Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-11167-6

Does face restoration improve face verification?

André Sobiecki[1] · Julius van Dijk[2] · Hidde Folkertsma[2] · Alexandru Telea[2,3]

Received: 3 November 2019 / Revised: 12 April 2021 / Accepted: 5 May 2021 /
Published online: 09 August 2021

Abstract
Methods for face verification works reasonably well on face images with standardized (frontal) face positions and good spatial resolution. However such methods have significant challenges on poor resolution images, poor lighting conditions and not standard (frontal) face positions. In this paper, we survey the capability of existing face restoration and verification methods, with the aim of understanding how useful face restoration methods are for face verification. We propose a qualitative and quantitative comparison benchmark, and apply it on eight methods for face restoration and six methods for face verification, on several real-world low-quality images from a surveillance context, and outline observed advantages and limitations. Experiments shows that each restoration method can affect each face verification method differently, with fewer than the half of face restoration methods helping face verification. Interestingly, some face restoration methods with less good qualitative evaluation helped face verification the most. Experiments also show that face verification works less good if the resolution decreases.

Keywords Face verification · Face restoration · Face recognition · Image processing · Machine learning · Pattern recognition

1 Introduction

Methods for face recognition and verification (FRV) are becoming increasingly important in many application areas such as healthcare, the gaming industry, user studies, and homeland defense [40, 42]. Such methods work reasonably well on face images with standardized (frontal) face positions, good spatial resolution, and controlled lighting conditions, such as found in many imaging benchmarks. However, images are often acquired in very different conditions, *e.g.*, varying viewpoints, poor lighting, low resolution, and motion blur.

Face recognition and verification with poor resolution images without labels and ground truths is an important task for forensic investigation. Terrorists and other criminals have

✉ André Sobiecki
sobieckiandre@gmail.com

[1] São Bento do Sul city, Santa Catarina State, Brazil

[2] University of Groningen, Groningen, The Netherlands

[3] Utrecht University, Utrecht, The Netherlands

posted thousand of videos and images in the internet where the face images are of poor quality. Everyday new illegal videos and images are posted in the internet and many of these videos and images doesn't have the minimum required quality for the methods of face verification. In many cases is not possible to have one person to watch each image and/or video for doing face verification and/or recognition, because the investigators already are overwhelmed with digital data. Therefore is important that face images with poor digital quality are restored and/or verified and/or recognized.

Over the past decades many methods for image restoration have been proposed. The most methods work on any images but there are some methods made specifically for restoration of one type of image, such as face images. The evaluation of these methods is done mostly qualitatively only. When quantitative metrics are used, these typically only include PSNR [8], which requires a ground truth image, that may not always be available. Most of the proposed restoration methods are tested with images having artificial noise (blur or salt and pepper noise), which are quite different from real-world noise. According to our knowledge the methods of face restoration have never been tested in algorithms of face verification and/or recognition.

In this paper, we approach face recognition and verification (FRV) with two interrelated goals:

1. support face verification using face images with poor resolution and possibly a wide range of poses and other acquisition parameters;
2. study both qualitatively and quantitatively how face restoration can improve face verification.

To achieve the above, we design several experiments to evaluate several methods of face verification and recognition. We select seven face verification and six face recognition methods based on a set of criteria that describes what *practitioners* typically require from a method to be usable and useful, based on a survey of existing face recognition methods. We use face images with significantly poor resolution and no artificial noise from three databases. We test face verification using original and restored face images. We compare the results of face verification using the original low-quality face images with the restored face images. The structure of our experiments consists of two tests: we first test face recognition using originals and/or poor quality face images; next, we test face recognition using the restored images. This allows us to study if a restored face image can be easier recognized than an original, low-quality and resolution, face image.

The structure of this paper is as follows. Section 2 reviews related work. Section 3 presents the selected methods and the evaluation method. Section 4 present our experiments for face restoration, face recognition, and face recognition after face restoration. Section 5 discusses our findings. Section 6 concludes the paper.

2 Related work

As the number of papers in the scope of image restoration and verification is huge, we next focus only on results which are closest related to our specific goal. We review the methods of face restoration and face recognition. Firstly we want to evaluate the methods of face restoration and the methods of face recognition separately and secondly we want to see if the current methods of face restoration add value to face recognition.

2.1 Face or image restoration

We define face restoration as $f_{restor} : I \rightarrow I$ where I is the set of all possible (face) images and any given $x \in I$, $f_{restor}(x)$ is closer to a 'clean' image of the face of the person that was captured in I than I itself. Since we cannot in general measure ground-truth (we don't have 'ideal' images of the faces of persons in I), we evaluate the quality of f_{restor} by using 'proxy' quality metrics that look for typical things found in good-quality images, like contrast, details, texture and the results of algorithms of face recognition. Most of the methods of image restoration works on any image but there there are some methods that are made only for face image restoration such as [6, 30] and [29].

The first proposed methods for face restoration come from the more general class of image super-resolution generation. Bicubic interpolation, introduced by [12], has become the *de facto* standard reference method in image super-resolution papers. The method proposes a k by k kernel to create a bicubic spline interpolation to the input image, aiming at generating super-resolution details. The method resizes sharp, low-resolution images into a smooth, larger ones. Matsushita et al. [20] propose an algorithm that performs both deblurring and multiple-image super-resolution. While the method is fast (linear in the number of processed pixels) and simple to implement, it is only demonstrated on three images in the original paper.

Other popular approaches include deconvolution-based methods such as Wiener filter, Lucy-Richardson filters, and Tikhonov regularization [3]. While such methods work well on a broad spectrum of blurred images and are computationally fast, some do require one to specify the type of blurring kernel that affected the image, *e.g.*, out-of-focus, Gaussian, or motion blur. SmartDeblur is a recent open source implementation covering such approaches [41].

Lately the advent of deep learning methods and in particular convolutional neural networks (CNNs) spawned multiple approaches for super-resolution image generation and restoration. While many such methods achieve high quality (measured *e.g.* in peak signal-to-noise ratio, PSNR), their results often lack finer details, which is easily noticeable by humans. To alleviate this, [16] introduce SRGANs (Super-Resolution Using a Generative Adversarial Networks), the first GAN-based super-resolution imaging method that aims to hallucinate (*i.e.*, synthesize) such details. Dong et al. [7] propose the first method in which a CNN is used to perform single-image super-resolution. In contrast to earlier work, this method reconstructs all (RGB) image channels simultaneously, and can handle real-world images in seconds on a typical PC. However, this approach is not tuned at face images and their specific type of details.

Unlike other deep learning methods, [44] explicitly handles low-resolution images which have been degraded by multiple factors (*e.g.* blur kernel types and noise level). Compared to other methods, this approach visually performs better when the degradation is more complex than just bicubic downsampling. Kupyn et al. [14] propose a deep learning-based blind deblurring method. Similar to [16], this method also relies on GANs. The method achieves an average structural similarity (SSIM) score of 0.816, which is impressive. In addition, the authors test their method by running object detection (using the YOLO real-time detector and associated benchmarks [24]) on the generated images. Deblurring results in more objects being correctly detected.

