International Journal of Computer Vision (2019) 127:801–823
https://doi.org/10.1007/s11263-018-1138-7

Motion Deblurring of Faces

Grigorios G. Chrysos[1] · Paolo Favaro[2] · Stefanos Zafeiriou[1]

Received: 22 February 2018 / Accepted: 26 November 2018 / Published online: 17 December 2018

Abstract

Face analysis lies at the heart of computer vision with remarkable progress in the past decades. Face recognition and tracking are tackled by building invariance to fundamental modes of variation such as illumination, 3D pose. A much less standing mode of variation is motion deblurring, which however presents substantial challenges in face analysis. Recent approaches either make oversimplifying assumptions, e.g. in cases of joint optimization with other tasks, or fail to preserve the highly structured shape/identity information. We introduce a two-step architecture tailored to the challenges of motion deblurring: the first step restores the low frequencies; the second restores the high frequencies, while ensuring that the outputs span the natural images manifold. Both steps are implemented with a supervised data-driven method; to train those we devise a method for creating realistic motion blur by averaging a variable number of frames. The averaged images originate from the $2MF^2$ dataset with 19 million facial frames, which we introduce for the task. Considering deblurring as an intermediate step, we conduct a thorough experimentation on high-level face analysis tasks, i.e. landmark localization and face verification, on blurred images. The experimental evaluation demonstrates the superiority of our method.

Keywords Learning motion deblurring · Face deblurring · Data-driven networks

1 Introduction

Defocus and motion blur are the two most common types of blur.[1] Motion blur is caused by a change of the relative position of the camera-scene system during the sensor exposure, i.e. (i) camera shake and/or (ii) object movement. In most cases, the relative movement is unknown (blind deblurring); we know the blurry image and we want to recover the corresponding sharp one. Deblurring is significant for both high-end systems (e.g. human vision) and for computational tasks. In this work we focus on deblurring faces, as face analysis presents significant real-world applications. We introduce a method for deblurring facial images which suffer from (severe) motion blur.

Frequently, face analysis tasks involve (implicitly) learning a low-dimensional space, which is invariant to modes of variations influencing the performance of the task, e.g. rotation-invariant face recognition. Such invariance has been largely achieved for fundamental modes of variation, e.g. rotation, illumination (Tran et al. 2017). Recent works for approaching blur invariance have emerged, e.g. Ding and Tao (2018), however invariance in this mode is yet to be accomplished. This can be partly attributed to the underdetermined nature of the task; there are infinite combinations for sharp images, blur models and non-linear functions in the process of image formation. From the plausible combinations, we are interested only in the images that span the manifold of the natural images; deblurring methods capitalize on this restriction with various ways, e.g. priors or domain-specific knowledge.

[1] Defocus has significant applications, however it can be simulated with less complicated kernels than motion blur, e.g. in Ding and Tao (2018). Thus we focus exclusively on motion blur in this work.

Communicated by Rama Chellappa, Xiaoming Liu, Tae-Kyun Kim, Fernando De la Torre, Chen Change Loy.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11263-018-1138-7) contains supplementary material, which is available to authorized users.

✉ Grigorios G. Chrysos
g.chrysos@imperial.ac.uk

Paolo Favaro
paolo.favaro@inf.unibe.ch

Stefanos Zafeiriou
s.zafeiriou@imperial.ac.uk

[1] Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK

[2] Department of Informatics, University of Bern, Neubruckstrasse 10, 3012 Bern, Switzerland

Deblurring methods can be divided into two classes: (i) generic object deblurring, (ii) domain-specific methods. Even though the task of generic object deblurring is relatively well-studied (Tekalp et al. 1986; Cho and Don 1991; Levin et al. 2009; Nah et al. 2017), generic methods yield suboptimal results on faces. The reason is that generic methods typically rely on gradient-based information which fail in an object with many flat regions, e.g. human face. Moreover, the face includes highly structured shape, which is not utilized by the priors of generic object deblurring. Therefore, domain-specific methods could offer a better alternative for deblurring faces.

The domain-specific methods in face deblurring can be classified in two categories: (i) joint optimization methods (Liao et al. 2016; Nguyen et al. 2015; Ding and Tao 2018), (ii) geometric-based methods, e.g. Pan et al. (2014a). The former methods optimize over a face analysis task, e.g. recognition, and either include deblurring related priors in their cost function or add blurry examples in their training set, i.e. implicitly learn blur-invariant representations. However, joint optimization methods use oversimplifying assumptions for blur to enable (easier) convergence. The geometric-based methods utilize the contour or a sparse shape of the face to guide their optimization. Success has been demonstrated under synthetic/mild blurs, however extracting geometric cue information for real-world blurry images is not trivial.

In contrast to the aforementioned domain-specific methods, our work aims to constrain the solutions by projecting the outputs to the natural images' manifold. We introduce a two-step architecture; the first step consists of a strong discriminative network that restores the low-frequencies, while the second step restores the image and encourages the constraint to the natural images' manifold. The latter is achieved by a novel network, based on the conditional GAN (cGAN) of Mirza and Osindero (2014). Specifically, we replace the generator's pathway with two pathways; the first pathway accepts the blurry image, while the second extracts facial representations (similar to an auto-encoder). By sharing the weights in part of the two pathways, the first pathway is encouraged to have representations similar to those of the second. Both steps consist of data-driven supervised networks, which require pairs of blurry and ground-truth (sharp) images for training (Fig. 1).

Collecting pairs of blurry/sharp images for training is a laborious and expensive process (it requires specialized hardware). The efforts of Su et al. (2017), Nah et al. (2017), Kim et al. (2018), Noroozi et al. (2017) are noteworthy, however they include spatial and temporal limitations, i.e. capturing covers only a specific time-span and is restricted geographically. On the other hand, generating such pairs computationally can be achieved by (i) convolving sharp images with synthetic blur kernels, (ii) simulating motion blur by averaging sharp frames; in such case vast amount of data available online can be the source. Unfortunately, synthetic blurs (Hradiš et al. 2015) cannot capture natural facial deformations, while their generalization to real-world blurry images remains questionable (Lai et al. 2016). Contrarily, the main requirement for simulating motion blur is to have a video with a high fps.[2] The motion blur is achieved by averaging sequential frames. We introduce such an averaging scheme, where we consider the averaging as a multivariate function that depends on the number of frames averaged, the overlap of fiducial points, the optical flow and the image quality.

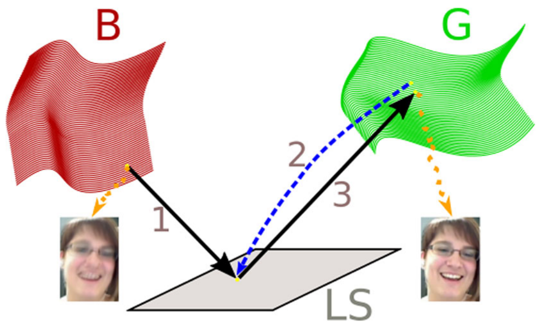


Fig. 1 The recent approaches in deblurring, e.g. Kupyn et al. (2018), project an image from the manifold of blurry images $\boldsymbol{B}$ to a latent space $\boldsymbol{LS}$ (step 1) and then to the manifold of sharp images $\boldsymbol{G}$ (step 3). In our approach we insert step 2, i.e. during training time we project from $\boldsymbol{G}$ to the latent space. A pair of blurry/sharp images, sampled from the $\boldsymbol{B}$ and $\boldsymbol{G}$ manifolds, are depicted (dashed orange line). All figures in this paper are best viewed in color

A direct drawback of the averaging scheme is the requirement for vast amount of frames. The lack of such facial data is partially the reason why face deblurring is understudied. To that end, we introduce $2MF^2$, a dataset with 11,590 videos which accumulate to 19 million frames. We use $2MF^2$ videos to generate blurry images and then train our system. $2MF^2$ includes the largest number of long videos of faces (each video has a minimum of 500 frames), while it includes multiple videos of the same identity in different scenes and conditions.

Following recent trends in deblurring (Hradiš et al. 2015; Kupyn et al. 2018), we further assess our method by considering deblurring as an intermediate task. That is, apart from the typical image quality metrics, we perform a thorough experimentation by utilizing the deblurred outcomes for two different tasks. We perform landmark localization, and face verification on two different datasets. Both tasks have solid quantitative figures of merit and can be used to evaluate the final result, hence verify implicitly whether the deblurred images indeed resemble the samples from the distribution of sharp images.

This work is an extension of Chrysos and Zafeiriou (2017b), where we introduced the first network used for

[2] Most commercial cameras capture at least 30 fps, which is reasonably fast.

Table 1 Summary of primary symbols

Symbol	Dimen.	Definition
$\boldsymbol{I}_s$	$\mathbb{R}^{hxw}$	Latent sharp image
$\boldsymbol{I}_{bl}$	$\mathbb{R}^{hxw}$	Blurry image
L	$\mathbb{R}$	# frames averaged (motion blur)
$\lambda_{\{ci,cg,p,r\}}$	$\mathbb{R}$	Regularization hyper-parameters

'#' abbreviates the 'number of'

deblurring facial images. The current work has substantial extensions. First of all, in the original work there was no explicit effort to constrain the output to the natural images' manifold. In addition, the sparse shape utilized for the original work did not work well under severe blur; we instead allow the network to extract the meaningful representations in a data-driven way. The architecture has been redesigned from scratch; the ResNet of He et al. (2016) in the previous version is much different than the current architecture. Last but not least, the experimental section has been completely redesigned; in this work in addition to the standard quality metric for deblurring we utilize the deblurred images as an intermediate step for experimenting with higher level tasks in face analysis.

Our contributions can be summarized as:

- The first learning-based architecture for motion deblurring facial images is introduced. To train that network, a new way to simulate motion blur from videos is proposed.
- We introduce $2MF^2$ dataset that includes over 19 million frames; the frames are utilized for simulating motion blur.
- We conduct a thorough experimental evaluation (i) with image quality metrics and (ii) by utilizing the deblurred images in other tasks. The deblurred images are compared in sparse regression and classification in face analysis tasks. Our comparisons involve deblurring over 60,000 images for each method, which consists one of the largest testsets used for comparing deblurring methods.

We consider our proposed method as a valuable addition to the research community, hence the blurry/sharp pairs along with the frames of $2MF^2$ will be released upon the acceptance of the paper.[3]

Notation A small (capital) bold letter represents a vector (matrix); a plain letter designates a scalar number. Table 1 describes the primary symbols used in the manuscript.

2 Related work

We initially provide an overview of the recent advances with Generative Adversarial Networks (core component of our architecture), then recap the literature on deblurring and sequentially study how blur is studied in face analysis tasks.

2.1 Generative Adversarial Network

Generative Adversarial Networks (GANs) by Goodfellow et al. (2014) have received wide attention. GANs sample noise from a predefined distribution (e.g. Gaussian) and learn a mapping with a signal from the domain of interest. Several extensions have emerged, like using convolutional layers instead of fully connected in Radford et al. (2015), feeding a (Laplacian) pyramid for coarse-to-fine generation in Denton et al. (2015), jointly training a GAN with an inference network in Dumoulin et al. (2016), learning hierarchical representations in Huang et al. (2017). Alternative cost functions and divergence metrics have been proposed (Nowozin et al. 2016; Arjovsky et al. 2017; Mao et al. 2017). In addition, several approaches for improving the training of GAN's have appeared (Salimans et al. 2016; Dosovitskiy and Brox 2016). GANs have been used for unsupervised (Radford et al. 2015; Arjovsky et al. 2017), semi-supervised (Odena 2016) and supervised learning (Mirza and Osindero 2014; Ledig et al. 2017; Isola et al. 2017; Tulyakov et al. 2017). The proliferation of the works with GANs can be attributed to their ability to preserve high texture details and model highly complex distributions.

2.2 Deblurring

Deblurring defines the computational task of reversing the unknown blur that has been inflicted to a sharp image $\boldsymbol{I}_s$. In the previous few decades, the problem was formulated as an energy minimization with heuristically defined priors, which reflect image-based statistics or domain-specific knowledge. However, aside of the computational cost of these optimization methods (typically they require over a minute for deblurring an image of 300 × 300 resolution), their prior consists their Achilles heel. Deep learning methods alleviate that by learning from data.

