective of this work is to protect our face images from being matched by automatic face matching algorithms. This kind of work is known as de-identifying works, and the process is called de-identification of faces. Some of the earliest face de-identifying methods [39, 96, 41] blur faces which prevent even us from identifying the person. For example, Fig. 1.4 shows some of the de-identified images from [41]. The short-coming with these approaches is the faces after de-identification loses their beauty and it is even difficult for humans to identify. In contrast, the aim of our work is to maintain the quality of faces as good as before but to make the automatic face matching algorithms fail. This will encourage users of social networks to use this tool before uploading their pictures in the web. We address this question by drawing a parallel between face de-identification and oracle attacks in digital watermarking [29, 30, 42]. In our case, the identity of the face is seen as the watermark to be removed. Inspired by oracle attacks, we forge de-identified faces by superimposing a collection of carefully designed noise patterns onto the original face. The modification of the image is controlled to minimize the probability of good identity matching while minimizing the distortion. In addition to this, these de-identified images are – by construction – made robust to counter attacks such as blurring. We present an

experimental validation in which we de-identify LFW faces and show that resulting images are still recognized by human beings while deceiving one of the state-of-the-art face recognition algorithms. We will explain our contribution and related works in Chapter 6.

Chapter 2

Hierarchical Metric Learning

Contents

2.1 Introduction

This chapter and the following one (Ch. 3) present our works on large scale face retrieval. This chapter focusses on our work on the automatic organization large face databases for efficient and accurate face retrieval while Chapter 3 presents our method to learn parameters from related tasks in multitask setups for efficient large scale face retrieval.

The task of identity-based face retrieval can be described as follows: given a query face image, retrieve the face(s) of the same person from a large database of known faces with large changes in face appearances due to pose, expression, illumination, etc. This task finds numerous applications, particularly in indexing and searching large video archives and surveillance videos and in controlling access to resources.

Metric learning has been quite popular to learn compact and discriminative representations of faces [52, 118, 93]. Such metric learning can be seen as a *global* approach where a single linear projection is learned to discriminate all types of faces. Instead of learning

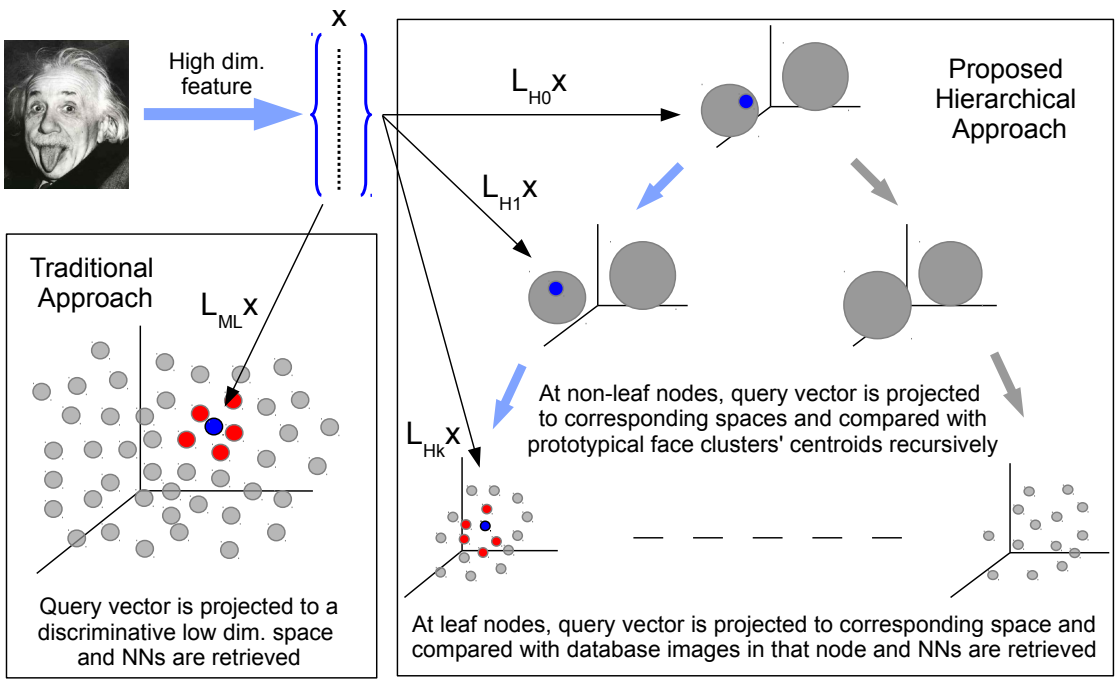


Figure 2.1: Principle of the proposed method, in contrast with the traditional metric learning based approach. While the traditional approach learns a single projection (L_{ML}) the proposed approach works hierarchically and learns different projection matrices (L_{H_n}) for different nodes. See Sec. 2.3 for details.

single set of parameters, Zhang *et al.* [156] proposed to learn a collection of *local* (linear) discriminative models. This approach outperformed the performance of single global model for the task of visual classification. Also, recent work from Kumar *et al.* 's attribute-based works on facial analysis [82, 83] hint towards the presence of local modes in the (attribute transformed) space of faces. In the same way, Verma *et al.* [139] proposed a novel framework to learn similarity metrics using class taxonomies, showing that nearest neighbor classifiers using the learned metrics get improved performance over the best discriminative methods. Inspired by these previous works, we propose to organize large face databases hierarchically using locally and discriminatively learned projections. More concretely, we propose a semi-supervised hierarchical clustering algorithm, alternating between the two steps of (i) learning local projections and (ii) clustering for splitting the faces into sets of more localized regions in face space. Intuitively, we expect such a hierarchical setup to capture coarse differences, *e.g.* gender, at the top levels and then specialize the different projections at the bottom levels to finer differences between the faces. Fig. 2.1 gives an overview of our approach in contrast to traditional metric learning. One big difference with [82, 83] or [139] is that our approach does not need any face taxonomy nor predefined set of attributes. Both are automatically discovered.

In the following, we set the context for our work in Sec. 2.2 and then describe our approach in detail in Sec. 2.3. We discuss our approach in relation to the most closely related works in Sec. 2.2.1. We then give qualitative and quantitative experimental results validating our approach in Sec. 2.4 and conclude this chapter in Sec. 2.5.

2.2 Context and related works

Comparing face images of different persons with large variations in appearance, pose, illumination, *etc.* , is a challenging problem. Locally computed features like Local Binary Patterns (LBP), Local Ternary Patterns (LTP) and Local quantized patterns (LQP) have been quite successful to address these kinds of problems [1, 132, 65]. One of the recent state-of-art methods [23] on Labeled Faces in the Wild (LFW) [64], the most challenging face verfication dataset, computes very high dimensional LBP (of dimension as high as 100k). In the recent years, several other variants of LBP have been introduced for different computer vision tasks (*e.g.* [107, 62, 149, 99]). In this work, we use the standard LBP descriptor for a good efficiency and performance trade-off.

Many other recent works address the problem with novel approaches, *e.g.* discriminative part-based approach by Berg and Belhumeur [15], probabilistic elastic model by Li *et al.* [87], Fisher vectors with metric learning by Simonyan *et al.* [118], novel regularization for similarity metric learning by Cao *et al.* [18], fusion of many descriptors using multiple metric learning by Cui *et al.* [31], deep learning by Sun *et al.* [127], method using fast high dimensional vector multiplication by Barkan *et al.* [9] or robust feature set matching for partial face recognition by Weng *et al.* [146]. Many of the most competitive approaches on LFW combine different features, *e.g.* [54, 147, 97] and/or use external data, *e.g.* [82, 14].

As we mentioned before metric learning has been quite successful recently on very diverse computer vision tasks and few more to mention here [8, 32, 46, 144, 151]. We refer the reader to Bellet *et al.* [12] for an excellent survey on Metric Learning. More specifically, methods based on metric learning have been reported to improve accuracy for face verification, either on static images [18, 54, 93, 118] or on videos [27]. We recall the key idea, metric learning is to learn a Mahalanobis like metric of the form $D^2_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)$, parametrized by the symmetric positive semi-definite (PSD) matrix M, to compare any two faces (described with some features) $\mathbf{x}_i$ and $\mathbf{x}_j$. The learning is based on optimizing some loss function which penalizes high distance between positives and small distance between negative pairs (see [12] for a survey of different metric learning methods/objectives). Since maintaining M as PSD is usually computationally expensive, M is often factorized as $M = L^\top L$. Then the problem can be seen as a linear embedding problem where the features are embedded in the row space of L and compared with the Euclidean distance there:

$$D^2_L(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top L^\top L(\mathbf{x}_i - \mathbf{x}_j) = \|L\mathbf{x}_i - L\mathbf{x}_j\|^2_2. \tag{2.1}$$

Local metric learning, *e.g.* learning a metric as a function of input vector, has also been studied [141]. However, this is expensive, specially in large scale as comparison with every instance will require projecting the query with a different matrix *vs.* only one pro-

jection in the case of a global metric.

Closely related to our work, hierarchically organized (metric) learning systems have also been explored in the past, *e.g.* the works by Hwang *et al.* [66], Deng *et al.* [33], Zheng *et al.* [159], Verma *et al.* [139]. However, they assume the presence of a taxonomy (most often a natural semantic taxonomy), while here we do not assume any such information. Our method is also related to clustering in general and with side information in particular [155, 124, 143, 71], the side information here being in the form of (sparse) pairwise *must-link* and *must-not-link* constraints. The goal of many of these works is to learn a metric to improve the performance of clustering with an implicit assumption that the constraints relate directly to the clusters. While in the current work, the metric learning with constraints relates to a first level of embedding which can be thought of a person identity space and then the clustering is done in such identity space. So, unlike previous works, it will be normal in our approach that two must-not-link vectors (faces of different persons) get assigned to same cluster as long as these different people share similar facial traits.

We are interested in the problem of comparing faces using learned metrics. In particular, we are interested in identity-based face retrieval with a focus on accuracy and efficiency of the setup for large-scale scenarios, *i.e.* with hundreds of thousands of distractors. As such, in addition to the above mentioned works on facial analysis, our method is also related to the SVM-KNN method of Zhang *et al.* [156] and to works on large scale image retrieval using product quantization of Jégou *et al.* [73]. We postpone discussing our method in the context of these methods to Sec. 2.2.1, after describing our method in the next section.

2.2.1 Relation with closely related works

Zhang *et al.* [156] proposed the SVM-KNN method, which for a test example creates on-the-fly a local discriminative support vector machine (SVM) classifier, based on its nearest neighbors. The motivation is that a complex non-linear decision boundary could be approximated with a piece-wise linear decision boundary. Also recently, many works based on 'local' comparisons, *e.g.* attribute based works of Kumar *et al.* [82, 83] where the faces are represented as vectors of confidences for the presence of attribute like long hairs, open mouth, *etc.* , have been shown to be important. We could imagine that the faces with such attributes would occupy a local region (or perhaps manifold) in the full face space and, thus, the success of such facial analysis system motivates us to work locally in the face space. Also, the success of SVM-KNN reassures us of the merit of a local strategy. In our case, such locality is automatically discovered in a data driven way. In the upper levels of the tree, the Vonoroi cells, corresponding to the clustering in the respective discriminative spaces of the nodes, can be interpreted as such local regions where the faces are similar

in a coarse way, *e.g.* one node could be of female faces *vs.* another of that of males. While as we go down the levels we expect such differences to become more and more subtle. We show later that qualitative results support our intuition. Hence, we could hope that concentrating on a local region (towards the bottom of the tree) where faces differ very slightly could help us discriminate better, perhaps even at a cheaper cost.

Another closely related but complementary stream of work is that of product quantization by Jégou *et al.* [73]. They propose to learn, in an unsupervised fashion, very compact binary codes to represent images and do very fast nearest neighbor retrieval at large scale. The key point is that they assume/expect the feature space to be Euclidean. However, face retrieval by directly comparing the image representations with Euclidean distance is not optimal and learning a Mahalanobis metric or equivalently a projection is required. Upon projecting the faces to such a space, Euclidean distance can be used and hence product quantization can be applied. As we have already discussed before, the proposed method can be seen as learning different projections for different local regions, we could use different product quantizations in corresponding different local regions found by the proposed method. Hence, the proposed method and product quantization are complementary to each other.

Finally, it worth comparing our approach to the recent work of Verma *et al.* [139], who proposed a framework for learning hierarchical similarity metrics using class taxonomies. Interestingly, they show that nearest neighbor classifiers using the learned metrics get improved performance over Euclidean distance-based k-NN and over discriminative methods. Our approach bears similarity with [139] as we also learn a hierarchy of similarity metrics. However, a notable difference is that our approach does not require any taxonomy. This is a big advantage as defining a taxonomy of faces would be more than challenging. Providing sufficient training annotations (*i.e.* sufficient number of faces for each level of the hierarchy) would be another complication.

One of the early works [123] on clustering proposed to do it in hierarchical fashion. This work assumes that the features are already optimised for the objective taken into consideration. Similarly, the recent work [105] proposed to learn multiple local metrices instead of only learning a single global linear projection matrix. Their approach is based on unsupervised clustering followed by learning of multiple local metrices. The major difference of our work to these work is, we propose to discover local regions on learned feature subspace than hand-crafted or feature not optimised for the task taken into consideration.

2.3 Approach

We work in the semi-supervised scenario where we have some annotated training pairs $\mathcal{A} = \{(\mathbf{x}_i, \mathbf{x}_j), y_{ij}\}$ with $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$ being features for face examples i, j resp. (*e.g.* , Local Binary Patterns [98]) and $y_{ij} = 1$ if the image pairs are of the same person and $y_{ij} = -1$ otherwise. We propose to learn a hierarchical organization of the faces for efficient face retrieval. Note that we assume the annotations are sparse, in the sense that only a very small fraction of pairs in the database is annotated.

We aim at exploiting the similarities between faces of different persons. In our hierarchical layout, we would like to first split the faces into groups based on coarse appearance similarities, *e.g.* gender, and then, at finer level, we would like to learn to discriminate between finer details in coarsely similar faces. We now discuss the case of a binary tree but the method could be applied for arbitrary k-ary trees. We start by taking all the faces into one node and learn a discriminative subspace using margin maximizing metric learning: we minimize a logistic loss function using the recently proposed Pairwise Constrained Principal Components (PCCA) [93] approach. In particular, we solve the following optimization,

$$\min_L E(L) = \sum_{\{(i,j)\}} \ell_\beta \left(y_{i,j} (\mathbf{D}_L^2(\mathbf{x}_i, \mathbf{x}_j) - 1) \right), \tag{2.2}$$

where $\ell_\beta(x) = \frac{1}{\beta} \log(1 + e^{\beta x})$ is the generalized logistic loss,

$$\mathbf{D}_L^2(\mathbf{x}_i, \mathbf{x}_j) = \|L(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 \tag{2.3}$$

is the distance in the row space of the projection matrix L and sum is taken over all labeled face pairs. The intuition of such metric learning formulation is that we would like to find a subspace (parametrized by the projection matrix L) where the distance between the positive pairs is small and that between the negative pairs is large.

We then obtain the projected features $X_p = LX$, where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ is the matrix of all face features in the database, and use k-means to cluster X_p into two clusters in the projected space. By doing this we hope to cluster the faces based on relatively coarse similarities. Once we have the clustering, we create two child nodes of the root containing only the faces from the two clusters respectively. We then repeat the process at each of the child nodes, working with faces in the current node only. At each node we save the indices of the faces which belong to the node along with the current projection matrix and cluster centroids (for the non-leaf nodes). We continue the process until a certain depth, which is a free parameter, is achieved. Algorithm 2.1 gives the pseudocode for the learning algorithm.

Once the hierarchical structure is built, the retrieval for a new query face is done by

Algorithm 2.1 Learning local metrics and organizing face database hierarchically.

Input: (i) Set of face features $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$, (ii) Sparse pairwise annotation $\mathcal{A}$, (iii) Height of the tree h, (iv) Dimensions of local projection subspaces at different depths/levels $\{D_0, \ldots, D_h\}$
Initialize: $n \leftarrow 0$, $\mathtt{1} \leftarrow (1, \ldots, N)$, $\mathtt{tree} \leftarrow \emptyset$
queue.add(n, 1) {Tree construction in a breadth-first manner}
while $n < 2^h - 1$ **do**
  $n, \mathtt{1} \leftarrow$ queue.pop()
  $\ell \leftarrow \lceil \log_2 n \rceil$ {Current level/depth}
  $L_n \leftarrow \mathtt{learn\_metric}(X[:, \mathtt{1}], \mathcal{A}[\mathtt{1}], D_\ell)$
  if $\ell < h$ **then**
    $C_1, C_2 \leftarrow \mathtt{cluster}(LX[:, \mathtt{1}], 2)$
    $\mathtt{1}_1, \mathtt{1}_2 \leftarrow \mathtt{cluster\_assign}(X[:, \mathtt{1}], C_1, C_2)$
    queue.add($n + 1, \mathtt{1}_1$)
    queue.add($n + 2, \mathtt{1}_2$)
  else
    $C_1, C_2 \leftarrow \emptyset$
  end if
  tree.add_node($\{n, L_n, \mathtt{1}, C_1, C_2\}$)
end while

traversing the tree with the following decision rule at each node: if it is a non-leaf node, project the face into its subspace and compare with the centroids and move to the closest child node (recall there is a child node for every cluster). If it is a leaf node, then project the face to its subspace and compare with all the faces in that node (projected onto the same subspace) and return the list of the nearest neighbors. Fig. 2.1 gives an illustration of the retrieval process.

2.4 Experimental results

Metric Used. We are interested in the task of identity based face retrieval, *i.e.* given a query face images, retrieving face(s) of the same person from a large database of known face images. Our objective is to find the same person and so, for us, it suffices if at least one of the retrieved faces is of the same person. In the ideal case, the top ranked retrieved face would be of the same person, but it would make a practical system if the correct face is ranked in the top n images, for a small value of n, as they can be manually verified by an operator. Hence, we propose to evaluate the method for k-call@n [24] (with $k = 1$): the metric is 1 if at least k of the top n retrieved results are relevant. We average this over our queries and report the mean 1-call@n.

Database and query set. We use the aligned version [147] of the Labeled Faces in the Wild (LFW) database by Huang *et al.* [64]. The dataset has more than 13000 images of over 4000 persons. In addition to LFW, for large-scale experiments, we add up to one million distractor faces that were obtained by crawling Flickr.com and retaining face de-

tection with high confidences. We select the persons/identities in LFW which have at least five example images and randomly sample one image each from them to use as our query set. We use all the LFW images except the query set to learn our system. The results are reported as the mean performance (1-call@n) over all the queries. All the evaluation is done with LFW annotations and, as the distractor images are from personal image collections from the internet while LFW images are that of well-known/celibrities, it is assumed that the distractors do not have the same identities as the query images.

Image description. To describe the images we use the Local Binary Pattern (LBP) descriptors of Ojala *et al.* [98]. We use grayscale images and centre crop them to size 170×100 pixels and do not do any other preprocessing. We use the publicly available `vlfeat` [137] library for LBP, with cell size parameter set to 10, of dimension 9860 for a face image.

Baseline parameter. To set the dimension of the baseline projection matrix we did preliminary experiments, with the standard protocol of LFW dataset, with values in $\{16, 32, 64, 128\}$ and found the performance (verification on LFW test set) saturated for d greater than 32. Hence we fixed the projection dimension to 32.

Tree parameters. We fixed the learned tree to be a binary tree and also fixed the dimension of the projection at successive levels to differ by a multiplicative factor of 2. But the proposed method can be easily extended to n-ary tree. We performed our experiment with k=2. It is because, binary is the simplest and elegant tree architecture to start with. Thus, the two parameters for the proposed hierarchical organization are the tree depth and the starting projection dimension. We report experiments with depths of 3 and 4, and with starting projection dimension of 128 and 256, leading to leaf nodes with dimensions 32 (same as baseline) in two cases and 16 (half of baseline) in one case. We discuss further in the Sec. 2.4.2.

2.4.1 Qualitative Results

Fig. 2.2 shows some example images from the 16 nodes obtained with a tree of depth 4. The clusters shown correspond to the ordering of the leaf nodes at the bottom, *i.e.* every odd cluster and its next neighbor were grouped together in the previous level in the tree and so on. We can note how similar faces are grouped together successively in the different levels of the tree. Cluster 1–12 are predominantly male faces, cluster 13–16 are females. Cluster 15 seems to specialize to females with bangs (hair over the forehead) and 14 on short hair and smiling females. Cluster 2 seems to have bald (or with very little hair) males who wear glasses while cluster 11 has males with smiling faces. With such semantically interpretable visual qualitative results, we conclude that the method seems to perform an attribute-based clustering.

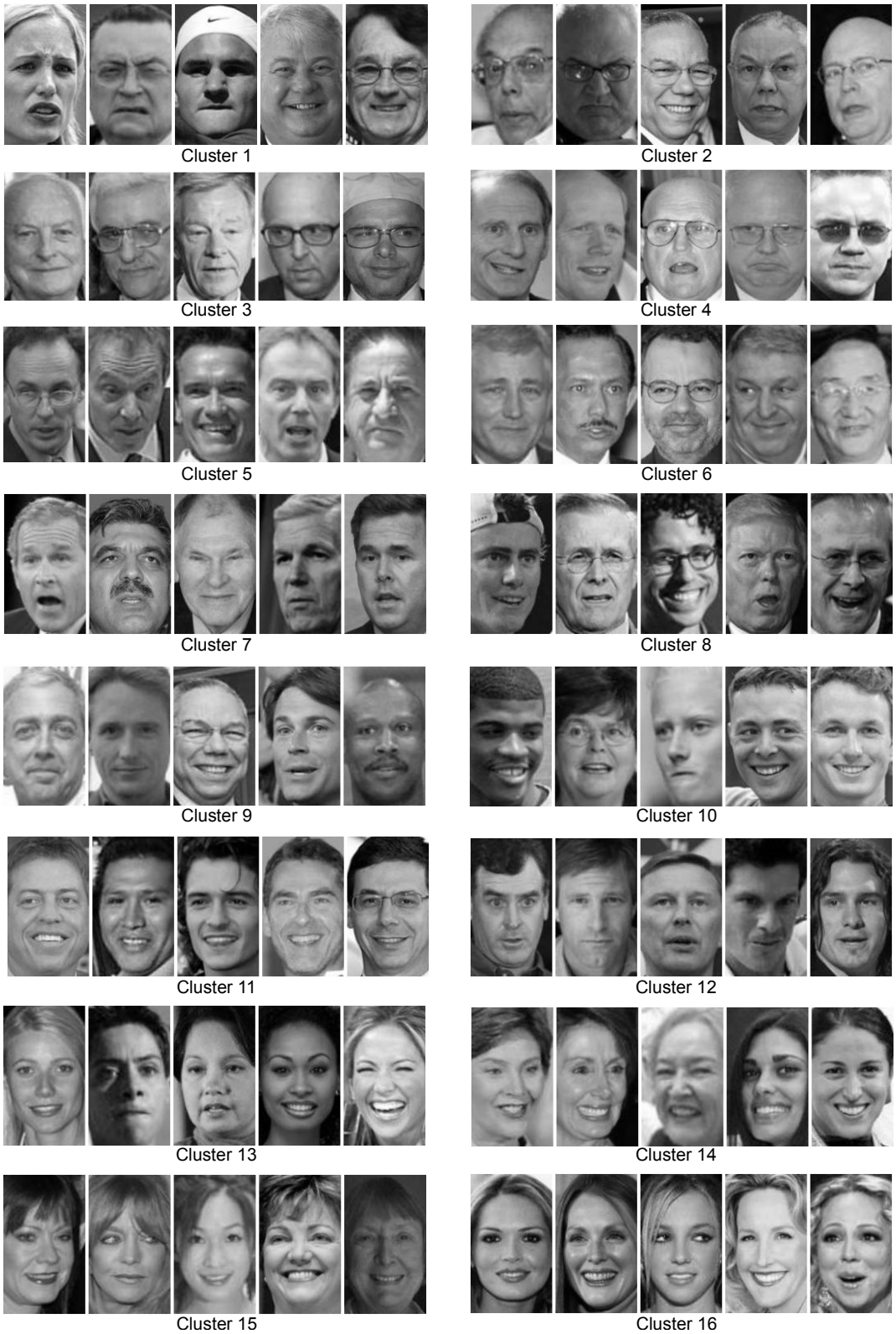


Figure 2.2: Visualization the clustering obtained at leaf nodes for a tree of depth 4. The clusters are ordered from left to right and top to bottom, i.e. *top eight (bottom eight) clusters together form the left (right) node at the first split. Images are randomly selected.*

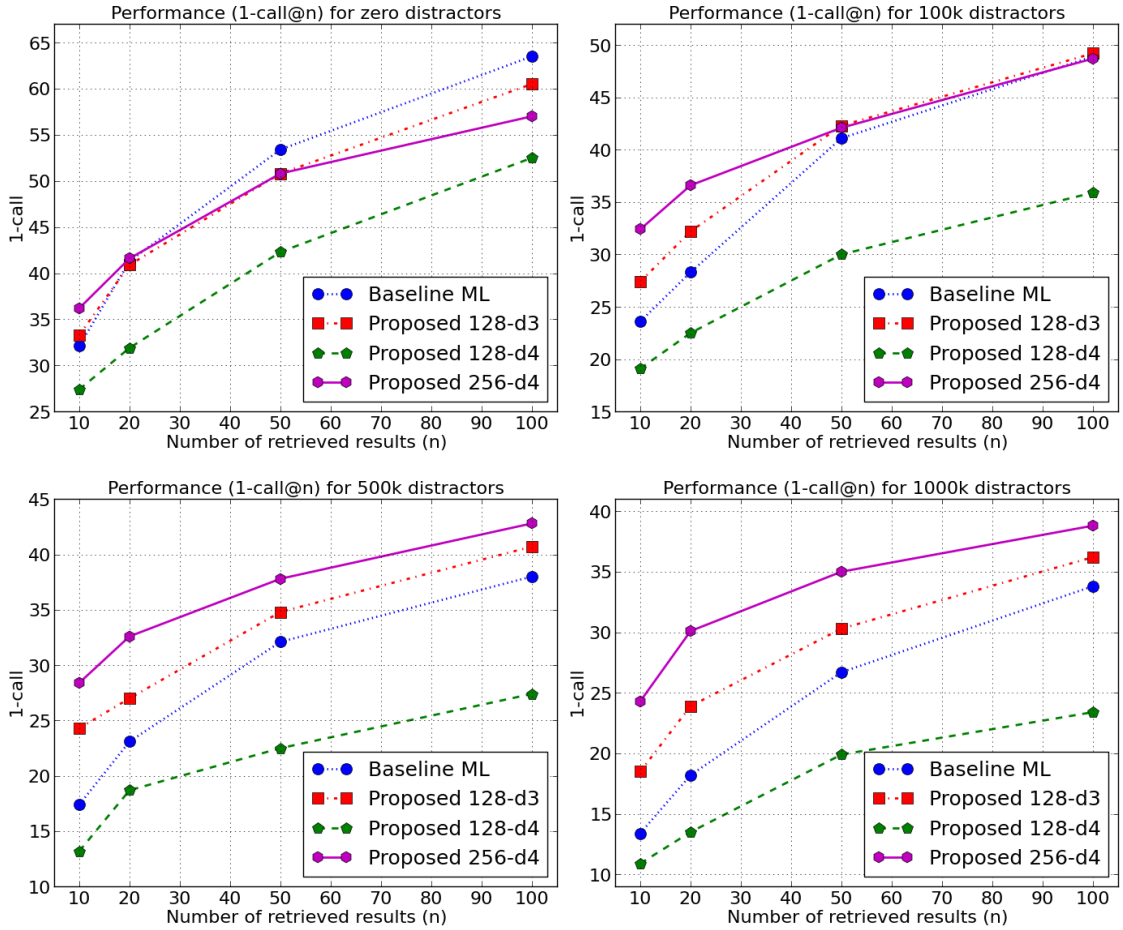


Figure 2.3: The performance of the baseline method and that of the proposed method for three different combinations of parameters (starting projection dimension and tree depth) for different numbers of distractors (0, 100k, 500k and 1m) at different operating points.