Related to our goal of handling (very) low quality input images, [6] propose FSRNet, a face-specific single-image super-resolution method. This method is specifically developed for very low resolution images – good results are demonstrated on images up to 16×16 pixels. Key to this method is a multi-stage approach, in which coarse facial features are identified by a deep network and next refined by a separate network.

Very close to our goal, [10] propose a method specifically developed for faces obtained from video surveillance content. While not a deep learning method per se, this technique makes use of sparse coding: Eigenfaces similar to the input are retrieved from a training set and next refined with local details using an approximate nearest-neighbor query. While results look in general good, they show a variable degree of hallucinations which may look unnatural. Closer to applications, [5] is an open-source project which integrates a wide range of techniques for performing super-resolution and deblurring of images. The key added-value of this work is offering the possibility to mix-and-match techniques presented in four recent papers on the topic [11, 13, 16, 28], as well as the open-source availability and good documentation.

2.2 Face verification

We see that face verification and face recogniton are very related to each other. Face verification aims to compare two face images to output a Boolean value (faces match, i.e. are of the same person, or do not match). Face recognition searches a given query face image in a set of images (face database) and returns the most similar (best matching) images. We are actually interested in face verification and we found that most methods of face verification use a method of face recognition. We define a function for face verification $f_{verif} : I_1 \rightarrow I_2$ where I_1 and I_2 are the possible face images. Thus, a face verification method gets two images and outputs $f_{verif}(x)$ being the label of that face, i.e. if two face images are the same person.

A considerable amount of methods for face recognition have proposed over the past few decades. Turk et al. [34] presented a paper describing a method for detection and identification of faces in near real time conditions. The method is called Eigenfaces. Before this approach the leading methods were based around the relationship between facial features such as the position of the eyes, nose and mouth. However, research has shown that the direct relationships between these features is not sufficient to achieve the same level of face identification as humans [4]. Turk et al. [34] took a different approach and were inspired by information theory. The algorithm works by constructing Eigenfaces, which can be thought of as a set of features which together characterize the variation between face images. First the algorithm has to be trained, training the algorithm uses known images of identities to construct the corresponding Eigenfaces. When testing the identity of an unknown image it will try to approximate the image based on linear combinations of the Eigenfaces that were generated at the training stage.

Linear discriminant analysis (LDA) was proposed by [33]. They used it to solve a taxonomic problem, classifying flowers. [2] proposed using LDA for face recognition by the so-called Fisherfaces. Their extensive testing showed that their method produces a lower error rate than Eigenfaces. They tested both an Eigenfaces implementation as their Fisherfaces implementation on a dataset with a lot of variation in the lighting of faces as they assumed that Eigenfaces would not behave well to such changes.

Ahonen et al. [1] proposed an adaptation of local binary patterns (LBP), which was used to extract texture features, such that it could be used for face recognition [37]. The local binary pattern operator works by taking a block of a certain size NxN, the kernel, and applying a threshold such that a binary pattern is obtained. This improves upon methods like Eigenfaces and Fisherfaces by extracting features on a smaller scale than the entire face, such that many extracted features are the same for different face representations of the same person.

Many methods of face recognition using machine learning have been proposed in last decade. [39] introduce a framework for light convolutional neural networks for face recognition and face verification know as LightCNN. Since it is a framework multiple models are available that make use of this structure. In the paper they define Light CNN-4, CNN-9 and CNN-29 where the number indicates the number of convolution and max-pooling layers. Light CNN-9 consists of 9 convolution layers and 9 max-pooling layers. The models are trained on the MS-CELEB-1M dataset.

FaceNet [26] computes the Euclidean distance of each face. FaceNet learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors. FaceNet uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches.

Liu et al. [18] address a specific face recognition problem by their *SphereFace* method. They argue that, at the time of writing, there were few suitable loss functions in models to specifically compare facial features. They introduce a variation on the softmax loss function, a-softmax. This loss function is able to make a better comparison between extracted facial features than regular softmax.

Ning et al. [21] develops a new dimensionality reduction method, named Biomimetic Uncorrelated Locality Discriminant Projection (BULDP), for face recognition. It is based on unsupervised discriminant projection and two human bionic characteristics: principle of homology continuity and principle of heterogeneous similarity. The performance of the proposed algorithms is evaluated and compared with the state-of-the-art methods on four public benchmarks for face recognition. It is a promising method for face recognition using face images with high resolution because the method uses a technique of dimensionality reduction which is necessary to reduce the feature size if the captured image data lies in a high-dimensional feature space. This is not a requirement for us because we apply face restoration and face verification in face images with poor resolution. Separately, [23] disscuss seven diferent methods for the task of ID photo verification. However, in contrast to our work, they use images of relatively good quality and, given the nature of the photos (IDs), also images which have a standardized pose and very limited occlusion.

Ning et al. [22] present a survey of methods used for the generation of multi-view frontal images of human faces. Besides discussing the challenges of multi-view generation, this survey also discusses the challenges of *generating* such views. These challenges are quite related to our challenges of *interpreting* – that is, recognizing and verifying – faces that are provided from multiple views.

3 Method selection and comparison

Comprehensively comparing FRV methods is extremely daunting. On the one hand, a simple search on GitHub reveals over 300 methods related to image restoration and more than thousands for face verification and/or recognition, most of which are, at least technically, applicable to our task. On the other hand, while recent papers on the topic show very good results, replicating their findings is far from trivial.

We approach the selection of candidate methods to compare from a *practical* perspective. For this, we define eight criteria that methods should satisfy for them to be good candidates

Table 1 Criteria used to select super-resolution methods to test for low-quality in a FRV context

Criterion	Detailed explanation
Input type	The method uses multiple (M) images as input for output one restored image or the method uses one single (S) image as input for output one restored image.
Validation	The method has multiple promising results presented in its paper and/or supplementary material. Ideally, the method performs well on face images and is demonstrated on multiple real-world datasets, *e.g.*, Set5 and Set14 [9] and CelebA [19].
Replicability	One can replicate the results presented together with the method on the same datasets relatively easily.
Availability	The method has available source code or can be easily implemented. For machine learning approaches, available pre-trained models are preferred. If code is available, code coming from the method's authors rather than from third parties is preferred.
Speed	The method is reasonably fast to compute (less than one minute per image). For machine learning approaches, either the training completes within one hour, or pre-trained models are available.
Usability	The method does not require fine tuning of many parameters to get acceptable results. Ideally, the method comes with parameter presets which deliver the desired results without requiring further tuning.
Documentation	The method is well documented so one can build, tune, and run the software with reasonable effort.
Works on our data	The method should be able to handle our own datasets, besides datasets from third-party benchmarks or datasets supplied with the method itself.

for being used in a practical (rather than academic) context (Table 1). The same methodology has been used earlier for ranking other imaging tools [35] and visualization tools [27] from a practical perspective. We rank all criteria from Table 1, except input type and replicability, using a 5-point Likert scale (with values denoted by --, -, +/-, + or ++). Here, -- indicates that the method scores very poorly at that particular criterion, while ++ indicates a very good score. Input type has two categorical values: (S)ingle image and (M)ultiple images. Replicability is binary (yes/no).

