Dataset bias exposed in face verification

Eric López-López, Xosé M. Pardo, Carlos V. Regueiro, Roberto Iglesias and Fernando E. Casado

Dataset Bias Exposed in Face Verification

Eric Lopez-Lopez*[1], Xosé M. Pardo[2], Carlos V. Regueiro[1], Roberto Iglesias[2], and Fernando E. Casado[2]

[1] *Universidade da Coruña, CITIC, Spain*
[2] *CiTIUS, Universidade de Santiago de Compostela, Spain*

Submitted: October 2018

Abstract

Most facial verification methods assume that training and testing sets contain independent and identically distributed samples, although, in many real applications, this assumption does not hold. Whenever gathering a representative dataset in the target domain is unfeasible, it is necessary to choose one of the already available (source domain) datasets.

In this paper, a study was performed over the differences among six public datasets, and how this impacts on the performance of the learned methods. In the considered scenario of mobile devices, the individual of interest is enrolled using a few facial images taken in the operational domain, while training impostors are drawn from one of the public available datasets.

This work tried to shed light on the inherent differences among the datasets, and potential harms that should be considered when they are combined for training and testing. Results indicate that a drop in performance occurs whenever training and testing are done on different datasets compared to the case of using the same dataset in both phases. However, the decay strongly depends on the kind of features. Besides, the representation of samples in the feature space reveals insights into to what extent bias is an endogenous or an exogenous factor.

1 Introduction

One of the first decisions to make when addressing a new visual classification problem is about whether building a new dataset, or using some of the already available ones, to build good representations and train strong classifiers from labelled data. Whenever gathering a representative set of samples in the target domain is an unfeasible option, it is necessary to choose the most convenient one among others already available (source domain) datasets. Unfortunately, this selection is not a minor task, because most of these datasets are created by extensive human effort and there is a high chance that they do not cover the diversity of real-world scenarios [24]. Therefore, source and target domains distributions can be quite different [36, 39, 42]. Indeed, several studies have demonstrated that system based on the analysis of facial features can discriminate based and gender, due to the substantial disparities in the accuracy of classifying different demographic cohorts as a consequence of a bias in datasets [2].

In the case of face recognition, systems have to learn to select the identity-carrying features μ (facial characteristics, ethnicity, gender, etc.) for further classification, while discard the identity-irrelevant ones, ϵ (haircut, makeup, injuries, aging, illumination, pose, etc.) [4]. Several face datasets have been built over the last decades which have been of paramount importance for the progress in the general field of face recognition, and have paved the way to the development of (data intensive) deep learning methods, which have achieved impressive performance [4, 7, 13, 17]. Nonetheless, face recognition in unconstrained and data-scarce real-world domains remains a challenging task. Particularly, in face verification on mobile devices, where given two images the objective is to determine whether they belong to the same person, specific difficulties arise. First, it is necessary to deal with a limited number of samples available during enrolment, or even just one, to build each target model. Second, whenever environments are non-stationary in terms of target-individual appearance, the hardware itself, or both, the probabilistic properties of the data change over time, and demand a continuous learning process. And third, even the largest datasets often cannot provide comprehensive coverage of the characteristic of interest for these specific contexts, i.e. they are biased regarding the specific context of the application.

Face verification can be applied to very different target domains (e.g. biometrics in mobile devices, video-surveillance, or mugshot verification) whose data distributions are not only different from the one in the public domain face datasets but are also different among them. For instance, images generated by the users of a mobile device, equipped with a non-collaborative face verification system, are dependent on users' behaviour and habits. This can generate a bias related to ϵ, towards specific perspective distortion caused by specific pose (predominance of a certain way of holding the device), or preference for wearing makeup, just to mention a few examples. For its part, in the context of video recognition, the spectrum of

*Corresponding author: eric.lopez@udc.es camera poses and resolutions, scales or blurring effects make the distribution of features of frames captured by video-surveillance cameras so different from the one of the high-quality and pose-constrained mugshots.

If a system aimed to authenticate the owner of a mobile device, were trained using negative samples drawn from a general (i.e. with a diverse pose, illumination and capturing conditions) face dataset, some really identity-irrelevant features could be misleadingly identified as relevant ones. Thus, some μ features would be considered as part of ϵ features, and verification would be partially based on context-specific features, thereby leading to poor performance.

To tackle these issues, datasets are usually augmented by different means (generating synthetic faces images [12, 18]; applying several transformations to enrich the dataset with new simulated poses, resolutions, blurring effects, or lighting [6, 27, 30]), or alternatively, combined through domain-adaptation approaches [21, 35, 42, 45].

Although the problem of dataset bias was not too much studied, some authors have explored the topic in the case of object recognition [38, 39] . Nevertheless, detecting bias in face datasets is more challenging than in general context of object datasets. While, finding datasets aimed at training visual classification systems which share object categories (e.g. car, tree, or building) is easy, this is no the case for face datasets, as they usually do not share identities (categories). Hence, performing a cross-dataset analysis to explore how the same identities are represented over different face datasets, and how to exploit their differences, is not possible, in contrast to what happen with objects datasets [15, 32].

In summary, although most of the facial verification methods proposed so far assume that training and testing sets are built by gathering independent and identically distributed samples, however, in many real applications, this assumption does not hold, even each one of these sets could contain samples drawn from different populations. Motivated by these facts, the main goal of this work is to shed light on the inherent differences among face datasets commonly used for training face verification approaches, and the potential harms that should be considered when combining different datasets for training, testing or both processes. In particular, the main contributions of this work are:

- A study of the impact on performance of dataset bias, when we use different face datasets for training SVM models for face verification purposes (Sections 2.1 and 5).

- A novel insight into the face dataset bias by a study of their distribution on feature spaces. We have approached this issue from a geometrical perspective by looking into the datasets' feature vectors distribution (Sections 2.2 and 6).

- A comparison of the behaviour of different feature descriptors (Section 4), both handcrafted and learned, and their robustness against dataset bias.

The rest of the paper is organized as follows. First, in Section 2, we will explore the nature of our problem describing how bias in datasets can affect to face verification. Then, the datasets and the kind of features involved in this study are presented in sections 3 and 4. Section 5 and 6 describe our two different experiments with their results, and, finally, Section 7 presents the final conclusions.

2 The Nature of Dataset Bias

Dataset bias becomes evident when classifiers performances vary with different training and/or testing datasets. It could be said that bias is everywhere, but what are the specific causes of bias? and how can this problem be tackled?