2.4.2 Quantitative Results

Fig. 2.3 shows the performances of the baseline *vs.* the proposed method for three different configurations of (i) starting projection dimension 128 with tree depth 3, denoted '128-d3', (ii) starting projection dimension 128 with tree depth 4, denoted '128-d4', and (iii) starting projection dimension 256 with tree depth 4, denoted '256-d4'.

We note that the different configurations of the proposed method give different time complexities. The 128-d3 and 256-d4 trees have leaf node projection dimensions of 32 (same as baseline) with 4 and 8 leaf nodes respectively while the 128-d4 tree has a projection dimension of 16 with 8 nodes. The time complexity for the proposed method depends on (i) projection and Euclidean distance computation with two centroids at non-leaf nodes (repeated $(h-1)$ times, where h is the height of the tree) and (ii) projection and Euclidean distance computation with all the database vectors in leaf nodes. The leaf nodes have about the same number of database vectors and hence a tree with same leaf

node projection dimension (of 32) as baseline but with 4 (8) nodes is expected to be $4\times$ ($8\times$) faster than baseline as the bottleneck in large-scale scenario is the computation of Euclidean distances with a large number of (compressed) database vectors.

We observe that as more and more distractors are added the proposed method performs better. In the presence of large number of distractors, 100 nearest neighbor are expected to lie in a smaller region around the query points and hence an explanation for the better performance of the method could be that it is better adapted to local neighborhood. In the zero distractor case, we observe that the proposed method is better in the case of small n, *i.e.* it is able to do relatively better retrieval when smaller neighborhoods are considered, while the baseline performs better when n is large and hence larger neighborhoods are considered. The success of the method in the presence of a large number of disctractors underlines the need for locally adapted metrics for identity based face retrieval, especially in a large scale scenario.

Time complexity. The proposed method is expected to be faster in the large scale setting where the number of vectors in the database is greater than the feature dimension. In that case the cost of projecting the query becomes negligible compared to the cost of computing the nearest neighbors in the projected space. Assuming the database vectors uniformly occupy the leaf nodes, a tree with N leaves is then expected to give an N fold speed-up. We carried out all our experiments on a computer with Intel Xeon 2.8 GHz CPU running linux. Empirically we obtain speedups of about $2.8\times$, $5.9\times$ and $10.2\times$ for trees with $4, 8$ and 16 nodes respectively, with our unoptimized Python implementation for the experiments with one million distractors, with all computations being timed with data in RAM.

2.5 Conclusions

We presented a method for accurate and efficient identity based face retrieval, which relies on a hierarchical organization of the face database. The method is motivated by the recent works on local learning of discriminative decision boundaries and of metrics, and works based on attributes. We showed quantitatively that organizing faces hierarchically, with automatically learned hierarchy, leads to an attribute based clustering of faces. Further, we showed quantitatively that the method is capable of better retrieval at a better time complexity compared to the baseline method in large-scale setting.

Chapter 3

Multi-task Metric Learning

Contents

3.1 Introduction

In this chapter we address the problem of face retrieval in large scale scenario as we did in our previous work (Chapter 2). In our previous work, our primary focus was efficiency and our secondary focus was accuracy. In this chapter, we propose a novel multi-task metric learning method which allows us to learn the parameters minimizing the objective functions from training examples of several related tasks at a time. We address the important problems of identity based facial analysis such as age and expression variation as auxiliary tasks.

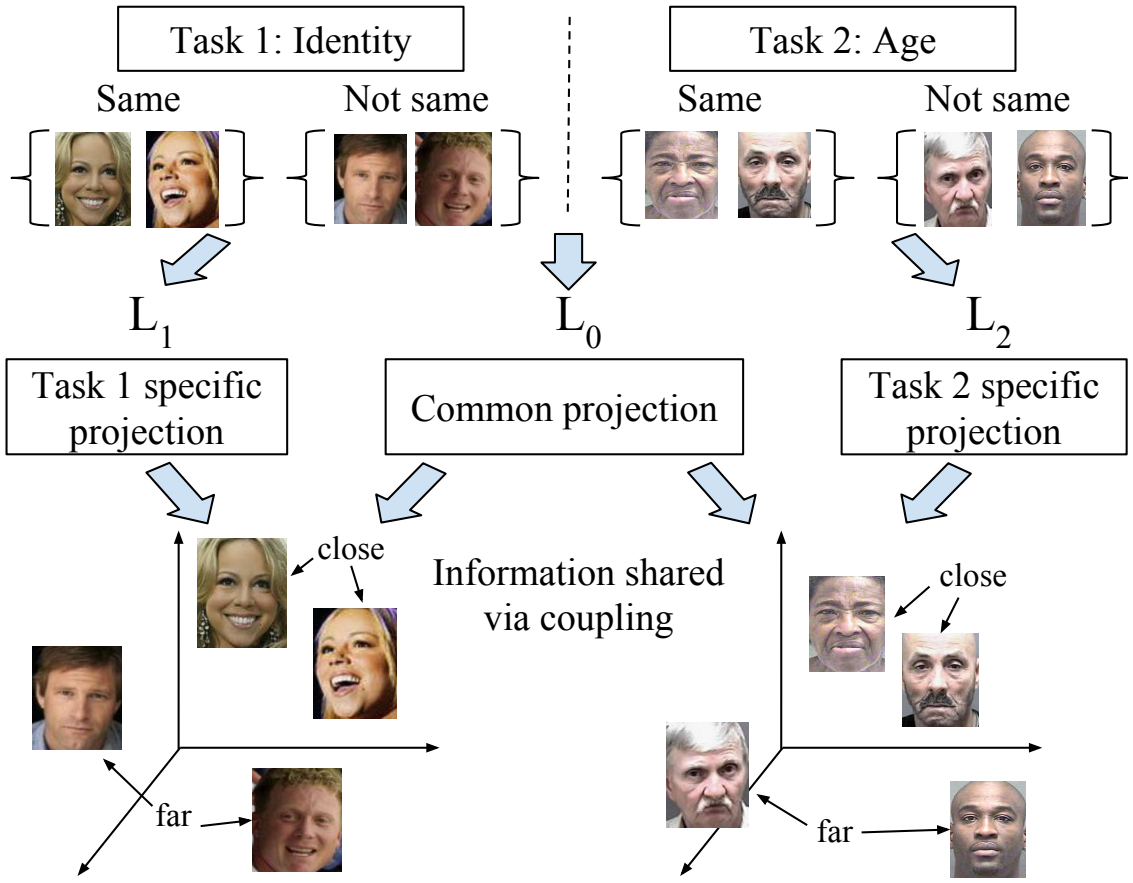


Figure 3.1: Illustration of the proposed method. We propose a multi-task metric learning method which learns a distance function as a projection into a low dimensional Euclidean space, from pairwise (dis-)similarity constraints. It learns two types of projections jointly: (i) a common projection shared by all the tasks and (ii) task related specific projections. The final projection for each task is given by a combination of the common projection and the task specific projection. By coupling the projections and learning them jointly, the information shared between the related tasks can lead to improved performance.

Many computer vision algorithms heavily rely on a distance function over image signatures and their performance strongly depends on the quality of the metric. As mentioned in our previous chapters, Metric Learning (ML) *i.e.* learning an optimal distance function for a given task, using annotated training data, is in such cases, a key to good performance. Hence, ML has been a very active topic of interest in the machine learning community and has been widely used in many computer vision algorithms for image annotation [52], person re-identification [10] or face matching [54], to mention a few of them.

Similar to our previous work (Ch. 2), this work focuses on the task of face matching *i.e.* comparing images of two faces with respect to different criteria such as identity, expression or age. More precisely, the task is to retrieve faces similar to a query, according to the given criteria (*e.g.* identity) and rank them using their distances to the query.

One key contribution of this work is the introduction of a cross-dataset multi-task ML approach. The main advantage of multi-task ML is leveraging the performance of single task ML by combining data coming from different but related tasks. While many recent works on classification have shown that learning metrics for related tasks together using multi-task learning approaches can lead to improvements in performance [7, 20, 84, 91, 110, 158], most of earlier works on face matching are based on a single task. In addition, there are only a few works on multi-task ML [100, 142, 153], with most of the multi-task approaches being focussed on multi-task classification. In addition, the previous multi-task ML methods have been shown to work on the same dataset but not on cross dataset problems. Finally, none of the mentioned approaches have been shown to be scalable to millions of images with features of thousands of dimensions.

The goal of our work is hence to develop a scalable multi-task ML method, using linear embeddings for dimensionality reduction, able to leverage related tasks from heterogeneous datasets/sources of faces. Such challenging multi-task heterogeneous dataset setting, while being a very practical setting, has received almost negligible attention in the literature. Towards that goal, here we present a novel Coupled Projection multi-task Metric Learning method (CP-mtML) for learning better distance metrics for a main task by utilizing additional data from related auxiliary tasks. The method works with pairwise supervision of similar and dissimilar faces – in terms of different aspects *e.g.* identity, age and expression – and does not require exhaustive annotation with presence or absence of classes for all images. We pose the metric learning task as the one of learning coupled low dimensional projections, one for each task, where the final distance is given by the Euclidean distance in the respective projection spaces.

The projections are coupled with each other by enforcing them to be a combination of a common projection and a task specific one. The common projection is expected to capture the commonalities in the different tasks, while the task specific components are expected to specialize to the specificities of the corresponding tasks. The projections are jointly learned using, at the same time, training data from different datasets containing different tasks.

The proposed approach is experimentally validated with challenging publicly available datasets for facial analysis based on identity, age and expression. The task of semantic face retrieval is evaluated in a large scale setting, *i.e.* in the presence of order of millions of distractors, and compared with challenging baselines based on state-of-the-art unsupervised and supervised projection learning methods. The proposed model consistently improves over the baselines. The experimental section also provides qualitative results visually demonstrating the improvement of the method over the most challenging baselines.

3.2 Related Work

As said in the introduction, because of its key role in many problems, ML has received lot of attention in the literature. The reader can refer to [12, 79] for comprehensive surveys on ML approaches in general. Among the possible classes of distances, the Mahalanobis-like one is certainly the most widely studied [93, 112, 144, 150] and has been very successful in variety of face matching tasks [16, 52, 54, 119].

The various Mahalanobis-like methods differ in their objective functions which are themselves related to the type of constraints provided by the training data. The constraints can be given at class level (*i.e.* same-class vectors have to be close from one another after projection) [112], under the form of triplet constraints *i.e.* $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ with $\mathbf{x}_i$ relatively closer to $\mathbf{x}_j$ compared to $\mathbf{x}_k$ [144], or finally by pairwise constraints $(\mathbf{x}_i, \mathbf{x}_j, y_{ij})$ such that $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar (dissimilar) if $y_{ij} = +1$ ($y_{ij} = -1$) [93, 119].

While the above mentioned works considered only a single task, multi-task ML has recently been shown to be advantageous, allowing to learn the metrics for several related tasks jointly [100, 152, 153]. Multi-task Large Margin Nearest Neighbor (mt-LMNN) [100], which is an extension of the (single task) LMNN method [144], was one of the earliest multi-task ML methods. Given T related tasks, mt-LMNN learns $T+1$ Mahlanobis-like metrics parametrized by matrices $M_0, \{M_t\}_{t=1}^{T}$. M_0 encodes the general information common to all tasks while M_t's encode the task specific information. Since a full rank matrix is learned, the method scales poorly with feature dimensions. Pre-processing with unsupervised compression techniques such as PCA is usually required, which potentially leads to loss of information beforehand. Similarly, Wang *et al.* [142] proposed a multi-feature multi-task learning approach inspired by mt-LMNN. In general, mt-LMNN suffers from overfitting. To overcome overfitting, Yang *et al.* [152] proposed a regularizer based on Bregman matrix divergence [35]. In contrast with these works, Yang *et al.* [153] proposed a different but related approach aiming at learning projection matrices $L_t \in \mathbb{R}^{d \times D}$ with $d \ll D$. They factorized these matrices as $L_t = R_t^\top L_0$, where L_0 is common transformation matrix for all the tasks and R_t are task specific matrices. Their method is an extension of the Large Margin Component Analysis (LMCA) [134]. It is important to note that LMCA requires k-nearest neighbors for every classes in their objective function, and hence does not allow to handle tasks in which only pairwise (dis-)similarity constraints are available. Furthermore, computing the k-nearest neighbors is computationally expensive.

In contrast to the works exploiting related tasks, Romera-Paredes *et al.* [110] proposed a multitask learning method which utilises a set of unrelated tasks, enforcing via constraints that these tasks must not share any common structure. Similarly, Du *et al.* [40] used age verification as an auxiliary task to select discriminative features for face verification. They use the auxiliary task to remove age sensitive features, with feature interaction encour-

aged via an orthogonal regularization. Other works such as [72, 88, 108] discourage the sharing of features between the unrelated set of tasks.

The application considered in this work is similar to the one we presented in previous Chapter, *i.e.* face retrieval, requires encoding face images by visual descriptors. This is another problem, widely addressed by the literature. Many different and successful face features have been proposed such as [65, 98, 116, 132]. In the present work, we use signatures based on (i) Local Binary Patterns (LBP) [98] which are very fast to compute and have had a lot of success in face and texture recognition, and (ii) Convolutional Neural Networks (CNN) [78] which have been shown to be very effective for face matching [130]. The computation of face signatures is usually done after cropping and normalizing the regions of the images corresponding to the faces. We do it by first locating face landmarks using the approach of Cao et al. [19].

3.3 Approach

As stated in the introduction, the proposed method aims at jointly learning Mahalanobis-like distances for T different but related tasks, using positive and negative pairs from the different tasks. The motivation is to exploit the relations between the tasks and potentially improve performance. In such a case, the distance metric between vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$ can be written as

$$d^2_{M_t}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M_t (\mathbf{x}_i - \mathbf{x}_j) \tag{3.1}$$

where $M_t \in \mathbb{R}^{D\times D}$ is a task specific parameter matrix (in the following, subscript t denotes task t). To be a valid metric, M must be positive semi-definite and hence can be factorized as $M = L^\top L$. Following [93, 144] we decompose M as the square of a *low rank* matrix $L \in \mathbb{R}^{d\times D}$, with $\mathrm{rank}(L) \leq d \ll D$. This has the advantage that the distance metric can now be seen as a projection to a Euclidean space of dimension $d \ll D$ *i.e.*

$$d^2_{L_t}(\mathbf{x}_i, \mathbf{x}_j) = \|L_t\mathbf{x}_i - L_t\mathbf{x}_j\|^2, \tag{3.2}$$

thus resulting in a discriminative task-adaptive compression of the data. However, it has the drawback that the optimization problem becomes non-convex in L $\forall d < D$, even if it was convex in M [144]. Nonetheless, it has been observed that even if convergence to global maximum is not guaranteed anymore, the optimization of this cost function is usually not an issue and, in practice, very good results can be obtained [54, 93].

We consider an unconstrained setting with diverse but related tasks, coming from possibly different heterogenous datasets. Training data consists of sets of annotated positive and negative pairs from the different task related training sets, denoted as $\mathcal{T}_t = \{(\mathbf{x}_i, \mathbf{x}_j, y_{ij})\} \subset \mathbb{R}^D \times \mathbb{R}^D \times \{-1, +1\}$. In the case of face matching, $\mathbf{x}_i$ and $\mathbf{x}_j$ are the face

Algorithm 3.1 SGD for proposed CP-mtML

Given: $\{\mathcal{T}_t | t = 1, \ldots, T\}, \eta_0, \eta$
Initialize: $b_t = 1$, $L_i \leftarrow \text{wpca}(\mathcal{T}_i), L_0 \leftarrow L_1$
for all $i = 0, \ldots,$`niters`-1 **do**
  for all $t = 0, \ldots, T-1$ **do**
    if $\text{mod}(i, T) == t$ **then**
      Randomly sample $(\mathbf{x}_i, \mathbf{x}_j, y_{ij}) \in \mathcal{T}_t$
      Compute $d_t^2(\mathbf{x}_i, \mathbf{x}_j)$ using Eq. 3.3
      if $y_{ij}(b_t - d_t^2(\mathbf{x}_i, \mathbf{x}_j)) < 1$ **then**
        $L_0 \leftarrow L_0 - \eta_0 y_{ij} L_0 (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$
        $L_t \leftarrow L_t - \eta y_{ij} L_t (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$
        $b_t \leftarrow b_t + 0.1 \times \eta y_{ij}$
      end if
    end if
  end for
end for

signatures while $y_{ij} = +1$ (-1) indicates that the faces are similar (dissimilar) for the considered task *e.g.* they are of the same person (for identity retrieval) or they are of the same age (for age retrieval) or they both are smiling (for expression retrieval).

The main challenge here is to exploit the common information between the tasks *e.g.* learning for age matching might rely on some structure which is also beneficial for identity matching. Such structures may or may not exist, as not only the tasks but also the datasets themselves are different.

Towards this goal, we propose to couple the projections as follows: we define a generic global projection L_0 which is common for all the tasks, and, in addition, we introduce T additional task-specific projections $\{L_t | t = 1, \ldots, T\}$. The distance metric for task t is then given as

$$\begin{aligned} d_t^2(\mathbf{x}_i, \mathbf{x}_j) &= d_{L_0}^2(\mathbf{x}_i, \mathbf{x}_j) + d_{L_t}^2(\mathbf{x}_i, \mathbf{x}_j) \\ &= \|L_0\mathbf{x}_i - L_0\mathbf{x}_j\|^2 + \|L_t\mathbf{x}_i - L_t\mathbf{x}_j\|^2. \end{aligned} \tag{3.3}$$

With this definition of d_t we learn the projections $\{L_0, L_1, \ldots, L_t\}$ *jointly* for all the tasks.

Learning the parameters of our CP-mtML model, *i.e.* the projection matrices $\{L_0, L_1, \ldots, L_t\}$, is done by minimizing the total pairwise hinge loss given by:

$$\underset{L_0, \{L_t, b_t\}_{t=1}^T}{\text{argmin}} \quad \sum_{t=1}^{T} \sum_{\mathcal{T}_t} [1 - y_{ij}(b_t - d_t^2(\mathbf{x}_i, \mathbf{x}_j))]_+, \tag{3.4}$$

with $[a]_+ = \max(0, a)$, $b \in \mathbb{R}$ being the bias, for all training pairs from all tasks. We optimize this function jointly *w.r.t.* all the projections, ensuring information sharing between the different tasks.

In practice, stochastic gradient descent (SGD) is used for doing this optimization. In each iteration, we randomly pick a pair of images from a task, project them in (i) the common and (ii) the corresponding task specific spaces and then compute the square of the Euclidean distance between image descriptors in the respective sub-spaces. If the sum of distances violates the true (dis-)similarity constraint, we update both matrices. To update the matrices, we use the closed-form expression of the partial derivatives of the distance function d_t *w.r.t.* L_0, L_t, given by

$$\frac{\partial d_t^2(\mathbf{x}_i, \mathbf{x}_j)}{\partial L_k} = L_k(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \forall k = 0, \ldots, T \tag{3.5}$$

Alg. 3.1 summarizes this learning procedure.

The learning rates of the different projections are set as explained in the following. η_0 and η adapts the learning rate of common projection matrix and task specific matrix respectively. Regarding the update of the common projection matrix, we can note that the update is done for every violating training example of every task, while other projection matrices are updated much less frequently. Based on this observation, the learning rate for task specific projection matrices is set to a common value denoted as η while the learning rate for the common projection matrix, denoted as η_0, is set as a fractional multiple of η *i.e.* $\eta_0 = \gamma\eta$, where, $\gamma \in [0, 1]$ is a hyper-parameter of the model. The biases b_t are task specific and are the thresholds on the distances separating positive and negative pairs.

Advantage over mt-LMNN [100]

The proposed distance function (Eq. 3.3) can be rearranged and written as $d_t^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top (L_0^\top L_0 + L_t^\top L_t)(\mathbf{x}_i - \mathbf{x}_j)$ and thus bears resemblance to the distances learned with mt-LMNN [100], where $d_t^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top (M_0 + M_t)(\mathbf{x}_i - \mathbf{x}_j)$. However, the proposed model as well as the learning procedure are significantly different from [100]. First, the objective function of mt-LMNN is based on triplets (while our is based on pairs) *i.e.* after projection a vector should be closer to another vector of the same class than to a vector of a different class. The learning procedure of mt-LMNN requires triplets which is in general more difficult to collect and annotate than pairs. Second, despite the fact that mt-LMNN leads to a semidefinite program which is convex, the proposed model has many practical advantages. Since a low rank projection is learnt, there is no need for an explicit regularization as limiting the rank acts as a regularizer. Another advantage is that the low dimensional projections lead to a discriminative task-adaptive compression, which helps us do very efficient retrieval. Third, the proposed SGD based learning algorithm is highly scalable and can work with tens of thousands of examples in thousands of dimensional spaces, without any compression/pre-processing of the features. Finally, another big advantage of our approach is that it can work in an online setting where the data streams with time.