We next gathered a set of over 50 candidate methods from both the literature and GitHub, including all methods mentioned in Section 2, and scored these using the criteria in Table 1. During this process, we used an 'early elimination' procedure, *i.e.*, eliminated from further analysis methods that score unacceptably low on at least one criterion, so as to make their practical usage impossible. Examples hereof are methods which do not have a public (or easily replicable) implementation; methods which contain only very limited results presented in the respective paper(s) and thus raise serious concerns on their validation; or methods whose documentation is so limited as it makes replicating the presented results hardly possible.

After performing this step, we were left with 14 face restoration and 11 face verification methods to study in further detail. Table 2 shows these methods and their scores along the criteria in Table 1. For completeness *vs* the related work discussion (Section 2), we also included here the methods of [10, 20, 25], and [21], for which we did not find an implementation. Apart from the above four, all methods in Table 2 score highest on availability (++), except [41], whose availability we ranked (+/-) as its open-source variant lacks a blur kernel recognition functionality. Four methods can handle multiple images; the rest work only for single images. Most methods show good to very good results in their respective papers, except [12]. However, we included this method in the further analysis given its standard

Table 2 Scores of restoration and face verification techniques selected to further study *vs* the desirable criteria outlined in Table 1

Algorithm	Restoration or Verification	Input type	Validation	Replicability	Availability	Speed	Usability	Documentation	Works on our data
[20]	Restoration	M	+/-	no	-	++	+	-	no
[5]	Restoration	S	+	yes	++	+/-	-	++	yes
[7]	Restoration	S	+	no	++	-	-	++	no
[44]	Restoration	S	+	no	++	-	–	++	no
[16]	Restoration	S	+	no	++	-	-	++	no
[10]	Restoration	M	+	no	-	++	-	-	no
[32]	Restoration	M	+	yes	++	+/-	-	++	no
[25]	Restoration	M	+	no	-	+	-	-	no
[14]	Restoration	S	+	yes	++	+/-	-	++	yes
[6]	Restoration	S	+	yes	++	+/-	-	++	yes
[12]	Restoration	S	-	yes	++	++	++	++	yes
[41]	Restoration	S	+	yes	+/-	+	+	+/-	yes
[30]	Restoration	S	+	yes	+/-	+	+	+	yes
[29]	Restoration	S	+	yes	+/-	+	+	+	yes
[34]	Verification	S	++	yes	+	+	+	+/-	yes
[2]	Verification	S	++	yes	+	+	+	+/-	yes
[1]	Verification	S	++	yes	+	+	+	+/-	yes
[39]	Verification	S	++	yes	+	+	+	+/-	yes
[26]	Verification	S	++	yes	+	+	+	+/-	yes
[18]	Verification	S	++	yes	+	+	+	+/-	yes
[31]	Verification	S	++	no	-	+	+	+/-	yes
[38]	Verification	S	++	no	+	+	+/-	+/-	yes
[36]	Verification	S	++	no	++	+	+/-	+/-	yes
[21]	Verification	S	++	no	- -	+	+/-	+/-	yes
[23]	Verification	S	- -	+/-	+/-	+/-	+	+/-	yes

quality and long-standing in the literature. Finally, all of the CNN-based methods are scored as "poor" on usability because their training (and sometimes testing) requires the non-trivial setting of several hyperparameters.

For the methods in Table 2, we next tested that we can (1) build them from source code, (2) replicate the results given by their authors, and (3) have them working on our own datasets. The methods of [7] and [16] did not work due to software dependency issues and lack of documentation. [44] did not work because it relies on the user to input a blur kernel and noise level; these are unknown for our dataset. [32] worked as expected on the images given by its authors, but not on our dataset. As such, we had to exclude these methods from further evaluation.

This leaves us with thirteen methods – 7 for face verification and 6 for face recognition – plus our proposed method to evaluate in detail as shown in Table 3). In detail: FSRNet uses a CNN for face-specific super-resolution. Neural Enhance does the same, but for general-purpose super-resolution. DeblurGAN uses a special type of conditional GAN for its training. Unlike the previous two methods, it does so for deblurring. The more traditional approaches to deblurring and super-resolution are comprised of Smart Deblur and bicubic interpolation, respectively. We found that usually face verification methods have better quality of source code and documentation available than methods of image restoration.

3.1 Mixing FSRNet with the original image

Mixing FSRNet's result I_{FSRNet} with the original image $I_{original}$ can improve upon I_{FSRNet}. Specifically, we noticed that I_{FSRNet} shows good results in uniform-color areas but less good results in non-uniform areas. The idea of mixing is that the uniform-color areas use more from I_{FSRNet} and the nonuniform-color areas use more from the $I_{original}$. We achieve this mixing by computing

$$I_{result} = I_{FSRNet} \cdot (1 - \|\nabla I_{FSRNet}\|) + I_{original} \cdot \|\nabla I_{FSRNet}\|. \quad (1)$$

where $\|\nabla I_{FSRNet}\|$ is the gradient of I_{FSRNet}, computed by central differences, and normalized to [0, 1]. High values of $\|\nabla I_{FSRNet}\|$ indicate edges or textures in the image; low values

Table 3 Details on the seven face verification and six face recognition methods selected for in-depth evaluation

Authors	Restoration or verification	Method name	Kind of method	Face-specific
[5]	restoration (SR and deblur)	*Neural Enhance*	CNN	no
[41]	restoration deblur	*Smart Deblur*	Traditional	no
[6]	restoration SR	*FSRNet*	CNN	yes
[12]	restoration SR	*Bicubic interpolation*	Traditional	no
[30]	restoration deblur	*Beter digitalization*	Segmentation, inpaint	yes
[29]	restoration deblur	*Face Inpaint*	Segmentation, inpaint	yes
[14]	restoration deblur	*DeblurGAN*	CNN	no
[34]	verification	*Eigenfaces*	Covariance matrix	yes
[2]	verification	*Fisherfaces*	Linear discriminant analysis	yes
[1]	verification	*local binary patterns*	CNN	yes
[39]	verification	*LightCNN*	CNN	yes
[26]	verification	*FaceNet*	CNN	yes
[18]	verification	*SphereFace*	CNN	yes

indicate uniform color areas. In our experiments we next compare I_{result} with I_{FSRNet} and with all other methods for face restoration.

3.2 Implementation of Face Verification

For executing the different experiments we adopted the implementation of the methods such that they have a uniform application programming interface (API). We have created a single pipeline which can use any method adhering to our API as shown in Fig. 1.