Energy Optimization Methods The blurry image $\boldsymbol{I}_{bl}$ is assumed as the convolution of the latent sharp image $\boldsymbol{I}_s$ and a (uniform) kernel $\boldsymbol{K}$, mathematically expressed as $\boldsymbol{I}_{bl} = \boldsymbol{I}_s * \boldsymbol{K} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ denotes the noise. Deblurring is then formulated as minimization of the cost function

$$\boldsymbol{I}_s = \arg\min_{\tilde{\boldsymbol{I}}_s} \left(||\boldsymbol{I}_{bl} - \tilde{\boldsymbol{I}}_s * \boldsymbol{K}||_2^2 + f(\boldsymbol{I}_{bl}, \boldsymbol{K}) \right). \tag{1}$$

with $f(\boldsymbol{I}_{bl}, \boldsymbol{K})$ a set of priors based on generic image statistics or domain-specific priors. These methods are applied in a coarse-to-fine manner; they estimate the kernel and then perform non-blind deconvolution.

[3] For all matters concerning the data of this paper, please send your mails/requests to 2mf2.db@gmail.com.

The blur kernel $\boldsymbol{K}$ and the latent image $\boldsymbol{I}_s$ are estimated in an alternating manner, which might lead to a blurry result if a joint MAP optimization is performed (Levin et al. 2009). They suggest instead to solve a MAP on the kernel with a gradient-based prior that reflects natural image statistics. Pan et al. (2014b) apply an ℓ_0 norm as a sparse prior on both the intensity values and the image gradient for deblurring text. Hacohen et al. (2013) support that the gradient-based prior alone is not sufficient. They introduce instead a prior that locates dense correspondences between the blurry image and a similar sharp image, while they iteratively refine the correspondence estimation, the kernel and the sharp image estimation. Their core idea relies on the existence of a similar reference image, which is not always available. A generalization of Hacohen et al. (2013) is the work of Pan et al. (2014a), which relaxes the existence of a similar image with an exemplar dataset. The assumption is that there is an image with a similar contour in the exemplar dataset. However, the contour of an unconstrained object or the similarities between contours are not trivially found, hence Pan et al. (2014a) restrict the task to face deblurring to profit from the constrained shape structure. At test time, a search in the dataset with the exemplar images is performed; the exemplar image with a contour similar to the test image is then used to initialize the blind estimation iterations. Pan et al. (2014a) demonstrate how this leads to an improved performance. Unfortunately, the noisy contour matching process along with the obligatory presence of a similar contour in the dataset limit the applications of this work. Huang et al. (2015) recognize the deficiencies of this approach and propose to perform landmark localization before extracting the contour. They effectively replace the exemplar dataset matching by training a localization technique with blurry images, however their approach still suffers in more complex cases (they use a few synthetic kernels) or even more in severe blur cases.

In contrast to the gradient-based priors, Pan et al. (2016) introduce a prior based on the sparsity of the dark channel. The dark channel is defined as the pixel with the lowest intensity in a spatial neighborhood. Pan et al. (2016) prove that the intensity of the dark channel is increased from the blurring process; they demonstrate how the sparsity of the dark channel leads to improved results.

Even though the aforementioned methods provably minimize the energy of a blurred image, their strong assumptions (e.g. non-informative, hard-coded priors) consist them both computationally inefficient[4] and with poor generalization to real-world blurry images (Lai et al. 2016).

Learning-Based Methods for Motion Blur Removal The experimental superiority of neural networks as function approximators have fuelled the proliferation of deblurring methods learned from data. The blurring process includes several non-linearities, e.g. the camera response function, lens saturation, depth variation, which the aforementioned optimization methods cannot handle. Conversely, neural networks can approximate these non-linearities and learn how to reverse the blur by feeding them pairs of sharp and blurry images.

There are two dominant approaches: (i) use a data-driven method to learn an estimate; then refine the kernel/image estimation with classic methods (Sun et al. 2015; Chakrabarti 2016), (ii) let the network explicitly model the whole process and obtain the deblurred result (Hradiš et al. 2015).

In the former approach, Schuler et al. (2016) design a network that imitates the optimization-based methods, i.e. it iteratively extracts features, estimates the kernel and the sharp image. Sun et al. (2015) learn a convolutional neural network (CNN) to recognize few predefined motion kernels and then perform a non-blind deconvolution. Chakrabarti (2016) proposes a patch-based network that estimates the frequency information for uniform motion blur removal. Gong et al. (2017) train a network to estimate the motion flow (hence the per-pixel blur) and then perform non-blind deconvolution.

The second approach, i.e. modelling the whole process with a network, is increasingly used due to the increased capacity of the networks. Noroozi et al. (2017); Nah et al. (2017) introduce multi-scale CNNs and learn an end-to-end approach where they feed the blurry image and obtain the deblurred outcome. Nah et al. (2017) also include an adversarial loss in their loss function. A number of very recent works utilize adversarial learning to learn an end-to-end mapping between blurry and sharp images (Ramakrishnan et al. 2017; Kupyn et al. 2018).

The works utilizing adversarial learning are the closest to our methods, however there a number of significant differences in our case. First of all, we constrain our outputs to span the natural images manifold; we also approach the task as a two step process where in the first step we restore the low frequency components and then refine the high frequency details by adversarial learning.

2.3 Face Analysis Under Blur

As face analysis consists a core application of computer vision, the need for studying the blurring process is increasingly emphasized, e.g. in Zafeiriou et al. (2017). The face detector of Liao et al. (2016) introduces the NPD features and experimentally verify how robust they are under different blur conditions. Estimating the age from a blurry face is the task of Nguyen et al. (2015), who introduce an opti-

[4] The coarse-to-fine procedure is executed in a loop hundreds or even thousands of times to return a deblurred image; as indicated by Chakrabarti (2016) some of them might even require hours for deblurring a single image.

mization method that estimates the motion blur; classify the blurry image based on the blur and deblur accordingly based on the category.

Facial blur poses a major challenge in face recognition, hence few efforts explicitly include blurry images in the learning process. Nishiyama et al. (2011) construct a identity-invariant feature space, which has the property that images degraded by similar blur are clustered together. Then, cluster-based deblurring is performed. Even though the idea works well for synthetic images, the clusters of real-world blurs are less distinct. Gopalan et al. (2012) derive a blur-robust descriptor for face recognition, based on simplifying assumptions about (i) the convolution with a specific size kernel, (ii) no-noise case. Ding and Tao (2018) learn blur-invariant representations by feeding simultaneously a blurry and a sharp image to the network.

Face hallucination, i.e. generating a high resolution image from low-resolution input, includes similar approaches to face deblurring. Recent approaches in hallucination include learning-based approaches similar to deblurring. Zhu et al. (2016) explore a cascade method that performs jointly hallucination and dense correspondence estimation. A different approach is considered by Cao et al. (2017), who utilize reinforcement learning to sequentially discover patches to enhance. Lee et al. (2018) recently explored using attributes to assist face hallucination. Their method is a conditional GAN (Mirza and Osindero 2014) with two conditioning labels: the input (low-resolution) image along with the attributes. In contrast to hallucination, deblurring methods need to restore the high-frequency details instead of synthesizing a plausible match, which makes hallucination more approachable. In addition, in hallucination the data are readily available as a single image can be downscaled to obtain the input image, while in deblurring different blurring methods still emerge.

3 Method

In this section we introduce the network that we employ for the task; we additionally outline the process of generating realistic training pairs for our network.

Blurring is a challenging process to be reversed by few convolutional layers; we confirmed that with our prior work in Chrysos and Zafeiriou (2017b). To that end, we create a two step process; the first step includes the strong-performing hourglass network (HG) by Newell et al. (2016) to restore the low and mid-frequencies. The second step includes a variant of conditional GAN (cGAN), which restores the high-frequency details. The hourglass network is briefly described in Sect. 3.1; the conditional GAN in Sect. 3.2, while the final network in Sect. 3.3.

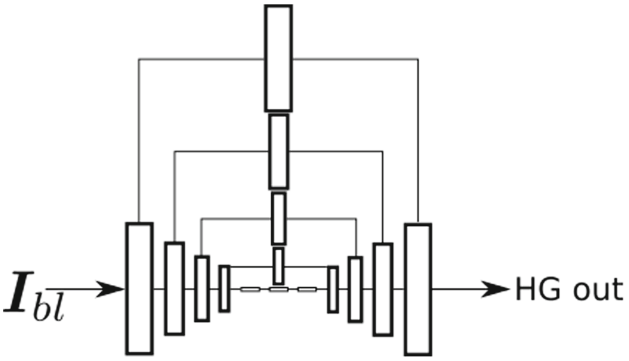


Fig. 2 Schematic of the hourglass network (Sect. 3.1)

3.1 Hourglass Network

The hourglass network (HG) is a strong-performing deep convolutional network; HG has a top-down and bottom-up approach and combines low-level features from different resolutions. The architecture consists of stacked resnet layers which form resnet blocks (He et al. 2016). A resnet layer has a convolutional layer, a batch normalization and a non-linear activation unit, while a resnet block consists of three resnet layers. The stacked resnet blocks are structured in the form of encoder-decoder network; after each resnet block there is a lateral connection from the decoder to the encoder. Each lateral connection includes a resnet block for filtering the signal. The schematic for the HG is depicted in Fig. 2.

The hourglass network has been primarily used for tackling geometric-based tasks, e.g. pose estimation in Newell et al. (2016) or estimating the facial fiducial points in Bulat and Tzimiropoulos (2017). HG has been also used for pixel-wise predictions in depth estimation Chen et al. (2016). Similarly to the latter work, our deblurring HG makes pixel-level prediction.

We utilize the vanilla HG trained with an ℓ_1 loss:

$$\mathcal{L}_{HG} = ||\boldsymbol{I}_s - H(\boldsymbol{I}_{bl})||_{\ell_1} \qquad (2)$$

where $H(\boldsymbol{I}_{bl})$ denotes the output of the HG.

3.2 Conditional GAN

GAN consists of a generator G and a discriminator D network commonly optimized with alternating gradient descent methods. The generator tries to model the true distribution of the data p_d; specifically it samples from a low-dimensional distribution (noise) and outputs samples in the target space. The discriminator tries to distinguish between real (sampled from the true distribution) and fake signals (sampled from the model's distribution). Conditional GAN (cGAN) by Mirza and Osindero (2014) extends the formulation by condition-

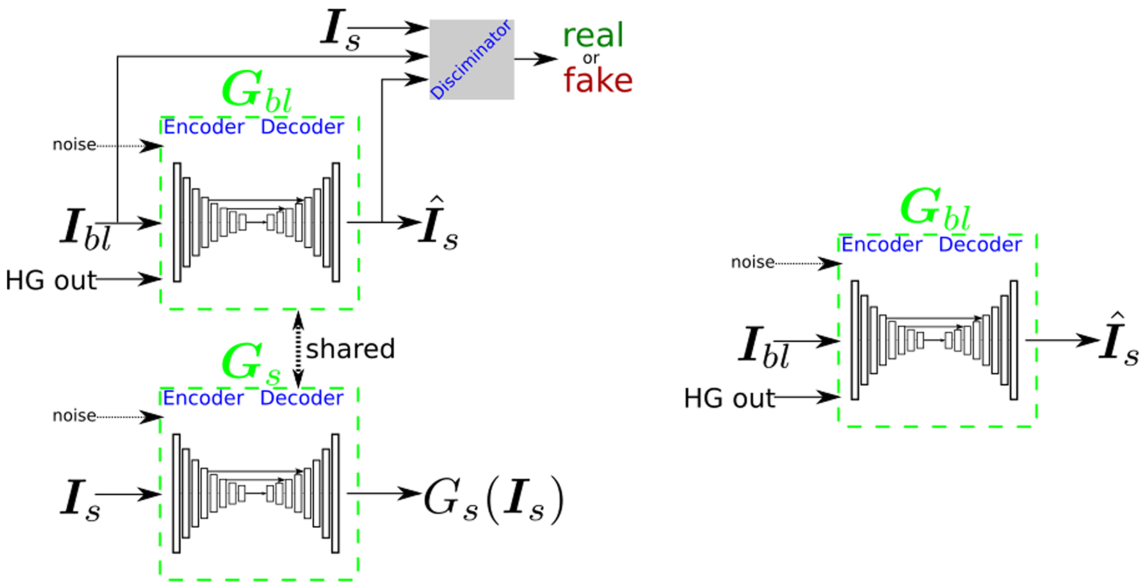


Fig. 3 Schematic of our cGAN (Sect. 3.2). The training time cGAN is depicted in the left. 'HG out' signifies the additional label used as input, i.e. the output of the HG network, while the weight sharing of the two decoders is denoted with the dashed line. During prediction (testing time), the network is greatly simplified; it is depicted in the right part of the figure

ing the distributions with additional labels. If p_z denotes the distribution of the noise, $\boldsymbol{s}$ the conditioning label and $\boldsymbol{y}$ the data, the objective function is expressed as:

$$\begin{aligned}&\mathcal{L}_{cGAN}(G, D)\\&\quad= \mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}[\log D(\boldsymbol{s}, \boldsymbol{y})]\\&\qquad+ \mathbb{E}_{\boldsymbol{s}\sim p_d(\boldsymbol{s}),\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(\boldsymbol{s}, G(\boldsymbol{s}, \boldsymbol{z})))]\end{aligned} \tag{3}$$

This objective function is optimized in an iterative manner, as

$$\begin{aligned}&\min_{\boldsymbol{w}_G}\max_{\boldsymbol{w}_D} \mathcal{L}_{cGAN}(G, D)\\&\quad= \mathbb{E}_{\boldsymbol{s},\boldsymbol{y}\sim p_d(\boldsymbol{s},\boldsymbol{y})}[\log D(\boldsymbol{s}, \boldsymbol{y}; \boldsymbol{w}_D)]\\&\qquad+ \mathbb{E}_{\boldsymbol{s}\sim p_d(\boldsymbol{s}),\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(\boldsymbol{s}, G(\boldsymbol{s}, \boldsymbol{z}; \boldsymbol{w}_G); \boldsymbol{w}_D))]\end{aligned}$$

where $\boldsymbol{w}_G$, $\boldsymbol{w}_D$ denote the generator's and the discriminator's parameters respectively.