3.4 Experimental Results

We now report the experiments we conducted to validate the proposed method for the task of face retrieval based on traits which can be inferred from faces, including identity, age and expressions. Such a task constitutes an important application domain of face based visual analysis methods. They find application in security and surveillance systems as well as searching large human centered image collections. In our experiments we focus on the two main tasks of identity and age based face retrieval. For the former, we use age and expressions prediction tasks as auxiliary tasks while for the later, we use identity prediction as the auxiliary task. We also evaluate identity based retrieval at a very large scale, by adding a million of distractor faces collected independently from the web.

We now give details of the datasets we used for the evaluation, followed by the features and implementation details and then discuss the results we obtain.

CASIA Web [154] dataset consists of 494,414 images with weak annotations for 10,575 identities. We use this dataset to train Convolutional Neural Network (CNN) for faces.

Labeled Faces in the Wild (LFW) [64] is a standard benchmark for faces, with more than 13,000 images and around 5,000 identities.

MORPH(II) [109] is a benchmark dataset for age estimation. It has around 55,000 images annotated with both age and identity. There are around 13,000 identities, with an average of 4 images per person, each at different ages. We use a subset of around 13,000 images for our experiment. We use this dataset for age matching across identities and hence randomly subsample it and select one image per identity.

FACES [43] is a dataset of facial expressions with 2052 images of 171 identities. Each identity has 6 different expressions (neutral, happy, angry, in fear, disgusted, and sad) with 2 images of each. Here again, we sample one image from each of the expression of every person, and carefully avoid identity based pairings.

SECULAR [16] is a dataset having one million face images extracted from Flickr. These are randomly crawled images and these images are not biased to any of the tasks or datasets. We use these images as distractors during retrieval.

3.4.1 Implementation details

All our experiments are done with grayscale images. The CNN model (details below) is trained with normalized images of CASIA dataset. We use Viola and Jones [140] face detector for other datasets. For detecting facial key points and aligning the faces, we use the publicly available implementation[1] of the facial keypoints detector of [19]. Faces are encoded using the following two features.

[1] https://github.com/soundsilence/FaceAlignment

Local Binary Patterns (LBP). We use the publicly available `vlfeat` [137] to compute descriptors. We resized the aligned face images to 250×250 and centre cropped to 170×100. We set cell size equal to 10 for a descriptor of dimension 9860.

Convolutional Neural Networks (CNN). We use model trained on CASIA dataset with the architecture of Krizhevsky *et al.* [78] to compute the feature of faces. Before computing the features, the images are normalized similar to CASIA. We use the publicly available Caffe [74] deep learning framework to train the model. The weights of the `fc7` layer are taken as the features (4096 dimensions) and are ℓ_2 normalized. As a reference, our features give a verification rate of 88.4 ± 1.4 on the LFW dataset with unsupervised training setting ($+10\%$ compared to Fisher Vectors (FV) [119]) and 92.9 ± 1.1 with supervised metric learning with heavy compression (4096 dimensions to 32 dimensions) *c.f.* 91.4% for $16\times$ longer FVs.

3.4.2 Compared methods.

We compared with the following three challenging methods for discriminative compression, using the same features, same compressions and same experimental protocol for all methods for a fair comparison.

WPCA has been shown to be very competitive method for facial analysis – even comparable to many supervised methods [65]. We compute the Whitened PCA from randomly sampled subset of training examples from the main task.

Single Task Metric Learning (stML) learns a discriminative low dimensional projection for each of the task independently. In Alg. 3.1, we only have a global projection, with no tasks, *i.e.* $T = 0$, reducing it to single task metric learning which we use as a baseline. This is one of the state-of-art stML methods [119] for face verification.

Metric Learning with Union of Tasks (utML). We also learn a metric with the union of all tasks to verify that we need different metrics for different tasks instead of a global metric. We take all pairwise training data from all tasks and learn a single metric as in stML above.

mtLMNN. We did experiments with publicly available code of [100] but obtained results only slightly better than WPCA and hence do not report them.

mtLMCA. We implemented existing state-of-art multitask metric learning method [153] and have compared its performance with our approach. We will discuss about it along with additional qualitiative results in Section 3.5.

3.4.3 Experimental Protocol

We report results on two semantic face retrieval tasks, (i) identity based face retrieval and (ii) age based face retrieval. We now give the details of the experimental protocol *i.e.* details of metric used, main experiments and how we create the training data for the tasks.

Performance measure. We report the 1-call@Kmetric averaged over all the queries. n-call@$K \in [0, 1]$ is an information retrieval metric [25] which is 1 when at least n of the top K results retrieved are relevant. With $n = 1$, this metric is relevant for evaluating real systems, *e.g.* in security and surveillance applications, where at least one of the top scoring K retrievals should be the person of interest, which can be further validated and used by an actual operator.

Identity based retrieval. We use the LFW as the main dataset for identity based retrieval experiments and MORPH (for age matching) and FACES (for expressions matching) as the auxiliary datasets. We use $10,000$ (positive and negative) training pairs from LFW, disjoint from the query images. For auxillary tasks, of expression and age matching, we randomly sample $40,000$ positive and negative pairs, each. This setting is used to demonstrate performance improvements, when the data available for auxiliary task is more than that for the main task. To compare our identity retrieval performance with existing state-of-art rank boosting metric learning [94], we randomly sampled $25,000$ positive and negative pairs (*c.f.* $\sim 32,000$ by [94]) and take the same sets of constraints as before from auxiliary tasks.

Following Bhattarai *et al.* [16], we choose one random image from the identities which have more than five images, as query images and the rest as training images. This gives us 423 query images in total. We use these images to do Euclidean distance, in the projection space, based nearest neighbor retrieval from the rest of the images, one by one. The non-query images are used to make identity based positive and negative pairs for the main task. We use two auxiliary tasks, (i) age matching using MORPH and (ii) expressions matching using FACES.

Age based retrieval. We use the MORPH dataset as the main dataset and the LFW dataset as the auxiliary dataset. We randomly split the dataset into two disjoint parts as train+validation and test sets. In the test set, one image from each age class is taken as the probe query while the rest make the gallery set for retrieval. We take $10,000$ age pairs and $30,000$ of identity pairs.

Large scale retrieval with 1M distractors. We use the SECULAR dataset for distractors. We make the assumption that, as these faces are crawled from Flickr accounts of randomly selected common users, they do not have any identity present in LFW, which is a dataset of famous people. With this assumption, we can use these as distractors for the large scale identity based retrieval task and report performances with the annotations on the main

Projection	$K = 2$	5	10	20
L_0	30.3	38.1	43.3	51.8
L_1	35.0	46.6	55.8	64.8
L_2	4.5	7.6	10.4	13.0
$L_0 + L_1$	**43.5**	**55.6**	**63.6**	**69.5**

Table 3.1: Performance (1-call@K) of different projections matrices learned with proposed CP-mtML (LBP features, $d = 64$) for identity retrieval with auxiliary task of expression matching.

dataset, since all of the distractors will be negatives. However, we can not make the same assumption about age and hence we do not use distractors for age retrieval experiments.

Parameter settings. We choose the values for the parameters (η, η_0, `niters`) by splitting the train set into two parts and training on one and validating on the other *i.e.* these sets were disjoint from all of the test sets used in the experiments.

3.4.4 Quantitative Results

We now present the quantitative results of our experiments. We first evaluate the contributions of the different projections learnt, *i.e.* the common projection L_0 and the task specific projection L_t, in terms of performance on the main task. We then show the performance of the proposed CP-mtML *w.r.t.* the compared methods on the two experiments on (i) identity based and (ii) age based face retrieval. We mention the auxiliary task in brackets *e.g.* CP-mtML (expr) means that the auxiliary task was expression matching, with the main task being clear from context.

Contributions of projections. Tab. 3.1 gives the performance of the different projections for the task of identity based retrieval task with expression matching as the auxiliary task. We observe an expected trend; the combination of the common projection L_0 with the task specific one L_1 performs the best at 69.5 at $K = 20$. The projection for the auxiliary task L_2 expectedly does comparatively badly at 13.0, as it specializes on the auxiliary task and not on the main task. The projection L_1 specializing on the main task is better than the common projection L_0 (64.8 *vs.* 51.8) while their combination is the best (69.5). The trend was similar for the auxiliary task. This demonstrates that the projection learning follows the expected trend, the global projection captures commonalities and in combination with the task specific projections performs better for the respective tasks.

Identity based retrieval. We evaluate identity based face retrieval with two different features *i.e.* LBP and CNN, both with and without one million distractors. Tab. 3.2 and 3.3 give the performances of the different methods for different values of K (the number of top scoring images considered). First of all we notice the general trend that the performances are increasing with K, which is expected. We see that, both in the presence and absence of distractors, the proposed method performs consistently the best compared to

		No distractors				1M distractors			
Method	Aux	$K=2$	5	10	20	$K=2$	5	10	20
WPCA	n/a	30.0	37.4	43.3	51.3	24.6	28.8	33.8	39.0
stML	n/a	38.1	51.1	60.5	69.3	26.0	37.4	43.3	48.7
utML	expr	31.0	38.1	48.5	57.9	20.3	25.8	31.9	38.5
CP-mtML	expr	**43.5**	**55.6**	**63.6**	**69.5**	**33.1**	**43.3**	**51.1**	**55.3**
utML	age	21.7	31.4	41.1	53.0	12.8	18.9	24.6	31.7
CP-mtML	age	**46.1**	**56.0**	**63.4**	68.3	**35.7**	**43.5**	**47.8**	**52.2**

Table 3.2: Identity based face retrieval performance (1-call@K for different K) with and without distractors with LBP features. Auxiliary task is either Age or Expression matching. Projection dimension, $d = 64$

		No distractors				1M distractors			
Method	Aux	$K=2$	5	10	20	$K=2$	5	10	20
WPCA	n/a	72.1	80.4	83.7	89.1	65.2	72.1	75.9	78.7
stML	n/a	**76.8**	85.1	89.6	92.0	70.7	78.0	82.0	84.2
utML	expr	73.5	82.3	87.2	90.3	67.1	76.8	79.0	82.0
CP-mtML	expr	**76.8**	**86.5**	**90.3**	**93.4**	**71.2**	**79.7**	**83.2**	**85.3**
utML	age	73.0	82.0	88.2	91.0	68.1	76.1	81.1	82.7
CP-mtML	age	**76.8**	**85.8**	**90.3**	**93.6**	**71.2**	**79.0**	**83.0**	**85.1**

Table 3.3: Identity based face retrieval performance (1-call@K for different K) with and without distractors with CNN features. Auxiliary task is either Age or Expression matching. Projection dimension, $d = 64$

all other methods. In the case of LBP features, the performance gains are slightly more when the auxiliary task is age prediction *e.g.* 46.1 for CP-mtML (age) *vs.* 43.5 for CP-mtML (expr) at $K = 2$, both these values are much better than WPCA and stML (30.0 and 38.1) respectively. Interestingly, when we take all the tasks together and learn only a single projection, *i.e.* utML, it degrades for both age and expression as auxiliary tasks, but more so for age (21.7 *vs.* 31). This happens because the utML projection brings similar age people closer and hence confuses identity more, as age is more likely to be shared compared to expressions which are characteristic of different people. The proposed CP-mtML is not only able to recover this loss but also leverages the extra information from the auxiliary task to improve performance of the main task.

When distractors are added the performances generally go down *e.g.* 68.3 to 52.2 for LBP and 93.6 to 85.1 for CNN with CP-mtML (age). However, even in the presence of

		No distractor			1M distractors		
Method	Aux	$d=32$	64	128	$d=32$	64	128
WPCA	-	34.3	43.3	52.5	23.4	33.8	40.4
stML	-	50.1	60.5	63.6	33.3	43.3	51.3
utML	expr	44.2	48.5	57.4	25.3	31.9	31.9
CP-mtML	expr	**55.6**	**63.6**	**70.2**	**37.6**	**51.1**	**54.6**
utML	age	37.6	41.1	51.5	17.5	24.6	34.0
CP-mtML	age	**52.5**	**63.4**	**69.0**	**34.3**	**47.8**	**53.9**

		No distractor			1M distractors		
Method	Aux	$d=32$	64	128	$d=32$	64	128
WPCA	-	83.9	83.7	85.6	74.5	75.9	75.2
stML	-	88.4	89.6	88.7	80.6	82.0	**81.6**
utML	expr	85.1	87.2	86.3	73.0	79.0	78.3
CP-mtML	expr	**88.7**	**90.3**	**89.4**	**81.3**	**83.2**	81.1
utML	age	85.3	88.2	86.5	76.6	81.1	79.2
CP-mtML	age	88.2	**90.3**	**89.6**	**80.9**	**83.0**	**81.6**

Table 3.4: Identity based face retrieval, 1-call@10 at different projection dimension, d, (left) using LBP and (right) CNN features.

distractors the performance of the proposed CP-mtML is better than all other methods, particularly stML *e.g.* 43.3 for CP-mtML (expr) *vs.* 37.4 for stML at $K = 5$ with LBP and 79.7 for CP-mtML (expr) *vs.* 78.0 for stML with CNN.

The performances of the two different features are quite different. The lightweight unsupervised LBP features perform lower than the more discriminative CNN features, which are trained on large amounts of extra data *e.g.* 86.5 *vs.* 55.6 at $K = 5$ for CP-mtML (expr). The performance gains for the proposed method are larger for LBP compared to CNN features *e.g.* $+4.5$ *vs.* $+1.4$ at $K = 5$ for CP-mtML (expr) *c.f.* stML. While such improvements are modest for CNN features, they are consistent for all the cases. Parallely, the improvements for LBP features are substantial, especially in the presence of distractors *e.g.* $+7.8$ for CP-mtML (expr) *vs.* stML at $K = 10$. While it may seem that using stronger feature should then be preferred over using a stronger model, we note that this may not be always preferable. In a surveillance scenario, for instance, where a camera just records hours of videos and we need to find a specific face after some incident, using time efficient features as a first step for filtering and then using the stronger feature on a sufficiently small set of filtered examples is advantageous. This is highlighted by the time complexities of the features; in practice LBP are much faster than CNN to compute. While CNN features roughly take 450 milliseconds, the LBP features take only a few milliseconds on a 2.5 GHz processor.

Further, Tab. 3.4 presents the 1-call@10 while varying the projection dimension, which is directly proportional to the amount of compression. We observe that all methods gain performance when increasing the projection dimension, however, with diminishing returns. In the presence of one million distractors, CP-mtML (expr) improves by $+13.5$ when going from $d = 32$ to $d = 64$ and $+3.5$ when going from $d = 64$ to $d = 128$ for LBP. The results for larger d were saturated for LBPs with a slight increase. The performance changes with varying d in the presence of distractors for CNN features are more modest. CNN with distractors gets $+1.9$ for $d = 32$ to $d = 64$ and -2.1 for $d = 64$ to $d = 128$ *i.e.* the algorithm starts over-fitting at higher dimensions for the stronger CNN features. As an idea of space complexity, at compression to $d = 32$ dimensional single precision vector per face, storing ten million faces would require one gigabytes of space, after projection. Interestingly, the proposed method is better than stML in all but one case (CNN features with $d = 128$) which is a saturated case anyway.

Tab. 3.5 gives the comparisons (with LBP features and $d = 32$) with MLBoost [94]. At $K = 10$ CP-mtML obtains $61.5, 58.9$ with age and expressions as auxiliary taks, respectively, while the MLBoost method stays at 54.1. Hence the proposed method is better than the results reported in the literature. As said before, we also used the publicly available code of mtLMNN [100]. We obtained results only slightly better than WPCA and hence do not report them.

With the above results we conclude the following. The proposed method effectively lever-

Figure 3.2: The 5 top scoring images (LBP & no distractors) for three queries for the different methods (auxiliary task in brackets). True (resp. False) Positive are marked with a green (resp. red) border (best viewed in color).

		No distractors		1M distractors	
Method	Aux	K=10	20	10	20
MLBoost	n/a	54.1	63.4	34.3	39.5
CP-mtML	expr	**58.9**	**69.5**	**38.1**	**45.6**
CP-mtML	age	**61.5**	**70.7**	**39.7**	**47.8**

Table 3.5: Performance comparison with existing MLBoost [94] (for LBP features and $d = 32$).

ages the additional complementary information in the related tasks of age and expression matchings, for the task of identity based face retrieval. It consistently improves over the unsupervised WPCA, supervised stML which does not use additional tasks and also utML which combines all the data. It is also better than these methods at a range of projection dimensions (*i.e.* compressions), deteriorating only at the saturated case of high dimensions with strong CNN features.

Age based retrieval. Fig. 3.3 presents some results for face retrieval based on age for the different methods, with the auxiliary task being that of identity matching. In this task CP-mtML outperforms all other methods by a significant margin with LBP features. These results are different and interesting from the identity based retrieval experiments above, as they show the limitation of CNN features, learnt on identities, to generalize to other tasks — the performances with LBP features are higher than those with CNN features.

While the trend is similar for LBP features *i.e.* CP-mtML is better than stML, it is reversed for CNN features. With CNN features, stML learns to distinguish between ages when trained with such data, however, CP-mtML ends up being biased, due to its construction, towards identity matching and degrades age retrieval performance when auxiliary task is identity matching. However, the performance of CPmtML with LBP features is much higher than of any of the methods with CNN features.

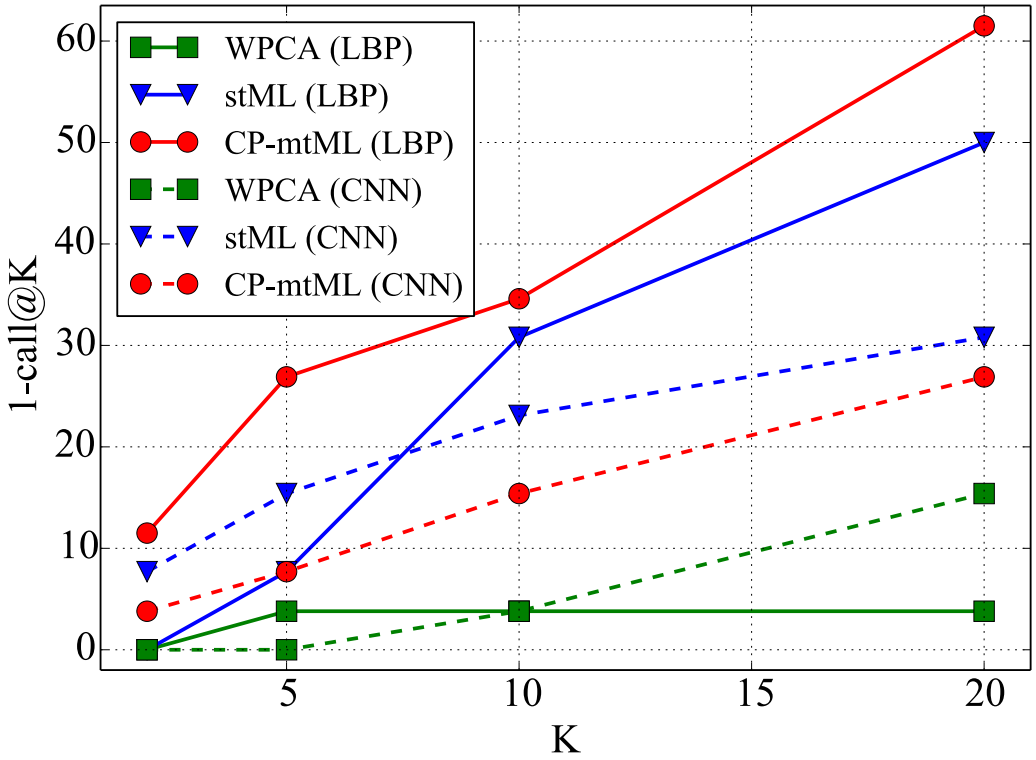


Figure 3.3: Age retrieval performance (1-call@K) for different K with auxiliary task of identity matching. The dimension of projection is $d = 32$

3.4.5 Qualitative results

We now present some qualitative comparisons between the proposed CP-mtML, with age and expression matching as auxiliary tasks, with the competitive stML method. Fig. 3.2 shows the top five retrieved faces for three different queries for stML and the proposed CP-mtML with age and expression matching as auxiliary tasks. The results qualitatively demonstrate the better performance obtained by the proposed method. In the first query (left) all the methods were able to find correct matches in the top five. While stML found two correct matches at ranks 1 and 4, CP-mtML (age) also found two but with improved ranks *i.e.* 1 and 2 and CP-mtML (expression) found three correct matches with ranks $1, 2$ and 5. While the first query was a relatively simple query, *i.e.* frontal face, the other two queries are more challenging due to non-frontal pose and deformations due to expression. We see that stML completely fails in these cases (for $K = 5$) while the proposed CP-mtML is able to retrieve one correct image with ranks $1, 3$ (middle) and $5, 2$ (right) when used with age and expression matching as auxiliary tasks, respectively. It is interesting to note that with challenging pose and expression the appearances of the faces returned by the methods are quite different (right query) which demonstrates that CP-mtML projection differs from that learned by stML.

		No distractors				1M distractors			
Method	Aux	$K=2$	5	10	20	$K=2$	5	10	20
WPCA	n/a	30.0	37.4	43.3	51.3	24.6	28.8	33.8	39.0
stML	n/a	38.1	51.1	60.5	69.3	26.0	37.4	43.3	48.7
utML	expr	31.0	38.1	48.5	57.9	20.3	25.8	31.9	38.5
mtLMCA	expr	29.3	40.7	48.0	61.0	19.9	28.4	34.8	40.0
CP-mtML	expr	**43.5**	**55.6**	**63.6**	**69.5**	**33.1**	**43.3**	**51.1**	**55.3**
utML	age	21.7	31.4	41.1	53.0	12.8	18.9	24.6	31.7
mtLMCA	age	27.4	39.7	50.4	61.0	18.7	24.6	29.8	35.5
CP-mtML	age	**46.1**	**56.0**	**63.4**	68.3	**35.7**	**43.5**	**47.8**	**52.2**

Table 3.6: Identity based face retrieval performance (1-call@K for different K) with and without distractors with LBP features. Auxiliary task is either Age or Expression matching. Projection dimension, $d = 64$

3.5 Additional Results

In this section we present additional both quantiative and qualitative results.

3.5.1 Quantitative Results

In this section, we compare performance of existing state-of-art multitask metric learning method, mtLMCA of Yang *et al.* [153] with the performance of the proposed method and other baselines. In addition to it, we present the in-depth analysis of the proposed algorithm such as its complexity and scalability. We then present the optimization curves of loss functions of our method and mtLMCA.

Comparisons with mtLMCA. We implemented the existing mtLMCA and compare the performance with the proposed method. For mtLMCA, we initialized the the common projection, L_0 and task specific, R_t matrices with identity matrices as explained in this chaper before. Whereas, for rest of the cases, as stated in the Alg. 3.1 with the WPCA.

Tab. 3.6 shows the performance comparison. In comparison with mtLMCA, we observe that the proposed CP-mtML outperforms mtLMCA by a significant margin. We explain it as follows. Without loss of generality consider task 1 (*e.g.* identity matching), the projection by proposed method is given by a common L_0 and a task specific L_1 while that by mtLMCA is given by common L_0 and task specific R_1 . While L_0 , L_1 are both $d \times D$ matrices R_1 is $d \times d$. Hence in CP-mtML there are dD common (across tasks) parameters and dD task specific parameters, while mtLMCA has same dD common parameters but only d^2 task specific parameters. We suspect that with equal number of task specific and common parameters CP-mtML is able to exploit the shared as well as task specific information well while for mtLMCA the small number of task specific parameters are not able to do so effectively e.g. for the specific case of 9860D LBP features projected to 64D, while 50% of the parameters are task specific for CP-mtML, only $64^2/(9860 \times 64) = 0.7\%$

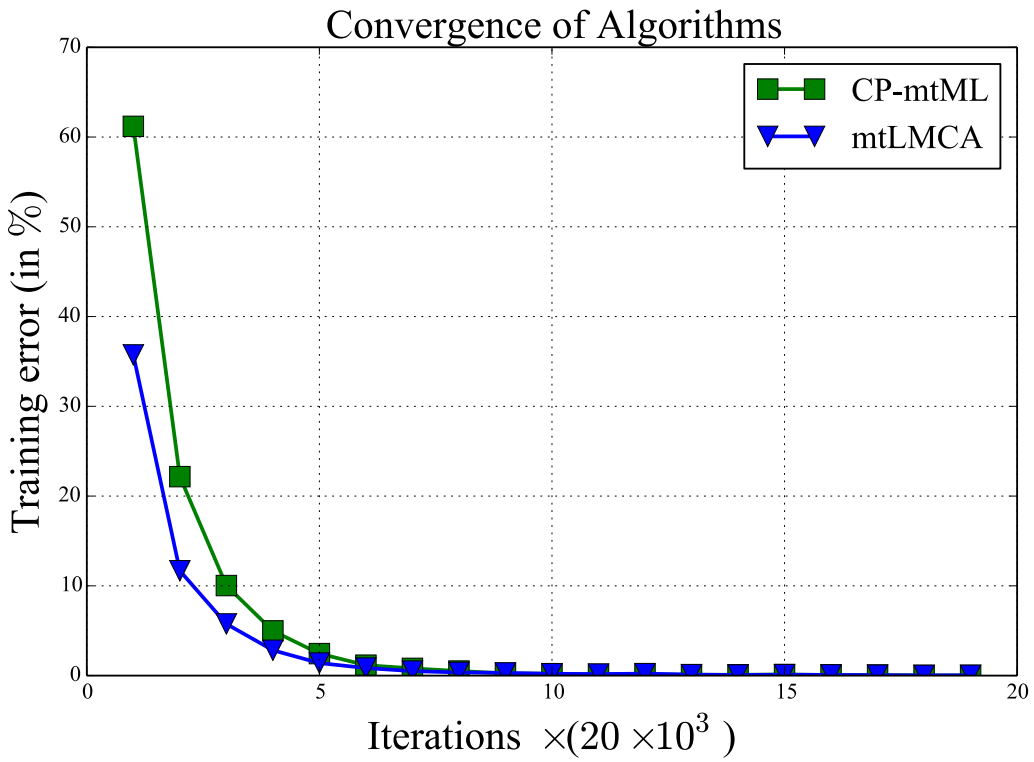


Figure 3.4: Minimization of objective function

are task specific in mtLMCA. In addition to it, we could see this method as utML with a very small fraction of task specific parameters. As mentioned before, utML learns a common projection matrix taking training examples from both the domains. From the performance also, it supports our argument. We can see that the performance of mtLMCA is slightly better than utML. This is due to the small separate task specific parameters in mtLMCA. Our proposed method, CP-mtML is capable of learning large task specific parameters maintaining the same projection dimension as that of other methods, which ultimately gives the improved performance.