3.3 Databases

For our testing, we use four databases of images:

- **Terrorists:** 20 faces images and of 10 persons. For each person there are two images: (A) a very poor quality image (about 45x45 pixels resolution) and (B) a slightly higher quality face image (about 200 x 200 pixels resolution). The images A are all normalized to 128 x 128 pixels, otherwise face detection doesn't work. These face images come from videos of potential terrorists where we used some face detection. The images have been acquired by running facial detection algorithms proposed by [43] on videos of potential members of the IS terrorist group that were posted on YouTube. Most images suffer from low resolution, blur, and/or noise. In addition, face details are sometimes obscured by cap covers and/or facial hair. Based on expertise from a video surveillance company, we selected these images to be typical of those that (a) are typical in surveillance tasks, but (b) surveillance software have difficulty in analyzing and recognizing.
- **Terrorists restoration results:** 80 images. We have eight methods of face restoration. From each method we got ten images as results
- **LFW database:** 13233 images, is one of the most used databases in other papers [15].
- **Lena:** Two synthetically degraded *Lena* images (bicubically downsampled and motion blurred, respectively);

We next analyzed the results produced by the eight evaluated methods both qualitatively and quantitatively.

4 Results

To answer the questions proposed in the introduction we have structured our results into three sections. First, we test the six selected methods using an experiment that uses our low

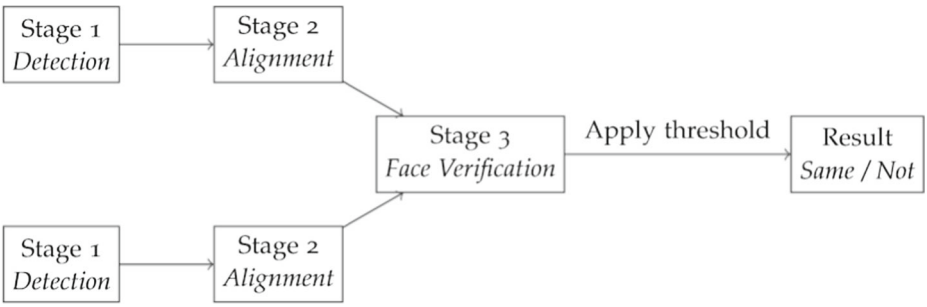

Fig. 1 Pipeline used for all face verification experiments

resolution dataset (Section 4.1).Next, we test eight different methods of face image restoration (Section 4.2). Thirdly, we test face verification after face restoration by repeating the previous test but on the datasets that have been constructed by applying image restoration techniques to the low resolution dataset (Section 4.3). Section 4.3.1 tests the influence of poor resolution images on face verification. Finally, we test the robustness of face verification methods using multiple photos of the same individual in different facial positions (Section 4.3.2). For all the 14 evaluated methods (Table 3), parameter settings are fixed per-method and identical to those proposed by the respective authors.

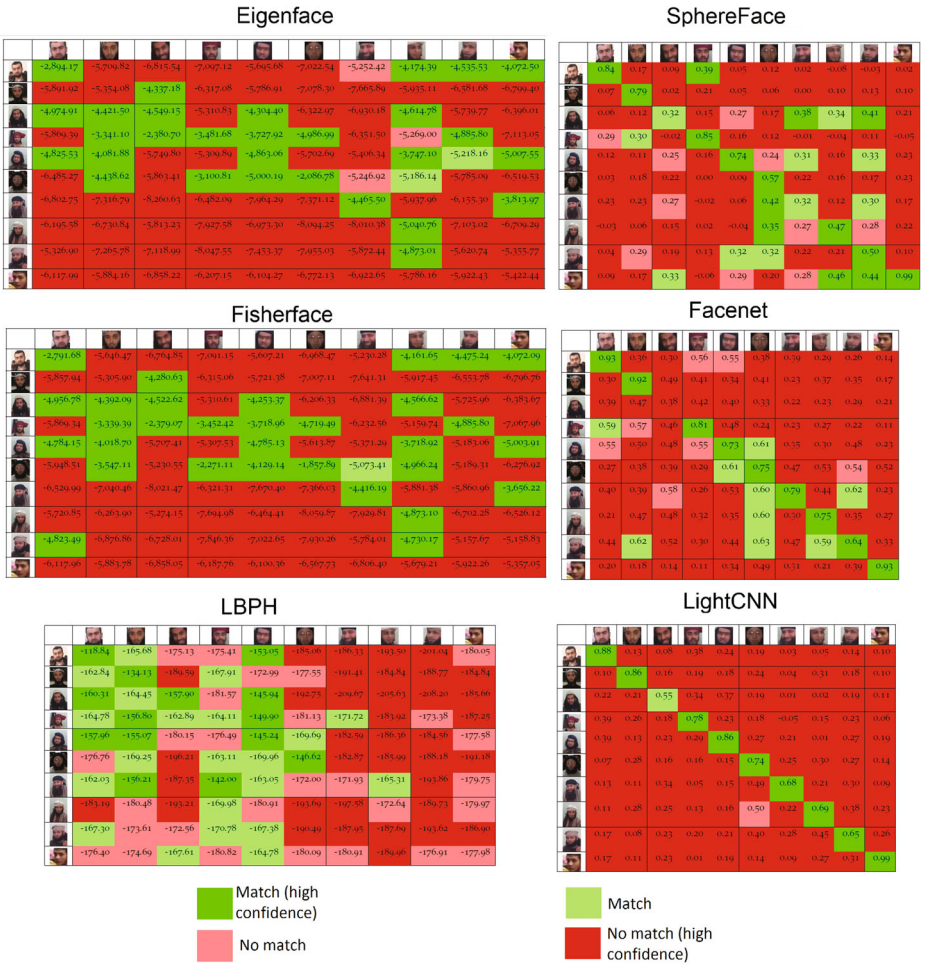

Fig. 2 Matrix of face verification using the *terrorists* database comparing six methods of face verification.

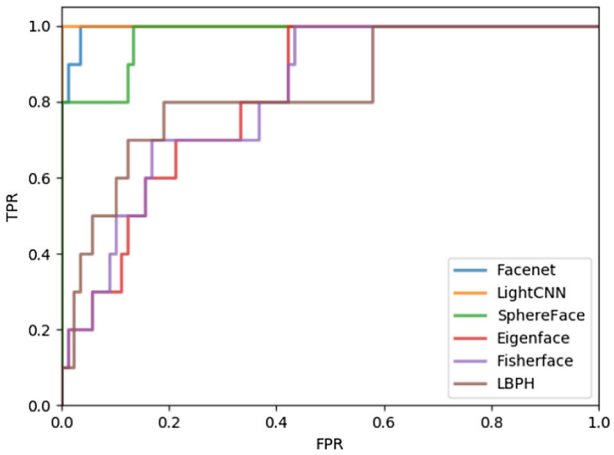

Fig. 3 ROC curves of the database *Terrorists (A)* and the database *Terrorists (B)*

4.1 Face verification

For this experiment we extracted a low resolution sample, combined with a higher resolution sample of the same individual from videos. The low resolution samples have all been up-scaled to 128x128 pixels using bi-cubic interpolation. For each identity we compare a high resolution image against a low resolution image of all 10 identities. Different cell colors represent whether the score will be accepted as a match under a certain threshold. We have chosen to use the threshold associated with the lowest equal error rate (EER) as opposed to choosing the threshold associated with the best accuracy as that would result in a minimal accuracy of 90% since the test contains only 10 true samples and 90 false samples.