In our case the target space (deblurred outcomes) are dependent on the input space (blurred input); we utilize a cGAN that is precisely defined for such a task. Similarly to our case, conditional GANs have been applied to diverse image processing tasks, since they output photo-realistic images. Recent applications include photo-realistic image synthesis by Ledig et al. (2017), style transfer by Yoo et al. (2016), inpainting by Pathak et al. (2016), image-to-image mappings by Isola et al. (2017), video generation by decoupling content from motion in Tulyakov et al. (2017), image hallucination in Xu et al. (2017).

3.3 Model Architecture

In order to tackle a challenging task like deblurring we introduce an architecture with two stacked networks; the first one is an hourglass as described in Sect. 3.1, while the second network is a novel structure based on cGAN.

The second network in our architecture is a novel type of conditional GAN. The original cGAN includes a generator module that accepts the input label/image and produces the output with a single pathway encoder-decoder network. However, since the generator is data-driven, there is no restriction in the target space, which might lead to regressing to values far from the natural images' manifold. Our motivation lies in implicitly restricting the output to span the desired manifold. We achieve that goal by augmenting the generator's pathway by an additional pathway. The original pathway, denoted as G_{bl}, remains the same; the new pathway, denoted as G_s, works as an auto-encoder in the target space. The two pathways share the same architecture layer-wise, while we share the weights of the two decoders. The shared weights encourage the latent representations of the sharp and blurry images to be similar. A schematic of our cGAN is visualized in Fig. 3.

Our cGAN inherits all the losses of the original cGAN, while we can add additional terms to further encourage the deblurred outputs to span the desired manifold. We design a loss function with four terms. Aside of the adversarial loss $\mathcal{L}_{cGAN}(G_{bl}, D)$, which is computed based on the G_{bl} generator's output, we add a content loss, a projection loss and a reconstruction loss.

The content loss consists of two terms that compute the per-pixel difference between the generator's output ($G_{bl}(\boldsymbol{I}_{bl})$) and the sharp (ground-truth) image. The two terms are (i) the ℓ_1 loss between the ground-truth image and the output of the generator, (ii) the ℓ_1 of their gradients; mathematically expressed as:

$$\mathcal{L}_c = \lambda_{ci}||G_{bl}(\boldsymbol{I}_{bl}) - \boldsymbol{I}_s||_{\ell_1} + \lambda_{cg}||\nabla G_{bl}(\boldsymbol{I}_{bl}) - \nabla \boldsymbol{I}_s||_{\ell_1} \quad (4)$$

where $\lambda_{ci}, \lambda_{cg}$ are two hyper-parameters.

The projection loss[5] enables the network to match the data and the model's distribution faster. The intuition is that to match the high-dimensional distribution of the data with the model one, we can encourage their projections in lower-dimensional spaces to be similar. To avoid adding extra parameters or designing a hard-coded projection, we utilize the projection of the discriminator. If π denotes the projected features from the penultimate layer of the discriminator, then:

$$\mathcal{L}_p = ||\pi(G_{bl}(\boldsymbol{I}_{bl})) - \pi(\boldsymbol{I}_s)||_{\ell_1} \quad (5)$$

Last but not least, in the G_s pathway, we include a reconstruction loss (typicall for auto-encoders). We penalize any dissimilarities between the reconstructed image from the target image, i.e.

$$\mathcal{L}_r = ||G_s(\boldsymbol{I}_s) - \boldsymbol{I}_s||_{\ell_1} \quad (6)$$

The total loss function of our cGAN is expressed as:

$$\mathcal{L} = \mathcal{L}_{cGAN} + \mathcal{L}_c + \lambda_p \mathcal{L}_p + \lambda_r \mathcal{L}_r \quad (7)$$

where λ_p, λ_r are hyper-parameters.

Our cGAN is the second step in our architecture; the input is the output of the HG. We experimentally verified that conditioning on the original blurry image improved the outcomes. We hypothesize that the output of the HG has lost the high-frequency details, in contrast to the original blurry image which contains the frequencies (they are scrambled though), hence the cGAN benefits from having this as input. The implementation details are analyzed in Sect. 5.

During the prediction (testing) step, the architecture is simplified, which makes it appropriate for real-world applications. The schematic of the prediction architecture is visualized in Fig. 4.

3.4 Training Data

Pairs of sharp images along with their corresponding motion blurred images are required to learn the model. To understand how to create such pairs, we examine briefly how the motion blur is generated. To capture an image, the aperture is opened, accumulates light from the dynamic scene and then creates the integrated result. The process can be formulated as an integration of all the light accumulated, or for the discrete machines the sum of all values as following:

$$\boldsymbol{I}_{bl} = \psi\left(\int_0^T \boldsymbol{I}_s(t)dt\right) \approx \frac{1}{K+1}\psi\left(\sum_{k=0}^{K} \boldsymbol{I}_s[k]\right) \quad (8)$$

where $\boldsymbol{I}_{bl}$ denotes the blurry image, $\boldsymbol{I}_s$ the latent sharp image, T, K the duration in continuous/discrete time that the aperture remains open. The function ψ expresses the unknown non-linear relationships that the imaging process includes, for instance lens saturation, sensor sensitivity. We cannot analytically compute the function; we can only approximate it, but this remains challenging as studied by Grossberg and Nayar (2003), Tai et al. (2013).

The aforementioned blurring process can be approximated computationally. The three dominant approaches for creating pairs of sharp/blurry images are the following: (a) use specialized equipment/hardware setup, (b) simulate motion blur by averaging frames, (c) convolve image with synthetic kernels. Synthetic motion blur was created by kernels in the past, please find a thorough review in Lai et al. (2016). Such synthetic blurs assume a static scene and are only applied in the 2D image plane, which consists them very simplistic. In addition, deblurring synthetically blurred images does not have a high correlation with deblurring on unconstrained conditions (Lai et al. 2016).

Using specialized equipment has appeared in the recent works of Su et al. (2017), Nah et al. (2017), Kim et al. (2018); they utilized GoPro Hero cameras to capture videos at 240 fps. The frames are post-processed to remove the blurry ones; sequential frames are then averaged to simulate the motion blur. The main benefit of such high-end devices is that they can reduce the amount of motion blur per frame; virtually ensuring that there will be minimal relative motion between the 3D scene and the camera at each capture. However, the spatial and temporal constraints of such a collection consist a major limitation. Even though a significant effort to capture diverse scenes might be required, the collection spans small variety of scenes with a constrained time duration (limited number of samples). Additionally, only specific high-fps capturing devices can be used and even those under restricted conditions, e.g. good lighting conditions.

On the other hand, the simulation entails averaging sequential frames of videos captured from commodity cameras (30–60 fps). Such videos are abundant in online platforms, for instance in YouTube hundreds of hours of new content are uploaded per minute. Content covers both short amateur clips and professional videos, while the scenes vary

[5] The projection loss is also referred to as feature matching loss in Salimans et al. (2016), Xu et al. (2017); we find the feature matching a very broad term.

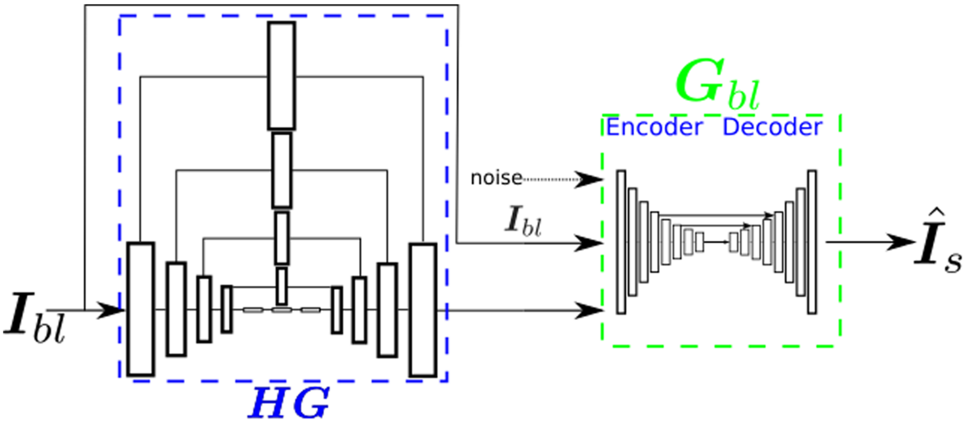


Fig. 4 Schematic of the architecture during prediction (testing). The cGAN is significantly simplified in this step

from outdoor clips in extreme capturing conditions to controlled office conditions. Previously to our work, Wieschollek et al. (2017) utilize such sources to simulate motion blur. For each pair of sequential frames the authors utilize a bidirectional optical flow to generate a number of subframes, which are then averaged to generate the blurry–sharp pair.

In our case, we utilize videos captured with commodity cameras and are available in the internet. We do not resort to any frame warping as this might lead to artifacts not present in the real-world motion blur cases. In lieu, we average frames from the training videos, more formally the precise blur is computed as:

$$\hat{\boldsymbol{I}}_{bl} = \frac{1}{L} \sum_{l=-\frac{L-1}{2}}^{\frac{L-1}{2}} \hat{\psi}(\boldsymbol{I}_s[l]) \tag{9}$$

where L denotes the number of frames in the moving average, $\hat{\psi}$ a function that is learned and approximates ψ.

The number of frames summed, i.e. L, varies and depends on the cumulative relative displacement of the face/camera positions. The number L is dynamically decided based on the current frames; effectively we generate a sub-sequence of L frames, we average the intensities of those to obtain the blurry image and consider the middle as the ground-truth image; the process is illustrated in Fig. 5. We continue adding frames to the sub-sequence until stop conditions are met. The conditions that affect the choice of L are the following: (i) there should be an overlap of some of the facial semantic parts between frames, (ii) the quality of the running average versus the current middle image of the sequence, (iii) motion flow. The first condition requires the first and the last frame of the sequence to have at least partial overlap, the second demands the blurry frame to be related to the sharp frame content-wise. The last condition avoids oscillation (or other failure) cases. We have experimentally found that such a moving average is superior to the constant sum blurring.

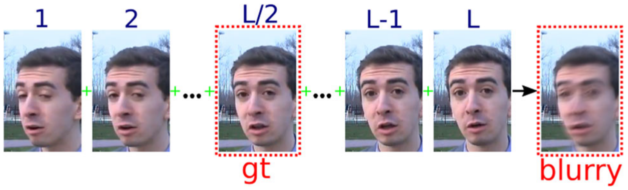


Fig. 5 Averaging scheme to simulate the motion blur. Averaging L frames generates a realistic motion blur and we can consider the middle frame $(L/2)$ as the gt

4 $2MF^2$ Dataset

As it was empirically proved the last few years, the scale of data has a tremendous effect in the performance of systems based on deep convolutional neural networks. Meanwhile, there is still a lack of a large scale database of real-world video data with faces (Ding and Tao 2018).