Time Complexity and Scalability. CP-mtML is about 2.5× slower to train than stML – specifically it takes 40 minutes to train CP-mtML with 50, 000 training pairs while compressing 9860D LBP features to 64D on a single core of 2.5 Ghz system running Linux. The training time is linear in the number of training examples. As the 64D features are real vectors it takes 256 bytes (with 4 bytes per real) to index one face or about a manageable 1.8 TB to index the current human population of about 7 billion people; hence we claim scalability.

Convergences of Algorithms. Fig. 3.4 shows the convergences of CP-mtML and mtLMCA. From the figure, we see that both the algorithms are converged well.

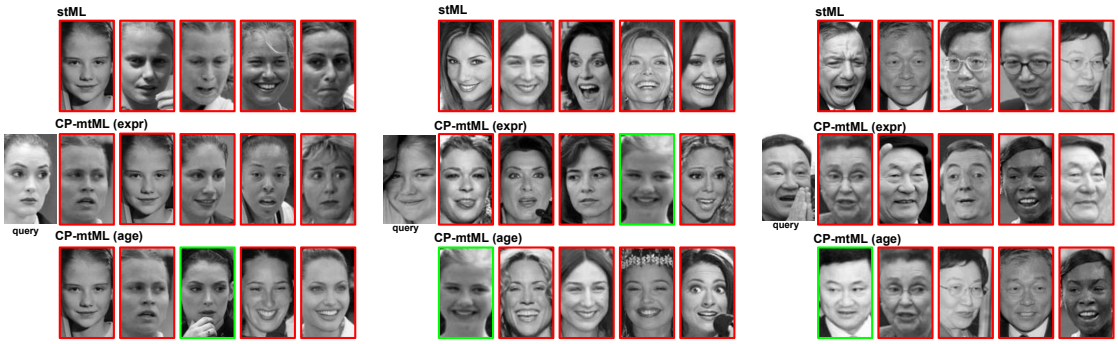


Figure 3.5: ***Sample set of queries for which CP-mtML (age) performs better than CP-mtML (expr) and stML.*** *The 5 top scoring images (LBP & no distractors) for the queries for the different methods. True (resp. false) positives are marked with a green (resp. red) border. Best viewed in color.*

3.5.2 Qualitative Results

We present some more qualitative results to compare the proposed Coupled Projection multi-task Metric Learning (CP-mtML) with the most competitive baseline *i.e.* Single Task Metric Learning (stML). The main task here is that of identity based face retrieval while the auxiliary tasks are expression (expr) and age (age) based matching.

We can make the following observations

(i) Fig. 3.5 shows some queries for which CP-mtML (age) does better than CP-mtML (expr) and stML. The results suggest that adding information based on age matching makes identity matching more robust to high variations due to challenging pose (left) and occlusions (hair and hand in the middle and right examples).

(ii) Fig. 3.6 shows some queries for which CP-mtML (expr) does better than CP-mtML (age) and stML. The results suggest that adding information based on expression matching makes identity matching more robust to challenging expressions.

(iii) Fig. 3.7 shows some queries for which CP-mtML (expr) and CP-mtML (age) do better than stML. These cases are really challenging and the results retrieved by stML, while being sensible, are incorrect. Adding more information based on age and/or expression matching improves results.

(iv) Fig. 3.8 shows some queries for which all three methods do well. These are queries with either neutral expression and frontal pose or with characteristic appearances *e.g.* moustache, baseball cap, glasses, hairstyle *etc.* which occur for the same person in the gallery set as well.

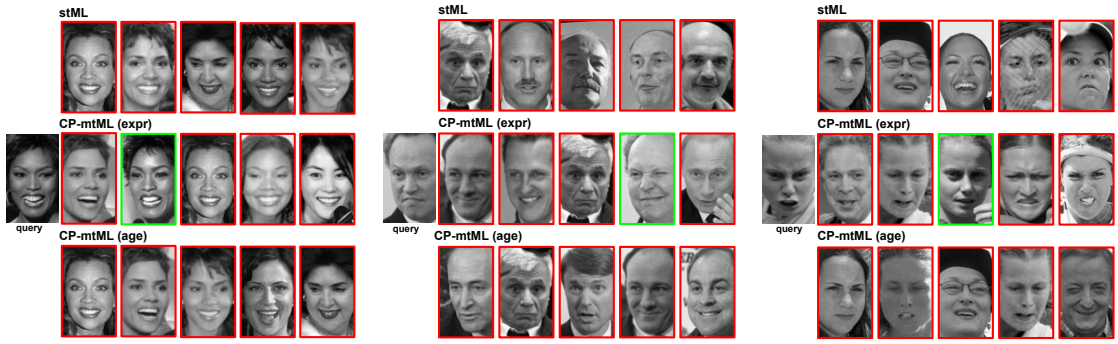


Figure 3.6: ***Sample set of queries for which CP-mtML (expr) performs better than CP-mtML (age) and stML.*** *The 5 top scoring images (LBP & no distractors) for the queries for the different methods. True (resp. false) positives are marked with a green (resp. red) border. Best viewed in color.*

Figure 3.7: ***Sample set of queries for which CP-mtML (expr) and CP-mtML (age) both perform better than stML.*** *The 5 top scoring images (LBP & no distractors) for the queries for the different methods. True (resp. false) positives are marked with a green (resp. red) border. Best viewed in color.*

3.6 Conclusions

We presented a novel Coupled Projection multi-task Metric Learning (CP-mtML) method for leveraging information from related tasks in a metric learning framework. The method factorizes the information into different projections, one global projection shared by all tasks and T task specific projections, one for each task. We proposed a max-margin hinge loss minimization objective based on pairwise constraints between training data. To optimize the objective we use an efficient stochastic gradient based algorithm. We jointly learn all the projections in a holistic framework which leads to sharing of information between the tasks. We validated the proposed method on challenging tasks of identity and age based image retrieval with different auxiliary tasks, expression and age matching for the former and identity matching in the later. We showed that the method improves performance when compared to competitive existing approaches. We analysed the qualitative results, which also supported the improvements obtained by the method.

Figure 3.8: ***Sample set of queries for which all of CP-mtML (expr), CP-mtML (age) and stML perform well.*** *The 5 top scoring images (LBP & no distractors) for the queries for the different methods. True (resp. false) positives are marked with a green (resp. red) border. Best viewed in color.*

Chapter 4

Cross Domain Age Estimation

Contents

4.1 Introduction and related work

In this chapter and in the following chapter (Ch. 5), we present our works on features alignment to address the problem of domain adaptation. In this chapter, we are addresing the problem of bringing data points from different domains into a common subspace, whereas in the Ch. 5, we address the problem of projecting different types of features into the same subspace.

Automatic age estimation from face images has become a popular research problem [60, 26, 121, 133, 21]. It has various important applications such as age specific human-computer interaction [48], business intelligence [114], etc. Previous studies [56, 57, 58, 55] have shown that the rate of ageing among different groups of people is different. This is because, ageing patterns are directly affected by genes, dieting habits, culture, weather, race, gender etc. Thus, it has been more challenging to design an age prediction model which generalizes for people from such different categories. In addition, it has

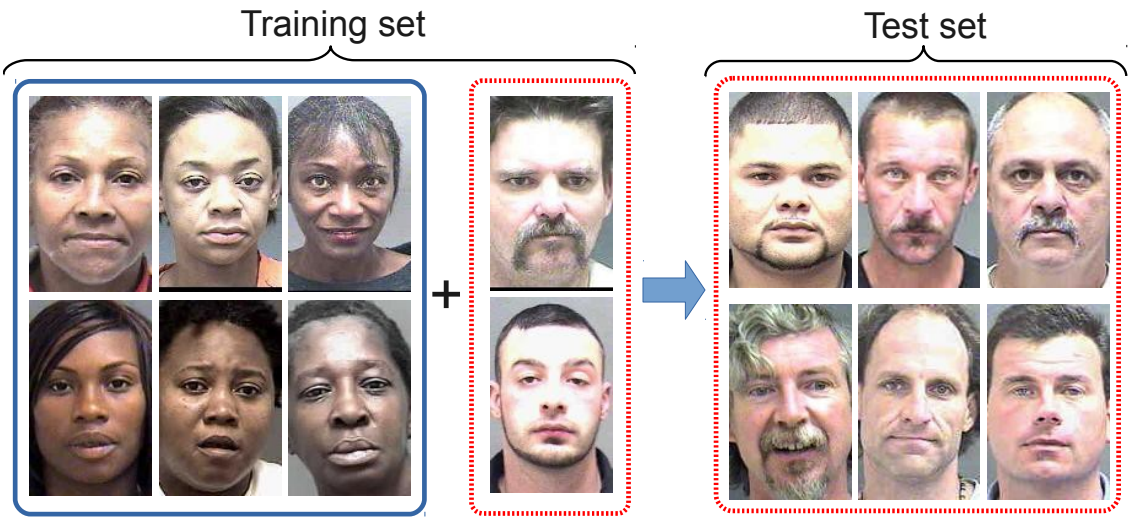


Figure 4.1: Illustration of the proposed setting of cross domain age estimation. The algorithm learns a projection and a regressor jointly, to align source and target face domains and predict ages in the target domain. The training is mainly with source domain examples complemented very few target domain examples, while testing is done on target domain images only. The source and target domains may differ in age range, sex, race etc.

been shown [57] that, training a single model on all different groups together, affect the performance that separate specialized models for different groups can give, due to the differences in ageing patterns.

Training separate model for each and every groups of people has its own limitations. It is difficult, expensive and time consuming to collect and annotate face images. Moreover, due to privacy issues and related concerns, people may not be keen to share their biometrics information such as ages, race *etc.* Thus, it would be ideal to utilise the training examples available for one group of people to improve performance in another group which has a very limited number of training examples. In this chapter we are interested in such a setting, illustrated by Fig. 4.1.

As explained before, we are interested in the problem of estimating age from face images, in a cross-population setting *i.e.* assuming that there are a large number of training examples available in one domain (the source domain) but only a very few ones in another domain (the target domain). We would like to utilise the training examples of the source domain to improve the performance of age estimation on the target domain. This problem was first well posed and addressed by Guo *et al.* [59]. In their approach, they used a variant of LDA (Linear Discriminant Analysis) to learn common projection matrix which align ageing patterns from source and target. However, they need a large number of target instances to learn target domain ageing pattern, which are often not available in practice. Similarly, Alnajar *et al.* [4] proposed a method to do cross expression age estimation. But, the datasets they used for their experiments, FACES and LifeSpan are rather small and does not reflect the situation where abundant training data is available

in the source domain.

This chapter proposes a joint learning method which (i) learns a subspace aligning features from source and target domain and (ii) learns a regressor in this subspace for predicting ages. Our projection matrix learning approach is similar to the metric learning method of Mignon and Jurie and Simonyan *et al.* [93, 119] – the projection matrix is learnt to satisfy sparse pairwise (dis)similar constraints and age prediction based constraints simultaneously. We show empirically that the proposed method is consistently better than several strong baselines including those based on discriminative metric learning. We attain state-of-the-art performance on the largest publicly available age estimation dataset. In the following, we discuss about the proposed method in Sec. 4.2. In Sec. 4.3 we provide the experimental results and in Sec. 4.4 we conclude this chapter.

4.2 Proposed methods

We will now detail the proposed method. We will first introduce Metric Learning (ML) in general and then we will explain how we it can be used for learning a projection to align features from source and target domains. Finally, we will explain the proposed Joint Learning (JL) algorithm.

4.2.1 Metric Learning and its application to cross-domain classification

As we explained in our previous chapters, Metric Learning (ML) has been quite successful in various facial analysis tasks such as Face Recognition [93, 53], Face Retrieval [16] etc. Mahalanobis-like ML can be seen as learning a projection to map high dimensional features into a lower dimensional subspace where the constraints are better satisfied. For a pair of descriptors, $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^{d'}$, ML involves the task of learning a Mahalanobis like metric of the form $D^2_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M(\mathbf{x}_i - \mathbf{x}_j)$, parameterized by positive semi-definite matrix, M. As M is PSD, it can be decomposed as $M = L^\top L$. The problem can then be re-formulated as that of finding a linear subspace, into which features are first mapped and then compared as

$$\begin{aligned} D^2_L(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i - \mathbf{x}_j)^\top L^\top L(\mathbf{x}_i - \mathbf{x}_j) \\ &= ||L\mathbf{x}_i - L\mathbf{x}_j||^2_2 \end{aligned} \tag{4.1}$$

In the present case, we are given a training set of face images represented by their feature vectors and annotated with their ages *i.e.*

$$\mathcal{T} = \{(X, Y) : X \in \mathbb{R}^{d' \times N}, Y \in \mathbb{N}^N\} \tag{4.2}$$

Algorithm 4.1 Joint learning of projection and regressor

Input: (i) Projection matrix, L ; Regressor $\mathbf{w}$, ii) Set of face features $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d' \times N}$, Set of age annotations $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^N$ (ii) Sparse pairwise age annotation $\mathbf{S}, \mathbf{D}$ (iii) maximum iterations `max-iters`, ϵ, α, β, γ, `learn-rate:`r
Output: L, $\mathbf{w}$
while $it <$ `max-iters` **do**
  $\Delta y_i \leftarrow |\mathbf{w}L\mathbf{x}_i - y_i|$
  if $\Delta y_i > \epsilon$ **then**
    $L_{it} \leftarrow L_{it-1} - \beta r \mathbf{w}_{it-1}\mathbf{x}_i^\top$
    $\mathbf{w}_{it} \leftarrow \mathbf{w}_{it-1} - r(\beta L\mathbf{x}_i + \lambda \mathbf{w}_{it-1})$
  end if
  $\Delta y_j \leftarrow |\mathbf{w}L\mathbf{x}_j - y_j|$
  if $\Delta y_j > \epsilon$ **then**
    $L_{it} \leftarrow L_{it-1} - \beta r \mathbf{w}_{it-1}\mathbf{x}_j^\top$
    $\mathbf{w}_{it} \leftarrow \mathbf{w}_{it-1} - r(\beta L\mathbf{x}_j + \lambda \mathbf{w}_{it-1})$
  end if
  $D_L^2(\mathbf{x}_i, \mathbf{x}_j) \leftarrow \|L\mathbf{x}_i - L\mathbf{x}_j\|^2$
  if $y_{ij}(1 - D_L^2(\mathbf{x}_i, \mathbf{x}_j)) < 0.2$ **then**
    $L_{it} \leftarrow L_{(it-1)} - \gamma.r.y_{ij}L_{(it-1)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$
  end if
end while

We construct two other sets from this information, set of *similar* vectors **S** and that of *dissimilar* ones **D**, given by

$$\mathbf{S} = \{(i, j) : |y_i - y_j| \leq \delta\} \tag{4.3}$$

$$\mathbf{D} = \{(i, j) : |y_i - y_j| > \delta\} \tag{4.4}$$

with $\delta = 0$. We are interested in learning a mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^+$ to predict the age of new test faces. We impose pairwise similarity and dissimilarity constraints, in the present case, and formulate the learning similar to the approach of Mignon *et al.* and Simonyan *et al.* [93, 119] *i.e.* optimize the objective function given in Equation 4.6 using stochastic gradient descent. We generate the pairwise constraints from the large number of examples from source domain and a limited number of examples from the target domain. This is similar to the approach of Saenko *et al.* [111], who use ML for cross-domain image classification. It is important to note here that, the pairs they generated were from the examples belonging to two different domains. In [111], after learning projection matrix, training examples are projected into this subspace and classifier is trained in this subspace.

$$\ell_L(\mathbf{x}_i, \mathbf{x}_j, y_{ij}) = \max[0, 1 - y_{ij}(m - D_L^2(\mathbf{x}_i, \mathbf{x}_j))] \tag{4.5}$$

$$\min_L \mathcal{L}(\mathcal{T}, \mathbf{S}, \mathbf{D}; L) = \sum_{\mathbf{S} \cup \mathbf{D}} \ell_L(\mathbf{x}_i, \mathbf{x}_j, y_{ij}) \tag{4.6}$$

4.2.2 Proposed joint learning for cross-domain regression

An immediate extension of the approach of Saenko *et al.* [111] for regression could be similar ML projection followed by regressor learning. The problem with such approach is that it would not directly address the main goal of minimizing the absolute age difference between the ground truth age and predicted age. Moreover, pairwise constraints try to bring images belonging to same age categories together but push away the images belonging to different age categories. They push dissimilar pair away equally *i.e.* without taking into consideration the difference in their ages. For example, two pairs of images with the ages $(25, 26)$ and $(25, 55)$ are equally pushed apart. Unlike classification tasks, it is important to address this issue in regression tasks. Incorporating the regressor while learning projection matrix address this problem by pushing the ages with lesser difference comparatively less farther.

We are thus interested in learning a projection L and a regressor w, in the resulting space, jointly. We propose to minimize the following objective for learning $\mathbf{w}, L$,

$$\begin{aligned}\min_{L,\mathbf{w}} \mathcal{L}(\mathcal{T}, \mathbf{S}, \mathbf{D}; L, \mathbf{w}) = &\frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \beta \sum_k \ell_{\mathbf{w}}(L\mathbf{x}_k, y_k) \\ &+ \gamma \sum_{\mathbf{S}\cup\mathbf{D}} \ell_L(\mathbf{x}_i, \mathbf{x}_j, y_{ij})\end{aligned} \tag{4.7}$$

where, the first term is ℓ^2 regularization on $\mathbf{w}$, $\lambda, \beta, \gamma \in \mathbb{R}$ are free parameters controlling the relative contributions of the different terms, $\ell_{\mathbf{w}}$ is the support vector regression loss which aims to bring the predicted age within $\pm\epsilon \in \mathbb{R}^+$ of the true age, given by:

$$\ell_{\mathbf{w}}(L\mathbf{x}, y) = \max(0, |\mathbf{w}^\top L\mathbf{x} - y| - \epsilon) \tag{4.8}$$

where $\ell_L(\mathbf{x}_i, \mathbf{x}_j, y_{ij})$ is the loss which aims at bringing similar age pairs together while pushing dissimilar age pairs away from each other. In practice, we optimize the objective using a stochastic gradient based solver, which is detailed in Alg. 4.1.

4.3 Experiments

Dataset. We use the largest publicly available dataset for age estimation, the MORPH-II dataset, to evaluate the proposed method. We followed the experimental setup of Guo *et al.* [59] and compared the performance of our method with their method. We computed Local Binary Patterns (LBP) [107] of face images instead of Biologically Inspired Features (BIF) [58] which they used for their experiments. The database contains around 55×10^3 images from different races ('Black', 'White', 'Caucasian', *etc.*) and genders ('Male', 'Female'). Similar to [59], we took randomly sampled subsets of the database for the experiments. We took images from two races 'Black', and 'White', and two genders

'Male', and 'Female'. This subset contains $2,570$ White Female (WF), $7,960$ White Male (WM), $2,570$ Black Female (BF), and $7,960$ Black Male (BM) face images. Each of these categories is called a domain. From each of these domains, 50% of randomly sampled images are used for training and validation purposes and rest 50% are used for testing. We used SVM regressor for predicting ages. The performance is calculated by Mean Absolute Error (MAE). MAE is the mean of absolute difference between the ground truth age and the predicted age.

Face Description. We used Viola and Jones face detector [140] to compute the bounding boxes of faces. These bounding boxes were resized to the size of 250×250. We computed facial landmarks using publicly available state-of-art facial landmark detector [19][1]. With the help of these facial landmarks we align the faces if required. The aligned faces are then centre cropped into the size of 160×100. We then compute local binary patterns (LBP) for each of these images using the publicly available `vlfeat` [137] library. We set cell size is equal to 10 as parameter and obtain signature for each of the images which are of 9280 dimensions. Note however, the proposed method can work with other types of features *e.g.* LQP [65], LHS [116] or Fisher Vectors [118].

4.3.1 Baselines

As a first reference we used the full features without any projection learning and hence without any compression. In addition, we compared with the following competitive baselines.

Unsupervised compression. We used Whitened Principal Components (WPCA) to compress high dimensional LBP to 64 dimensions. For training and testing, these representations are very efficient but suboptimal, as they may remove some discriminative information for age prediction.

Supervised Compression with ML. We used ML to learn compact representation of images which retains some discriminative information. We initialized with WPCA and learned the projection with stochastic gradient descent. This approach not only samples features that are useful for age estimation, but also aligns the features between the source and target domains.

After compressing, and potentially aligning the domains, for all these baselines, we use the publicly available SVR from `scikit-learn` [102] to learn the model on projected features to predict the ages. For all the experiments reported, we chose a linear kernel. We split train set into two halves for cross-validation. We set $\epsilon = 0.1$ and select the C parameter for SVR by cross-validation.

[1] https://github.com/soundsilence/FaceAlignment

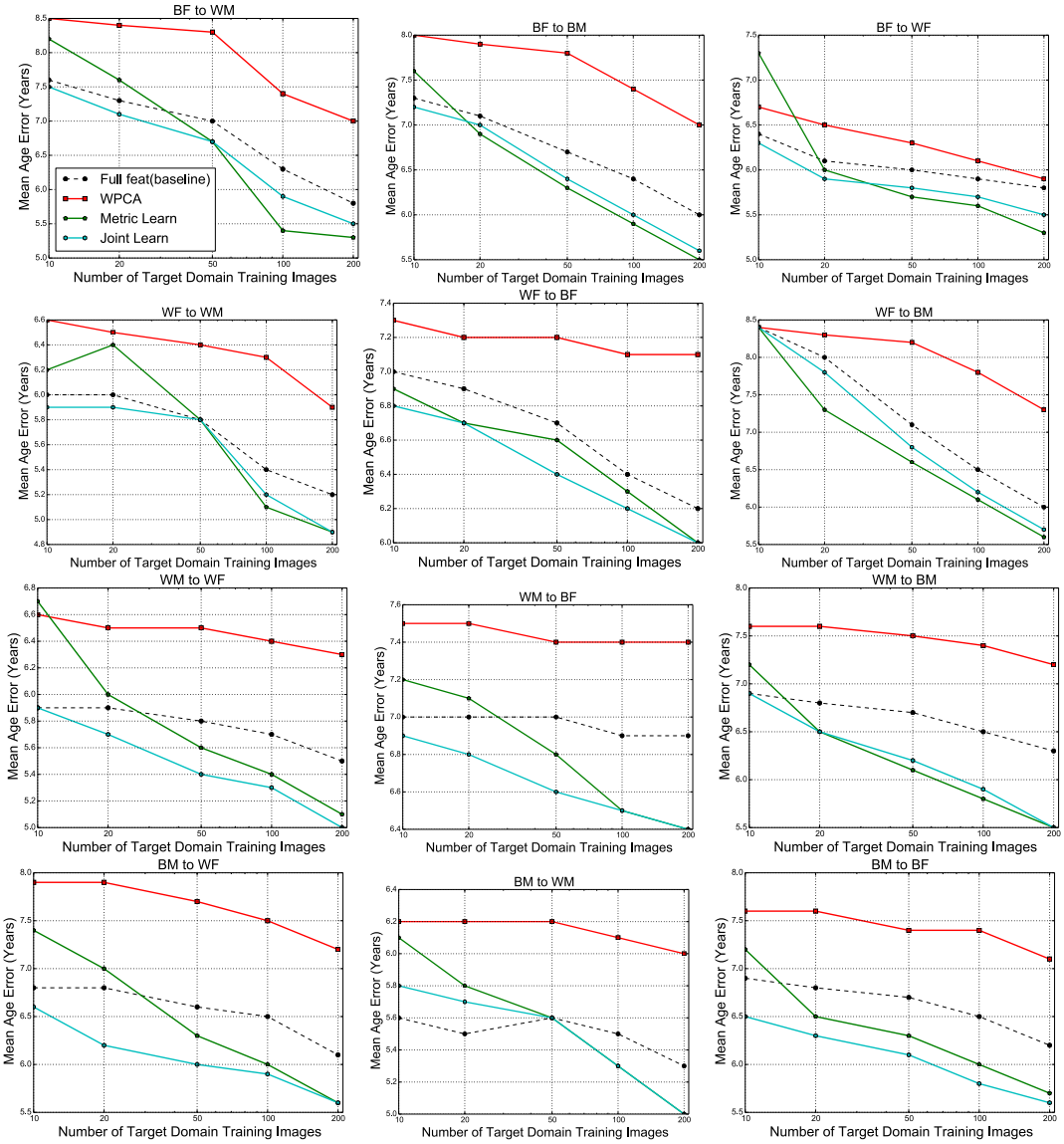


Figure 4.2: Graphs showing performance of different approaches vs. the number of target training examples.