Figure 2 presents the matrix of experiments with six methods of face verification. We test the database *Terrorists (B)* with the database *Terrorists (A)* and we choose ourselfs the

Fig. 4 Evaluation on *Lena* image: **a** original, **b** original downsampled to 64x64, **c** Bicubic Interpolation, **d** Smart Deblur, **e** Neural Enhance, **f** DeblurGAN, **g** FSRNet, **h** [30], **i** [29] and **j** mixing $FSRNet$ with $original$

best threshold in this experiment. According to this experiments LightCNN discriminates *matches* and *nomatches* better. As second place FaceNet has presented some false positive and false negative results.

Figure 3 shows the ROC curves tof testing the database *Terrorists (B)* with the database *Terrorists (A)* as presented in Fig. 2. The ROC curves clearly denote that the newer, CNN based, methods perform better on this experiment. When comparing the matrices of the different methods we observe that Eigenface and Fisherface provide similar results. This is not surprising as the methods work in a similar way. SphereFace and FaceNet look a lot worse than LightCNN. This is mostly because of choosing a threshold that results in an equal error rate. When choosing a slightly higher threshold, the number of false positives will drop dramatically increasing the overal accuracy.

Fig. 5 Surveillance dataset: **a** original, **b** bicubic interpolation, **c** Smart Deblur, **d** Neural Enhance, **e** DeblurGAN, **f** FSRNet, **g** [30], **h** [29] and **i** mixing $FSRNet$ with $original$

4.2 Face restoration

In this subsection we evaluate the methods of face restoration. Figure 4 shows the evaluation results for the *Lena* test image, which we manually downsampled from the original (512×512 pixels to 64×64 pixels). As visible, Smart Deblur and Neural Enhance yield the visually best resolution results in terms of clarity, crispness, and reconstruction of details. Interestingly, FSRNet performs the poorest, as it creates a high amount of blur.

Figure 5 shows an example subset of 10 faces from our surveillance dataset. The original input video images range between 46×46 and 64×64 pixels. We used the tested methods to upsample these images to 128×128 pixels, which is a resolution deemed sufficient by our surveillance experts to either manually detect specific persons, or else run face detection software to do this automatically. In stark contrast to the *Lena* image (Fig. 4), FSRNet performs here arguably the best, being able to produce images very close to the original input. However, FSRNet also exhibits the tendency to hallucinate certain details not present in the original image, such as the faint trace resembling spectacles (Fig. 5, right column, row 6 from top), and the peculiar unnatural details added to th eye (Fig. 5, right column, row 4 from top) and respectively beard edges (Fig. 5, right column, bottom row). Occasionally, FSRNet also changes the overall facial tint (Fig. 5, right column, row 6 from top). Separately, we see that DeblurGAN tends to increase the noisiness level significantly more than all other methods (Fig. 5, second-right column).

Figure 6 zooms in on several images from Fig. 5, for further insights. The high noise level yielded by DeblurGAN becomes now clearer. Also, we see that FSRNet appears to achieve the highest contrast from all tested methods, see *e.g.* the nose highlights (first character from top) and the right-eye arcade highlights and dark brow (second character from bottom).

4.3 Face verification after restoration

In this experiment we repeat the face verifcation experiment but instead of comparing the high resolution images with the low resolution images we use the result of face restoration

Fig. 6 Surveillance dataset zoom-in: **a** original full-image, **b** original zoom in, **c** bicubic interpolation, **d** Smart Deblur, **e** Neural Enhance, **f** DeblurGAN **g** FSRNet, **h** [30], **i** [29] and **j** mixing $FSRNet$ with $original$

methods. We have tested the database *Terrorists (B)* with the database *Terrorists (A)* and the database *Terrorists (B)* with the database *Terrorists restoration results*. The barplots in Fig. 7 present the results of testing *matches* and *nomatches* faces. We have tested the database *Terrorists (B) vs* the database *Terrorists (A)* and the database *Terrorists (B) vs* the database *Terrorists restoration results*.

For all methods except LightCNN a discrepancy between the highest scoring methods based on AUC and Accuracy is present. As explained in the previous section the accuracy at the equal error rate is not necessarily the best possible accuracy while the AUC is a performance metric independent of a chosen threshold. Thus we can assume that in the case of FaceNet, not only DeblurGAN and FSRNet have a improved accuracy but also the mixing method. SphereFace present the largest discrepancy, at the equal error rate only the accuracy improves with DeblurGAN but the AUC scores better with [30], NeuralEnhance, DeblurGAN and [29]. For FaceNet the difference in accuracy between DeblurGAN and the resized images is 3%.

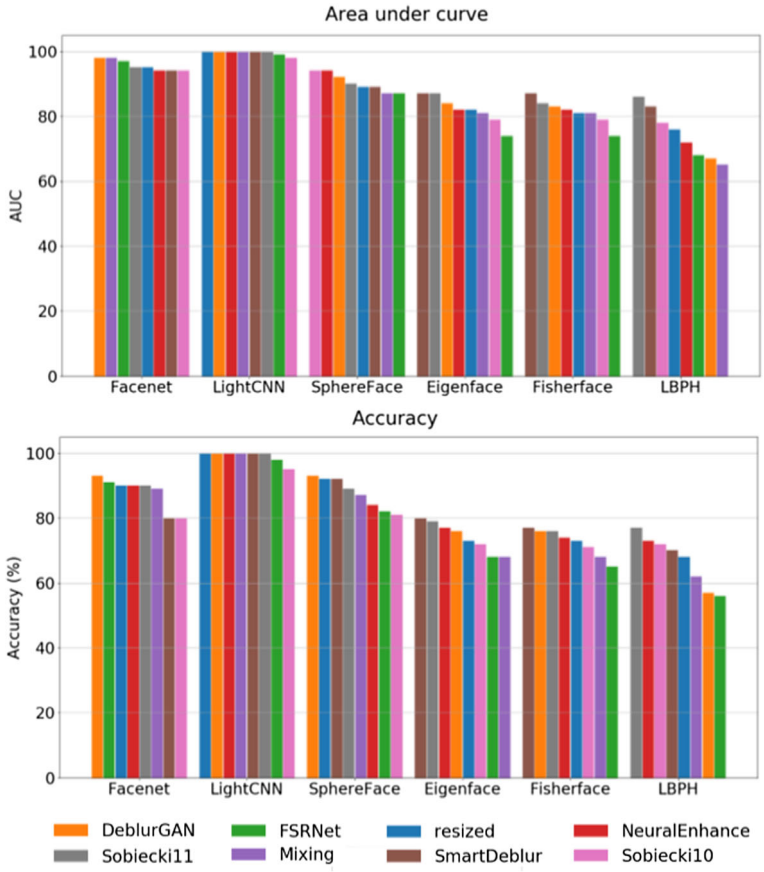

Fig. 7 Barplots with accuracy and area under the ROC shows how each face restoration method affects each face verification method (*matches* and *nomatches* cases)

Fig. 8 Left: original image taken from LFW. Right: our poor-quality variation of the left image

4.3.1 Face verification after downsampling

In this experiment as shows Fig. 8 we have lowered the resolution, using bi-cubic interpolation, of each image in the LFW dataset [15] to 50x50 pixels, afterwards we increased the resolution back to 128x128 pixels such that we could use it as input for each method. We will refer to this set as the low quality dataset.