Various databases have been previously published, however all of them have restrictions that jeopardize the pair generation we require. Youtube Faces (YTF) database Wolf et al. (2011) includes 3425 videos, several of which are of low resolution and low length (very few frames) and with restricted movement. UMD Faces of Bansal et al. (2017) and IJC-B of Whitelam et al. (2017) include several thousands of videos, however they do not include sequential frames, which does not enable us to use them for averaging sequential frames. Chung and Zisserman (2016) introduce a dataset for lip-reading, however apart from the constrained conditions of the videos (BBC studios), all the clips include the same duration and a single word is mentioned per clip. Shen et al. (2015) introduce 300VW for landmark tracking, however this include very few videos (identities) for our case.

The requirement for high resolution and high fps videos led us to create $2MF^2$ dataset. The dataset was created by downloading public videos from YouTube. Some sample images can be viewed in Fig. 6; a video with the first frames from every video is depicted in https://youtu.be/iQ7-80eg3u4, while an accompanying video depicting some

Fig. 6 Visualization of $2MF^2$ dataset samples

Table 2 A comparison of $2MF^2$ to other benchmarks with facial videos in unconstrained conditions

Dataset name	# References	# of videos	# of identities	# of frames	Seq. frames	Identity info
YTF	Wolf et al. (2011)	3425	1595	62,095	✓	✓
300VW	Shen et al. (2015)	114	∼ 100	218,595	✓	x
IJC-B	Whitelam et al. (2017)	7011	1845	55,026	x	✓
UMD-Faces V.	Bansal et al. (2017)	22,075	3107	3,735,476	x	✓
BBC Lip-reading	Chung and Zisserman (2016)	538,496	∼ 1000	15,616,384	✓	x
$2MF^2$	(This work)	11,590	850	19,202,440	✓	✓

The column ‘Seq. frames’ denotes whether those videos consist of sequential frames or randomly sampled ones from a video; the column ‘Identity info’ refers to the meta-data available for identifying different identities

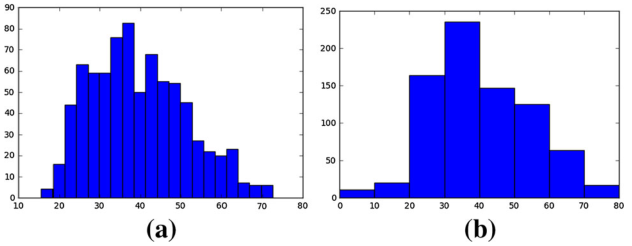


Fig. 7 Histograms corresponding to the age estimation of the identities in $2MF^2$. **a** Estimation by DEX (each bin corresponds to ∼ 4 years), **b** estimation by four annotators (each bin corresponds to a decade). Note the similarities of the two histograms; they both indicate that the majority of the identities belong in the age group of 30−40 years old, while they both demonstrate that there are several people from all age groups. **a** System estimation **b** Human estimation

videos along with the popular 68 shape mark-up can be found in https://youtu.be/Mz0918XdDew. Since the original version in Chrysos and Zafeiriou (2017b), we have significantly extended the database, both in number of videos and in number of identities. We have made an effort to create a database with large variations, e.g. to include people with different ages, ethnic groups. The database includes 19 million frames from 11,590 videos of 850 different identities. The different identities were manually annotated by two different people. On average each identity appears in 11,760 frames. Each person appears in multiple videos, which enables us to capture a wide variation of expressions and external conditions, e.g. illumination, background (Table 2).

We have collected annotations for the age of each unique identity by (i) human annotators, (ii) utilizing the widely-used DEX of Rothe et al. (2018). Four human annotators were asked to estimate the age of each unique identity based on the first frame of the video; eight non-overlapping options were available, i.e. 1−10, 11−20, etc. The first frame of the video was additionally fed into the popular DEX that estimates the real age of the identity. The automatically derived ages were separated into 20 bins (each bin approximately corresponds to 4 years). The resulting plots for both cases are visualized in Fig. 7. The histograms demonstrate that $2MF^2$ includes several samples from each age group.[6]

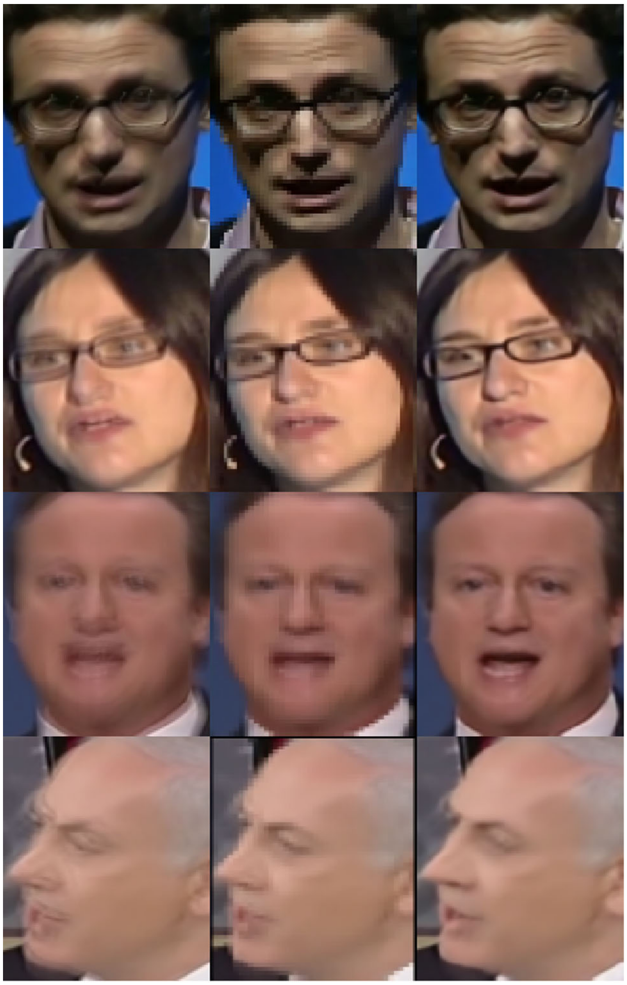
5 Experiments

We systematically scrutinized the performance of our proposed method, both internally evaluating the proposed adaptations and against the majority of the publicly available implementations for deblurring. Not only we utilized SSIM as standard metric for image quality, but we also considered deburring as a proxy task and evauated the performance on higher-level tasks in face analysis. Initially, we introduce the implementation and experimental setup and sequentially summarize the experimental results.

5.1 Implementation Details

Architecture Details An off-the-shelf implementation of the HG network as described in Newell et al. (2016) is used. The generators G_{bl} and G_s share the same architecture, apart from the first layer that G_{bl} accepts a 6-channel input image, i.e. the concatenation of the blurry image and the output of

[6] Few samples exist from the children less than 10 and the people older than 70 years old; that is because the videos including such groups are scarce.



(a) blurred **(b)** HG **(c)** final-ours

Fig. 8 Outputs from our two-step architecture. In each row: **a** blurred image, **b** output of the HG, **c** final output of our architecture. The blurry parts are significantly reduced in **b**, however the high-frequency details are only restored in **c**

Table 3 Details of the conditional Generative Adversarial Network employed

(a) Encoder

Layer	Filter size	Stride	BN
Conv. 1	$4 \times 4 \times 64$	1	×
Conv. 2	$4 \times 4 \times 128$	2	✓
Conv. 3	$4 \times 4 \times 256$	2	✓
Conv. 4	$4 \times 4 \times 512$	2	✓
Conv. 5	$4 \times 4 \times 512$	2	✓
Conv. 6	$4 \times 4 \times 512$	2	✓
Conv. 7	$4 \times 4 \times 512$	2	✓
Conv. 8	$1 \times 1 \times 256$	2	✓

(b) Decoder

Layer	Filter size	Stride	BN	Dropout
F-Conv. 1	$1 \times 1 \times 1024$	2	✓	0.5
F-Conv. 2	$4 \times 4 \times 512$	2	✓	0.5
F-Conv. 3	$4 \times 4 \times 512$	2	✓	0.5
F-Conv. 4	$4 \times 4 \times 512$	2	✓	1.0
F-Conv. 5	$4 \times 4 \times 256$	2	✓	1.0
F-Conv. 6	$4 \times 4 \times 128$	2	✓	1.0
F-Conv. 7	$4 \times 4 \times 64$	2	✓	1.0
F-Conv. 8	$4 \times 4 \times \{1, 3\}$	1	×	1.0

'Filter size' denotes the size of the filters for the convolutions; the last number denotes the number of output filters. BN stands for batch normalization. Conv denotes a convolutional layer. F-Conv denotes a transposed convolutional layer with fractional-stride. In the conditional GAN, dropout denotes the probability that this layer is kept

the HG network. Each generator includes an encoder and a decoder along with skip connections. Both the encoder and the decoder are composed by 8 layers; each convolutional layer is followed by a RELU and batch normalization (Ioffe and Szegedy 2015). The discriminator consists of 5 layers, while the input images in all sub-networks have 256×256 shape. Visual results of the architecture are depicted in Fig. 8.

A number of skip connections are added in the generators. Those consist of residual connections (He et al. 2016) and U-net style connections (Ronneberger et al. 2015). The residual connections are added after the first, the third and the fifth layer of the encoder and skip two layers each; the intuition is to propagate the lower level features explicitly without filtering. In the decoder, two similar residual connections are added after the first and the third layers of the decoder. Two U-net style skip connections from the encoder to the decoder are added. Those skip connections enforce the network to learn the residual between the features corresponding to the blurred and the sharp images. This has the impact of faster convergence as empirically detected (Table 3).

Training Data The HG network was trained by synthetically blurred images while the cGAN with the simulated motion blur developed in Sect. 3.4. Three million images from MS-Celeb of Guo et al. (2016) were blurred using random camera-shake kernels from Hradiš et al. (2015). The HG was trained to convergence and then the weights were kept as constants in the training of the cGAN. Conversely, the videos of (i) $2MF^2$, (ii) BBC trainset of Chung and Zisserman (2016) (sub-sampled to keep every 10th video) were utilized to generate the simulated motion blur for cGAN training; 300,000 pairs from $2MF^2$ and 60,000 pairs from BBC. 100,000 more pairs were generated by parsing the clips of $2MF^2$ in reverse (temporal) order.[7]

[7] Each reverse blurry frame was compared with the frames exported from the forward pass; the frames that included essentially the same blur were skipped in the reverse order pass.

Fig. 9 Indicative frames from the databases used for testing. **a** 300VW **b** YTF

Table 4 Self evaluation results on 300VW

Method	PrAvg 7		
	AUC	Failure rate (%)	SSIM
cGAN-noHG-only2mf2	0.751	8.800	0.92
cGAN-noHG	0.759	7.895	0.88
HG-disc	0.766	5.261	0.89
HG-disc-VGG	0.771	5.383	0.90
HG-vanilla-cGAN	0.783	5.962	0.93
vanilla-cGAN-VGG	0.756	7.009	0.90
HG-vanilla-cGAN-VGG	0.786	5.207	0.93
Final	0.798	3.947	0.95
Nr. of test images	17,125		

The results indicate that the methods with a simple conditional GAN (cGAN-noHG-only2mf2, cGAN-noHG, HG-disc, HG-disc-VGG, vanilla-cGAN-VGG) are outperformed by the two-step architectures. The vanilla cGAN (single pathway) with identity loss does not perform as well our proposed method ('final'). This is expressed qualitatively in Fig. 10

Training Details The network ran for 60 epochs, trained in a single GPU. We tried to minimize the hyper-parameter tuning by setting $\lambda_{ci} = \lambda_{cg} = \lambda_r$. The values were set experimentally as $\lambda_r = 100$, $\lambda_p = 8$.

5.2 Experimental Setup

We have included experiments in two popular tasks in face analysis, i.e. landmark localization and face recognition. The benchmarks used for each task were the following:

- *Landmark Localization* The benchmark of 300 videos in-the-wild (300VW) of Shen et al. (2015), Chrysos et al. (2015) was utilized; this is currently the most established public benchmark for landmark tracking (Chrysos and Zafeiriou 2017a). This database includes 114 videos; the testset comprises of 64 videos, which are divided into three categories with different degrees of difficulty. Each video is approximately a minute long; each frame includes a single person with 68 markup annotations (Gross et al. 2010) in the face. The 64 videos for testing include 120,000 frames. Such an amount of frames provides a sufficient pool for averaging schemes.
- *Face Verification* The Youtube Faces (YTF) of Wolf et al. (2011) includes 3425 videos of 1595 identities. YTF has been the most popular public benchmark for verification the last few years. Each video includes a single person; there are multiple videos corresponding to the same identity, however the movement in each video is restricted. The length of the videos varies from 50 frames to over 6000 with 181 frames as average duration.