One of the first studies that properly explored the first question was presented in [39]. This work is centered in the object recognition field and has been done before the deep learning revolution [19], so the study is restricted to handcrafted descriptors, namely the HOG descriptor. Posterior works, involving learned features, have also concluded that dataset bias remains a relevant problem for object recognition [38]. They both distinguish four different kind of bias according to their nature:

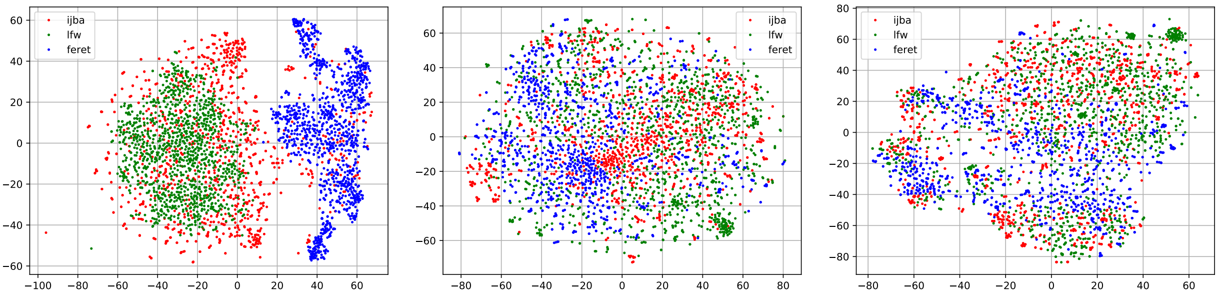
- *Selection bias* is related to the way in which images are collected (keywords search, manual selection, crowd-sourcing collection, etc.).

- *Capture bias* comes from how images are captured (type of device, context of acquisition, etc). A very important aspect when working with mobile devices in a non-collaborative interaction. Sometimes, we refer to this bias as *context*.

- *Label bias* is related to a poor semantic annotation of the dataset, where any labeling mechanism could assign a different label to the same object ("screen" vs. "TV"; "grass" vs. "lawn"). This is something which not apply in face verification, since labels here are perfectly defined.

- *Negative set bias* is related with the way of how the *rest of the world* (the *rest of faces* in our specific case) is sampled in the dataset. If this set is imbalanced or not representative, the model generated with it will have problems to generalize.

Regarding the second question, domain adaptation methods have often been presented as a way of solving dataset bias problems [8, 40]. A domain is a more general concept than a dataset; for example, in face verification, different domains can be defined as the one that only includes grey-scale images, the one with wide coverage of face poses, the one specially built for video-surveillance recognition, etc. Domain adaptation tools help to move or transfer knowledge between different domains. In the specific case of face recognition, face-frontalization methods (methods that convert any face to its frontal view, allowing us to eliminate the pose



(a) LBP.

(b) VGG.

(c) ResNet.

Figure 1: t-SNE representation of features of a random subset of samples drawn from three datasets with three feature descriptors: (Red) IJB-A; (Green) LFW; (Blue) FERET.

bias) [1, 9, 10] represent a way of domain adaptation to work just with frontal faces.

Since we could define a domain like the one that represents a certain dataset, it is easy to understand how these approaches help us to deal with the problem. A domain adaptation technique could be used in order to move between datasets. Nevertheless, we cannot say that domain adaptation techniques solve dataset problems. For example, if we have two different datasets with two different bias, we could use a domain adaptation technique in order to transfer a knowledge learned in one of them to the other one or even to a common one; but we could still have a biased model. At the end of the day, since we need samples of the visual world in order to learn, an unbiased dataset will be needed anyway.

In the next two sections, we present two of the main concerns with the case of face verification on mobile devices, namely bias in cross-dataset training and in dataset's feature space.

2.1 Bias in Cross-Dataset Training

In the development of a face verification system for a specific context (e.g. biometrics for mobile devices in a non-collaborative context) the ideal scenario would be to build an ad-hoc representative dataset of the target domain. Nevertheless, high costs and privacy-related issues usually make its construction unfeasible for most researchers. Take into account the previous specific scenario of face verification in mobile devices. In this case, the ideal scenario would require to build an entire dataset for each possible device (with its genuine and impostors sets). Thus, images from another richer source dataset are usually used to perform the experiments (henceforth *Dataset B*).

Due to the fact that, in the real scenario, every positive (identity of interest) sample collected to build the verification model will be necessarily taken under the proper operational domain (henceforth *Dataset A*), images taken from the richer source dataset are used as negative samples. Thus, the main idea is to explore whether the system trained with this configuration is able to generalize enough to distinguish between faces from the operational domain (same context as *Dataset A*) or, instead, the differences between the two datasets prevent the system from proper learning.

It could be expected that the use of a very rich source domain dataset (oriented for a broad range of cases) will be worse for the system performance than the use of a smaller dataset with a feature distribution closer to that of the target domain. Besides, the influence of the dataset bias could be assessed by measurements of the performance drop. This set-up resembles the one used in [39] for the analysis of the negative set bias, where the impact on the performance of using samples drawn from different datasets, to build the negative training set, was explored in the object recognition field.

2.2 Bias in Dataset's Feature Space

Given an input feature vector, $\mathbf{x}_q$ and a claimed identity, I, the verification process tries to determine whether $\mathbf{x}_q$ is genuinely user I, or it is an impostor. Performance of the verification operation depends both on the feature vectors, and on the classifier (distance measure).

The effect of the dataset bias can be analyzed by considering all the system as a black-box, only the impact on classifier performance or, instead, by analyzing just the feature vectors. The dissociation between feature extractor and classifier is something widely done in the literature. In a face verification context, there are works that use pools of feature extractors to establish a benchmark [23]. The performance of newly designed features is often compared in similar benchmarks to test their quality [20, 37]. The exchange between hand-crafted and learned features within the same system is even used [5] to push performance to state-of-the-art values.

At any rate, both, hand-crafted features, that are not specific of any context, and deep features, which were learned in source domains that are different from the target domain, have to be classified to determine whether a given facial image belongs to the identity of interest. Yet, it is crucial to reduce the intra-class variations while enlarging the inter-class differences for face verification. A SVM classifier is commonly used for transfer learning in these data scarce domains [44]. Besides, it is worth recalling that in our specific context of applications, positive (identity of interest) samples are generated in the target domain, but negative samples could come from different source domains. So, in this work, we will consider that the classifier is always based on SVM, and the differences are in the feature extractor side.

Given the fact that all datasets suffer from some kind of bias, their data distribution in feature space are expected to be different, as the t-SNE [22] representation of data in Fig. 1 suggests.

As the shapes of actual distributions are usually unknown, (dis)similarity among datasets can be estimated from distances among feature vectors of different datasets. Our hypothesis is that vectors from two datasets with similar distributions should tend to have the same probability of finding nearby vectors from the same and the other dataset.

On the other hand, distances are something especially relevant in the case of face verification. The distance between two feature vectors is often used to verify or not an identity. Feature extractors are designed so that same identities tend to be closer. For example, in some deep learning methods, a triplet loss is used in the training phase in order to keep same identities closer in a simple Euclidean space [28, 33], and similar work is done with handcrafted features too [23]. Distances between feature vectors are used also in [37] for both verification as other related tasks as attribute detection (ethnic, male/female, age, etc.). A cosine dissimilarity metric is also used in [20] over different kinds of descriptors to compare the performance of a certain face descriptor. So, we expect that anomalies in these distances provoked by datasets will directly lead to problems in real-world performance.

This neighbour search can be seen from multiples points of view. First, it can be seen as a way of knowing how datasets are distributed in the feature space, and if their distributions are equivalent. It also can be seen as the *Name that Dataset!* experiment of [39] where the dataset to which a sample belongs to can be guessed using just its neighbours. Finally, another interpretation can be to find the dataset origin of hard negatives, in other words, the samples that are more difficult to classify. This is important because these elements are the most important ones in order to build any model.