It is important to note that this will also have detrimental effects to the face detection algorithm, which is used for aligning the images before running face verification methods. We have opted to use the original face detection and alignment landmarks such that we can run the test in the same manner as the authors of the dataset recommend.

The LFW dataset contains guidelines for testing the accuracy of methods. They include 10 test sets, which each contain 300 matches and 300 non match cases. We will now refer to the 10 test sets as a single test. This test has been run three times per method. Once testing the original LFW dataset against the original dataset. Once using the original LFW dataset against the low quality variation. Once using the low quality dataset against the low quality set. The graphs in Fig. 9 shows poor image quality causes lower accuracies of face verification.

Table 4 shows that the accuracy decreases if resolution decreases to all methods. All methods are having better results with *O-O*. We see that four methods are having better results with *O-L* than *L-L* and only LBPH has better results with *L-L* than *O-L*.

4.3.2 Robustness of face verification

In the first and third subsection we only tested two images per identity. In this next experiment we aim to determine how the methods perform when encountering a variation of face expressions and lighting conditions. We plot the output score of each image, as verified with the reference image, such that we can compare how the variations influence the score.

Figure 10 tests face verification when the faces changes position and expression. We test the reference image with ten images from the same person and ten images from other persons. The ten images of the same person present different expression and/or head position.

The threshold that results in the best accuracy is plotted using a black line. All the results above this line are accepted as a match, while the results below are not. Surprisingly each method achieved a 100% accuracy on this experiment.

Apparently all methods are robust to occluding objects, such as microphones. There are microphones in the images with same person and in the images with different person.

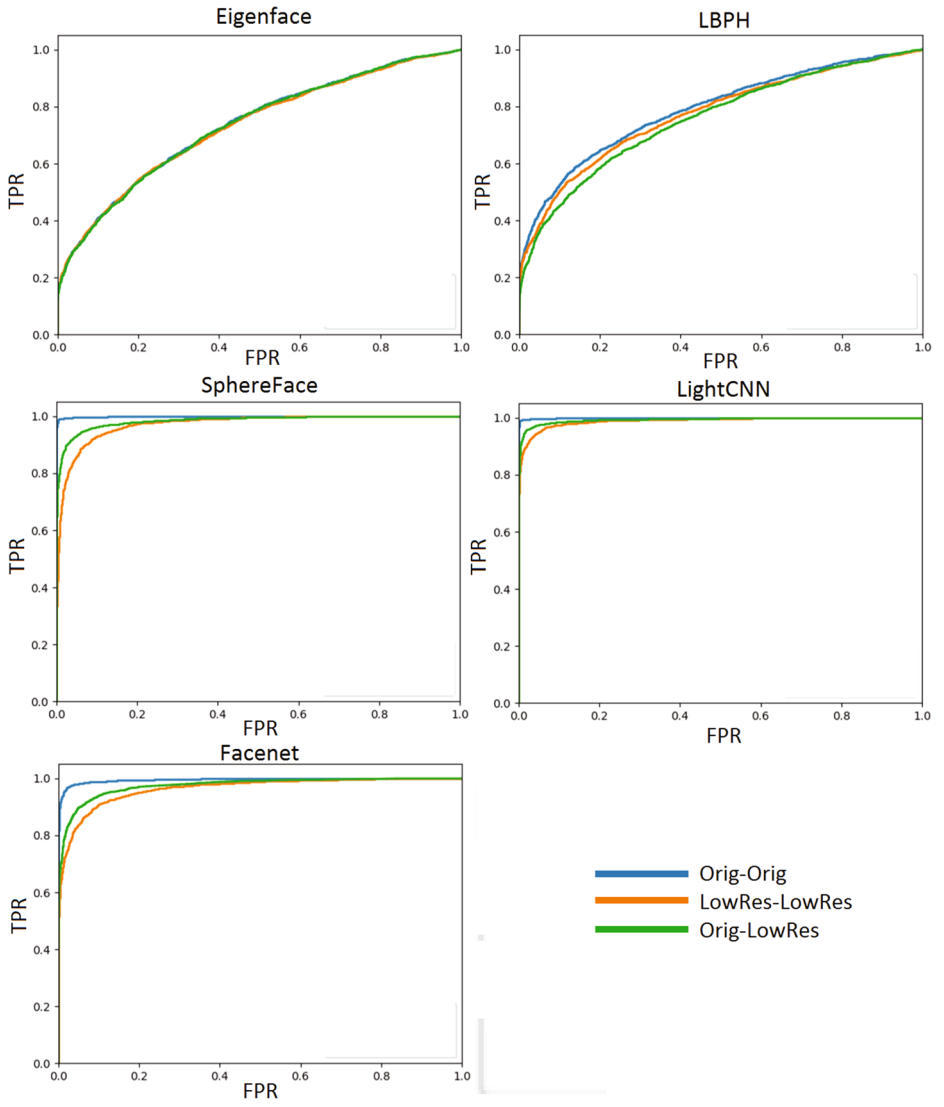

Fig. 9 ROC curves of four methods of face verification. $Orig$ means original images and $LowRes$ means low resolution images

5 Discussion

We distill several points from our experiments, as follows.

Face restoration: There are many methods of face verification. According to our experiments the face restoration method *DeblurGAN* gives the greater contribution to methods of face verification. The accuracy of face verification *FaceNet* gets about 3% higher using face restoration *DeblurGAN*. We have seen that each method of face restoration affects each method of face verification differently. Half of the methods for face restoration improves

Table 4 Accuracies at the EER for each method and test executed

Algorithm	O-O Acc @ EER ($\pm\sigma$)	O-L Acc @ EER ($\pm\sigma$)	L-L Acc @ EER ($\pm\sigma$)
Eigenface	68.1% ($\pm$2.3%)	68.0% ($\pm$2.4%)	67.9% ($\pm$1.8%)
LBPH	73.3% ($\pm$1.4%)	70.4% ($\pm$1.6%)	71.9% ($\pm$2.0%)
FaceNet	97.5% ($\pm$0.8%)	92.8% ($\pm$0.9%)	90.7% ($\pm$0.9%)
LightCNN	99.4% ($\pm$0.5%)	97.15% ($\pm$0.5%)	95.5% ($\pm$0.4%)
SphereFace	99.2% ($\pm$0.5%)	94.5% ($\pm$0.5%)	92.0% ($\pm$1.0%)

O-O denotes the original dataset compared with the original dataset. *O-L* denotes the original dataset compared with the low quality dataset. *L-L* denotes the low quality dataset compared with the low quality dataset

the results of face verification methods. FSRNet causes smoothness and sometimes some small noise. A mix between original and restored images have show more quality close to reality. Our experiments also shows that poor quality images negatively affects the results of face verification and this concludes that there are room for methods of face restoration be improved. In our opinion traditional filters of image processing will not give the restoration results. In the future the good methods of face restoration will be the methods that have some prior information about physiognomy and facial characteristics. We don't know if the future methods will be more machine learning or others.