4.3.2 Proposed joint approach

Joint Learning (JL) learns the regressor and projection in with an integrated objective function. The advantage of JL in comparison to ML is that it takes care of dissimilarity constraint between the ages. As mentioned before in the Section 4.2, ML pushes the dissimilar images equally farther irrespective of difference between the ages. We trained JL identically cf. ML; we used the same training pairs that were used for ML and initialized the projection matrix with WPCA and regressor by mean of the principal components of WPCA. Since we learned a projection matrix of dimensions 64, our regressor has 64 dimensions. The initial values of regressor are mean values of 64 principal components. We set learning rate to 0.001 and the number of maximum iterations to 2×10^5. For the regressor, we set $\epsilon = 0.1$, similar to that of standard SVR we used for all the baselines.

4.3.3 Experimental Results

Fig. 4.2 shows the performance of all the baselines and the one of our approach *w.r.t.* the size of the number of target training examples in all the domain pairs. Tab. 4.1 shows the performances of our method along with those of the baselines and the current state-of-art method of Guo *et al.* [59]. The values in the table shows the Mean (over 12 domain pairs) of the MAE (mean average error over examples) in years in relation with the number of Target Training Examples (TTE) used. It usually requires large number of labeled examples per class to compute scatter matrix using LDA, so we assume Guo *et al.* used more than 200 examples. In the domains, WF and BF, 200 examples counts around $(200/1285) \times 100 = 15.6\%$ and in WM, BM, it counts $(200/3980) \times 100 = 5\%$ of the training examples.

We note that, in comparison to the baselines *i.e.* LBP and WPCA, the proposed method consistently performs better. In comparison to ML, it performs better when the training examples from target domain is very small; whereas ML performs even worse than WPCA in such case (*e.g.* source target pair WM and WF). ML overfits when the positive training pairs are very small in number. This is an important practical use case, as often obtaining annotated examples of a new target domain is expensive. With the increasing size of target examples, the performance of ML ultimately converges to that of JL. Finally, the proposed approach clearly out-performs previous state-of-the-art method [59] by just taking 20 training examples from target domain.

Method	LBP	WPCA	ML	JL
Dimensions	9280	64	64	64

Total Target Examples	Method	Mean of MAE (y)
>200	[59]	**6.6 ± 1.0**
0	LBP	6.8 ± 0.8
	WPCA	7.4 ± 0.7
10	LBP	6.8 ± 0.7
	WPCA	7.4 ± 0.7
	ML	7.2 ± 0.7
	JL	**6.7 ± 0.7**
20	LBP	6.7 ± 0.7
	WPCA	7.3 ± 0.7
	ML	6.7 ± 0.5
	JL	**6.5 ± 0.6**
50	LBP	6.5 ± 0.5
	WPCA	7.3 ± 0.7
	ML	6.2 ± 0.4
	JL	**6.1 ± 0.4**
100	LBP	6.2 ± 0.4
	WPCA	7.0 ± 0.6
	ML	**5.8 ± 0.4**
	JL	**5.8 ± 0.4**
200	LBP	5.9 ± 0.5
	WPCA	6.8 ± 0.6
	ML	**5.5 ± 0.4**
	JL	**5.5 ± 0.4**

Table 4.1: Performance comparison between different baselines, our approach and previous state-of-art method [59].

4.4 Conclusions

We propose a novel joint learning method for cross-domain age estimation. We have evaluated our method on the largest publicly available dataset. The proposed experimental validation shows that our method outperforms wide ranges of strong baselines, improves the performance over the previous state-of-art algorithm and attains a state-of-art performance.

Chapter 5

Deep Fusion of Visual Signatures

Contents

5.1 Introduction

As we mentioned before, at the beginning of Ch. 4, this chapter presents a work mainly focused in aligning different feature types in a common sub-space.

More precisely, we present a novel multi-input hybrid deep learning method to align different types of features in common subspace and fuse them. Our approach can be easily deployed in a heterogeneous server-client framework for the challenging and important task of analyzing images of faces. Facial analysis is a key ingredient for assistive computer vision and human-machine interaction methods, and systems and incorporating high performing methods in daily life devices is a challenging task. The objective of this work is

to develop state-of-the-art technologies for recognizing facial expressions and facial attributes on mobile and low cost devices. Depending on their computing resources, the clients (i.e. the devices on which the face image is taken) are capable of computing different types of face signatures, from the simplest ones (e.g. LPB) to the most complex ones (e.g. very deep CNN features), and should be able to eventually combine them into a single rich signature. Moreover, it is convenient if the face analyzer, which might require significant computing resources, is implemented on a server receiving face signatures and computing facial expressions and attributes from these signatures. Keeping the computation of the signatures on the client is safer in terms of privacy, as the original images are not transmitted, and keeping the analysis part on the server is also beneficial for easy model upgrades in the future. To limit the transmission costs, the signatures have to be made as compact as possible. In summary, the technology needed for this scenario has to be able to merge the different available features – the number of features available at test time is not known in advance but is dependent on the computing resources available on the client – producing a unique rich and compact signature of the face, which can be transmitted and analyzed by a server. Ideally, we would like the universal signature to have the following properties: when all the features are available, we would like the performance of the signature to be better than the one of a system specifically optimized for any single type of feature. In addition, we would like to have reasonable performance when only one type of feature is available at test time.

For developing such a system, we propose a *hybrid deep neural network* and give a method to carefully fine-tune the network parameters while learning with all or a subset of features available. Thus, the proposed network can process a number of wide ranges of feature types such as hand-crafted LBP and FV, or even CNN features which are learned end-to-end.

While CNNs have been quite successful in computer vision [78], representing images with CNN features is relatively time consuming, much more than some simple hand-crafted features such as LBP. Thus, the use of CNN in real time applications is still not feasible. In addition, the use of robust hand-crafted features such as FV in hybrid architectures can give performance comparable to Deep CNN features [104]. The main advantage of learning hybrid architectures is to avoid having large numbers of convolutional and pooling layers. Again from [104], we can also observe that hybrid architectures improve the performance of hand-crafted features e.g. FVs. Therefore, hybrid architectures are useful for the cases where only hand-crafted features, and not the original images, are available during training and testing time. This scenario is useful when it is not possible to share training images due to copyright or privacy issues.

Hybrid networks are particularly adapted to our client-server setting. The client may send image descriptors either in the form of some hand-crafted features or CNN features or all of them, depending on the available computing power. The server has to make correct

Figure 5.1: Randomly sampled images of CelebA and a subset of attributes. Green color attributes are relevant for the image whereas red color attributes are irrelevant (better viewed in color).

predictions with any number and combination of features from the client. The naive solution would be to train classification model for the type of features as well as for any of their combinations and place them in the server. This will increase the number of model parameters exponentially with the number of different feature types. The proposed hybrid network aligns the different feature before fusing them in a unique signature. The main contribution of the work presented in this Chapeter is a novel multi-features fusion hybrid deep network, which can accept a number of wide ranges of feature types and fuse them in an optimal way. The proposed network first processes the different features with feature specific layers which are then followed by layers shared by all feature types. The former layer(s) generate(s) compact and discriminative signatures while the later ones process the signatures to make predictions for the faces. We learn both feature specific parameters and shared parameters to minimize the loss function using backpropagation in such a way that all the component features are aligned in a shared discriminative subspace. During test time, even if all the features are not available, e.g. due to computation limitations, the network can make good predictions with graceful degradations depending on the number of features missing. The thorough experimental validation provided, demonstrates that the proposed architecture gives state-of-the art result on attributes prediction on the CelabA dataset when all the features are available. The method also performs competitively when the number of features available is less i.e.

in a resource-constrained situation.

The rest of this Chapter is organized as follows: Sec. 2 presents the related works, Sec. 3 gives the details of our approach while Sec. 4 presents the experimental validation.

5.2 Related Works

In this section we review some of the works which are, on one side, related to hybrid architectures or, on the other side, related to face attribute classification.

Hybrid Architectures. One of the closest works to our work is from Perronnin et al. [104]. The main idea behind their work is to use Fisher Vectors as input to Neural Networks (NN) having few fully connected (supervised) layers (up to 3) and to learn the parameters of these layers to minimize the loss function. The parameters are optimized using backpropagation. Unlike their architecture, our network takes a number of wide range of hand-crafted features including FVs, but not only. In addition, our architecture is also equipped with both feature specific parameters and common parameters. We have designed our network in such a way that the input features are aligned to each other in their sub-spaces. The advantage of such alignments is that our system can give good performance even when a single type of feature is present at test time. Moreover, such ability makes our system feature independent i.e. it can properly handle any types of features it encounters.

There are some works, such as [136], which, instead of taking hand-crafted features as input, takes CNN features and compute FVs in the context of efficient image retrieval and image tagging. This approach improves the performance of CNNs and attains state-of-art performance, showing that not only FVs but also CNNs benefit from hybrid architecture.

Face Attribute Classification. Some of the earliest and seminal work on facial attribute classification is the works from Kumar *et al.* [81, 82]. Both of their works use hand-crafted low-level features to represent faces, sampled with AdaBoost in order to discover the most discriminative ones for a given attribute, and train binary SVM classifiers on this subset of features to perform attribute classification. The current state-of-art method of Liu *et al.* [89] uses two deep networks, one for face localization and another for identity based face classification. The penultimate layer of the identity classification network is taken as the face representation, and a binary SVM classifier is trained to perform an attribute classification. Some other recent state-of-the-art methods such as PANDA [157], Gated ConvNet [77] etc. also use deep learning to learn the image representation and do attribute classifications on it. From these works, we can observe that either hand-crafted features or CNN features are used for attribute classification. From our knowledge, the proposed method is the first to learn a hybrid structure combining multiple hand-crafted and CNN features for facial attribute classification. Moreover, most of the mentioned

works here are performing binary attribute classification while we are predicting multiple attributes of faces.

Multi-mode fusion. Recently Neverova *et al.* [95] proposed a method called *Mod-Drop* to fuse information from multiple sources. Their main idea is to take a batch of examples from one source at a time and feed into the network to learn the parameters, instead of taking examples from all the sources. The main drawbacks of their approach is, when new source is encountered and need to fuse, it requires to re-run the whole network. Some other recent works such as [76, 122, 148] fuse multiple source of information to improve the performance of end result. None of these works evaluated the performance of component sources or their possible combinations after fusion.

5.3 Approach

As mentioned before, a key challenge addressed in this Chapter is to learn an optimal way to fuse several image features into a common signature, through the use of a hybrid fully connected deep network. This section presents the proposed method in detail, explains how to learn the parameters and gives technical details regarding the architecture.

5.3.1 Network architecture

Fig. 5.2 shows a schematic diagram of the proposed network. A, B and C denote the different feature types to be aligned and fused, which are the input to the network. We recall that all or only a subset of the features can be available depending on the computing resources of the client. While we show a network with 3 features types, more can be used with similar layers for the new features. The key idea here is to train a single network which consists of feature specific layers (shown in blue), to be implanted on the clients, and common layers (shown in black), to be implanted on the server. The activations of the middle layer, obtained after merging the feature specific layers, gives the universal signature which will be transmitted from the client to the server. Each layer is fully connected with its parents in the network. In our application the output of the network are the facial expressions/attributes to be recognized, one neuron per expression/attribute, with the final values indicating the score for the presence of these attributes.

5.3.2 Learning the parameters of the network

Carefully setting up the learning of such hybrid network is the main issue for competitive performance. We propose to learn the parameters of this network with a multistage

approach. We start by learning an initialization of the common parameters. To do this we work with the most discriminate feature type (e.g. A, B or C). For example, suppose we observed that A is the most discriminate for our application (as discussed in the experiment section, we will see that for our application FVs are the most discriminant features). Thus we start learning the parameters of the network corresponding to both (i) the feature specific parameters of network A (blue layers) and (ii) the part of the network common to all features (black layers). Then we fix the common parameters and learn the feature specific parameters of the feature B taking training examples encoded with B. In our case, the task is same but the features are different during each training round. By repeating the same procedure, we learn the feature specific parameters of the network for each of the remaining type of features. In the end, all the features are aligned into a common signature which can then be transmitted to the server for the computation.

The major advantage of this strategy is that although we are mapping all the features into same feature space, we do not require feature to feature correspondence e.g. we are not using a certain feature type to estimate or mimic any other feature type. Moreover, when we encounter a new feature type, we can easily branch out the existing network and learn its parameter without hindering the performance of other feature types. Thus the proposed learning strategy, while performing very well, also avoids the retraining of the whole network upon addition of a new features type. This is a major advantage of this our approach over existing Mod-Drop [95] algorithm. Finally, since there are fewer parameters to optimize than training one distinct network per feature, the computations required are less and the training is faster.

Another alternative, that we explored, is to learn the parameters of the whole network first with all the available feature types, and then fix the common parameters and fine-tune the feature specific parameters. The reason behind this approach is to make shared subspace more discriminative than with the one learned with the single most discriminative feature so that we can align all the component features in this subspace and improve the overall performance. We found the performance obtained with this approach is slightly better than the one we discussed before. However, this alternative requires feature to feature correspondence mapping. Moreover, training with all the features at a time requires more computing resource and also leads to slow convergence and longer training time. We compare the performances of these methods in more details in the experiment section.

5.3.3 Details of the architecture

The proposed network is composed of only fully connected (FC) layers. Once the features are fed into the network, they undergo feature specific linear projections followed by

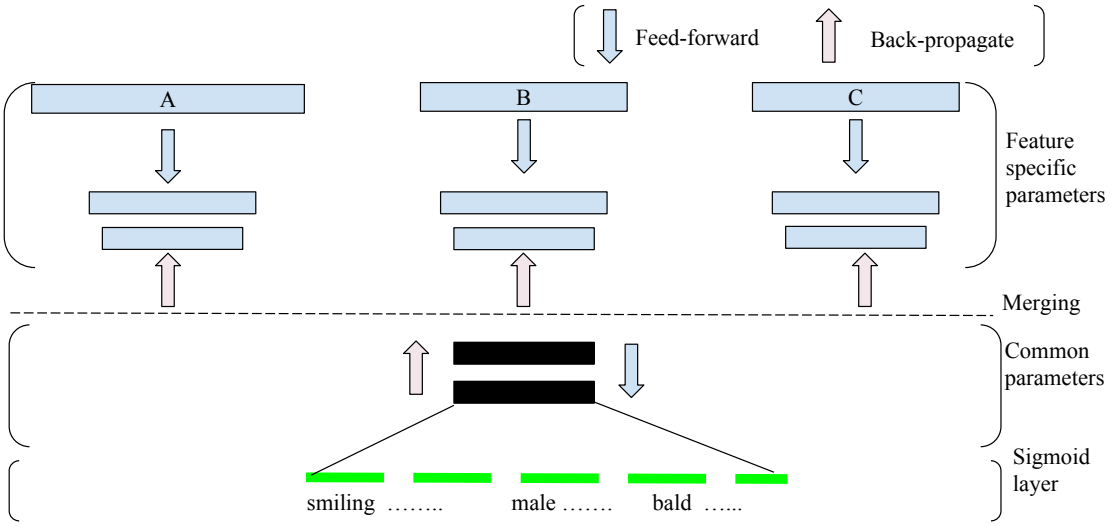


Figure 5.2: Illustration of proposed method.

processing with Rectified Linear Units (ReLU). Eq. 5.1 gives the feature-specific transformations, where σ is the non-linear transformation function i.e. ReLU, W_A, W_B, W_C and $\mathbf{b}_A, \mathbf{b}_B, \mathbf{b}_C$ are projection matrices and biases for the input features of the networks A, B, and C respectively. These representations further go into linear projections followed by ReLU depending upon the depth of the network.

$$\begin{aligned} h^A &= \sigma(\mathbf{x}_A W_A + \mathbf{b}_A) \\ h^B &= \sigma(\mathbf{x}_B W_B + \mathbf{b}_B) \\ h^C &= \sigma(\mathbf{x}_C W_C + \mathbf{b}_C) \end{aligned} \tag{5.1}$$

When the network takes more than one type of features at a time, it first transforms them with the FC and ReLU layers and then sums them and feeds into the common part of the network. We call this step as *merging*, as shown in the diagram. We further call the vector obtained at this point, after merging, as the signature of the face.

In the common part of the network, intermediate hidden layers are projected into linear space followed by ReLU. The final layer of the network is a sigmoid layer. Since we are doing multilabel predictions, sigmoid will assign higher probabilities to the ground truth classes. We learn the parameters to minimize the sum of binary cross-entropy of all the predictions of the sigmoid layer. We minimize the loss function using Stochastic Gradient Descent (SGD) with standard backpropagation method for network training.

In the heterogeneous client-server setting, the client is expected to compute the signature and send it to the server for processing. Since different clients can have very different computing capabilities they can compute their signature with different types and number of features – in the worst case with just one feature. The method allows for such diver-

Parameters Type	Layer Type	A	B	C
Feature Specific	Input	$\mathbf{x}_A$	$\mathbf{x}_B$	$\mathbf{x}_C$
	FC(ReLU)	4096	4096	4096
	FC(ReLU)	1024	1024	1024
Merge	Add	1024		
Common Parameters	FC(ReLU)	1024		
	FC(ReLU)	1024		
	Sigmoid	40		

Table 5.1: Detail parameters of proposed network

sity among clients and as the server side works with the provided signature while being agnostic about what and how many features were used to make it.

5.4 Experiments

We now present the experimental validation of the proposed method on the task of facial attribute classification. All the quantitative evaluation is done on the CelebA dataset [89], the largest publicly available dataset annotated with facial attributes. There are more than 200,000 face images annotated with 40 facial attributes. This dataset is split into train, val, and test sets. We use train and val set for training and parameter selection respectively, and we report the results on the test set.

In the rest of the section, we first give the implementation details and then discuss the results we obtained.

5.4.1 Implementation details

We have performed all our experiments with the publicly available aligned and cropped version of the CelebA[1] [89] dataset (without any further pre-processing). We assume that up to 3 different types of features can be computed, namely, Local Binary Patterns, Fisher Vectors and Convolutional Neural Networks features, as described below.

Local Binary Patterns (LBP). We use the publicly available `vlfeat` [137] library to compute the LBP descriptors. The images are cropped to 218×178 pixels. We set cell size equal to 20, which yields a descriptor of dimension 4640.

Fisher Vectors (FV). We compute Fisher Vectors following Simoyan et al [119]. We compute dense SIFTs at multiple scales, and compress them to a dimension of 64 using Principal Component Analysis. We use a Gaussian mixture model with 256 Gaussian

[1] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

components. Thus, the dimension of the FV feature is of 32,768 (2×256×64). The performance of this descriptor is $77.6 \pm 1.2\%$ on LFW for the task of face verification, with unsupervised setting, which is comparable to the one reported [119].

Convolutional Neural Networks (CNN). We use the publicly available state-of-art CNN mode trained on millions of faces presented in [101], to compute the CNN features. The dimension of CNN feature is of 4096. Our implementation of this feature gives $94.5 \pm 1.1\%$ on LFW for verification in unsupervised setting. Here, these features are computed without flipping and/or multiples of cropping of faces.

5.4.2 Baseline methods.

We report two different types of baselines. In the first one, the network is trained with a given feature type (e.g. LBP) while the same type of feature is used at test time (e.g. LBP again). We call this type of network as *Dedicated Networks*. In the second setting, we allow the set of features at train time and the one used at test time to differ. Such networks are adapted to different sets of features. This is the particular situation we are interested in. More precisely, we experimented with 3 different dedicated networks (one per feature type) and 2 adapted networks, as detailed below, all such are considered as baselines.

LBPNet/FVNet/CNNNet. These baseline networks use only LBP, FV or CNN features, respectively, for both training and testing. They provide the single feature performances, assuming that no other feature is available either at training or testing.

All Feature Training Network (AllFeatNet). In this setting, all the available features are used to train the network. At test time, one or more than one type of features can be used, depending on its availability. For us, the available features are as described before FVs, CNNs, and LBPs.

Mod-Drop. This is currently the best method for learning cross-modal architectures, inspired by [95]. It consists, at train time, in randomly sampling a batch of examples including only one type of features at a time, instead of taking all the available features, and learn the parameters in a stochastic manner. We refer the reader to the original work [95] for more details.

5.4.3 The proposed method.

On the basis of which we fix the parameters of the common shared subspace, we categorize the proposed methods into two:

FVNetInit. Tab. 5.2 shows the individual performance of different features we used for

Method	Avg. Precision
Random	23.1%
FVNet	**69.0%**
CNNNet	68.7%
LBPNet	64.3%

Table 5.2: Average Precision (AP) of single feature type baselines

Method	mean Avg. Precision
AllFeatNet	63.4 $\pm$ 9.46 %
Mod-Drop	67.8 $\pm$ 3.67 %
Ours(FVNetInit)	68.8 $\pm$ 2.98%
Ours(AllFeatNetInit)	**69.0 $\pm$ 3.42%**

Table 5.3: mean AP(mAP) of multi-feature baselines

our experiments. From the table we can see that, FVs are most discriminative for our application. Thus, we choose to take few top layer's parameters (please refer Tab. 5.1 of for the number of layers in shared subspace) of FVNet as common shared parameters of proposed network. Once we fix this, we learn the feature specific parameters for CNNs and LBPs to minimize the loss function. Fig. 5.4 shows the evolution of performances of FVs, LBPs, and CNNs with the number of training epochs.

AllFeatNetInit. In this case, we use the common part of AllFeatNet as a starting point. Then we fix these parameters and learn the feature specific parameters of FVs, LBPs and CNNs to minimize the loss the function.

5.4.4 Quantitative results

We now present the results of the experiments we do to evaluate the proposed method. We measure the performance using average precision (AP) i.e. the area under the precision vs. recall curve. We do not consider attribute label imbalances for all the cases, unless explicitly stated.

Features	Dedicated Network	AllFeatNet	Mod-Drop	Ours (FVNetInit)	Ours (AllFeatNetInit)
FV	69.0%	64.2%(-4.7%)	70.0%(+1%)	69.0%(-0.0%)	68.8%(-0.2%)
CNN	68.7%	63.3%(-5.5%)	68.2%(-0.5%)	68.1%(-0.6%)	67.9%(-0.8%)
LBP	64.3%	42.5%(-21.8%)	59.6%(-4.7%)	62.1%(-2.2%)	61.5%(-2.8%)

Table 5.4: Performance comparison of proposed methods and other compared methods with the dedicated networks. The table shows that, the performance of proposed methods is competitive to that of dedicated networks while the performance of other compared methods is significantly low, particularly in the case of LBPs.