3 Datasets

In this work six datasets were considered (Fig. 2), five of them are well-known public face datasets, and the other is a dataset which was built for the specific context of non-collaborative face verification on mobile devices. Three of them were built gathering images captured with mobile devices (two with and one without users' collaboration), and the others contain images taken in a range of different (general) contexts. Here, the specific characteristics of each one (see Tab. 1) are described:

- **FERET** [29] is one of the first datasets that tried to become a standard for face recognition, both for training and testing processes. It consists of a total of 8,525 images of 1,109 people with a range of different (annotated) poses taken in a highly controlled environment. For this work, only the *dvd-1* data was used.
- **Labelled Faces in the Wild**[1] (LFW) [13] contains more than 13,000 face images of a total of 5,749 people collected from the web. Each face was annotated with the name of the person, and 1,680 of the identities have two or more distinct photos in the dataset. It is one of the first datasets aimed at coping with the *unconstrained* face recognition problem. Images were gathered from the internet, and the faces were detected using the Viola-Jones detector [41], which introduced a bias in the range of possible poses.
- **IARPA Janus Benchmark A*** (IJB-A) dataset [17] contains a total of 5,712 images of 500 identities (≈11 images per subject). The most distinctive characteristic of this dataset is the elimination of the bias in face detection due to the fact that the complete dataset was manually annotated using crowd-sourcing methods. It has been recently updated with the **IARPA Janus Benchmark B** dataset [43] in which the number of images has been increased to 21,728 (1,845 different identities), and the **IARPA Janus Benchmark C**, which even added up more images from video frames.
- **O2FN** dataset [31] contains 2,000 face images taken from 50 different subjects predominantly of Asian ethnic. Images are self-taken photos using a mobile phone in a collaborative context. Subjects were asked to take approximately 20 indoor images and 20 outdoor images, with limited variations in facial expression and out-plane rotations.
- **MobBIO** multimodal dataset [34] was specifically designed for biometrics. It contains data of faces, voice and iris of 105 identities. In the case of facial images, which is the part of our interest, there are a total of 1,640 photographs (≈16 images per identity) taken with mobile devices in a controlled environment, with a limited pose and illumination variations.
- **FaceSampler** (FS) dataset has been created using the frontal camera of mobile phones in a non-collaborative context. It consists of a total of 2102 images of a total of 15 different identities. The acquisition system was designed to use inertial information in order to maximize the probability

[1] It was necessary to eliminate the overlapped identities between IJB-A and LFW datasets. The procedure was to eliminate from LFW the 183 identities also present in IJB-A.

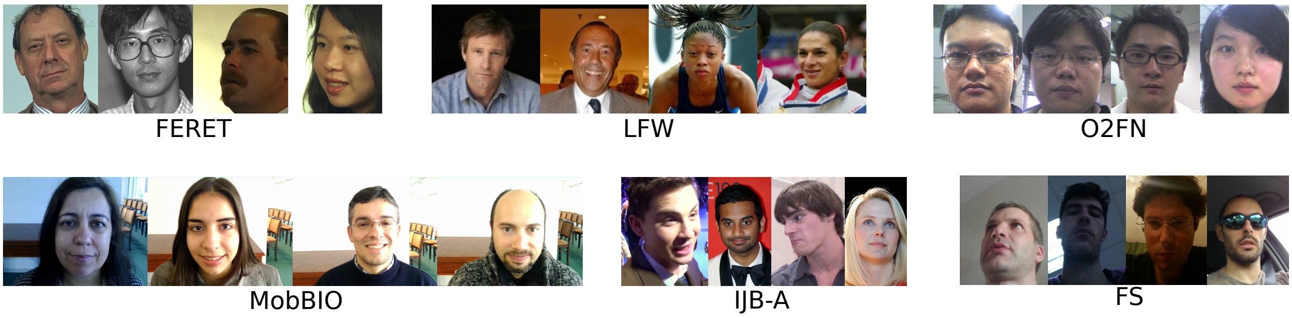


Figure 2: Sample images from each dataset used for the experiments.

Table 1: Summary of dataset characteristics.

Dataset	Context	Pose Variation	Controlled Environment	Illumination	Ethnicity
FERET	General	Full	High	Indoors	Caucasian
LFW	General	Limited	Low	Varied	Varied
IJB-A	General	Full	Low	Varied	Varied
O2FN	Mobile	Limited	Intermediate	Varied	Asian
MobBIO	Mobile	Limited	High	Indoors	Caucasian
FS	Mobile	Limited	Low	Varied	Caucasian

of the existence of faces in the frame, although only the images with a detected face were gathered. The Viola-Jones detector was used. Image collection was running in the background while users were using the mobile phone [3]. Up to our knowledge, this is the unique dataset which was built on the operational domain for the problem of non-collaborative verification of users of mobile devices.

3.1 Dataset Pre-processing

To the purpose of face detection, we have used the detector implemented in the Dlib library [16], which is based on HOG features, due to its perfect integration with facial landmark detector also implemented in the same library. Here, indeed, an important constraint was introduced. For example, the main contribution of IJB-A dataset is the fact that they eliminate the Viola-Jones bias by performing a manual annotation of the data.

Nevertheless, we can justify this fact for two reasons. First, the vast majority of face recognition methods (included the ones that are tested here) use this step in their pipelines. And second, by setting a fixed face detector it can be assumed that any behaviour of the data will not be related to the face detector. So, it can be taken as part of the context in which we are working; that can be called *universe of possible faces* detected by our face detector. We can see how each dataset remains after the face detection process in Tab. 2.

4 Features

The presence of highly imbalanced data with respect to the distribution of the characteristics of interest (in a given universe), makes feature extraction techniques to produce biased features. Sometimes this bias is good to characterize the specificity of the context of interest (target domain), but quite often the extracted features mislead the classifier.

Feature extractors can be more or less sensitive to different characteristics, so its election is of paramount importance. For example, a hypothetical feature extractor that is perfectly robust to face pose variations, would not reflect in the feature space differences between a dataset with just frontal faces and a dataset richer in poses. This does not mean that the former dataset is not biased towards a particular pose, but the feature extractor is able to ignore this fact. Besides, this behaviour can be desirable (in a general context) or not (in a specific context of application poses can be biased, for instance when looking at the screen of a smartphone).

This degree of robustness is almost impossible to achieve with general purpose hand-crafted features since they were not designed for this specific domain of application. When using these features, only the classifier can be adapted throughout the training phase. On the contrary, deep features are trained, so it will be possible to create a more adjustable feature set depending on the application.

In this study, we have used three feature extractors with different abstraction and generalization abilities. All of them have been used in works that achieved state-of-the-art results in face verification contexts, and are representative of hand-crafted and deep learning features: LBP over landmarks, VGG-Face and ResNet.

Table 2: Summary of samples of each dataset used after the face detection.