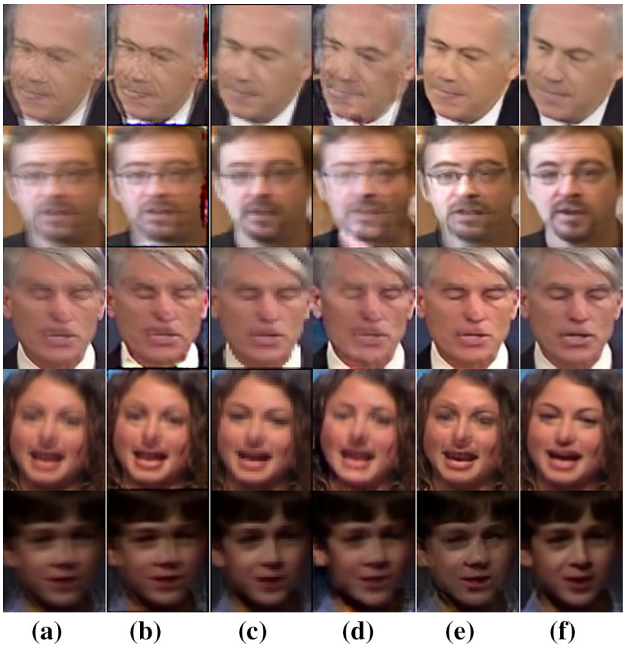


Fig. 10 Qualitative sample images for the self evaluation experiment. From left to the right, the columns correspond to the following case: **a** 'Blurred', **b** 'cGAN-noHG', **c** 'HG-disc-VGG', **d** 'vanilla-cGAN-VGG', **e** 'HG-vanilla-cGAN', **f** 'final'. Note how that the methods with a single network either cannot remove the blur or cannot restore the high frequency details. The 'HG-vanilla-cGAN' can restore the blur but does not always result in a natural image, while our proposed version (denoted as 'final') both restores the image and results in a natural image

The two datasets include another axis of variation, i.e. facial sizes. The mean size for 300VW is 135 × 135, while for YTF is 85 × 85. An indicative frame from the datasets is visualized in Fig. 9.

None of the aforementioned databases includes real-world blurred/deblurred pairs, so we have opted to simulate the motion blur with multiple methods mentioned in Sect. 3.4.

Table 5 Quantitative results on the experiment on landmark localization of Sect. 5.4

Method	PrAvg 7			PrAvg 9			PrAvg 11		
	AUC	*Failure Rate (%)*	*SSIM*	*AUC*	*Failure Rate (%)*	*SSIM*	*AUC*	*Failure Rate (%)*	*SSIM*
Krishnan et al. (2011)	0.706	11.013	0.79	0.734	5.801	0.78	0.728	6.294	0.79
Babacan et al. (2012)	0.730	9.950	0.93	0.735	7.796	0.92	0.728	7.993	0.92
Zhang et al. (2013)	0.692	11.276	0.80	0.713	6.956	0.79	0.711	7.129	0.80
Xu et al. (2013)	0.451	42.377	0.86	0.348	55.125	0.85	0.451	41.528	0.86
Pan et al. (2014b)	0.709	7.527	0.75	0.699	8.194	0.74	0.692	8.434	0.75
Pan et al. (2014a)	0.745	5.255	0.80	0.734	6.077	0.81	0.727	6.304	0.80
Pan et al. (2016)	0.768	4.502	0.87	0.756	4.968	0.85	0.749	5.440	0.86
Chakrabarti (2016)	0.504	37.080	0.89	0.480	39.708	0.88	0.466	41.403	0.88
Nah et al. (2017)	**0.792**	**3.235**	**0.94**	**0.779**	**3.837**	**0.93**	**0.771**	**4.471**	**0.93**
Kupyn et al. (2018)	0.766	5.647	0.89	0.749	6.902	0.89	0.739	7.503	0.89
Zhu et al. (2017)	**0.789**	**3.399**	**0.91**	0.778	3.860	0.90	**0.772**	**4.260**	**0.90**
Kupyn et al. (2018) + data	0.789	4.473	0.91	**0.779**	**5.083**	**0.90**	0.771	5.604	0.90
ours	**0.798**	**3.947**	**0.95**	**0.788**	**4.525**	**0.94**	**0.781**	**5.018**	**0.93**
Blurred	0.743	8.876	0.94	0.747	6.680	0.93	0.739	7.043	0.93
Oracle	0.827	1.816	–	0.829	1.605	–	0.829	1.612	–
Nr. of test images	17 125			13 083			10 422		

Colouring denotes the methods' performance ranking per category (based on AUC): 🟥 first 🟩 second 🟦 third

The first row dictates the blurring process. In this plot from left to right predefined averaging with (a) 7, (b) 9, (c) 11 frames

Table 6 Second part of the quantitative results for the landmark localization experiment of Sect. 5.4

Method	PrAvg 15			PrAvg 21			VLA		
	AUC	*Failure Rate (%)*	*SSIM*	*AUC*	*Failure Rate (%)*	*SSIM*	*AUC*	*Failure Rate (%)*	*SSIM*
Krishnan et al. (2011)	0.719	6.717	0.79	0.707	7.094	0.79	0.731	4.192	0.79
Babacan et al. (2012)	0.710	8.807	0.91	0.684	10.641	0.90	0.715	6.817	0.89
Zhang et al. (2013)	0.705	7.165	0.80	0.689	8.122	0.80	0.721	4.879	0.80
Xu et al. (2013)	0.446	41.552	0.86	0.438	41.435	0.85	0.390	48.459	0.85
Pan et al. (2014b)	0.681	8.536	0.75	0.665	9.170	0.74	0.692	6.852	0.76
Pan et al. (2014a)	0.716	6.609	0.79	0.694	7.960	0.78	0.727	4.280	0.78
Pan et al. (2016)	0.735	5.808	0.85	0.714	6.590	0.84	0.745	3.734	0.84
Chakrabarti (2016)	0.437	44.375	0.88	0.405	48.065	0.87	0.410	45.006	0.85
Nah et al. (2017)	**0.754**	**5.225**	**0.91**	0.730	7.255	0.90	**0.767**	**3.012**	**0.90**
Kupyn et al. (2018)	0.720	8.386	0.86	0.694	9.734	0.85	0.728	5.513	0.85
Zhu et al. (2017)	0.686	8.183	0.85	**0.736**	**5.903**	**0.88**	**0.771**	**2.713**	**0.88**
Kupyn et al. (2018) + data	**0.747**	**6.704**	**0.89**	**0.732**	**7.195**	**0.88**	0.765	4.316	0.88
ours	**0.769**	**5.482**	**0.91**	**0.748**	**6.751**	**0.90**	**0.786**	**3.118**	**0.90**
Blurred	0.721	7.667	0.92	0.695	9.270	0.90	0.725	5.866	0.90
Oracle	0.832	1.276	–	0.833	1.411	–	0.827	1.568	–
Nr. of test images	7 369			4 962			5 677		

Colouring denotes the methods' performance ranking per category (based on AUC): 🟥 first 🟩 second 🟦 third

Specifically, there are three types of blur that we utilize in our experiments:

1. *Synthetic Blur* The synthetic blur method of Kupyn et al. (2018) was utilized. Random trajectories are generated through a Markov process; the kernel is sampled from the trajectory points and the sharp image is convolved with this kernel.
2. *Predefined Averaging (PrAvg)* A predefined number L of frames are added to create the motion blur. The average intensity of the images summed is the blurry image, the ground-truth image is the one in the middle of the interval, i.e. $L/2$. The number L can be visually defined. Even though this is a generic method, it does not take into account neither the intra-class variation, e.g. the

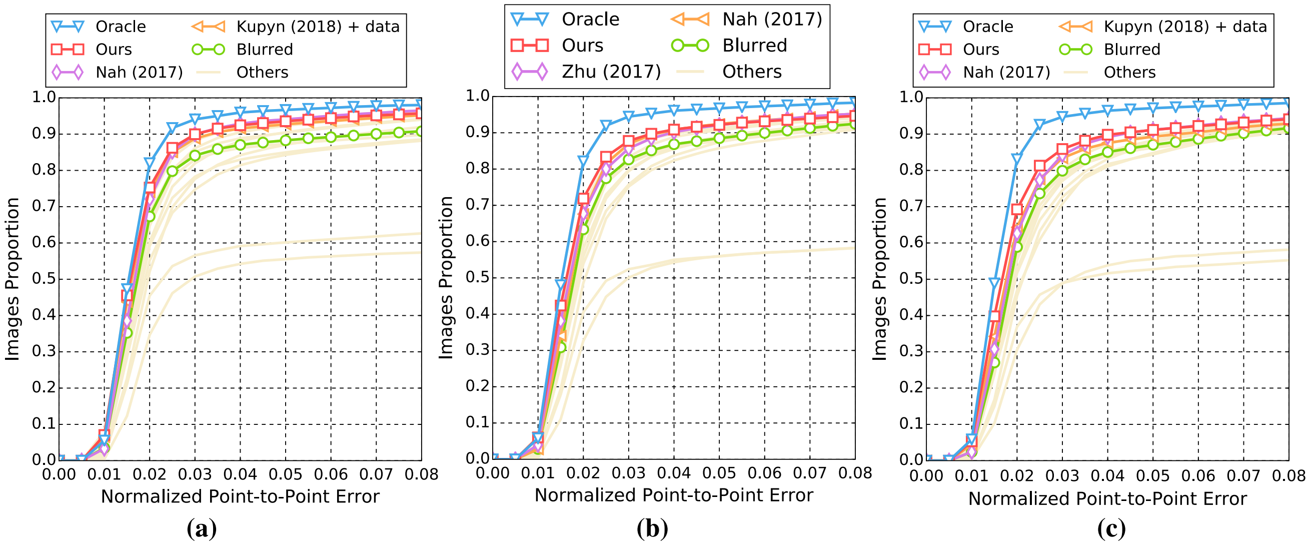


Fig. 11 CED plots for the landmark localization experiment of Sect. 5.4. To avoid cluttering the plot, only the top 3 methods along with the oracle and the original blur performance are plotted. From left to right the plots correspond to the experiments with blurring process: predefined averaging with **a** 7, **b** 11, **c** 15 frames

movement is not uniform temporally, nor the inter-class variation, e.g. the statistics of each clip are quite different.

3. *Variable Length Averaging (VLA)* This is the variable length averaging proposed in Sect. 3.4.

A wealth of methods were employed for our comparisons:

- The energy optimization methods of Krishnan et al. (2011), Babacan et al. (2012), Zhang et al. (2013), Xu et al. (2013), Pan et al. (2014b, 2016) were included.
- The strong-performing methods of Chakrabarti (2016), Kupyn et al. (2018), Nah et al. (2017) were compared.
- The domain-specific method of Pan et al. (2014a) (face deblurring) was included.
- Apart from the provided pre-trained model, we trained the method of Kupyn et al. (2018) with our data, to understand whether the improvement of our method can be solely provided by well-engineered training data. This method is denoted as 'Kupyn et al. (2018) + data' in the experiments.
- The recent, strong-performing method of Zhu et al. (2017) was trained with our data to demonstrate the strengths of our customized architecture.

The aforementioned methods include the majority of the deblurring methods, as well as two recent strong-performing methods trained with our data. A considerable computational effort was required to evaluate each of these methods for every single experiment,[8] since as reported by Chakrabarti (2016) the optimization-based methods require several minutes per frame.