Face verification: Methods of face verification perform different results in the Terrorists database than in the LFW database. According to the literature *FaceNet* has the highest accuracy on LFW dabase. With our terrorists database $lightCNN$ has the highest accuracy. Face detection and normalization are very important step for face verification and it deserves further studies.

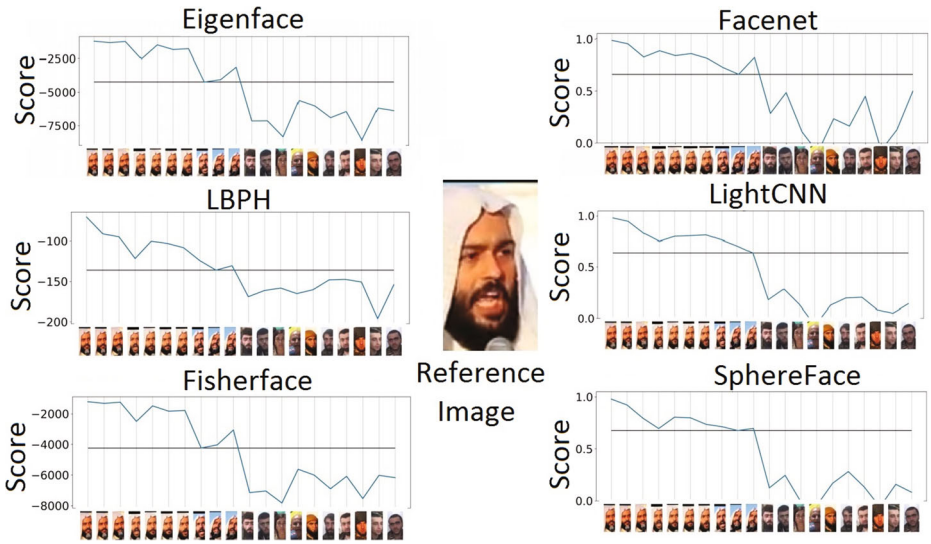

Fig. 10 Robustness to different position and expressions. The methods of face verification are robust to different kind of head position and face expression

Challenging input: Our real-world surveillance database consists of images that are at the same time low-resolution, noisy, and blurry. We notice that the tested methods set out to solve one, or possibly two, of these issues. This is also visible in several examples present in the literature on these methods. More specialized methods that aim to tackle all three issues jointly are needed.

Coverage: While we extensively scanned both the available literature and open-source repositories, we only found eight methods that we could actually apply – which, as explained in Section 3, involves obtaining their implementations, building them, testing the methods against 'ground truth' results provided with their distribution, setting their parameters, and actually running them on our images to obtain the output restored images. In particular, we could not find any multiple-frame super-resolution method that we could actually test. This indicates the (strong) need for more replicability in research related to restoration methods.

Specialization: The method that stood out in our evaluation, FSRNet, is trained on the CelebA dataset [19]. While also consisting of faces, this dataset contains quite different faces than the ones typically found in surveillance videos. Hence, for the specific goals related to surveillance, re-training of FSRNet on a more specific facial-image set could be a simple but effective way to achieve higher quality.

Face Detection: Face detection is an important step of face verification. We have used one single technique [43] in this paper. Face verification doesn't work without face detection and works limited on if the faces are not corrected aligned.

6 Conclusions

Our goal are face verification using low quality images and to know if face restoration adds value for face verification. We tested the performance of six methods on a real world dataset containing photos of terrorists, taken out of propaganda videos, and on a larger dataset to analyze the impact of low quality images on the verification process.

Generating facial images with better quality from surveillance videos is an important tool that can help both manual inspection and automated methods for face and person verification. In this paper, we conducted a study to assess the effectiveness of existing image restoration methods in this context. In contrast to other comparative studies of the same type, we have taken a strongly practical stance, focusing on methods which are available (implementation-wise), easily installable and configurable, have few or no parameters to tune, and can be applied to facial frames extracted from videos exhibiting poor resolution, noise, and blur. We performed an extensive literature and open-source repository search, which yielded in the end five methods that comply with these characteristics. We next tested these methods on a real-world set of facial images extracted from typical surveillance videos.

The experiments showed that restoration methods do influence the performance of verification methods, and they can improve the accuracy. However, the influence of a restoration method is not the same for all verification methods. One explanation is due to the fact how CNNs operate. The networks try to reduce the dimensionality of the input by extracting important features. Restoration methods can either enhance or distort these features, making

the method more accurate or less accurate. Qualitative analysis of the restoration methods showed that FSRNet the best restoration method. Hence, we expected the verification results to reflect that. The accuracy of FaceNet was only slightly raised by FSRNet, while it even lowered the accuracy of LightCNN and SphereFace dramatically. The best restoration method for LightCNN can not be determined as it achieved a perfect accuracy on the non-improved images. The best restoration method for FaceNet and SphereFace is DeblurGAN, which is surprising as the qualitative analysis classified this as one of the worst restoration methods. Again, a possible explanation lies in the way how CNNs operate. The networks extract certain features from the input image and does this in a different way than humans. What looks best to the human eye does not necessarily look the best for a CNN.

The test on the Labeled Faces in the Wild (LFW) has shown us the importance of good quality input images. On average the accuracy between the original images compared with the original image was 3.9% higher than the low quality images compared with the low quality images. Eigenfaces are less susceptible to the difference in quality. The CNNs are influenced in a large way by the lower quality images. The largest difference was reached by SphereFace: 7.2%.

From the obtained results, one method stands out – FSRNet [6] in terms of complying with all our criteria that capture practical usage. However, during our evaluation we also saw that FSRNet has the strongest tendency to hallucinate details which are not present in the input images, such as spectacles, facial details, highlights, or even skin tint. We believe this is an interesting finding since it suggests that a trade-off may exist in the current FSRNet design (and possibly the design of other related deep learning methods) between the *fidelity* of their output and its *realism*. From a forensics and surveillance perspective, hallucinating details is a topic to be treated with great care, as it may lead to incorrect identifications of the recorded persons. This opens the interesting future research direction of *constraining* the deep network in the types and extent of details that it is allowed to add to an input.

Our real world tests show promising results for using face restoration to enhance face verification. The dataset that we used consists mainly of unlabeled data, this makes running large tests impossible. Our current test only contained 10 true/match cases, and 90 false/no match cases. An argument can be made why this should be avoided. However, testing an equal amount of match as no match cases is also not representative of real world use cases [17]. An improvement can be made by running the test on more true matches, for instance by including more images per identity or by labeling more unique identities.