Dataset	**Identities**	**Images**
FERET	739	4,929
LFW	5,566	10,966
IJB-A	500	19,427
O2FN	50	1,720
MobBIO	100	1,599
FS	15	2,102

Table 3: Type and dimensionality of each feature detector.

	Type	Dimensions
LBP over landmarks	*Handcrafted*	*96,288*
VGG-Face	*CNN Learned*	*4,096*
ResNet	*CNN Learned*	*128*

4.1 LBP over Landmarks

Bayesian approaches are some of the few methods that while using handcrafted features to encode information of faces, still obtain state-of-the-art results ([5] 96.33% accuracy in LFW). So, we have included LBP features over facial landmarks in our analysis.

To obtain a face description, first of all, a facial landmark detection [14] was performed in order to locate a total of 68 points on face images. These landmarks were used to perform a similarity transformation in order to rectify the image. After that, patches centred around just 51 of the inner landmarks (Fig. 3 left) were extracted at two different scales. The side lengths of the image at the two scales were 180 and 118 pixels. The patch size was fixed to 40 × 40 in both scales. Each patch was divided into 4×4 non-overlapped cells (Fig. 3 right). Finally, an uniform-LBP vector was computed for each cell, and all of them were concatenated on the final feature vector of 96,288 dimensions (see Tab. 3), following [26].

4.2 VGG-Face

The VGG-Face CNN proposed in [28] for general face recognition has achieved an extraordinary value of accuracy of 98.95% on the LFW dataset. This CNN was trained using a large-scale dataset of 2.9 million images of 2600 people. The first layers of its architecture, discarding the last fully connected layer, were used in this work to extract 4,096 dimension features (Tab. 3).

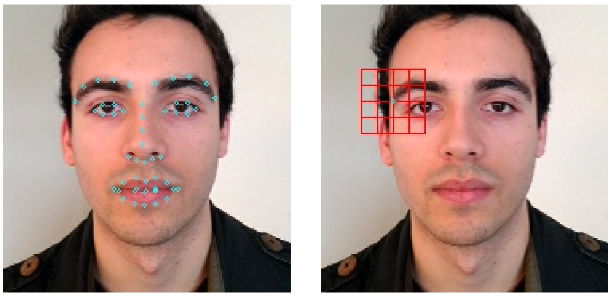


Figure 3: Extraction of LBP features. Left: locations of the 51 inner landmarks used for feature extraction. Right: 4x4 cell centered on one of the landmarks.

Table 4: Configuration of *Dataset A* and *Dataset B* in the training phase and in the two different tests (1 and 2).

Phase	Positive	Negative
Train	*A*	*B*
Test 1	*A*	*B*
Test 2	*A*	*A*

4.3 ResNet

Recently, deeper networks, with shorter features vectors, have achieved better results in classification than their predecessors. A version of the ResNet-34 [11], with a few layers removed and the number of filters per layer reduced by half, was used as well. This model was trained on the same dataset of the VGG-Face [28] and the face scrub dataset [25], apart from additional images taken from the Internet, amounting to a total of about 3 million images of 7,485 different identities. Again, pre-trained models (available in the Dlib library [16]), that was reported to achieve a 99.38% accuracy in the LFW dataset for the task of face verification, was used. As for the case of the VGG-Face, we have discarded the last fully connected layer in order to extract a 128 dimension feature vector (Tab. 3).

5 Cross-Dataset Training

With the scenario described in Section 2.1 in mind, this experiment was designed to explore the potential damage in the ability to distinguish samples of the same dataset when negative samples taken from a different dataset are used for the training phase.

In other words, could the trained classifier find inter-dataset differences much larger than intra-dataset ones? To address this question, two datasets, *Dataset A* and *Dataset B*, are defined as follows:

- *Dataset A* is a small dataset gathered in the target domain. It consists of a set of users' faces with at least 10 images per user. Considering each dataset has a different number of identities, it would be desirable to maximize uniformity in this sense. At the same time it would be also desirable to have a number of identities as relevant as possible to perform statistics. Thus, even though the minimum of identities is 15 in FS, we have set a maximum of 50 corresponding to the identities of O2FN (the second lowest value). With this maximum, when we use FS as *Dataset A* the results are averaged just over 15 users. The images of this dataset are drawn from one of the datasets presented in Section 3. Besides, images of each identity are split by half into train and test subsets.

- *Dataset B* is a large dataset used as a negative sample source. It is generated using one dataset, among the ones described in Section 3, different from the one used to generate *Dataset A*. For each user of *Dataset A* a *Dataset B* is generated with the rest of users of the original dataset. This set is split as well into train and test subsets without shared identities. The number of identities in both sets is the same.

For each identity in *Dataset A*, a Linear-SVM model was learned using the subset of samples of the specific user in *Dataset A*, as positive samples for training, and the training subset from *Dataset B* as negative samples.

The learned model was tested in two different ways in order to compare the performance:

- Test 1. Testing against other identities in *Dataset B* (same configuration as the training phase).

- Test 2. Testing against other identities in *Dataset A*.

The (dis)similarity between a query face pattern and the biometric model of the identity that the query pattern is verified against, is based on a threshold. Depending on the choice of this threshold, the efficiency system of the system varies both in the fraction of the falsely accepted impostors (False Acceptance Rate (FAR)), an in the rate of the truly accepted genuine user (True Positive Rate (TPR)). Both measures are complementary, as they are usually combined in a single one. In this work, the used performance measure was the True Acceptance Rate at 0.001 False Acceptance Rate (TAR @ 0.001 FAR), a common performance measure in biometrics.

The experiment is performed independently for each one of the features described in Section 4.

5.1 Experimental Results

Experimentation was done with different combinations of *Dataset A* and *B*, and features. Results are shown in Tab. 5, where *Datasets A* are represented in rows and *Datasets B* in columns. The small-size number present the TAR @ 0.001 FAR performance for Test 1 and Test 2. The normal-size number is the drop in performance between the two different testings. Finally, in the last column and in the last row there are the average drops row-wise and column-wise respectively.

The first thing we observe is that there is a general drop in performance between Test 1 and Test 2. This means that, at the same false positive rate, the system has a higher rate of false negatives acceptance when testing and training are done on the same dataset. This reveals that instead of just learning the identity information, what is learned is the bias of the dataset. This effect corroborates the important influence that dataset bias can have in performance.

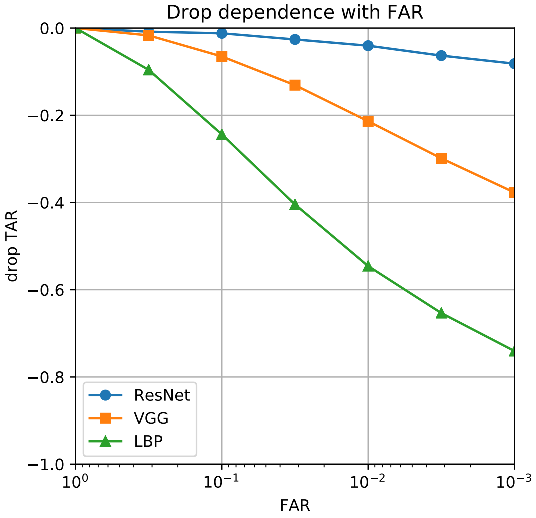


Figure 4: Average TAR drop between Test 1 and Test 2 respect to the FAR point in which we perform the measure, using FS just as *Dataset A*.