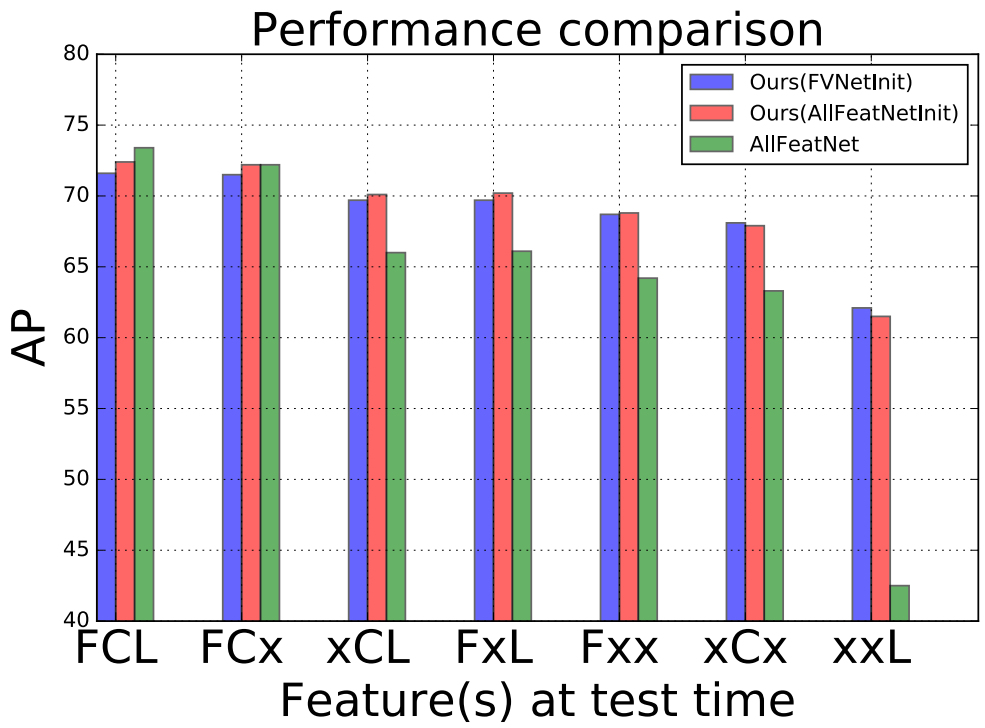


Figure 5.3: Performance comparison between different methods at the different combination of feature(s) at test time. FCL represents FVs, CNNs, and LBPs respectively. ***'x'*** *represents the absence of the feature corresponding to it's index.*

Our experiments are mainly focused on validating two aspects of the proposed method. First, we demonstrate that the performance due to individual features are retained after merging all the features in the same common subspace. Second, we demonstrate that the performance is improved in the presence of more information, i.e. presence of multiple types of features at a time.

Performance comparison with Dedicated Networks. Tab. 5.2 and Tab. 5.4 give the performance of single feature trained networks and their comparison with that of the multi-feature trained network (when, at test time, only one type of feature is present). From these tables, we can observe that, with both our approaches, the performance of the component features at test time is competitive to that of dedicated networks trained with those features only. Compared to existing methods such as Mod-Drop and AllFeatNet, the range of performance drops in comparison to dedicated networks is the least in our case. More precisely, the widest drop range for us is up to -2.8% w.r.t. that of LBPNet in AllFeatNetInit network. While for the same feature, it is up to -4.7% in Mod-Drop and up to -21.8% in AllFeatNet w.r.t. that of LBPNet. These results clearly demonstrate that our method is more robust in retaining the performances of individual features while projecting them in common subspace.

Performance comparison with Multi-feature Networks. Table 5.3 compares the mean

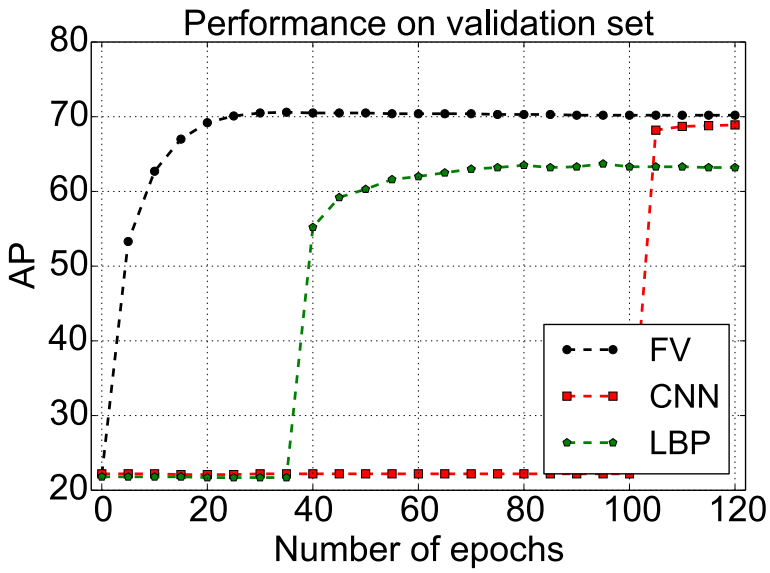


Figure 5.4: Performance of FVs, CNNs, and LBPs on the validation set.

average precision (mAP) of different multiple features based networks with the proposed method. For a network with 3 different types of input features, there are 7 different possible combinations of feature(s) at test time. The performance shown in the table is the mean AP obtained with all these combinations. The proposed method outperforms the other multi-feature based networks. This shows that the proposed network and the multi-stage training strategy is capable of making better predictions in the presence of more information i.e. multiple types of features at a time and are optimal to every combinations of features.

Fig. 5.3 shows the performance comparison between the proposed methods with AllFeatNet at different levels of feature combinations. From the bar-chart, we can observe that, when all the features are available at test time, AllFeatNet performs better than ours. It is expected too, because this approach is optimized only for this combination. But this is the most unlikely scenario for the applications we are addressing, due to constraints such as computing resources and time, etc. Out of other 6 cases, our method performs substantially better and gives similar performance in one case. This shows that our method leverages all the features available and when more information is present, gives better performance. Unlike AllFeatNet, the proposed method is optimal in every combinations of features too.

5.4.5 Qualitative results

Fig. 5.5 shows the qualitative performances comparison between the baselines and the proposed method. We randomly choose three different test images and used them for evaluation. Here, we consider LBPs (the simplest feature type) only for evaluation. Thus

	LBPNet		AllFeatNet		Ours(AllFeatNetInit)	
	blond hair	0.43	mouth stly open	0.17	wavy hair	0.88
	pointy nose	0.43	black hair	0.2	attractive	0.93
	Attractive	0.56	oval face	0.26	bushy eyebrows	0.94
	heavy makeup	0.70	pointy nose	0.39	heavy makeup	0.96
	w. lipstick	0.94	young	0.72	young	0.99
	young	0.95	no beard	0.99	w. lipstick	0.99
	no beard	1.0	male	0.99	no beard	1.0
	straight hair	0.38	blurry	0.16	attractive	0.85
	attractive	0.57	w. necktie	0.17	male	0.92
	black hair	0.61	mouth stly open	0.28	bushy eyebrows	0.94
	male	0.69	black hair	0.76	no beard	0.98
	young	0.88	young	0.81	black hair	0.99
	bangs	0.98	no beard	0.99	bangs	0.99
	no beard	0.99	male	1.0	young	0.99
	bags under eyes	0.23	high cheekbones	0.11	oval face	0.55
	oval face	0.28	pointy nose	0.27	mouth stly open	0.72
	male	0.53	oval face	0.41	bushy eyebrows	0.8
	young	0.69	young	0.56	no beard	0.85
	mouth stly open	0.70	mouth stly open	0.86	young	0.88
	w. hat	0.96	no beard	0.96	w. hat	0.93
	no beard	0.98	male	1.0	male	0.98

Figure 5.5: Qualitative results comparison of the proposed method with other methods. Top 7 attributes predicted by these methods are shown. As before green color indicates relevant attributes whereas red color indicates irrelevant attributes for the image. (Better viewed in color)

for both the single feature network (LBPNet) and multi-feature network (AllFeatNet and ours), only LBPs are available at test time. In the figure we can see the top 7 attributes predicted by the compared methods. For each of the attributes, the corresponding score shows the probability of an attribute being present in the given image. On the basis of the number of correct predicted attributes, the performances of LBPNet and and the proposed method is comparable in two cases (first two cases). While in the third case, our method (4 correct predictions) is even better than LBPNet (3 correct predictions). This further validates that the proposed method retains the property of component features. The performance of AllFeatNet is comparatively poorer than LBPNet and ours for all test images. Moreover, it is important to note that, the scores corresponding to the predicted attributes by AllFeatNet are small. This suggests that with this approach the predictive power of LBPs are masked by other strong features e.g. FV and CNNs.

5.5 Conclusions

We propose a novel hybrid deep neural network and a multistage training strategy, for facial attribute classification. We demonstrated, with extensive experiments, that the proposed method retains the performance of each of the component features while aligning

and merging all the features in the same subspace. In addition to it, when more than one feature type are present, it improves the performance and attains state-of-art performance. The proposed method is also easily adaptable to new features simply learning the feature specific parameters. This avoids retraining the existing network. Since the majority part of the network is shared among all the feature types, the proposed method reduces the number of parameters.

Chapter 6

Face De-identification

Contents

6.1 Introduction

In Chapters 2, 3, 4, 5 we presented works which aimed at automatically extracting different information such as identity, age, and other attributes and expressions from faces. In this chapter we present a work which deals with hiding identity information from faces. But, it is required that the image looks as similar as before on overall. As mentionned at the beginning of this thesis, this work is in a direction opposite to the one presented before, from the research point of view. We introduce below in this chapter the problem, describe the related works, our approach and experiments to validate the propsed method.

Posting photos of oneself and relatives is one of the main activity on social networks. Yet, these are pictures with visible faces, and faces are distinctive. Thanks to an automatic face recognition solution (and all the major actors in the field have recently acquired

Figure 6.1: Pairs of images from LFW. Left images are original images and right images are de-identified forgeries.

such technology), the network proposes the user to cross link faces on his pictures with profiles of his acquaintances. This functionality, called "Photo Tag Suggest" on Facebook for instance, uses already labeled photos and face recognition technology to identify individuals in new photos. If Yana Welinder doesn't lower the fantastic innovation and usefulness behind social networks, she points the numerous breaches of individual privacy bound to sharing face pictures on this medium [145]. Face recognition technology in conjunction with social networks shifts the anonymity paradigm. A priori anonymous faces are not only connected to names, of which there can be several, but also to all the information of social network profiles. Welinder shows that law alone cannot prevent these dangers, she stresses that it is up to the user to decide to benefit from this functionality or not, and that this user-centric privacy policy should be enforced together by legal and technological means.

This appeal for a technological privacy gatekeeper motivates our work. More precisely, we are investigating whether a user could post a picture with her/his face publicly visible on the social network such that friends recognize him, while, at the same time, the automatic face tagging technology fails.

Face tagging is usually done thanks to a face verification algorithm. To compare two faces, this algorithm encodes them into discriminative signatures, computes a 'distance' between the signatures, and compares this metric to a threshold. [85] provides a benchmark of recent face verification algorithms. In this context, de-identifying an image consists in altering it in such a way that its signature becomes different enough from the signature of a reference image (*i.e.* the distance to the reference signature is above the threshold).

Anonymizing faces in publicly available images is easy by masking or blurring them (*e.g.*

such as in Google Street View) if the goal is to make them non recognizable by humans. When the image quality has to be preserved (*i.e.* let a face looks like a face), the wording 'de-identification' is preferred to 'anonymization' since a human still recognize a relative on the picture. Better image processing than blurring have been proposed [39, 96]. Section 6.2 reviews these works and outlines three pitfalls. Their scenario is different as they address the sanitization of a database of face images before publication, whereas we de-identify a query image. Their very specific approach, *i.e.* a retro-engineering of the well-known eigenfaces representation, does not apply on modern face recognition schemes, which are much more non-linear. Their experimental work uses images from 'biometric' datasets which are quite different from face images published on social network.

We first outline that we do not target the same goal. These previous works make a strong connection between face recognition and database privacy. They aim at enforcing privacy concepts like k-anonymity to face images. Their context is the following: the owner of a database of pictures with visible faces would like to publish it with the guarantee that, later on, it will not be used for identifying people on some other photos. In other words, the goal is to sanitize the database before publishing it. These authors process the images of the database coherently to cluster k similar faces into one representative. In our context, it is up to the social network to perform this task, and its users have to trust the effectiveness of such operation. Our approach is different as we assume the social network has a collection of pictures which cannot be modified. It is up to the user to process his picture before posting it. In other words, we manipulate the query image not the database images.

Another weakness of these previous works is that they work with a specific signature extraction, which is almost always the well-known eigenfaces representation. In a nutshell, the signature is obtained by projecting the face image (after a normalizing process) onto some reference pattern called eigenfaces. Their approach was to retro-engineered the signature extraction: they pretendedly apply the minimum distortion on the face image which modifies its signature into a precise value. Consequently, their de-identification process is dedicated to this somehow outdated signature and cannot be generalized to modern representations. We believe that such retro-engineering is much more difficult if not impossible with recent extraction processes which are highly non-linear. For instance, it is not even sure that there exists an image whose signature exactly equals a given value. Section 6.2 carries on the analysis of these previous works outlining other differences with our approach.

However, because of the construction process, these images look blurred the trade-off between quality and de-indentification can not be controlled. Furthermore, previous works are focused on biometrics conditions (controlled illumination, fixed face poses, *etc.*) and used the simple Eigen-face recognition framework while much more powerful technologies exist at the moment.

The parallel between our de-identification scenario and oracle attacks in the context of digital watermarking [29, 30, 42] motivates our approach. A digital watermark is a kind of perceptually invisible marker embedded in multimedia contents. It is typically used to prove copyright ownership of still images. An attack refers to as an image processing partially removing the watermark in the sense that the detector no longer classifies the altered images as watermarked. This is governed by a trade-off between the probability of deceiving the watermark detection and the attack distortion, often measured by the averaged pixel distortion between the watermarked and the attacked images. In some applications, the pirate has access to the watermark detector as an oracle: the pirate has no knowledge about the watermarking technology ; the watermark detector is a black box, to which the pirate submits images and observes the binary decisions (presence or absence of a watermark). Oracle attacks benefit from this feedback to iteratively refine the quality of the attacked images.

Our problem is similar in the sense that the identifiability of a face is the equivalent of the detectability of a watermark to be hindered. The main proposal of our work is that this parallel opens the door to interesting avenues for face de-identification. First, the assumption of an oracle is valid: more and more face recognition tools are publicly available (*e.g.* Google Picasa or Apple iPhoto softwares provide this functionality), the user a priori knows the photos of himself present in the social network used as a reference to identify him, and he can freely manipulate his picture before publishing it on the network. Second, the user doesn't need to know all the internals of the face recognition technology, the oracle attack only needs the output of this black box: from an image with a face, the person is correctly identified or not.

Even guided by the feedback of an oracle, the strategy for altering the face image is of utmost importance. It turns out that the previous approach based on the modification of the eigenfaces projections deeply distorts the images to a point where recognition by humans is challenging (see figures in [39, 96]). On the other hand, hindering the recognition by a computer but not by a human is a well-known CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) ill-posed problem. The keystone idea is that human recognition is more invariant and robust than computer recognition against certain types of distortion. Geometrical distortion is wide-spread in text-based CAPTCHA, but it produces too many unpleasant artefacts when applied on face images. Our approach relies on background noise distortion and the ability of the human brain for sources separation [92]. The classical example is the "cocktail party problem", where many people are talking simultaneously in a room, and a listener tries to follow one of the discussions. Transposing this to images means that seeing a grid on a face image, the human brain spots the two 'sources', the grid and the face, that it easily separates.

Consequently, one key idea of this work is to combine face images with procedural noise

textures (*e.g.* Perlin's noise [103]), with the rational that (i) the stationarity of the noise texture lets human brains remove it, (ii) the parameters of the texture allows to minimize the alteration of the image unrecognized by a computer. An iterative procedure inspired by oracle attacks in digital watermarking controls the trade-off between image quality and de-identification. Another goal of our work is the robustness against counter-attacks. Indeed, the added noise should be robust to filtering (*e.g.* Gaussian filtering) otherwise removing it will be easy. This is explicitly taken into account in the optimization of the texture parameters.

The proposed approach is experimentally validated on the popular Labeled Faces on Wild (LFW) dataset [64]. Indeed, by using images coming from the internet in which people appears 'in the wild' (*i.e.* in uncontrolled situations), this dataset is close to the targeted use case. Both qualitative and quantitative experiments validate the approach. Qualitative experiments present altered images for visual inspection. Quantitative experiments evaluate the performance of our method against recent best performing face recognition algorithms, as well as their robustness to counter attacks. In the experiment section, we show that the accuracy of the face recognition system can be reduced from a state-of-the-art 85% to the level of chance while the alteration of images is acceptable for humans.

The remainder of this chapter is organized as follows. We describe some of the closely related works in Section 6.2. Similarly, our method is described in Section 6.3, experiments and conclusion are given in Section 6.4 and 6.5 respectively.

6.2 Related Work

Face recognition literature. Surprisingly, face de-identification has received very little attention in the computer vision literature. One of the pioneer work in this field is the early work of [96, 50]. Newton *et al.* [96] propose a privacy enhancing algorithm, called k-Same, transposing the concept of k-anonymity to face image databases. It aims at limiting the ability of a given face recognition system when working on a specific database. The algorithm first determines similarity between faces of the database, clusters similar faces, and creates a new face by aggregating the faces of a cluster. Gross *et al.* [51] proposes a factorization approach to separate identity and non-identity related factors, allowing to only replace the factors expressing the identity by the cluster's aggregation, while keeping the non-identity factors untouched to better preserve facial expressions. Dufaux and Ebrahimi [41] presents an effective scrambling techniques to foil face recognition. Recently, Driessen and Dürmuth [39] put the preservation of the human recognition as a top requirement. The idea is to find the modification of the image which has the lowest distortion (in the image space) while changing the signature to a desired value, *i.e.* the aggregation of the cluster's signatures. In practice, they work with the signature extraction based on the face image projection onto the manifold spanned by some eigen-

faces. Modifying the signature amounts to change this projection, and since this is a linear process, mapping back this change into the image space is simply achieved by modulating the eigenfaces components. The image part orthogonal to the space spanned by the eigenfaces is kept untouched. Another improvement is a relaxation of the k-Same principle: it is sufficient to push the signatures towards instead of exactly onto the cluster's aggregation.

We first outline that we do not target the same goal. The k-Same principle transposes the k-anonymity privacy concept. Their context is thus the following: the owner of a database of pictures with visible faces would like to publish it with the guarantee that, later on, it will not be used for identifying people represented on these images or some other pictures. In other words, the goal is to sanitize the database before publishing it. These authors process the images of the database coherently to cluster k similar faces into one representative. Consequently, the search for the most similar face of a query image will output k identities. In our context, it is up to the social network to perform this sanitization task, and its users have to trust the effectiveness of such operation. Our approach is different as we assume the social network has already published a collection of pictures, and it is up to the user to process his picture before posting it. In other words, we manipulate the query image not the database images.

The previous works with the well-known eigenfaces representation [96, 39], or some variants [51] degrade the quality of images. Fig. 6.2 and Fig. 6.3 show some of the representative de-identified face images from [39] and [96] respectively. The quality of forged images is not so good, either are blurred or have artifacts. However, the good point with these methods is that it eases the retro-engineering of the signature extraction, which consists in forging an image whose signature equals a desired value. Consequently, their de-identification process is dedicated to this somehow outdated signature. We believe that such retro-engineering is much more difficult if not impossible with recent extraction processes which are highly non-linear [65, 93, 119]. Our approach based on oracle attacks against digital watermarking does not rely on retro-engineering the signature extraction.

As surprising as it may sound, up to our knowledge, none of the previous works addresses the privacy of faces 'in the wild' *i.e.* as they are on the internet. These previous works are focused on datasets like FERET [106] or a subset of CSU multi-pie [17], where images are taken in a controlled environment (frontal illumination, frontal pose, *etc.*).

Image forensics literature. There are works in image forensics tackling the manipulation of SIFT local descriptors. The application scenario is to delude a forensics tool detecting copy-move forgery or near duplicates of object in images [5], or a CBIR system (Content Based Image Retrieval) [38]. In the latter, Do *et al.* investigate the forgery of quasi-copies of pictures belonging to a large database of natural images (*i.e.* not face images) in a way that the CBIR system either fails recognizing the content (false negative),

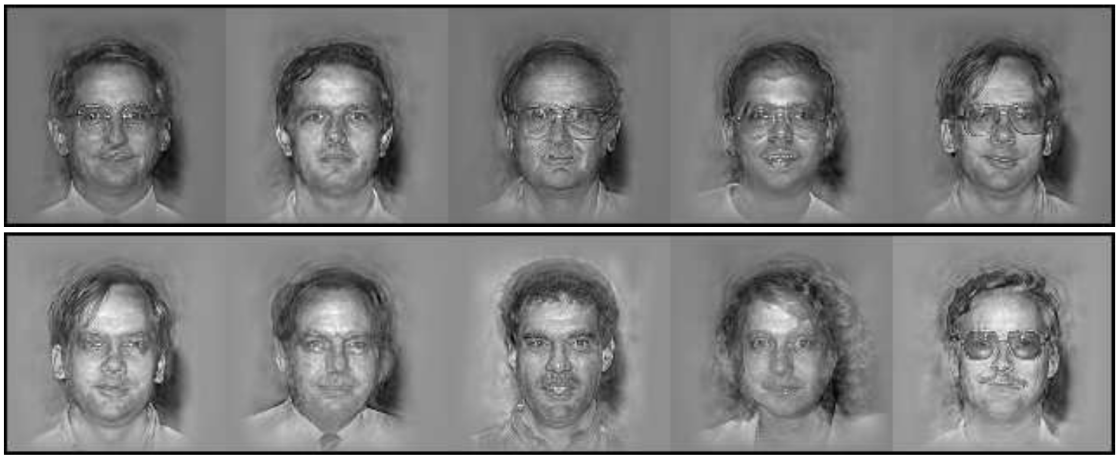


Figure 6.2: Sample de-identified images from [39]

Figure 6.3: Sample de-identified images from [96]

either recognizes another image (false positive).

These studies based on the SIFT local description shows that it is hard to remove keypoints at large scale in the forgery and that the retro-engineering (*i.e.* modifying the descriptors at will) is cumbersome, uncertain and yields severe local distortions on the image [37]. This is mainly due to the high non-linearity of SIFT. Indeed, for deluding a CBIR system, they conclude that the best strategy is to trigger a false positive by incorporating visual elements of another image into the query [36]. This approach is not relevant for our scenario because we don't use this description and we are not interested in false positive.

Digital watermarking literature. As explained in the introduction, our work is inspired by oracle attacks against digital watermarking. The detection of the presence (or the absence) of an invisible watermark in an image is sufficient in many applications. The embedding and the detection of the watermark are algorithms relying on a secret key. In some context, it is assumed that the pirate has a free access to a watermark detector in a black sealed box.

Let us consider images as points in the image space $\mathcal{I}$, and denote $\mathbf{x}_o$ ($\mathbf{x}_w$) the original image (resp. its watermarked version). The detection function $D : \mathcal{I} \to \{0, 1\}$ outputs 1

if the watermark is deemed present. We call $\mathcal{D} = \{\boldsymbol{z} \in \mathcal{I} | D(\boldsymbol{z}) = 1\}$ the set of images deemed as watermarked and $\mathcal{B}$ the boundary between $\mathcal{D}$ and $\bar{\mathcal{D}}$. The attacker aims at finding an image $\mathbf{y}$ such that $D(\mathbf{y}) = 0$ and as close as possible to $\mathbf{x}_w$. The distortion is measured by function $d(\mathbf{x}_w, \mathbf{y})$, say the Euclidean distance. Formally, the optimal attack can be written as: the following optimization problem:

$$\mathbf{y}^\star = \arg \min_{\mathbf{y}: D(\mathbf{y})=0} d(\mathbf{x}_w, \mathbf{y}) \tag{6.1}$$

Comesaña *et al.* [29], showed that (6.1) is equivalent to:

$$\mathbf{y}^\star = \arg \min_{\mathbf{y}} \delta(\mathbf{y}) \tag{6.2}$$

where $\delta(\mathbf{y}) = d(\mathbf{x}_w, h_{\mathcal{B}}(\mathbf{y}))$ and $h_{\mathcal{B}}(\cdot) : \mathcal{I} \to \mathcal{B}$ is a surjection onto the boundary of the detection region. The shape and location of this boundary in the image space depends on the watermarking scheme and the secret key. This prevents the pirate to implement it. Yet, thanks to the free access of the watermark detector, oracle attacks aims at solving this minimization problem. Several implementations have been proposed. They first differ by the implementation of the surjection. For example, when $\mathbf{y}$ is such that $D(\mathbf{y}) = 0$, a classical trick is to perform a line search over $\alpha \in (0, 1)$ such that $\hat{h}_{\mathcal{B}}(\mathbf{y}) = \mathbf{x}_w + \alpha(\mathbf{y} - \mathbf{x}_w)$ is close to $\mathcal{B}$. Another difference is the way the oracle attack locally explores $\mathcal{B}$: Comesaña estimates the gradient (sometimes together with the Hessian) of the function $\delta(\mathbf{y})$, which costs $O(N)$ (resp. $O(N^2)$) oracle calls, in order to perform a Newton-Raphson method [29, 30]. To save oracle calls, Earl [42] keeps on randomly drawing a new direction in the space and tests whether moving along this direction may decrease the functional $\delta(\mathbf{y})$.