The experiments on the LFW dataset show a discrepancy between the accuracy reported by the authors and our findings for several methods. A possible explanation is the variation in the alignment algorithm used. For each method we used the same detection and alignment algorithm, while the original implementations of the tested methods did not use the same detection and algorithm method or implementation. FaceNet, LightCNN and SphereFace all use the same method for detection, but different implementations which result in close (within 2-3px), but not matching output of the detection algorithm.

We believe that the future methods of face restoration should be able to consider some information about physiognomy and ethnicity. In particular to the case of forensic research we conclude that some wider database with face images of criminals are important to develop and test the methods of face verification.

When generating the low quality version of the LFW dataset the face detection algorithm failed on more than 20% of the cases. Without a working detection method, verification is impossible. We expect that face restoration methods might have a dramatic improvement on the detection of faces in low quality photos.

References

1. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: Application to face recognition. IEEE Trans PAMI:2037–2041
2. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans PAMI:1–10
3. Campisi P, Egiazarian K (2001) Blind image deconvolution: theory and applications. CRC Press
4. Carey S, Diamond R (1977) From piecemeal to configurational representation of faces. PubMed - NCBI
5. Champandard AJ (2016) Neural Enhance: super resolution for images using deep learning. https://github.com/alexjc/neural-enhance
6. Chen Y, Tai Y, Liu X, Shen C, Yang J (2018) FSRNet: End-to-end learning face super-resolution with facial priors. In: Proceedings of IEEE CVPR, pp 2492–2501
7. Dong C, Loy CC, He K, Tang X (2016feb) Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell 38(2):295–307
8. Horé A, Ziou D (2010) Image quality metrics: Psnr vs. ssim. 20th Int Conf Pattern Recogn:2366–2369
9. Huang JB, Singh A, Ahuja N (2015) Single image super-resolution from transformed self-exemplars. In: Proceedings of IEEE CVPR, pp 5197–5206
10. Jia Z, Zhao J, Wang H, Xiong Z, Finn A (2015) A two-step face hallucination approach for video surveillance applications. Multimed Tools Appl 74(6):1845–1862
11. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of ECCV
12. Keys R (1981) Cubic convolution interpolation for digital image processing. IEEE Trans Acoust Speech Signal Process 29(6):1153–1160
13. Kim J, Lee JK, Lee KM (2016) Deeply-recursive convolutional network for image super-resolution. In: Proceedings of IEEE CVPR
14. Kupyn O, Budzan V, Mykhailych M, Mishkin D, Matas J (2018) DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: Proceedings of IEEE CVPR
15. Learned-Miller E, Huang GB, Chowdhury AHL, Hua G (2016) Labeled faces in the wild: A survey. In: Kawulok M, Celebi ME, Smolka B (eds) Advances in Face Detection and Facial Image Analysis. Springer, pp 189–248
16. Ledig C, Theis L, Huszar F, Caballero J, Aitken AP, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of IEEE CVPR
17. Liao S, Lei Z, Yi D, Li SZ (2014) A benchmark study of large-scale unconstrained face recognition. IEEE Int Joint Conf Biomet:1–8
18. Liu W, Wen Y, Yu Z, Li M, Raj B, Song L (2017) Sphereface: Deep hypersphere embedding for face recognition. CVPR
19. Liu Z, Luo P, Wang X, Tang X (December 2015) Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV)
20. Matsushita Y, Kawasaki H, Ono S, Ikeuchi K (2014) Simultaneous deblur and super-resolution technique for video sequence captured by hand-held video camera. In: Proceedings of IEEE International Conference on Image Processing (ICIP), pp 4562–4566
21. Ning X, Li W, Tang B, He H (2018) Buldp: Biomimetic uncorrelated locality discriminant projection for feature extraction in face recognition. IEEE Trans Image Process 27:2575–2586
22. Ning X, Nan X, Yu L, Zhang L (2020) Multi-view frontal face image generation: A survey. Concurrency Computat Pract Exper - Wiley Online Library, pp 1–18
23. Praveenbalaji D, Srinivas R, Roopa S, Suresh M, Gayathri A (2020) Id photo verification by face recognition. 6th International Conference on Advanced Computing & Communication Systems (ICACCS), pp 1449–1453
24. Redmon J, Farhadi A (2018) YOLOv3: An incremental improvement. arXiv:1804.02767v1[cs.CV]
25. Sajjadi MSM, Vemulapalli R, Brown M (2018) Frame-recurrent video super-resolution. CoRR arXiv:1801.04590
26. Schroff F, Kalenchenko D, Philbin J (2015) Facenet a unified embedding for face recognition and clustering. CVPR:815–823
27. Sensalire M, Ogao P, Telea A (2008) Classifying desirable features of software visualization tools for corrective maintenance. In: Proceedings of ACM SoftVis
28. Shi W, Caballero J, Huszar F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of IEEE CVPR

29. Sobiecki A, Neves LAP, Giraldi GA, Gatts GJF, Thomaz CE (2011) Segmentacao e restauracao digital para eliminacao de artefatos em imagens frontais de face. VII Workshop de Visão Computacional, Curitiba
30. Sobiecki A, Neves LAP, Thomaz CE (2010) To a better digitalization and visualization of frontal face photographs. IV European Conference on Computational Mechanics, Paris
31. Sun Y, Wang X, Tang X (2014) Deep learning face representation from predicting 10,000 classes. CVPR:1891–1898
32. Tao X, Gao H, Liao R, Wang J, Jia J (2017) Detail-revealing deep video super-resolution. In: Proceedings of IEEE ICCV
33. Turk M, Pentland A (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7.2:179–188
34. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3.1:71–86
35. van Vliet S, Sobiecki A, Telea A (2018) Joint brightness and tone stabilization of capsule endoscopy videos. In: Proceedings of VISAPP
36. Wang F, Xiang X, Cheng J, Yuille AL (2017) Normface: L 2 hypersphere embedding for face verification. Proceedings of the 25th ACM International Conference on Multimedia, pp 1041–1049
37. Wang L, Dong-Chen H (2006) Texture classification using texture spectrum. IEEE Trans PAMI:905–910
38. Wen Y, Zhang K, Li Z, Qiao Y (2016) A discriminative feature learning approach for deep face recognition. ECCV:499–515
39. Wu X, He R, Sun Z, Tan T (2018) A light cnn for deep face representation with noisy labels. IEEE Trans Inf Forensic Secur 13(11):2884–2896
40. Yang M-H, Kriegman DJ, Ahuja N (2002) Detecting faces in images: a survey. IEEE TPAMI 24(1):34–58
41. Yuzhikov V (2015) SmartDeblur deconvolution software. http://smartdeblur.net
42. Zhang C, Zhang ZY (2010) A survey of recent advances in face detection
43. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
44. Zhang K, Zuo W, Zhang L (2018) Learning a single convolutional super-resolution network for multiple degradations. In: Proceedings of IEEE CVPR