Comparing the results for the different feature extractors it can be noted that the drop in performance is much stronger with the LBP features, reaching values up to +80% drop. Although a drop in performance is also observed in the case of deep features, it is much smaller, especially in the case of the ResNet features.

The drop in performance also depends on the FAR point where the TAR is measured (Fig. 4). The influence of the dataset bias in performance is correlated with the level of difficulty of the task. So, the higher the level of FAR requirements, the more notorious the effect is. Related to that, it is also remarkable that the best results are obtained when using MobBIO as *Dataset A*, a dataset that was gathered under highly controlled conditions. It seems that its limited amount of intra-class variations, makes the verification task easy, even using LBP features.

Comparing the behaviour of the assessed datasets as *Dataset B*, it should be noted that datasets oriented for the totally unconstrained problem of face verification are the ones that behave the best in this role (LFW and IJB-A). On the contrary, the most constrained ones (MobBIO, FS and O2FN) lead to the lowest performances (highest drop). This is something that could be expected and agrees with the idea that for a negative training set the more general the better.

In addition, and more specifically, it is remarkable the entanglement between LFW and IJB-A datasets. When one is used in the role of *Dataset A*, the best performance is achieved with the other one used as *Dataset B*. This indicates that when the face information for verification is taken in an unconstrained environment, the impostor information during the training should be also taken without constraints.

A similar, but weaker, entanglement is observed between FERET and IJB-A, possibly due to their wide range of user's poses (we must remember that pose effect is a bit limited using a quasi-frontal face detector).

Table 5: Drop in performance (TAR @ FAR 0.001) of Test 1 respect to Test 2 for each combination of *Dataset A* and *B* using different kind of features.

	LBP																		
B	**MobBIO**			**FS**			**O2FN**			**FERET**			**LFW**			**IJB-A**			
A	Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Av. drop
MobBIO				1.0000	0.9600	-0.0400	1.0000	0.9625	-0.0375	1.0000	0.9325	-0.0675	1.0000	0.9750	-0.0250	1.0000	0.9900	-0.0100	-0.0360
FS	0.9583	0.1697	-0.7886				0.9266	0.3242	-0.6023	0.9349	0.1371	-0.7978	0.9905	0.1163	-0.8742	0.9210	0.2814	-0.6397	-0.7405
O2FN	1.0000	0.6891	-0.3109	0.9821	0.6812	-0.3009				0.9857	0.5748	-0.4109	1.0000	0.2774	-0.7226	0.9967	0.7233	-0.2734	-0.4037
FERET	0.9800	0.0495	-0.9305	0.9274	0.1491	-0.7783	0.9769	0.1555	-0.8215				1.0000	0.0276	-0.9724	0.8964	0.2972	-0.5992	-0.8204
LFW	0.9875	0.2162	-0.7714	0.9999	0.1584	-0.8415	1.0000	0.0603	-0.9397	1.0000	0.0496	-0.9504				0.8996	0.6820	-0.2176	-0.7441
IJB-A	0.7959	0.2589	-0.5370	0.9026	0.2796	-0.6230	0.9294	0.1980	-0.7315	0.9256	0.1815	-0.7441	0.8362	0.3751	-0.4611				-0.6193
Av. Drop			-0.5167			-0.3480			-0.6111			-0.6266			-0.6677			-0.3713	

	VGG-Face																		
B	**MobBIO**			**FS**			**O2FN**			**FERET**			**LFW**			**IJB-A**			
A	Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Av. drop
MobBIO				0.9975	0.8582	-0.1393	1.0000	0.7657	-0.2343	1.0000	0.9775	-0.0225	1.0000	1.0000	0.0000	1.0000	0.9975	-0.0025	-0.0797
FS	0.9368	0.6179	-0.3598				0.9368	0.3220	-0.6148	0.9679	0.5666	-0.4014	0.9741	0.6950	-0.2791	0.9406	0.7104	-0.2302	-0.3771
O2FN	1.0000	0.6626	-0.3374	0.9820	0.2312	-0.7508				0.9861	0.8776	-0.1085	0.9807	0.9142	-0.0664	0.9870	0.8634	-0.1236	-0.2773
FERET	1.0000	0.6102	-0.3898	0.9810	0.3653	-0.6157	0.9754	0.3457	-0.6297				0.9852	0.7952	-0.1900	0.9387	0.8407	-0.0981	-0.3847
LFW	0.9973	0.7647	-0.2326	0.9898	0.3608	-0.6290	0.9942	0.3467	-0.6475	0.9685	0.7933	-0.1752				0.9008	0.8496	-0.0512	-0.3471
IJB-A	0.9821	0.4921	-0.4899	0.9051	0.2200	-0.6851	0.9848	0.2248	-0.7600	0.8964	0.5885	-0.3079	0.8550	0.7258	-0.1292				-0.4744
Av. Drop			-0.3619			-0.5640			-0.5773			-0.2031			-0.1329			-0.1011	

	ResNet																		
B	**MobBIO**			**FS**			**O2FN**			**FERET**			**LFW**			**IJB-A**			
A	Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Test 1	Test 2		Av. drop
MobBIO				1.0000	0.9975	-0.0025	1.0000	0.9850	-0.0150	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	-0.0035
FS	0.9454	0.9496	0.0042				0.9978	0.7014	-0.2964	0.9821	0.9401	-0.0420	0.9853	0.9524	-0.0329	0.9799	0.9388	-0.0412	-0.0817
O2FN	1.0000	0.7090	-0.2910	1.0000	0.7004	-0.2996				0.9959	0.9896	-0.0063	0.9987	0.9855	-0.0132	0.9978	0.9783	-0.0195	-0.1259
FERET	0.9967	0.7047	-0.2919	0.9860	0.6423	-0.3437	0.9853	0.6804	-0.3049				0.9589	0.9268	-0.0321	0.9621	0.9135	-0.0486	-0.2042
LFW	0.9978	0.8703	-0.1275	0.9978	0.8390	-0.1587	1.0000	0.7400	-0.2600	0.9954	0.9531	-0.0423				0.9784	0.9620	-0.0165	-0.1210
IJB-A	0.9696	0.6639	-0.3057	0.9812	0.5945	-0.3867	0.9919	0.5153	-0.4766	0.9380	0.8232	-0.1148	0.8257	0.8815	0.0558				-0.2456
Av. Drop			-0.2024			-0.2382			-0.2706			-0.0410			-0.0044			-0.0251	

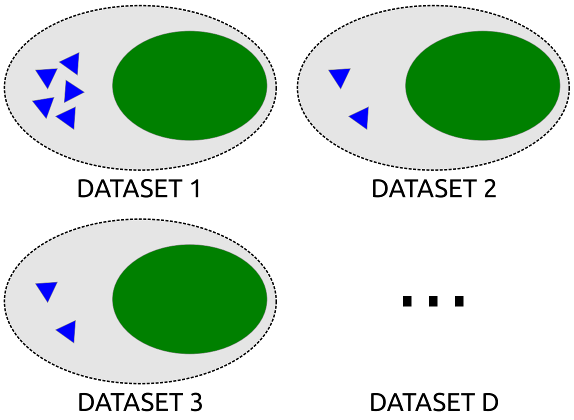


Figure 5: Scheme of the subsets generated for each dataset (Dataset 1, Dataset 2, ..., Dataset D_i, ..., Dataset N_D). Blue triangles represent *probe sets* and green ellipses represent the part of the dataset used to generate the *gallery set*. In this case, $N_p = 5$ *probe sets* were generated for the first dataset, and $N_p = 2$ *probe sets* for the rest.