An oracle attack thus 'travels' over the boundary until it finds a minimum of $\delta(\mathbf{y})$. This is a local minimum because the detection region is a priori not convex. It drastically reduces the average pixel distortion (around 10^{-4}, *i.e.* PSNR of 40dB) required for removing the watermark compared to blind attacks like a coarse JPEG compression (around 10^{-2}, *i.e.* PSNR of 20dB). The main criterion to compare oracle attacks is the number of calls to the watermarking detector.

6.3 Our method

We consider a given face verification algorithm determining if a face image represents the same person than a reference image $\mathbf{x}_o$. As this is usually done by comparing the distance between the signatures extracted from the images to a threshold, we model it as a binary function of the image space: $V_{\mathbf{x}_o}(\cdot) : \mathcal{I} \to \{0, 1\}$. We start from a face image $\mathbf{x}_f$ representing the same person than $\mathbf{x}_o$ (*i.e.* $V_{\mathbf{x}_o}(\mathbf{x}_f) = 1$). De-identifying consists in forging a new image $\mathbf{y}$ such that $V_{\mathbf{x}_o}(\mathbf{y}) = 0$ and $d(\mathbf{x}_f, \mathbf{y})$, the distortion metric between $\mathbf{x}_f$ and $\mathbf{y}$,

is small.

6.3.1 Oracle attack for face de-identification

We transpose the oracle attack to the field of face recognition by replacing the watermark detector $D(\cdot) : \mathcal{I} \to \{0,1\}$ by the face verification function $V_{\mathbf{x}_o}(\cdot) : \mathcal{I} \to \{0,1\}$. In the same way, we define $\mathcal{V}_{\mathbf{x}_o} = \{\mathbf{x} | V_{\mathbf{x}_o}(\mathbf{x}) = 1\}$ as the set of images detected as representing the person as in $\mathbf{x}_o$. We aim at optimizing the following problem:

$$\begin{aligned} \mathbf{y}^\star &= \arg \min_{y: V_{\mathbf{x}_o}(\mathbf{y})=0} d(\mathbf{x}_f, \mathbf{y}) && (6.3) \\ &= \arg\min d(\mathbf{x}_f, h_{\mathcal{B}_{\mathbf{x}_o}}(\mathbf{y})) && (6.4) \end{aligned}$$

Where $h_{\mathcal{B}_{\mathbf{x}_o}}(\mathbf{y})$ is a surjection onto the boundary $\mathcal{B}_{\mathbf{x}_o}$ between $\mathcal{V}_{\mathbf{x}_o}$ and $\overline{\mathcal{V}_{\mathbf{x}_o}}$

6.3.2 Main ingredients

We worked with the approach of Earl rather than the method of Comesaña to make less oracle calls (see Sect. 6.2). We present the way we explore the boundary $\mathcal{B}_{\mathbf{x}_o}$ by the synthesis of noise textures, the approximate surjection, and finally the main algorithm.

Texture parametrization. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ be a set of N texture images chosen for their statistical or visual properties detailed later on. The images have pixel values in the range $[0, 255]$. The noise t is computed as a linear combination of the textures of $\mathcal{T}$. Writing the images column wise, $T = [t_1, t_2, \ldots, t_N]$ is the matrix where the i-th column corresponds to the i-th texture image, and:

$$t = \sum_{i=1}^{N} \beta_i t_i = T\beta \tag{6.5}$$

where β contains the coefficients of the linear combination. To ensure that the pixel of t are in $[0, 255]$, we add the constraints: $\beta_i \in [0,1]$, $\forall i$ and $\sum_i \beta_i = 1$. In other words, β lies in the standard $(N-1)$-simplex of $\mathbb{R}^N$, denoted by Δ^{N-1}.

The surjection function. In our method, the forgery $\mathbf{y}$ is computed as the interpolation between the original image $\mathbf{x}_f$ and a noise image t: $\mathbf{y} = (1-\alpha)\mathbf{x}_f + \alpha t$. The distortion function $d(\mathbf{x}_f, \mathbf{y}) = \|\mathbf{x}_f - \mathbf{y}\|^2$ becomes:

$$d(\mathbf{x}_f, \mathbf{y}) = \alpha^2 \|T\beta - \mathbf{x}_f\|^2. \tag{6.6}$$

Algorithm 6.1 Greedy approximation over standard simplex

Input: a function $f(\beta)$
Ouput: an approximate solution β^*
$k \leftarrow 0$
$i_0 \leftarrow random(N)$
$\beta^0 \leftarrow e_{i_0}$
while not converged **do**
  $k \leftarrow k + 1$
  $i_k \leftarrow random(N)$
  $\eta_k \leftarrow linesearch(f(\beta^{k-1} + \eta(e_{i_k} - \beta^{k-1})))$
  $\beta^k \leftarrow \beta^{k-1} + \eta_k(e_{i_k} - \beta^{k-1})$
end while
return β^k

Since $V_{\mathbf{x}_o}(\mathbf{x}_f) = 1$ and $V_{\mathbf{x}_o}(t) = 0$, there exists a value $\alpha(\beta) \in (0, 1]$ which brings $\mathbf{y}$ on the boundary $\mathcal{B}_{\mathbf{x}_o}$. The surjection consists in finding this appropriate value. In practice, this value is approximated by a bisection process as in [30]. With these notations, we aim at solving the following problem:

$$\min_{\beta \in \Delta^{N-1}, V_{\mathbf{x}_o}(\mathbf{x}_f + \alpha(\beta)(T\beta - \mathbf{x}_f)) = 0} \alpha(\beta)^2 \|T\beta - \mathbf{x}_f\|^2 \tag{6.7}$$

Optimization over the standard $(N-1)$-simplex. The region $\mathcal{V}_{\mathbf{x}_o}$ is a priori not a convex set, hence neither $\alpha(\beta)$ nor the functional to be minimized in (6.7) are convex functions. This makes our problem difficult to solve exactly. Again, we follow the same path as Earl in [42] by resorting to a stochastic approximate optimization, detailed in Algorithm 6.1. This is a stochastic variant of coordinate descent and convergence guarantees can be given when the functional is convex [69, 70]. Again, this doesn't hold in our case, but we do observe a convergence to (likely) local minima. This shows that, if region $\mathcal{V}_{\mathbf{x}_o}$ may not be convex, it is certainly piecewise convex.

6.4 Experiments

The overall approach for validation is threefold. (1) very strict experiments are first performed in a so-called *self de-identification* setting. The objective is to forge face images, which, when compared to themselves (forged vs original image), are considered by the face recognition algorithm as representing two different persons. We compare the proposed approach to de-identification by blurring as well to jpeg compression. (2) A method for making our de-identification more robust to simple counter attacks, is proposed, here again within the same self de-identification context. (3) finally, a more realistic (while easier) set of experiments is proposed. The the goal is to de-identify positive face pairs (*i.e.* pairs including two different images representing the same person) by altering one

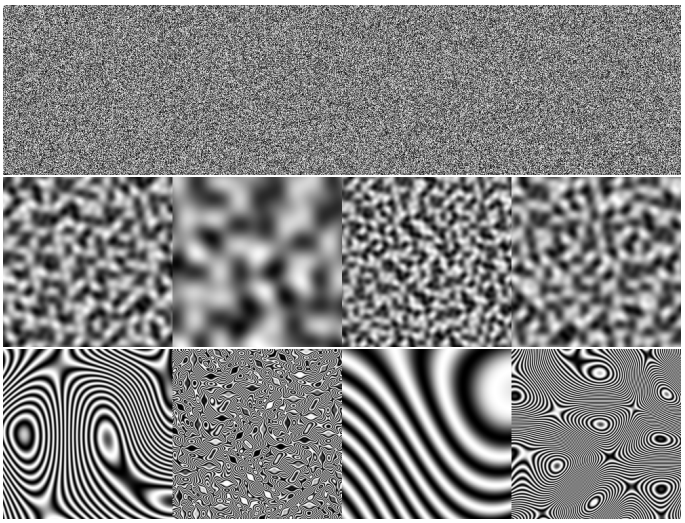


Figure 6.4: Example of noise images used for face de-identification. First row: white noise images. Middle row: Perlin noise images. Last row: examples of sine Perlin noise images.

of the two images, the other being considered as the reference image. In this context, the recognition rate of human subjects on de-identified images is also evaluated and compared to face recognition systems.

All the experimental validations are done with the Labeled Faces in the Wild (LFW) face database [64]. The choice of this database is led by the great variability of faces with respect to the pose, lighting and expression conditions compared to other popular databases like FERET [106]. Faces contained in this dataset are very close to the application context we are interested in, *i.e.* the privacy of face images on social networks.

Regarding face encoding, the I-LQP descriptor [65] is used. While recent methods [23, 119], give slightly better performances, I-LQP has the great advantage of being much faster to compute: about 10 times faster than the Fisher Vector based encoder of [119]. This is an important issue since any call to the oracle implies the signature extraction from a new image.

Finally, we use three types of image noise to alter the image. Since most modern face recognition algorithms rely on statistics of local features, we believe that textures good at de-identification should exhibit energetic components at the same scale as characteristic faces features. Another important point is that those textures should have either recognizable structure or strong stationarity (typically white noise), so that the human brain can easily separate the two channels (noise and face). We test different types of texture and report results for three of them, namely (i) white noise, (ii) Perlin noise and (iii) what we call sine Perlin noise. Some sample images are given in Fig. 6.4. Regarding white noise images, each pixel intensity is obtained by drawing uniform integer in the range $[0, 255]$. Perlin noise is obtained with the modified algorithm of Perlin proposed in [103]. This noise is widely used in computer graphics as the random seed to generate a wide range

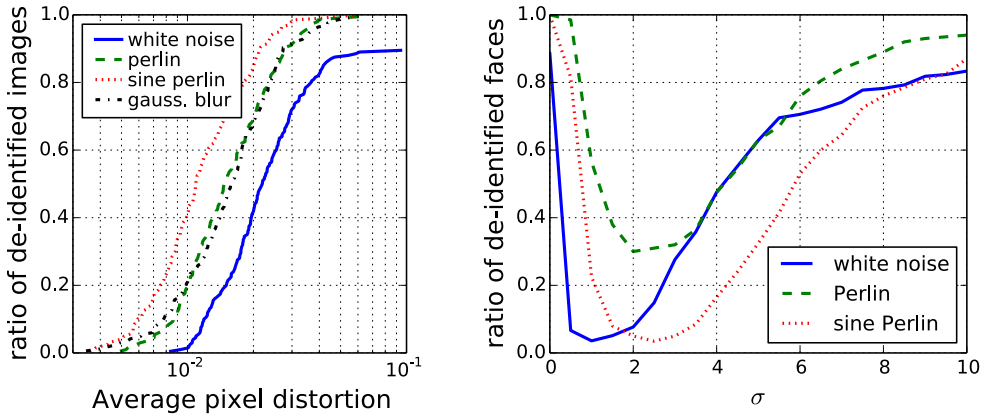


Figure 6.5: Left: Proportion of successfully de-identified faces as a function of the average pixel distortion. Right: Impact of low-pass filtering on textured forgeries.

of useful textures. We adapt the implementation of the `battlestar-tux` game project[1], which works by creating a fixed map *observed* at different locations and scales. There are 3 parameters namely (x, y, s) which correspond to the coordinates of the center of the texture patch and the scale of observation. The last family is a Perlin texture modulated by the sine function. It has the 3 same parameters plus the frequency of the sine. The succession of dark and light lines corresponds to the level sets of the original Perlin texture in Fig. 6.4.

For each image to be de-identified, we draw $N = 50$ texture images of the chosen family sharing the same dimensions as LFW images with randomly chosen sets of parameters. The line search tests 7 values of η anytime requiring a surjection onto the boundary approximated by a bisection with 10 iterations. In total, 3500 calls to the face verification system is needed to de-identify a face image.

6.4.1 Self de-identification

This first set of experiments is a direct application of the method presented in Sect. 6.3 to 220 randomly selected faces from LFW. In this particular case, the reference image $\mathbf{x}_o$ and the starting image $\mathbf{x}_f$ are the same, *i.e.* $\mathbf{x}_o = \mathbf{x}_f$. These experiments used the face recognition of [65], whose detection threshold is set as the optimal threshold on the view 1 of LFW. Note that, since the altered image obtained with our method are just beyond the boundary of the detection region $\mathcal{V}_{\mathbf{x}_o}$, slight variations could bring them back into $\mathcal{V}_{\mathbf{x}_0}$. To avoid this, we actually multiply the value of $\alpha(\beta)$ by a factor 1.1 before applying the final texture to the image. Fig. 6.5 (Left) shows the cumulative distribution of the proportion of de-identified images as a function of the average pixel distortion ($\frac{\|\mathbf{x}_f - \mathbf{y}\|^2}{255^2 P}$ with P the number of pixels).

[1] `http://code.google.com/p/battlestar-tux/`

Figure 6.6: Examples of de-identified images for, from top to bottom, white noise, Perline and sine Perlin textures. The bottom line shows images de-identified using gaussian blur.

The sine Perlin textures de-identify a larger amount of images with less distortion that the other textures. Note also that the white noise textures saturate on the graph at a value lower than 1.0. This means that some images could not be de-identified using white noise. Fig. 6.6 displays some sample images in this context. De-identification by blurring clearly prevents human identification. This is not the case with the oracle attack using Perlin sine noise. However, the visual quality is not so good. A simple trick could be to remove manually the noise except over the faces to be de-identified.

Overall, the oracle attack performs betters but achieves mitigated results when $\mathbf{x}_o = \mathbf{x}_f$. Nevertheless, this setup is very pessimistic: users usually post new pictures on social networks, and therefore these images cannot play the role of $\mathbf{x}_o$. In other words, it is hopeless to de-identify an already published picture while preserving visual quality because face verification algorithms are too robust.

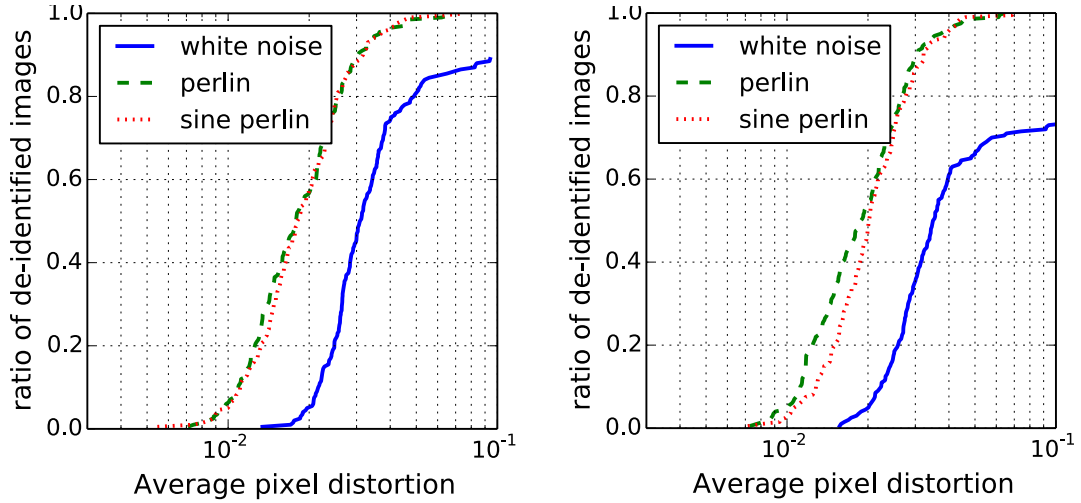


Figure 6.7: Proportion of successful de-identification as a function of the average pixel distortion, when robustness to low-pass filtering is enforced for $\Sigma = \{2\}$ (left) and $\Sigma = \{2, 4\}$ (right).

6.4.2 Improving robustness to simple counter attacks

To test the robustness of our de-identification process, we examine if simple counter attacks could be applied by the face verification system. Since our method is likely to introduce high frequency components, a simple low-pass filter could make the forged images recognizable again.

We test a simple gaussian filter parametrized by its standard deviation σ. The right part of Fig. 6.5 shows the evolution of the rate of forged images successfully identified as a function of the σ parameter for the three texture families. The rate falls down at small σ because the filter succeeds in partially removing the noise while preserving the face, and then rises up at bigger σ because the filter blurs so much the image that the face can't be recognized. Perlin noise is clearly more robust to this counter attack. Our explanation is that its spectrum (density of power over frequencies) resembles more the spectrum of the face images, so that the filter has more difficulty in separating the face and the noise.

We partially achieve robustness against this counter attack by explicitly taking it into account during the oracle attack. We choose a set of S scale parameters $\Sigma = \{\sigma_i\}_{i=1}^{S}$ for which we want to enforce de-identification, and the oracle attack considers this new verification region: $\mathcal{V}_{\mathbf{x}_o,\Sigma} = \mathcal{V}_{\mathbf{x}_o} \cup \mathcal{V}_{\mathbf{x}_o,\sigma_1} \cup \cdots \cup \mathcal{V}_{\mathbf{x}_o,\sigma_S}$ where $\mathcal{V}_{\mathbf{x}_o,\sigma}$ is the set of images which are identified as the person of $\mathbf{x}_o$ after the filtering by a Gaussian kernel of deviation σ. Since $\mathcal{V}_{\mathbf{x}_o} \subset \mathcal{V}_{\mathbf{x}_o,\Sigma}$, $\mathbf{y}^\star$ may get further away from $\mathbf{x}_f$ resulting in a greater or equal average pixel distortion, as shown in Fig. 6.7. When the standard deviation of the low-pass filter used at the face verification side belongs to Σ, we achieve a perfect robustness as the rate of de-identification equals 1, otherwise we drastically reduce the impact of the counter attack as shown in Fig. 6.8.

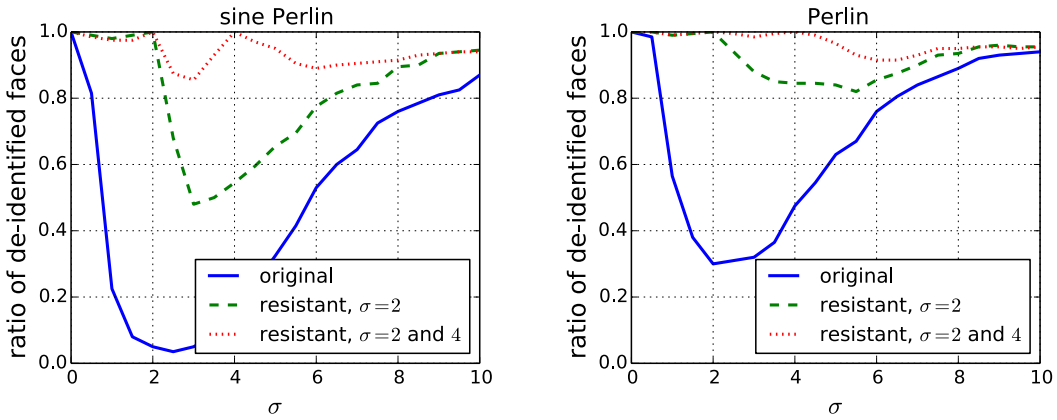


Figure 6.8: Impact of low-pass filtering on robust forgeries textured with Perlin and sine Perlin.

Figure 6.9: Images de-identified using a different reference image with Perlin noise. Rows correspond to images respectively taken around 25%, 50% and 75% of the average distortion distribution.

6.4.3 De-identification of image pairs

This new set of experiments is closer to the targeted use case. The objective is now to forge an image which can't be matched with a different image of the same person: $\mathbf{x}_f \neq \mathbf{x}_o$. These experiments take the image pairs of LFW, which are correctly detected as positive by the face verification algorithm [65]. The amount of noise necessary for the de-identification is lower than in Sect. 6.4.1 since the two face images are already different (in illumination, pose, etc.). Indeed, the noise is almost invisible for the pairs 'on the edge' of the face verification capacity (see the first row of Fig. 6.9). Comparatively, Fig. 6.10

Figure 6.10: Example of images not de-identified by JPEG compression even with the lowest quality factor. We show pairs of the reference and the compressed image of the same person.

shows that a JPEG compression not only introduces very annoying blocky artefacts, but also fails in de-identifying any image! The average pixel distortion spreads over a wider range of values (see Fig. 6.11 (Left)), and is globally weaker than under the $\mathbf{x}_o = \mathbf{x}_f$ setup: Half of the images are de-identified at a distortion lower than $\approx 2.10^{-3}$ whereas the median is at $\approx 2.10^{-2}$ in Fig. 6.7 (Right).

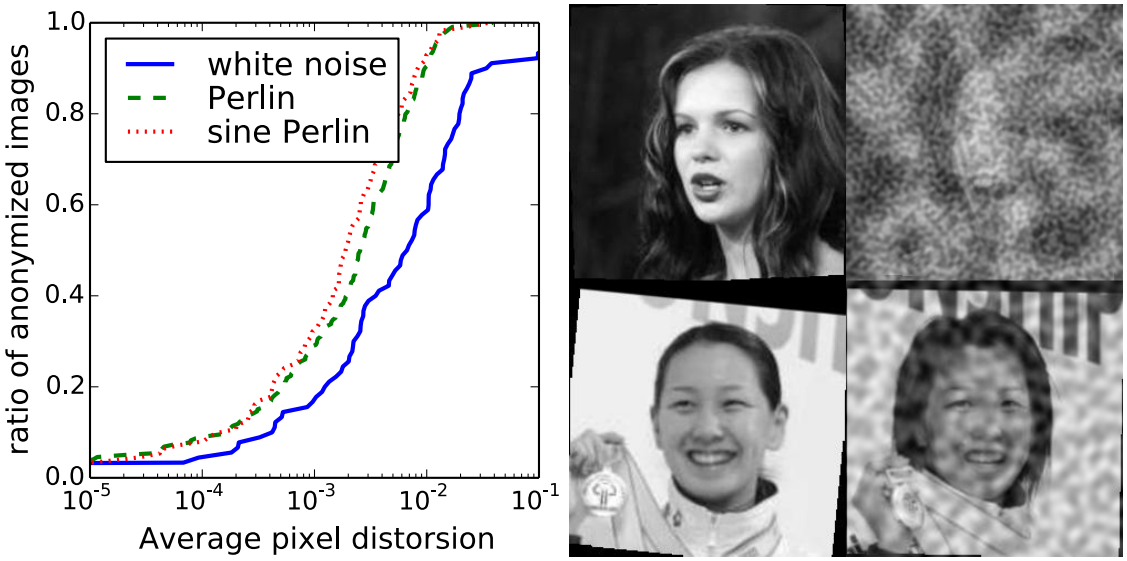


Figure 6.11: Left: Proportion of successfully de-identification vs. the average pixel distortion, with robustness to low pass filtering ($\Sigma = \{2, 4\}$). Right: Examples of pairs where human failed.

The other motivation for these experiments is to assess whether humans still recognize people in the images forged by our method. Pairs of images $(\mathbf{x}_o, \mathbf{y})$ are shown to humans, who are asked to evaluate if both the images are from same person or not. We display 200

	ground +	ground −
human +	93.2	7.2
human −	6.8	92.8

Table 6.1: Confusion matrix of the classification of 100 *positive and* 100 *negative pairs by a group of 117 people. Figures correspond to number of pairs averaged over all the humans.*

random pairs of images with an equal number of positive and negative pairs of images in random order to a group of 117 people.

From Table 6.1, we see that, on average, our method does not alter too much the facial features important for the human brain. However, there are few pairs where our method still damages too much the image: these are the pairs where the conditions are similar (same facial expression, same illumination, same pose *etc.*) as shown in Fig. 6.11 (Right). To our knowledge, such investigations have not been pursued before.

6.5 Conclusion

Motivated by the need of tools for privacy protection on the Internet, in this Chapter we presented a novel approach for the de-identification of face images, *i.e.* for preventing automatic matching with public face collections (*e.g.* faces found on the Internet) while preserving their visual aspect and letting them recognizable by human beings. Experiments show that our method achieves this goal most of the times, provided that the oracle attack uses a close enough version of the face verification system.