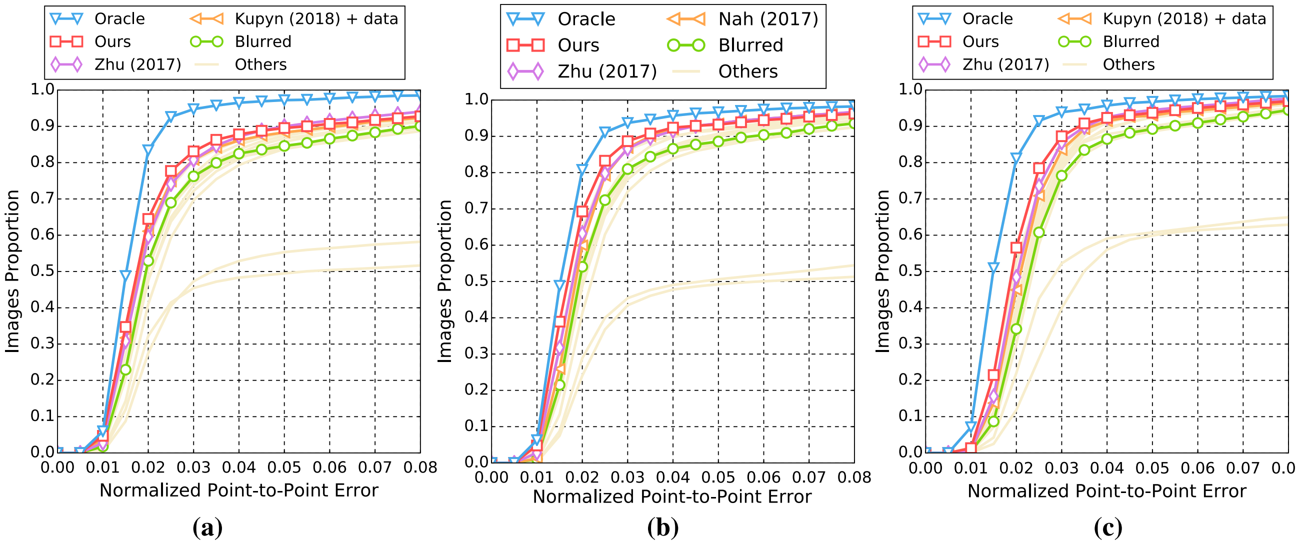
5.3 Self Evaluation

To analyze the various components of our method, we trained from scratch different variants, which were:

1. 'cGAN-noHG': Only our cGAN where the G_{bl} is not conditioned on the HG, i.e. it accepts only the blurry image.
2. 'cGAN-noHG-only2mf2': Same as the previous, but trained only on the $2MF^2$ forward data, i.e. 300,000 samples.
3. 'HG-disc': The HG plus a discriminator, i.e. a conditional GAN with the encoder-decoder being the HG.
4. 'HG-disc-VGG': The aforementioned 'HG-disc' trained with an additional identity loss term (pre-trained VGG).
5. 'HG-vanilla-cGAN': Same as our two-step architecture with our cGAN replaced by the original cGAN, i.e. single pathway not including the G_s.
6. 'vanilla-cGAN-VGG': The original cGAN trained with an additional identity loss term (pre-trained VGG).
7. 'HG-vanilla-cGAN-VGG': Similar to the aforementioned 'HG-vanilla-cGAN' with an additional identity loss term (pre-trained VGG).
8. 'final': The full proposed version.

To benchmark those variants we used the 300VW and landmark localization. Each blurry/sharp pair was produced by averaging 7 frames, which results in 17,125 test pairs. Each blurry image was deblurred with the four aforemen-

[8] The public implementations of all methods were used.



(a) **(b)** **(c)**

Fig. 12 Continuation of the CED plots of the landmark localization experiment. From left to right the plots correspond to the experiments with blurring process: **a** predefined averaging with 21 frames, **b** VLA, **c** synthetic blur

Table 7 Third (last) part of the quantitative results for the landmark localization experiment of Sect. 5.4

Method	Synthetic		
	AUC	*Failure Rate (%)*	*SSIM*
Krishnan et al. (2011)	**0.736**	**2.774**	**0.76720**
Babacan et al. (2012)	0.686	5.344	0.83956
Zhang et al. (2013)	0.730	2.850	0.77321
Xu et al. (2013)	0.437	36.845	0.76055
Pan et al. (2014b)	0.715	3.486	0.74478
Pan et al. (2014a)	0.718	3.639	0.72402
Pan et al. (2016)	0.727	2.723	0.77778
Chakrabarti (2016)	0.475	34.631	0.84699
Nah et al. (2017)	0.729	2.774	0.83791
Kupyn et al. (2018)	0.733	4.580	0.84477
Zhu et al. (2017)	**0.744**	**2.488**	**0.84404**
Kupyn et al. (2018) + data	0.735	3.053	0.83154
ours	**0.764**	**2.672**	**0.87141**
Blurred	0.697	4.656	0.85225
Oracle	0.830	1.425	–
Nr. of test images	3 930		

Ranking (based on AUC): 🟥 first 🟩 second 🟦 third

tioned variants and then landmark localization was performed in each one of those using the network of Yang et al. (2017). Standard quantitative metrics, i.e. AUC, failure rate, were used; the metrics are summarized in the localization experiment in Sect. 5.4.

The quantitative results are summarized in Table 4. The following were deduced based on the results: (i) the model did benefit from additional labels, (ii) the additional conditioning label, i.e. HG network, improved deblurring, (iii) the final model with the two pathway cGAN outperformed all the variants.

5.4 Landmark Localization

If the deblurred images indeed resemble the statistics of sharp facial images, then a localization method should be close to the localization of the original ground-truth images. To assess this similarity, we utilized the 300VW as a testbed to compare the localization of the deblurred images. The frames of 300VW were blurred and each comparison method was used to deblur them. Sequentially, the state-of-the-art landmark localization method of Yang et al. (2017), winner of the Menpo Challenge of Zafeiriou et al. (2017), was used to perform landmark localization in the deblurred images. Apart from the comparison methods, an ‘oracle’ was implemented. The ‘oracle’ represents the perfect deblurring method, i.e. the deblurred images are identical to the latent sharp images. We used the oracle to indicate the upper bound of the performance of the deblurring methods.

The following error metrics are reported for this experiment:

- cumulative error distribution (CED) plot: It depicts the percentage of images (y-axis) that have up to a certain percentage of error (x-axis).
- area under the curve (AUC): A scalar that is the area under the CED plot.
- failure rate: The localization error is cut-off at 0.08; any larger error is considered a failure to localize the fiducial points. Failure rate is the percentage of images with error larger than the cut-off error.
- structural similarity (SSIM) by Wang et al. (2004): Image quality metric typically used to report the quality of the deblurred images.

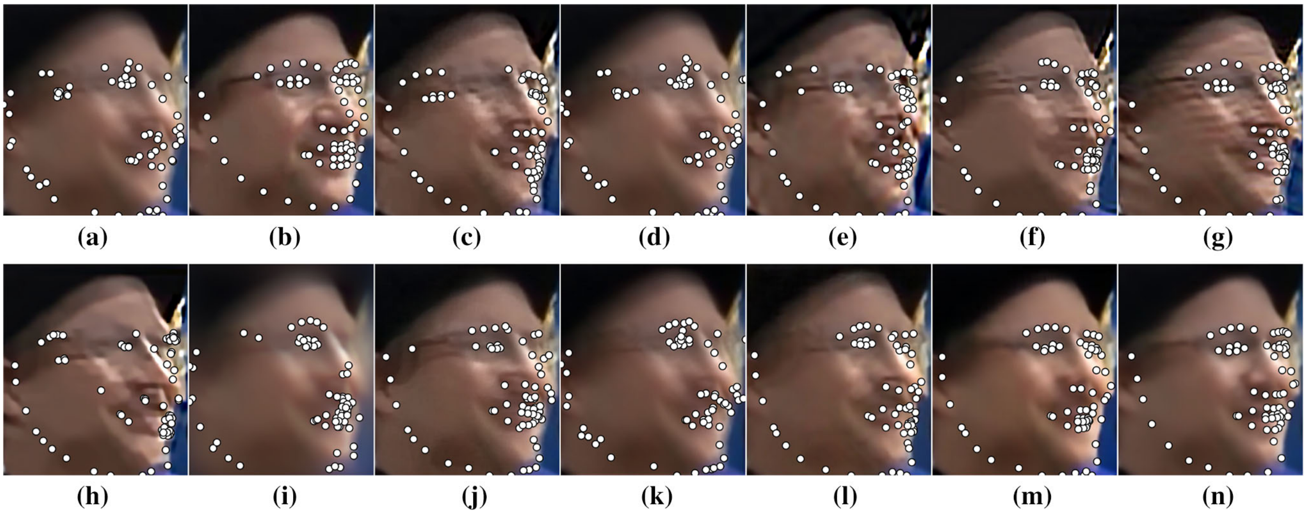


Fig. 13 Landmarks overlaid in the deblurred images as described in Sect. 5.4. **a** blurred, **b** gt, **c** Krishnan et al. (2011), **d** Babacan et al. (2012),**e** Zhang et al. (2013), **f** Pan et al. (2014b), **g** Pan et al. (2014a), **h** Pan et al. (2016), **i** Chakrabarti (2016), **j** Nah et al. (2017), **k** Kupyn et al. (2018), **l** Zhu et al. (2017), **m** Kupyn et al. (2018) + data, **n** ours

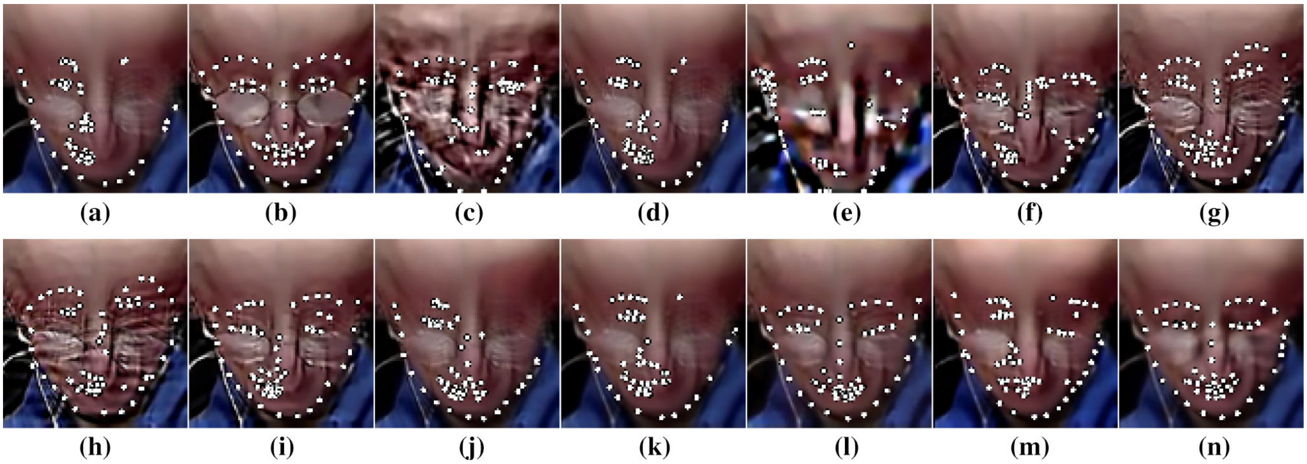


Fig. 14 Landmarks overlaid in the deblurred images as described in Sect. 5.4. **a** Blurred, **b** gt, **c** Krishnan et al. (2011), **d** Babacan et al. (2012), **e** Zhang et al. (2013), **f** Xu et al. (2013), **g** Pan et al. (2014b), **h** Pan et al. (2014a), **i** Pan et al. (2016), **j** Nah et al. (2017), **k** Kupyn et al. (2018), **l** Zhu et al. (2017), **m** Kupyn et al. (2018) + data, **n** ours

– cosine distance distribution plot: The embedding of the face is extracted per frame with faceNet of Schroff et al. (2015); similarly the embedding of the sharp image is extracted and the cosine distance of the two is computed. A cumulative distribution of those cosine distances is plotted.

The CED plot, AUC and Failure rate consist standard localization error metrics; we utilize the same conventions as in Chrysos et al. (2018), i.e. the error is the mean euclidean distance of the points, normalized by the diagonal of the ground-truth bounding box.

The following schemes for simulating blur were utilized: (i) predefined averaging with $L \in \{7, 9, 11, 15, 21\}$, (ii) VLA, (iii) synthetic blur in ground-truth images of VLA. The different options of predefined averaging considered allowed us to assess the robustness under mild differences in blurring averages.

The quantitative results for the predefined averaging cases are depicted in Tables 5, 6; the CED plots[9] of the top three performing methods (based on the AUC) are visualized in Figs. 11, 12. The complete CED plots are visualized in the supplementary material. The reported metrics validate that optimization methods did not perform as well as learning-

[9] The results of the averaging 7 or 9 frames are similar, hence the related CED plot is omitted.