Faces from the first are taken in a highly controlled environment whereas the case for the second one is the complete opposite. Despite this fact, when using FERET as *Dataset A*, the lowest drop, in 2 out of 3 cases, is observed with IJB-A in the role of *Dataset B*. This behaviour strengthens the previous statement of the necessity of having impostor data taken in the same conditions as the genuine one.

To sum up, when a model is trained using data from two different datasets (even when they were aimed at the same task) we have to take into account this potential drop in performance. Otherwise, the classifier may mislead the focus of finding useful patterns to verify the identity of interest, and instead, could learn to distinguish between datasets.

6 Dataset's Feature Space

As aforementioned in Section 2.2, the aim of this second experiment is to study the feature space in order to explore how different datasets are distributed in it. For this purpose, we have relied on a Nearest Neighbour search using two different metrics. Our premise is that, given a feature vector of a face of a certain user, the probability of the dataset to which its nearest neighbour belongs to (eliminating other images of that same user) should tend to be uniform for a equivalent datasets (non-biased between them).

First, we are going to explain the experimental setup: how the dataset is built, the metrics and the nearest neighbour search. Finally, we present the experimental results and their discussion.

6.1 Building the subsets

In order to explore the distribution of dataset samples over the feature space, two kinds of splits for each one of N_D datasets were made (Fig. 5): the *probe sets* and the *gallery set*. The idea is to seek the nearest neighbours of the elements of the *probe sets* among the elements of the *gallery set*. These sets have been created this way:

- **Probe sets.** A total of N_p different random subsets of n_p faces sampled without replacement from each dataset. N_p will depend on the number of samples of the dataset. The greater the number of samples in a dataset, the higher number of its *probe sets*.
- **Gallery set.** The union of the random subsets of n_g elements sampled without replacement from each dataset, generates the *gallery set* set, with $N_D \cdot n_g$ unique elements.

It is important to note that there will not be any common elements, neither identities, between any generated partitions.

6.2 Metrics in the Feature Space

Given a set of feature vectors $X = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, we denote by $\mathbf{x}_* \in X$ the nearest neighbour of $\mathbf{x}$ if:

$$\min d(\mathbf{x}, \mathbf{x_i}) = d(\mathbf{x}, \mathbf{x}_*) \quad i = 1, ..., n \tag{1}$$

The function $d(\mathbf{p}, \mathbf{q})$ represents a general metric. For our study, we have used the euclidean distance ($L2-$norm):

$$d_{L2}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{m} (p_i - q_i)^2} \tag{2}$$

and the Manhattan distance ($L1-$norm):

$$d_{L1}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{m} |p_i - q_i| \tag{3}$$

Where p and q are two feature vectors in a m-dimensional descriptor space.

6.3 Nearest Neighbour Search

For each *probe set*, we have taken each of their elements and drawn without replacement (in order to avoid to always take the same outlier) their nearest neighbours from the *gallery set*. Using the dataset membership of the nearest neighbours, we will generate a histogram for each probe set and average them over the N_p different probe sets of each dataset.

As it has been stated before, the premise is that the probability (P) of a sample in the *probe set* finding an element of the *gallery set* gs_j belonging to the D_i dataset, as its nearest neighbour, should tend to be the same for all $i \in \{1, \ldots, N_D\}$:

$$P(x_* = gs_j \in D_i) = \frac{1}{N_D} \tag{4}$$

Any distribution different from the uniform one may indicate that the datasets sample different spaces, so at least one of them could have some kind of data bias.

Table 6: Distribution of nearest neighbors over each dataset in % using different features.

	LBP											
	L2-norm						*L1-norm*					
Gallery Probe	**MobBIO**	**FS**	**O2FN**	**FERET**	**LFW**	**IJB-A**	**MobBIO**	**FS**	**O2FN**	**FERET**	**LFW**	**IJB-A**
MobBIO	**92.78**	0.00	5.00	0.83	0.28	1.11	**94.44**	0.00	2.22	2.50	0.00	0.83
FS	1.78	**66.22**	19.78	10.44	0.11	1.67	1.44	**67.56**	17.00	12.22	0.00	1.78
O2FN	1.11	1.11	**95.83**	1.67	0.00	0.28	0.00	2.22	**93.89**	3.89	0.00	0.00
FERET	0.00	0.67	1.22	**97.33**	0.00	0.78	0.00	0.22	0.44	**99.00**	0.00	0.33
LFW	0.78	0.44	0.33	0.00	**86.33**	12.11	1.44	0.56	0.44	0.22	**84.89**	12.44
IJB-A	4.44	1.00	2.56	7.22	41.56	**43.22**	4.44	2.56	2.67	13.33	35.89	**41.11**

	VGG-Face											
	L2-norm						*L1-norm*					
Gallery Probe	**MobBIO**	**FS**	**O2FN**	**FERET**	**LFW**	**IJB-A**	**MobBIO**	**FS**	**O2FN**	**FERET**	**LFW**	**IJB-A**
MobBIO	**55.83**	10.83	2.50	11.94	8.89	10.00	**60.28**	8.61	2.22	9.17	11.39	8.33
FS	10.78	**45.11**	13.89	16.78	4.00	9.44	10.11	**46.67**	14.78	15.44	3.89	9.11
O2FN	0.56	0.00	**82.50**	13.89	0.56	2.50	0.56	0.28	**82.78**	14.17	1.39	0.83
FERET	1.11	1.78	15.56	**62.78**	6.67	12.11	1.33	1.78	14.56	**63.11**	7.56	11.67
LFW	3.11	3.78	5.33	22.89	**34.56**	30.33	3.56	3.22	5.22	24.89	**33.22**	29.89
IJB-A	1.22	4.44	4.44	20.33	15.78	**53.78**	1.56	4.78	4.00	20.89	20.33	**48.44**

	ResNet											
	L2-norm						*L1-norm*					
Gallery Probe	**MobBIO**	**FS**	**O2FN**	**FERET**	**LFW**	**IJB-A**	**MobBIO**	**FS**	**O2FN**	**FERET**	**LFW**	**IJB-A**
MobBIO	**39.17**	38.06	1.11	11.67	4.72	5.28	**36.11**	35.83	1.11	12.50	5.56	8.89
FS	21.89	**50.22**	1.56	11.89	4.44	10.00	21.56	**50.78**	1.44	12.22	4.22	9.78
O2FN	1.67	0.00	**88.89**	7.78	1.11	0.56	1.11	0.00	**88.33**	8.89	1.11	0.56
FERET	3.78	8.67	16.00	**42.22**	8.78	20.56	4.67	8.56	15.78	**41.56**	11.33	18.11
LFW	4.56	5.67	3.33	19.11	31.11	**36.22**	4.56	6.78	3.00	19.33	31.11	**35.22**
IJB-A	5.67	5.11	1.67	13.00	22.67	**51.89**	5.22	6.22	1.78	12.67	23.33	**50.78**

In our experiments, we have worked with a total of $N_D = 6$ datasets of different sizes. We have used $N_p = 2$ for O2FN and MobBIO, and $N_p = 5$ for the rest.