Chapter 7

Conclusions and Future works

In this thesis we present some of the methods to represent faces for both identification (identity, age, expressions *etc.*) and de-identification (identity removal) tasks. Compact and discriminative features are required to work in large scale setups while alignment of features from multiple domains are necessary to generate domain invariant representations. Moreover, effective organization of face database is also necessary to work in large scale to reduce the time complexity. In the following, we summarize our contributions and discuss some extensions of the work.

7.1 Hierarchical Metric Learning

To address the problem of large search space in large scale identity based face-retrieval, in Chapter 2, we propose a hierarchical metric learning method to group faces based on their common characteristics. The degree of closeness between the faces ranges from coarse characteristics such as sex (*male or female*) to fine-grained characteristics such as persons looking alike (*round face, long face, square face* etc.). We learn multiple metrics in inverted tree fashions. With the increase in depth of the tree, the degree of common characters between the people increases. From the quantitative evaluations, we draw the conclusions that the proposed method is more accurate than the compared baseline metric learning method and is efficient by a large factor (up to $10\times$). Moreover, the qualitative results show that faces bearing common attributes such as *sex, wearing glasses, bald, bangs etc.* are in the same group.

Future Works. In this work, we learn the parameters of multiple metrics in an inverted binary tree fashion. Each metric learning is followed by unsupervised K-means clustering (except for leaves nodes where we learn only projection matrix). Hard clustering techniques such as K-means can assign wrong cluster label to the faces which are equally likely to belong in either of the clusters. This kind of wrong assignments of cluster results

into degrade in performance. In order to improve the accuracy, it will be interesting to exlplore different soft clustering techniques.

7.2 Multi-task Metric Learning

To overcome the problems due to variations in ages and expressions of a person in his/her face analysis, we proposed to address these issues at training time by including auxiliary tasks, in Chapter 3. We proposed a multi-task metric learning method. A common matrix, sharing parameters with all the tasks in addition to task-specific matrices were learned at train time to minimize a loss function. We evaluated the proposed method for identity and age-based face retrieval in large scale. From both quantitative and qualitative results, the proposed method outperforms the compared competitive baselines and existing state-of-the-art methods.

Future Works. In this work, we only used a pair of tasks (a main task and an auxiliary task) to learn the parameters. We wanted to see how expressions matching and age matching as auxiliary tasks will influence the performance of identity matching – the main task — and vice versa. It will be interesting to revisit this work with more auxiliary tasks and evaluate the impact on the performance. Moreover, applying deep learning framework on the proposed method will also make an interesting future work.

7.3 Cross-domain Age Estimation

In Chapter 4, we address the problem of domain discrepancy for age estimation among people from different races and sexes. It has been observed that the rate of ageing between the people from different races and sexes is different. As it is hard to collect annotated examples to learn an age prediction model for each and every group of people, we proposed a method to use annotated examples from a domain which is rich in it, called source domain, to improve the performance of another domain where the annotated examples are limited, called target domain. We proposed a joint metric / regressor learning method to address this problem. We generated cross-domain 'same' and 'different' pairs to learn the parameters of projection matrix and those of the regressor to minimize the mean age error jointly. Metric learning helps to generate compact, aligned features while the regressor learns to predict correct ages on these features. Our extensive experiments show that the proposed method outperforms existing state-of-the-art methods and competitive baselines by a large margin.

Future Works. In this work, we only considered the case of a source and a target in a pair. It would be interesting to explore, multi-source and multi-target pairs and train a

model for multiple domains. In addition to it, extending our method to deep-architecture will be an interesting direction to explore.

7.4 Deep Fusion of Visual Signatures

In Chapter 5, we proposed a novel deep neural network to fuse multiple types of features. This network can feed any number of any types of features at a time and the parameters of the network are still optimal to make correct predictions. Moreover, when an unseen type of feature is encountered and need to adapt in the network, it can be done simply by learning parameters corresponding to this feature with the reference to the common parameters of the network. This avoids the re-training of the whole network. We have evaluated the performance of our network for facial attributes prediction. Compared to competitive baselines and existing best performing method, the proposed method attains a state-of-the-art performance.

Future Works. An immediate extension to this method will be a network to fuse multiple types of features in multi-task setups. In addition to this, exploring other techniques to fuse features than uniformly weighted average makes an interesting future work.

7.5 Face De-identification

In Chapter 6 we proposed, a joint learning method to de-identify the faces from automatic face matching algorithms and still identifiable from human beings. Our experimental results, show that the proposed method is able to represent faces in such a way that faces are completely de-identified by state-of-the-art face matching algorithms while the human beings were able to identify easily.

Future Works. In future, exploring tools and techniques to improve the quality of de-identified images will be interesting. Moreover, since the robustness of automatic analysing technology is rapidly increasing, exploring novel techniques to counter such technologies will be an interesting direction of research.

Appendix A

Publications

Some of the empirical results, figures, algorithms *etc.* presented in thesis have already been published in the proceedings of major international conferences. These can be found in our following publications.

- Bhattarai, Binod, Gaurav Sharma, and Frédéric Jurie. **"CP-mtML: Coupled Projection multitask Metric Learning for Large Scale Face Retrieval"** IEEE conference on Computer Vision and Pattern Recognition (*CVPR*), 2016.
- Bhattarai, Binod, Gaurav Sharma, Alexis Lechervy, and Frédéric Jurie. **"A joint learning approach for cross-domain age estimation"** 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (*ICASSP*), 2016 (**Best Student Paper** on Image, Video Multidimensional (IVM) Signal Processing)
- Bhattarai, Binod, Gaurav Sharma, and Frédéric Jurie. **"Deep fusion of visual signatures for client-server facial analysis"** Indian conference on Computer Vision, Graphics, and Image Processing (*ICVGIP*), 2016 (**Best Paper Award Runner-up**)
- Bhattarai, Binod, Gaurav Sharma, Frédéric Jurie and Patrick Pérez.**"Some faces are more equal than others: Hierarchical organization for accurate and efficient large-scale identity-based face retrieval"** European Conference on Computer Vision (*ECCV*) Workshops, 2014.
- Bhattarai, Binod, Alexis Mignon, Frédéric Jurie and Teddy Furon. **"Puzzling face verification algorithms for privacy protection"** IEEE International Workshop on Information Forensics and Security (*WIFS*), 2014.

List of Figures

List of Tables

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, 2004.

[2] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *PAMI*, 28(12), 2006.

[3] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila. Recognition of blurred faces using local phase quantization. In *ICIP*, 2008.

[4] F. Alnajar, Z. Lou, J. Alvarez, and T. Gevers. Expression-invariant age estimation. In *British Machine Vision Conference (BMVC)*, 2014.

[5] I. Amerini, M. Barni, R. Caldelli, and A. Costanzo. SIFT keypoint removal and injection for countering matching-based image forensics. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, IH&MMSec '13, pages 123–130, New York, NY, USA, 2013. ACM.

[6] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 2005.

[7] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[8] A. Bar-Hillel, T. Hertz, N. Shental, D. Weinshall, and G. Ridgeway. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(6), 2005.

[9] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz. Fast high dimensional vector multiplication face recognition. In *International Conference on Computer Vision (ICCV)*, 2013.

[10] A. Bedagkar-Gala and S. K. Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing (IVC)*, 32(4):270–286, 2014.

[11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 1997.

[12] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *arXiv:1306.6709*, 2013.

[13] T. Berg and P. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[14] T. Berg and P. N. Belhumeur. Tom-vs-pete classifiers and identity-preserving alignment for face verification. In *BMVC*, volume 2, page 7. Citeseer, 2012.

[15] T. Berg and P. N. Belhumeur. POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[16] B. Bhattarai, G. Sharma, F. Jurie, and P. Pérez. Some faces are more equal than others: Hierarchical organization for accurate and efficient large-scale identity-based face retrieval. In *European Conference on Computer Vision (ECCV) Workshops*, pages 160–172, 2014.

[17] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B. A. Draper. The csu face identification evaluation system: its purpose, features, and structure. In *Proceedings of the 3rd international conference on Computer vision systems*, ICVS'03, pages 304–313, Berlin, Heidelberg, 2003. Springer-Verlag.

[18] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *International Conference on Computer Vision (ICCV)*, 2013.

[19] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision (IJCV)*, 107(2):177–190, 2014.

[20] R. Caruana. Multitask learning. *Journal of Machine Learning Research*, 1997.

[21] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. A ranking approach for human ages estimation based on face images. In *ICIP*, pages 3396–3399, 2010.

[22] B.-C. Chen, C.-S. Chen, and W. H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *European Conference on Computer Vision (ECCV)*, pages 768–783. Springer, 2014.

[23] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[24] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Special Interest Group in Information Retrieval*, 2006.

[25] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Special Interest Group in Information Retrieval*, 2006.

[26] K. Chen, S. Gong, T. Xiang, and C. C. Loy. Cumulative attribute space for age and crowd density estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2467–2474, 2013.

[27] R. G. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in tv video. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1559–1566. IEEE, 2011.

[28] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning (ICML)*, 2008.

[29] P. Comesaña, L. Pérez-Freire, and F. Pérez-González. The return of the sensitivity attack. In *Proceedings of the 4th international conference on Digital Watermarking*, IWDW'05, pages 260–274, Berlin, Heidelberg, 2005. Springer-Verlag.

[30] P. Comesaña, L. Pérez-Freire, and F. Pérez-González. Blind Newton Sensitivity Attack. *Information Security, IEE Proceedings*, 153(3):115–125, 2006.

[31] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[32] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning (ICML)*, 2007.

[33] J. Deng, A. C. Berg, and L. Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 785–792. IEEE, 2011.

[34] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 2011.

[35] I. S. Dhillon and J. A. Tropp. Matrix nearness problems with bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007.

[36] T.-T. Do, L. Amsaleg, E. Kijak, and T. Furon. Security-Oriented Picture-In-Picture Visual Modifications. In *ICMR - ACM International Conference on Multimedia Retrieval*, Hong-Kong, China, 2012.

[37] T.-T. Do, E. Kijak, L. Amsaleg, and T. Furon. Enlarging hacker's toolbox: deluding image recognition by attacking keypoint orientations. In *ICASSP - 37th International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, Mar. 2012. IEEE.

[38] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg. Deluding Image Recognition in SIFT-based CBIR Systems. In *ACM Multimedia in Forensics, Security and Intelligence*, Firenze, Italy, Oct. 2010. ACM.

[39] B. Driessen and M. Dürmuth. Achieving anonymity against major face recognition algorithms. In *Communications and Multimedia Security*, pages 18–33, 2013.

[40] L. Du and H. Ling. Cross-age face verification by coordinating with cross-face age verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[41] F. Dufaux and T. Ebrahimi. A framework for the validation of privacy protection solutions in video surveillance. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, 2010.

[42] J. W. Earl. Tangential sensitivity analysis of watermarks using prior information. In *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX*, pages 650519–650519–12, 2007.

[43] N. C. Ebner, M. Riediger, and U. Lindenberger. Faces—a database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior research methods*, 2010.

[44] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy"–automatic naming of characters in tv video. In *BMVC*, 2006.

[45] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, 2013.

[46] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[47] C. Geng and X. Jiang. Face recognition using sift features. In *ICIP*, 2009.

[48] X. Geng, Z.-H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *PAMI*, 29(12):2234–2240, 2007.

[49] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[50] R. Gross and L. Sweeney. Towards Real-World Face De-Identification. In *First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, 2007.

[51] R. Gross, L. Sweeney, F. D. la Torre, and S. Baker. Semi-supervised learning of multi-factor models for face de-identification. In *CVPR*. IEEE Computer Society, 2008.

[52] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

[53] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *International Conference on Computer Vision (ICCV)*, pages 498–505, 2009.

[54] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *International Conference on Computer Vision (ICCV)*, 2009.

[55] G. Guo and G. Mu. Human age estimation: What is the influence across race and gender? In *Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 71–78, 2010.

[56] G. Guo and G. Mu. Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression. In *Computer Vision and Pattern Recognition (CVPR)*, pages 657–664, 2011.

[57] G. Guo, G. Mu, Y. Fu, C. Dyer, and T. Huang. A study on automatic age estimation using a large database. In *International Conference on Computer Vision (ICCV)*, pages 1986–1991, 2009.

[58] G. Guo, G. Mu, Y. Fu, and T. S. Huang. Human age estimation using bio-inspired features. In *Computer Vision and Pattern Recognition (CVPR)*, pages 112–119, 2009.

[59] G. Guo and C. Zhang. A study on cross-population age estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4257–4263, 2014.

[60] H. Han, C. Otto, and A. K. Jain. Age estimation from face images: Human vs. machine performance. In *ICB*, pages 1–8, 2013.

[61] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *PAMI*, 2005.

[62] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer Vision, Graphics and Image Processing*, pages 58–69. Springer, 2006.

[63] Y. Hu, L. Manikonda, S. Kambhampati, et al. What we instagram: A first analysis of instagram photo content and user types. In *ICWSM*, 2014.

[64] 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. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[65] S. U. Hussain, T. Napoléon, and F. Jurie. Face recognition using local quantized patterns. In *British Machine Vision Conference (BMVC)*, 2012.

[66] S. J. Hwang, K. Grauman, and F. Sha. Semantic kernel forests from multiple taxonomies. In *Advances in Neural Information Processing Systems*, pages 1718–1726, 2012.

[67] Instagram. `https://www.instagram.com/press/`, 30/07/2016.

[68] T. S. Jaakkola, D. Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems (NIPS)*, 1999.

[69] M. Jaggi. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zürich, 2011.

[70] M. Jaggi. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *Proc. of Int. Conf. on Machine Learning (ICML)*, pages 427–435, 2013.

[71] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[72] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[73] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *PAMI*, 33(1):117–128, 2011.

[74] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, 2014.

[75] J. Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 2008.

[76] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, **c**C. Gül**c**cehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In *International conference on Multimodal Interaction*. ACM, 2013.

[77] S. Kang, D. Lee, and C. D. Yoo. Face attribute classification using attribute-aware correlation map and gated convolutional neural networks. In *ICIP*, 2015.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[79] B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

[80] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[81] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *European Conference on Computer Vision (ECCV)*. 2008.

[82] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *International Conference on Computer Vision (ICCV)*, 2009.

[83] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. *PAMI*, 33(10):1962–1977, 2011.

[84] M. Lapin, B. Schiele, and M. Hein. Scalable multitask representation learning for scene classification. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[85] E. Learned-Miller. `http://vis-www.cs.umass.edu/lfw/results.html`, 2013.

[86] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua. Labeled faces in the wild: A survey. In *Advances in Face Detection and Facial Image Analysis*. 2016.

[87] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic matching for pose variant face verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[88] P. Liu, J. T. Zhou, I. W.-H. Tsang, Z. Meng, S. Han, and Y. Tong. Feature disentangling machine-a novel approach of feature selection and disentangling in facial expression analysis. In *European Conference on Computer Vision (ECCV)*, 2014.

[89] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision (ICCV)*, 2015.

[90] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou. Neighborhood repulsed metric learning for kinship verification. *PAMI*, 2014.

[91] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *ICML*, 2013.

[92] N. Mesgarani and E. F. Chang. Selective cortical representation of attended speaker in multi-talker speech perception. *Nature*, 485(7397):233–236, 2012.

[93] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[94] R. Negrel, A. Lechervy, and F. Jurie. Boosted metric learning for efficient identity-based face retrieval. In *British Machine Vision Conference (BMVC)*, 2015.

[95] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. In *PAMI*, 2016.

[96] E. M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *Knowledge and Data Engineering, IEEE Transactions on*, 17(2):232–243, 2005.

[97] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Computer Vision–ACCV 2010*, pages 709–720. Springer, 2011.

[98] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002.

[99] V. Ojansivu. *Blur invariant pattern recognition and registration in the Fourier domain*. PhD thesis, 2009.

[100] S. Parameswaran and K. Q. Weinberger. Large margin multi-task metric learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[101] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference (BMVC)*, 2015.

[102] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[103] K. Perlin. Improving noise. *ACM Trans. Graph.*, 21(3):681–682, July 2002.

[104] F. Perronnin and D. Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[105] M. Perrot, A. Habrard, D. Muselet, and M. Sebban. Modeling perceptual color differences by local metric learning. In *European Conference on Computer Vision (ECCV)*, 2014.

[106] J. P. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[107] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer vision using local binary patterns*, volume 40. Springer, 2011.

[108] J. Pu, Y.-G. Jiang, J. Wang, and X. Xue. Which looks like which: Exploring inter-class relationships in fine-grained visual categorization. In *European Conference on Computer Vision (ECCV)*, 2014.

[109] K. Ricanek Jr and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. 2006.

[110] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. In *AISTATS*, 2012.

[111] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, pages 213–226. Springer, 2010.

[112] R. Salakhutdinov and G. E. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007.

[113] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[114] C. Shan, F. Porikli, T. Xiang, and S. Gong. *Video Analytics for Business Intelligence*, volume 409. Springer, 2012.

[115] G. Sharma and F. Jurie. Local higher-order statistics (LHS) describing images with statistics of local non-binarized pixel patterns. *Computer Vision and Image Understanding (CVIU)*, 2016.

[116] G. Sharma, S. ul Hussain, and F. Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. In *European Conference on Computer Vision (ECCV)*, 2012.

[117] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition*, 2002.

[118] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference (BMVC)*, 2013.

[119] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference (BMVC)*, 2013.

[120] F. Song, X. Tan, and S. Chen. Exploiting relationship between attributes for improved face verification. *Computer Vision and Image Understanding (CVIU)*, 2014.

[121] Z. Song, B. Ni, D. Guo, T. Sim, and S. Yan. Learning universal multi-view age estimator using video context. In *International Conference on Computer Vision (ICCV)*, pages 241–248, 2011.

[122] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[123] J. Struyf and S. Dvzeroski. Clustering trees with instance level constraints. In *ECML*, 2007.

[124] J.-H. Sublemontier, L. Martin, G. Cleuziou, and M. Exbrayat. Integrating pairwise constraints into clustering algorithms: optimization-based approaches. In *Data*

Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, pages 272–279. IEEE, 2011.

[125] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

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

[127] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *International Conference on Computer Vision (ICCV)*, 2013.

[128] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[129] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[130] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[131] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[132] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *TIP*, 19(6):1635–1650, 2010.

[133] P. Thukral, K. Mitra, and R. Chellappa. A hierarchical approach for human age estimation. In *ICASSP*, pages 1529–1532, 2012.

[134] L. Torresani and K.-c. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[135] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 1991.

[136] T. Uricchio, M. Bertini, L. Seidenari, and A. Bimbo. Fisher encoded convolutional bag-of-windows for efficient image retrieval and social image tagging. In *ICCVW*, 2015.

[137] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[138] Venturebeat. `http://venturebeat.com/2016/04/27/facebook-passes-1-65-...`/, 27/04/2016.

[139] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair. Learning hierarchical similarity metrics. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2280–2287. IEEE, 2012.

[140] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004.

[141] J. Wang, A. Kalousis, and A. Woznica. Parametric local metric learning for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2012.

[142] S. Wang, S. Jiang, Q. Huang, and Q. Tian. Multi-feature metric learning with knowledge transfer among semantics and social tagging. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[143] X. Wang and I. Davidson. Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 563–572. ACM, 2010.

[144] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 2009.

[145] Y. Welinder. A face tells more than a thousand posts: Developing face recognition privacy in social networks. *Harvard Journal of Law and Technology*, 26(1), 2012.

[146] R. Weng, J. Lu, J. Hu, G. Yang, and Y.-P. Tan. Robust feature set matching for partial face recognition. In *International Conference on Computer Vision (ICCV)*, December 2013.

[147] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Computer Vision–ACCV 2009*, pages 88–97. Springer, 2009.

[148] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue. Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014.

[149] X. Xie. A review of recent advances in surface defect detection using texture analysis techniques. *Electronic Letters on Computer Vision and Image Analysis*, 7(3):1–22, 2008.

[150] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[151] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[152] P. Yang, K. Huang, and C.-L. Liu. Geometry preserving multi-task metric learning. *Machine learning*, 92(1):133–175, 2013.

[153] P. Yang, K. Huang, and C.-L. Liu. A multi-task framework for metric learning with common subspace. *Neural Computing and Applications*, 22(7-8):1337–1347, 2013.

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

[155] H. Zeng, A. Song, and Y. M. Cheung. Improving clustering with pairwise constraints: a discriminative approach. *Knowledge and information systems*, 36(2):489–515, 2013.

[156] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[157] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[158] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision (ECCV)*, 2014.

[159] L. Zheng and T. Li. Semi-supervised hierarchical clustering. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 982–991. IEEE, 2011.

Développement de méthode de rapprochement physionomique par apprentissage machine.

Ce travail présenté dans cette thèse se déroule dans le contexte général de l'appariement de visage. Plus précisément, notre but est de concevoir et de développer de nouveaux algorithmes pour apprendre des représentations compactes, discriminatives, invariantes au domaine ou de prévenir l'identification de visages. La recherche et d'indexation de visages ouvre la porte à des nombreuses applications intéressantes. Cependant, cela est devenu, jour après jour, plus difficile en raison de la croissance rapide du nombre de visages à analyser. La représentation des visages par des caractéristiques compactes et discriminatives est, par conséquent, essentielle pour en traiter cette ensemble de données très volumineux. De plus, ce volume augmente sans limites apparentes; C'est pourquoi il est également pertinent de proposer des solutions pour organiser les visages de fa**c**con sémantique, afin de réduire l'espace de recherche et d'améliorer l'efficacité de la recherche.
Bien que le volume de visages disponibles sur Internet augmente, il est encore difficile de trouver des exemples annotés pour former des modèles pour chaque cas d'utilisation possible, par exemple, pour la classification de différentes races, sexes, etc. L'apprentissage d'un modèle avec des exemples construites à partir d'un groupe de personnes peut ne nécessairement pas prédire correctement les exemples d'un autre groupe en raison, par exemple, du taux inégal entre exu de changements de dimensions biométriques produites par le vieillissement. De même, un modèle obtenu d'un type de caractéristique peut échouer à faire de bonnes prédictions lorsqu'il est testé avec un autre type de fonctionnalité. Il serait idéal d'avoir des modèles produisant des représentations de visage qui seraient invariables à ces écarts. Apprendre des représentations communes aide finalement à réduire les paramètres spécifiques au domaine et, encore plus important, permet d'utiliser des exemples construites par un domaine et utilisés dans d'autres. Par conséquent, il est nécessaire de concevoir des algorithmes pour cartographier les caractéristiques de différents domaines à un sous-espace commun, qui amène des visages portant les mêmes propriétés à être représentes plus prochement.
D'autre part, comme les outils automatiques de mise en correspondance de visage sont de plus en plus intelligents, il y a une menace croissante sur la vie privée. La popularité du partage de photos sur les réseaux sociaux a exacerbé ce risque. Dans un tel contexte, modifier les représentations des visages de fa**c**con à ce que les visages ne puissent pas être identifiés par des correspondants automatiques - alors que les visages semblent ne pas être modifiés -est devenu une perspective intéressante en matière de protection de la vie privée. Il permet aux utilisateurs de limiter le risque de partager leurs photos dans les réseaux sociaux.
Dans tous ces scénarios, nous avons exploré comment l'utilisation des méthodes d'apprentissage métrique (Metric Learning) ainsi que celles d'apprentissage profond (Deep Learning) peuvent nous aider à apprendre les représentations compactes et discriminantes des visages. Nous construisons ces outils en proposant des représentations compactes, discriminatives, invariantes au domaine et capables de prévenir l'identification de visages.
Nous avons appliqué les méthodes proposées sur une large gamme d'applications d'analyse faciale. Ces applications comprennent: recherche de visages à grande échelle, estimation de l'âge, prédictions d'attribut et identification de l'identité. Nous avons évalué nos algorithmes sur des ensembles de données publics standard et stimulants tels que: LFW, CelebA, MORPH II etc. De plus, nous avons ajouté des visages 1M de Flicker.com à LFW et généré un jeu de données nouveau et plus difficile à évaluer nos algorithmes en grande-échelle. Nos expériences montrent que les méthodes proposées sont plus précises et plus efficaces que les méthodes de références comparées et les méthodes de l'état de l'art et atteignent de nouvelles performances de pointe.

Keywords:Analyse faciale; Apprentissage métrique; Apprentissage profond; Apprentissage conjoint; Apprentissage multi-tâches

Discipline: Informatique et applications
Laboratoire: Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen - GREYC CNRS UMR 6072, Sciences 3, Campus 2, Bd Marechal Juin, Université de Caen, 14032 Caen