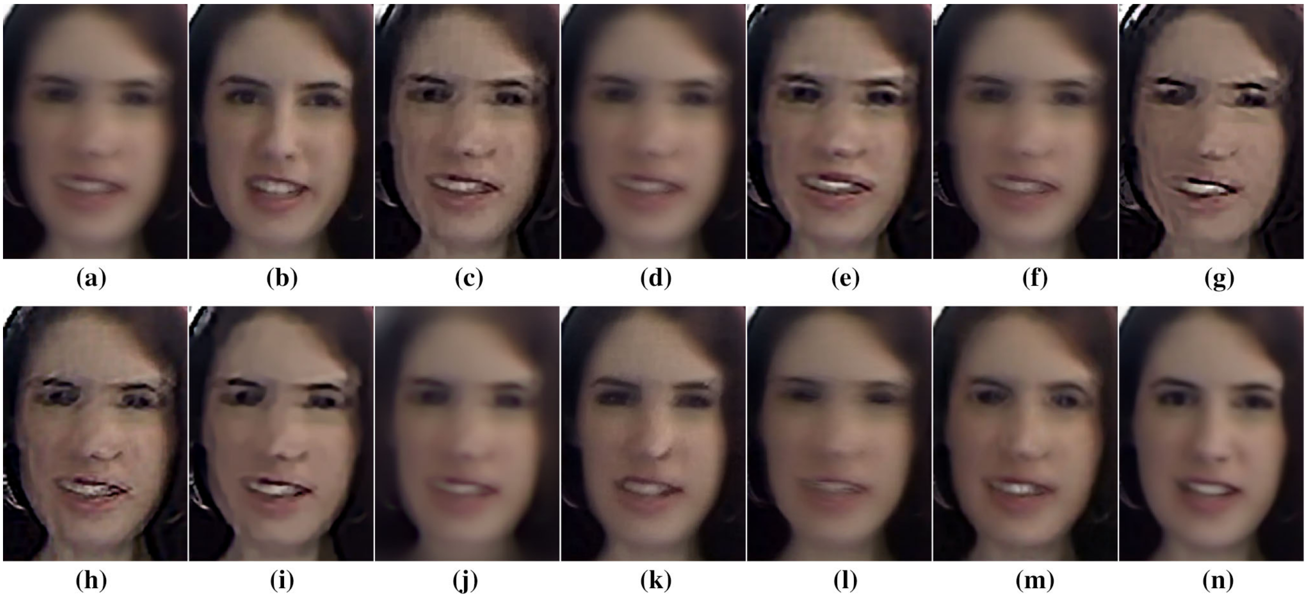


Fig. 15 Visual results. The majority of the existing methods fail to deblur the eyes and the nose; even the state-of-the-art method of Nah et al. (2017) does not manage to yield a realistic face. On the contrary, our method outputs a realistic face with both the eyes and the nose accurately deblurred. **a** Blurred, **b** gt, **c** Krishnan et al. (2011), **d** Babacan et al. (2012), **e** Zhang et al. (2013), **f** Xu et al. (2013), **g** Pan et al. (2014b), **h** Pan et al. (2014a), **i** Pan et al. (2016), **j** Chakrabarti (2016), **k** Nah et al. (2017), **l** Kupyn et al. (2018), **m** Zhu et al. (2017), **n** ours

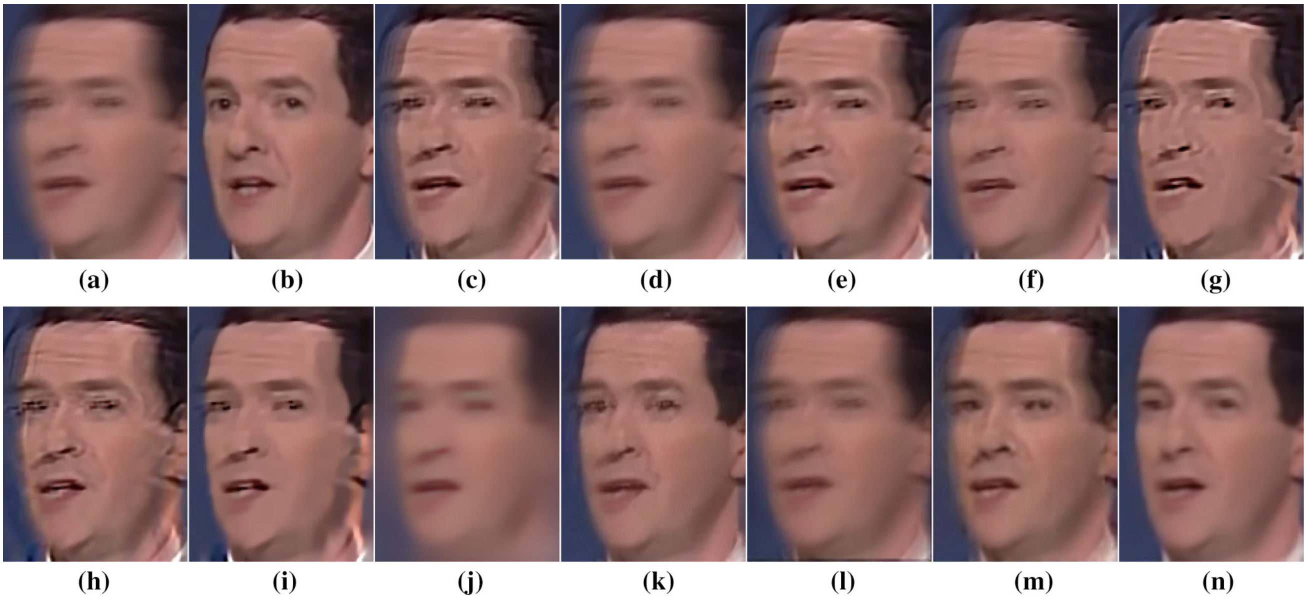


Fig. 16 The nature of the blur caused ghost artifacts in the outputs of the majority of the methods. Some of them are more subtle, e.g. in Pan et al. (2016); Nah et al. (2017), however they are visible by zooming-in the figures. Our method avoided such artifacts and returned a plausible face. **a** Blurred, **b** gt, **c** Krishnan et al. (2011), **d** Babacan et al. (2012), **e** Zhang et al. (2013), **f** Xu et al. (2013), **g** Pan et al. (2014b), **h** Pan et al. (2014a), **i** Pan et al. (2016), **j** Chakrabarti (2016), **k** Nah et al. (2017), **l** Kupyn et al. (2018), **m** Zhu et al. (2017), **n** ours

based methods; the only method that consistently performed well was Pan et al. (2016). Conversely, the learning-based methods improved the results of the blurred, while our method consistently outperformed the rest in all cases. As it is noticeable from the CED plots, in the region of small errors, i.e. < 0.02, a large number of averaged frames deteriorated the fitting considerably. Additionally, the majority of the methods were robust to small changes in the number of averaged frames, i.e. from 7 to 9 or from 9 to 11. On the contrary, the difference between averaging 7 and 21 frames

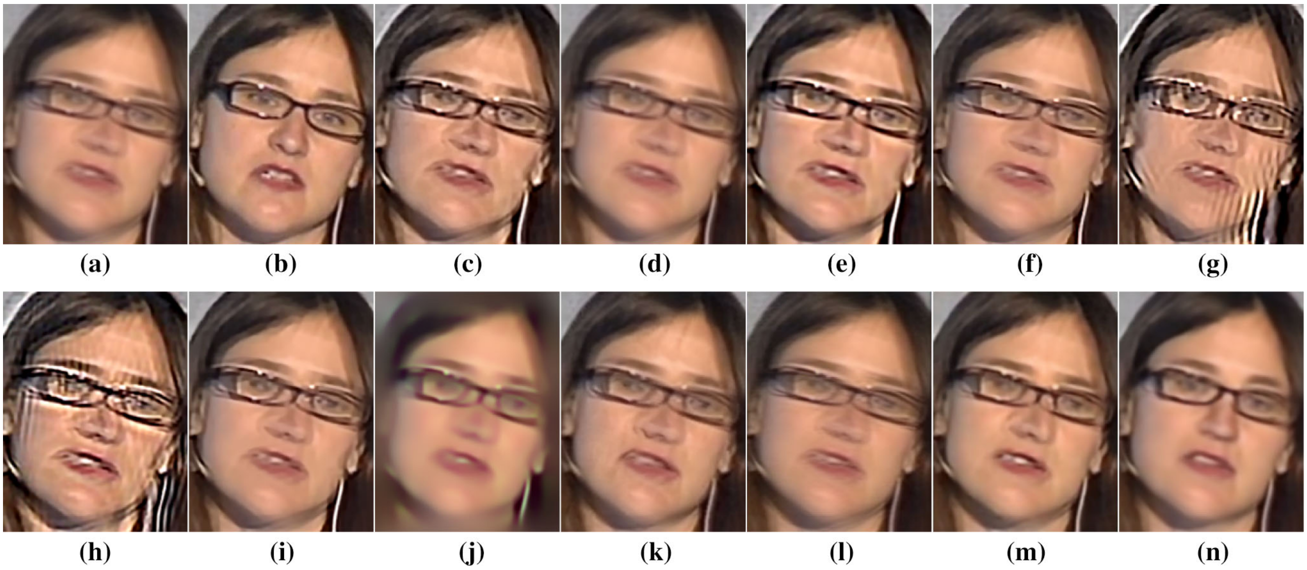


Fig. 17 The glasses are severely affected by the motion blur in this case; the compared method, even those for generic deblurring, fail to restore the glasses and the nose region, while our method returns a plausible outcome. **a** Blurred, **b** gt, **c** Krishnan et al. (2011), **d** Babacan et al. (2012), **e** Zhang et al. (2013), **f** Xu et al. (2013), **g** Pan et al. (2014b), **h** Pan et al. (2014a), **i** Pan et al. (2016), **j** Chakrabarti (2016), **k** Nah et al. (2017), **l** Kupyn et al. (2018), **m** Zhu et al. (2017), **n** ours

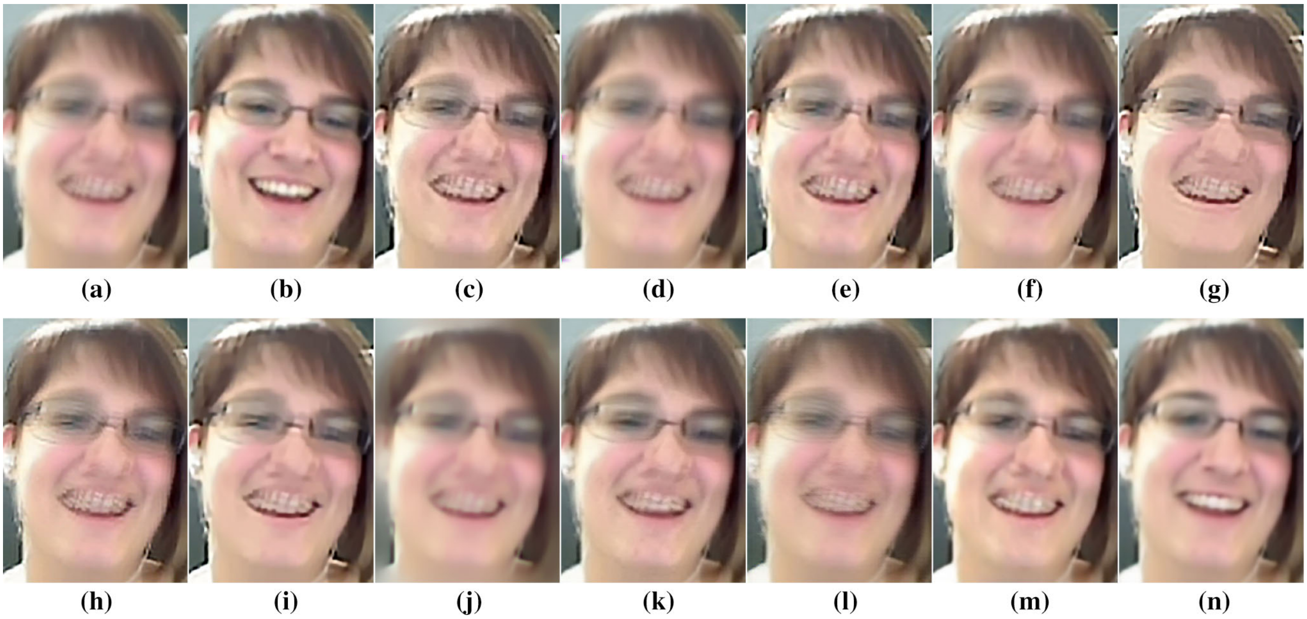


Fig. 18 The movement of the face caused severe blur in the mouth region. All the compared methods fail to deblur the mouth, our method does that accurately. **a** Blurred, **b** gt, **c** Krishnan et al. (2011), **d** Babacan et al. (2012), **e** Zhang et al. (2013), **f** Xu et al. (2013), **g** Pan et al. (2014b), **h** Pan et al. (2014a), **i** Pan et al. (2016), **j** Chakrabarti (2016), **k** Nah et al. (2017), **l** Kupyn et al. (2018), **m** Zhu et al. (2017), **n** ours

was noticeable in most methods; in our case the decrease in the performance was less than the compared methods.