As nearest neighbours are drawn without replacement, the prior probability change as elements are removed from the *gallery set* in each nearest neighbour search. In order to mitigate this effect, sizes of $n_p = 180$ and $n_g = 1200$ were fixed. This way the number of elements of the *probe set* will keep low (180) respect to the number of elements in the *gallery set* (7,200). This makes the effect of drawing nearest neighbours without replacement from the gallery set negligible.

Taking into account the prior probability (following Eq. 4 with $N_D = 6 \Rightarrow P \approx 16,7\%$) in the worst case scenario where the nearest neighbours always belongs to the same dataset, the prior probability of that dataset would decrease down to $\approx 15.2\%$.

6.4 Experimental Results

The second part of our experiments was aimed at observing the distribution of the nearest neighbour of the elements in a *probe set* with respect to elements in the *gallery set*. The distributions obtained for each case are shown in Tab. 6. Each row of the table contains the distribution (in %) among all the datasets (upper row), of the nearest neighbours to the elements in the *probe set* (leftmost column). Results are very similar for both L1 and L2 metrics.

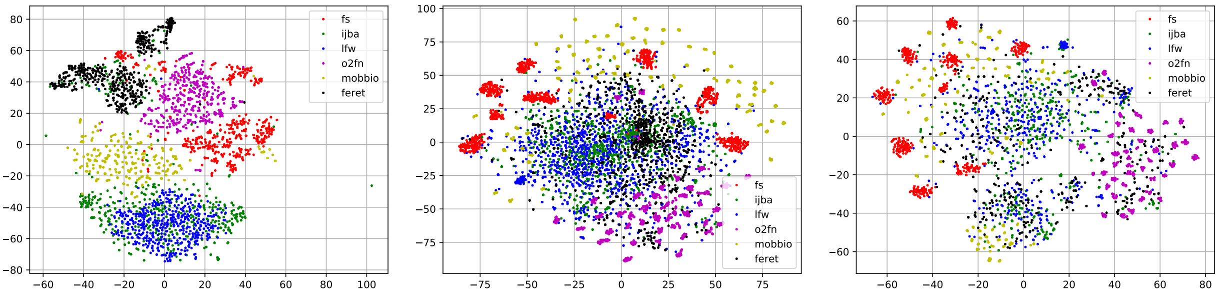
Next, we will analyse the obtained distributions from the two different points of dataset and features, in Sections 6.4.1 and 6.4.2, respectively. Finally, we will relate these results to the t-SNE data representation in Section 6.4.3.

6.4.1 Looking from the Dataset Side

The most evident fact that can be observed in the data is an important tendency to find nearest neighbours in the same dataset. Such effect reveals that the initial premise of a uniform distribution, Eq. (4), was false. It must be taken into account that LBP features, combined with the nearest neighbour classifier, are good enough (up to +90% accuracy) to guess the dataset membership (as done in the *Name that dataset!* challenge, [39]). It is also remarkable that the highest percentage of nearest neighbours are almost always achieved inside the same dataset, whatever the feature.

Before going any further with the discussion, we can divide datasets into two groups: the ones designed for the unconstrained the face recognition problem (LFW and IJB-A) and the ones designed for more specific applications, namely MobBIO, FS and O2FN datasets for mobile applications, and FERET dataset for controlled environments.

According to that, it can be observed that there is



(a) LBP.

(b) VGG.

(c) ResNet.

Figure 6: A t-SNE representation of the 6 *gallery sets* taken from each dataset of study for three feature descriptors. (Red) FS; (Green) IJB-A; (Blue) LFW; (Purple) O2FN; (Olive) MobBIO; (Black) FERET.

a certain entanglement between the two unconstrained datasets for every kind of feature and metric. *Probe sets* taken from LFW and IJB-A datasets seem to have the highest proportion of nearest neighbours within one of these two datasets (+60%). This fact suggests that both datasets were drawn from similar distributions. In spite of IJB-A being a more complete dataset, its main contribution is to eliminate the frontal face constraint. Our face detector (see Section 3.1) keeps having a certain tendency of detecting frontal faces. This seems to make the differences in the distribution between LFW and IJB-A to dilute. Such entanglement is not observed for the rest of datasets.

On the other hand, O2FN is the dataset with the highest mean rate of samples' nearest neighbours in the same dataset; probably, its specific ethnicity is crucial to that outcome. Meanwhile, MobBIO and FERET experienced the largest changes in the distribution of their nearest neighbours, according to features. For its part, FS dataset is one of the most stable in this respect.

6.4.2 Looking from the Feature Side

We can see that the same dataset pairing behaviour is quite strong in the case of the LBP features. We see a +80% pairing in for 4 of the 6 datasets. This means that the feature vector is not retaining the target information.

This pairing behaviour seems weaker when deep features are used. As we can expect, the training process that is performed in order to generate the CNN helps the system to discard more information not related to the identity.

The main difference in behaviour between the two deep descriptors is the fact that ResNet features break the rule of having always the highest frequency for self-pairing. This behaviour could indicate a certain correlation of the effect we are describing with the performance of the CNN.

Some important cues about the source of the bias can yet be found out in Tab. 6. The first cue is provided by the results obtained for O2FN. As aforementioned, the elements of this dataset are unique to get a +80% paring across the three features. Despite being reduced by the use of learned features it does not suffer a very strong drop like the other datasets. The second cue is given by the results obtained for FERET and MobBIO. In the case of LBP features, these datasets suffer a comparable paring with respect to O2FN. Nevertheless, the drop caused by learned features reduced the paring to a 40-60%, much lower than the case of O2FN.

The main characteristic that differentiates O2FN dataset from the others is the prevalent Asian ethnicity of their identities, a bias related to μ. On the other hand, the common characteristic of FERET and MobBIO is the similar controlled-environment condition in which both datasets were generated, a bias related to ϵ.

Consequently, we can state that deep features help to deal with bias in data, that is related to ϵ, better than LBP features. But, on the other side, in terms of μ related bias, the effect is more similar between deep and LBP features, because this kind of information is retained in both types of feature vectors. We just have to recall Section 4 to find a theoretical explanation for this fact. Deep features are created by a training process in order to retain the identity information (μ) and discard the non-identity one (ϵ). This is something much more difficult to achieve with hand-crafted features. So, this behaviour is an illustrative example of how different feature extractors can hide a bias present in the data.

6.4.3 t-SNE representation of the gallery sets

The high dimension of feature vectors makes the task of directly visualizing data impossible. Therefore, it is necessary to visualize data in a reduced space. One of the most sophisticated options is the t-SNE representation [22], which tries to preserve the local structure of the high-dimensional data as well as some of the more global structure.

We can represent our *gallery sets* using this representation in order to look for any cue of their distribution. The result can be seen in Fig. 6. The first thing that can be observed based on the representation is that each dataset distributes differently over the feature space. This fact is especially evident in the case of LBP features since its clusters are the most separable and compact.

Finally, it can also be observed how images from the same user cluster together when using deep features. For FS dataset where the gallery set has a limited amount of users, we can even easily count the number of users.

7 Conclusions

In this paper, we have performed a study over the differences between datasets oriented for face verification, from the point of view of the distribution of their elements into the feature space, and how this impacts on the performance of the learned systems when operating in real-world conditions. The considered scenario is of the one of a face verification system on a mobile device, where the individual of interest (genuine user) is enrolled using a few facial images (positive samples) taken in the operational domain, while impostors (negative samples) for training are drawn from public available datasets.

By using different combinations of positive (Dataset A) and negative (Dataset B) samples taken from different datasets, and different feature extractors, we have observed the impact of bias in verification performance.

It can be observed a drop in performance when Dataset A and Dataset B are taken from different source datasets. This indicates that elements from the same dataset tend to be more alike than elements taken from different datasets. Indeed, there is an important tendency to find elements within the same dataset as the nearest neighbours. This tendency can be so strong that, in the case of LBP features, we could be able to guess the dataset to which a sample belongs to with a +90% accuracy, by just looking at its nearest neighbour.

In terms of differences among datasets, it was observed that some bias are evident independently of the feature descriptor at hand, so we could talk about endogenous bias. However, other differences are very dependent on feature descriptor, what rather seems to be an induced, or exogenous, bias.

Finally, as it could be expected, in terms of differences between features, every experiment suggest better performance of deep features respect to handcrafted ones, especially with the ResNet architecture. Even though, the effect of the dataset bias is still observable.

Acknowledgements

This work has received financial support from the Xunta de Galicia, Consellería de Cultura, Educación e Ordenación Universitaria (Accreditation 2016-2019, EDG431G/01 and ED431G/08, and reference competitive group 2014-2017, GRC2014/030), the European Union: European Social Fund (ESF), European Regional Development Fund (ERDF) and FEDER funds and (AEI/FEDER, UE) grant number TIN2017-90135-R.

Eric López had received financial support from the Xunta de Galicia and the European Union (European Social Fund - ESF).

References

[1] S. Banerjee, J. Brogan, J. Krizaj, et al. To frontalize or not to frontalize: Do we really need elaborate pre-processing to improve face recognition? In *Winter Conference on Applications of Computer Vision (WACV)*, pages 20–29, March 2018.

[2] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, volume 81, pages 77–91. PMLR, 2018.

[3] F. Casado, C. Regueiro, R. Iglesias, X. Pardo, and E. López. Automatic selection of user samples for a non-collaborative face verification system. In *ROBOT 2017: Third Iberian Robotics Conference*, pages 555–566. Springer, 2018.

[4] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, pages 566–579, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[5] D. Chen, X. Cao, D. Wipf, F. Wen, and J. Sun. An efficient joint formulation for bayesian face verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):32–46, Jan 2017.

[6] D. Crispell, O. Biris, N. Crosswhite, J. Byrne, and J. Mundy. Dataset augmentation for pose and lighting invariant face recognition. *CoRR*, abs/1704.04326, 2017.

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[8] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, pages 2960–2967, 2013.

[9] T. Hassner. Viewing real-world faces in 3d. In *International Conference on Computer Vision (ICCV)*, pages 3607–3614, Dec 2013.

[10] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4295–4304, June 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[12] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek. Frankenstein: Learning deep face representations using small data. *IEEE Transactions on Image Processing*, 27(1):293–303, Jan 2018.

[13] G. 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.

[14] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874, 2014.

[15] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision (ECCV)*, pages 158–171. Springer, 2012.

[16] D. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009.

[17] B. F. Klare, B. Klein, E. Taborsky, et al. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1939, 2015.

[18] A. Kortylewski, B. Egger, A. Schneider, T. Gerig, A. Morel-Forster, and T. Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2174–217409, June 2018.

[19] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[20] Z. Lei, M. Pietikäinen, and S. Z. Li. Learning discriminant face descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):289–302, Feb 2014.

[21] Z. Luo, J. Hu, W. Deng, and H. Shen. Deep unsupervised domain adaptation for face recognition. In *International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 453–457, May 2018.

[22] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[23] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa. Active user authentication for smartphones: A challenge data set and benchmark results. In *International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8, Sept 2016.

[24] N. C. Mithun, R. Panda, and A. K. Roy-Chowdhury. Generating diverse image datasets with limited labeling. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 566–570, New York, NY, USA, 2016. ACM.

[25] H. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *International Conference on Image Processing (ICIP)*, pages 343–347, Oct 2014.

[26] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[27] M. Parchami, S. Bashbaghi, and E. Granger. Video-based face recognition using ensemble of haar-like deep convolutional neural networks. In *International Joint Conference on Neural Networks (IJCNN)*, pages 4625–4632, 2017.

[28] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In M. W. J. Xianghua Xie and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015.

[29] P. Phillips, H. Wechsler, J. Huang, and P. J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[30] A. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré. Learning to compose domain-specific transformations for data augmentation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 3236–3246. Curran Associates, Inc., 2017.

[31] J. Ren, X. Jiang, and J. Yuan. A complete and fully automated face verification system on mobile devices. *Pattern Recognition*, 46(1):45 – 56, 2013.

[32] 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.

[33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, June 2015.

[34] A. F. Sequeira, J. C. Monteiro, A. Rebelo, and H. P. Oliveira. Mobbio: A multimodal database captured with a portable handheld device. In *Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 133–139, Jan 2014.

[35] K. Sohn, S. Liu, G. Zhong, X. Yu, M. Yang, and M. Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *International Conference on Computer Vision (ICCV)*, pages 5917–5925, Oct 2017.

[36] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 443–450, Cham, 2016. Springer International Publishing.

[37] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *2015 IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 2892–2900, June 2015.

[38] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars. A deeper look at dataset bias. In *Domain Adaptation in Computer Vision Applications*, pages 37–55. Springer, 2017.

[39] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1528, June 2011.

[40] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, July 2017.

[41] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I. IEEE, 2001.

[42] G. Wen, H. Chen, D. Cai, and X. He. Improving face recognition with domain adaptation. *Neurocomputing*, 287(C):45–51, 2018.

[43] C. Whitelam, E. Taborsky, A. Blanton, et al. Iarpa janus benchmark-b face dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 592–600, July 2017.

[44] X. Xing, G. Xu, B. Cai, C. Qing, and X. Xu. Face verification based on feature transfer via pca-svm framework. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1086–1091, 2017.

[45] H. Xu, J. Zheng, A. Alavi, and R. Chellappa. Cross-domain visual recognition via domain adaptive dictionary learning. *CoRR*, abs/1804.04687, 2018.