Similar conclusions hold for the experiments with the VLA and the synthetic kernels schemes, the results of which exist in Tables 6 and 7. In both cases, our method increased the margin from the comparison methods. That is attributed to the very diverse number of blurs that our method was trained on. On the contrary, the prior art of Nah et al. (2017) suffered for slightly modified conditions than the predefined averaging.

In all seven cases with different blurs examined, we verified that our method was robust to mediocre and severe blurs.

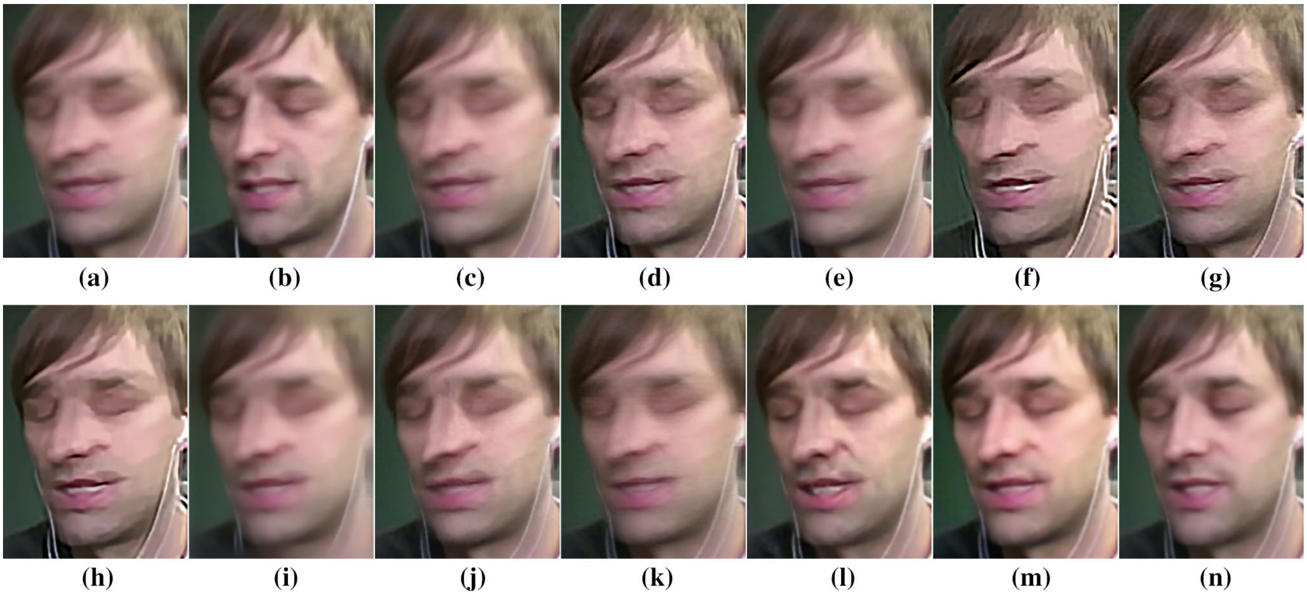


Fig. 19 The horizontal rotation caused severe blur in the nose, which our method deblurs successfully in comparison to the rest methods. **a** Blurred, **b** gt, **c** Babacan et al. (2012), **d** Zhang et al. (2013), **e** Xu et al. (2013), **f** Pan et al. (2014b), **g** Pan et al. (2014a), **h** Pan et al. (2016), **i** Chakrabarti (2016), **j** Nah et al. (2017), **k** Kupyn et al. (2018), **l** Zhu et al. (2017), **m** Kupyn et al. (2018), **n** ours

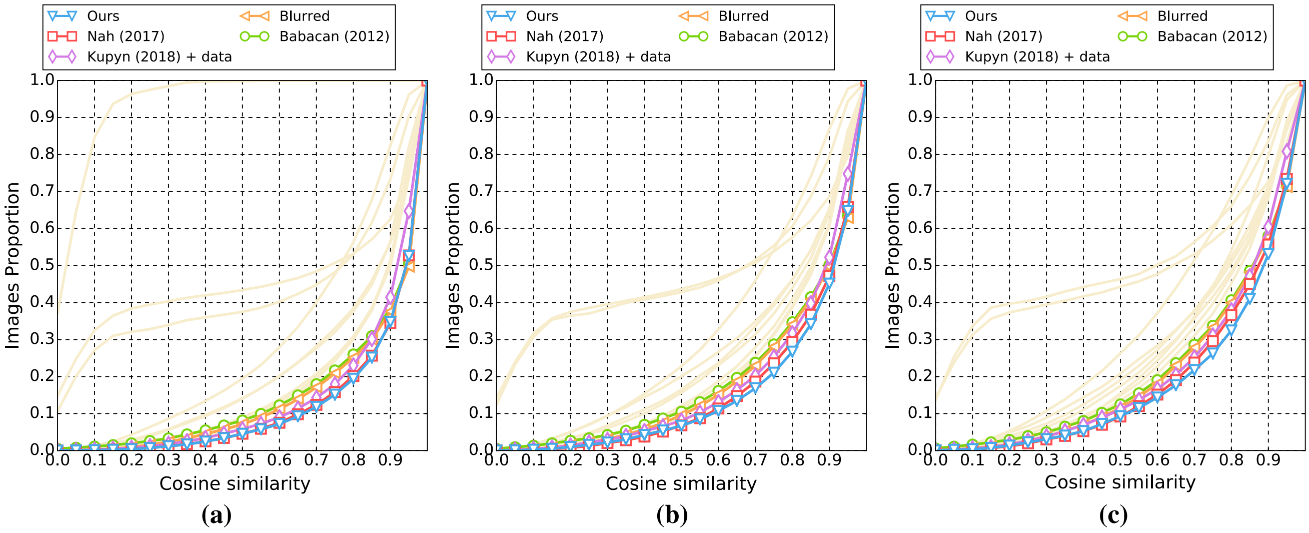


Fig. 20 Cosine distance distribution plots for the landmark localization experiment of Sect. 5.4. To avoid cluttering the plot, only the top four methods along with the original blur performance are plotted. The narrower distributions concentrated around one declare closer representation of the ground-truth identity. Please find further details in the text. From left to right the plots correspond to the experiments with blurring process: predefined averaging with **a** 7, **b** 11, **c** 15 frames

In Figs. 13, 14 two images with the landmarks overlaid are visualized, while in Figs. 15, 16, 17, 18 and 19 qualitative images with different cases are illustrated.[10]

Aside of the state-of-the-art network of Yang et al. (2017), we selected the top three performing methods and repeated two experiments with an alternative localization method. The method chosen was CFSS of Zhu et al. (2015), which was the top performing regression-based method. The results, which are in the supplementary material, ranked our method's deblurred images as the top performing ones.

[10] A supplementary video with deblurring examples is uploaded on https://youtu.be/Zb_6taC0Oi4.

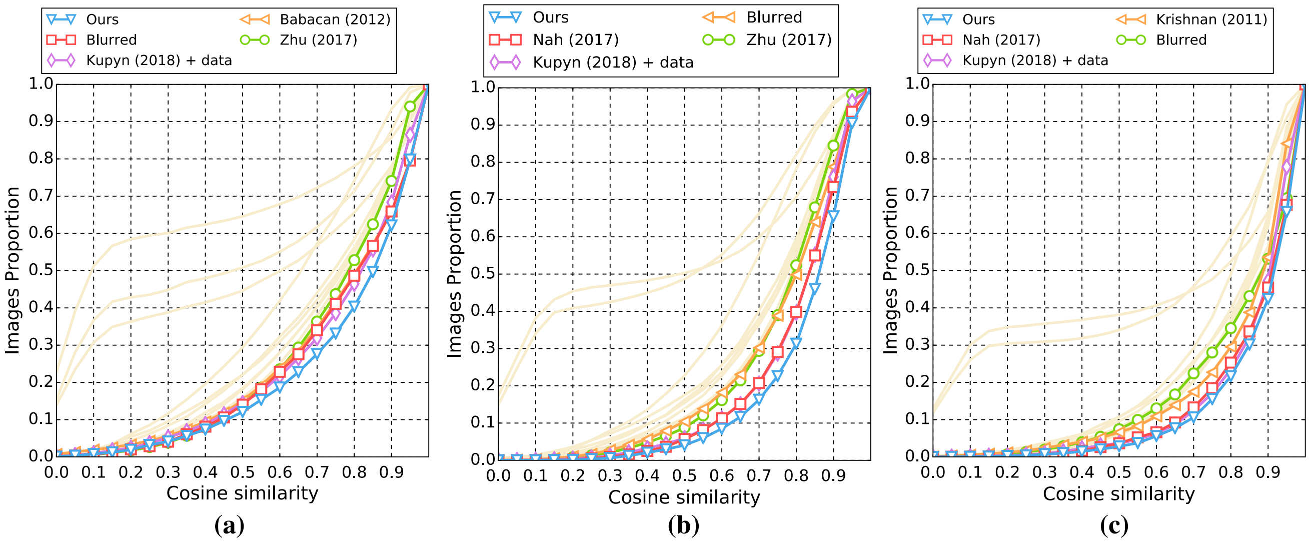


Fig. 21 Continuation of the cosine distance distribution plots from Fig. 20. From left to right the plots correspond to the experiments with blurring process: **a** predefined averaging with 21 frames, **b** VLA, **c** synthetic blur

Apart from the localization of the fiducial points, we wanted to assess the preservation of the identity in the various deblurring methods. Since there is no ground-truth information for the identity encoding, we opt to report a soft-metric instead. The embeddings of the widely used model of faceNet were adopted; we measured the cosine distance between each deblurred frame's embedding and the respective ground-truth's embedding. In the perfect identity preservation case, the cosine distance distribution plot should be a dirac delta around one; a narrow distribution centered to one denotes proximity to the embeddings of the ground-truth face. The results in Figs. 20, 21 indicate that our method is robust in different cases of blur and has a cosine distance distribution closer to the ground-truth than the compared methods.

5.5 Face Verification

We utilized the Youtube Faces (YTF) dataset for performing face verification. The video frames were averaged to generate the blurry/sharp pairs; each deblurring method was applied and then the deblurred outputs were used for face verification. Assessing the accuracy of each method would directly allow us to compare which method results in facial representations that maintain the identity.

The complete setup for the experiment was the following: The verification was performed in the deblurred images by extracting representation from faceNet of Schroff et al. (2015). We employed the predefined averaging with (i) 7, (ii) 11 frames.[11] The error metric used was the mean accuracy along with the computed standard deviation. The results are summarized in Table 8. The learning-based methods performed preferably to the optimization-based, while our method outperformed all the rest.

Table 8 Quantitative results in the face verification experiment of Sect. 5.5

Method	*PrAvg 7*	*PrAvg 11*
	Mean acc. ± std	*Mean acc. ± std*
Krishnan et al. (2011)	0.7344 ± 0.0155	0.7146 ± 0.0203
Babacan et al. (2012)	0.8194 ± 0.0193	0.8024 ± 0.0168
Zhang et al. (2013)	0.7002 ± 0.0219	0.6964 ± 0.0150
Xu et al. (2013)	0.5346 ± 0.0264	0.5348 ± 0.0176
Pan et al. (2014b)	0.6768 ± 0.0234	0.6654 ± 0.0213
Pan et al. (2014a)	0.7386 ± 0.0228	0.7190 ± 0.0145
Pan et al. (2016)	0.7648 ± 0.0241	0.7508 ± 0.0217
Chakrabarti (2016)	0.5510 ± 0.0272	0.5410 ± 0.0206
Nah et al. (2017)	**0.8434 ± 0.0188**	**0.8106 ± 0.0156**
Kupyn et al. (2018)	**0.8244 ± 0.0161**	0.7942 ± 0.0163
Zhu et al. (2017)	0.8204 ± 0.0222	0.7954 ± 0.0153
Kupyn et al. (2018) + data	0.8176 ± 0.0207	**0.8056 ± 0.0105**
ours	**0.8482 ± 0.0179**	**0.8212 ± 0.0087**
Blurred	0.8414 ± 0.0206	0.8154 ± 0.0136
Oracle	0.9006 ± 0.0118	0.9044 ± 0.0134

Ranking (based on mean accuracy): 🟥 first 🟩 second 🟦 third

5.6 Real-World Blurry Video

A video with extreme motion blur was captured with a high-precision camera; ground-truth frames are not available, we only have the 160 blurry frames. To allow a quantitative comparison, a sharp frame of the video was selected for extracting the identity embeddings with faceNet. Then each method deblurred the images, the embeddings were extracted

[11] We did not utilize the VLA method, since some of the videos were shorter than those required for the rolling averaging of VLA.

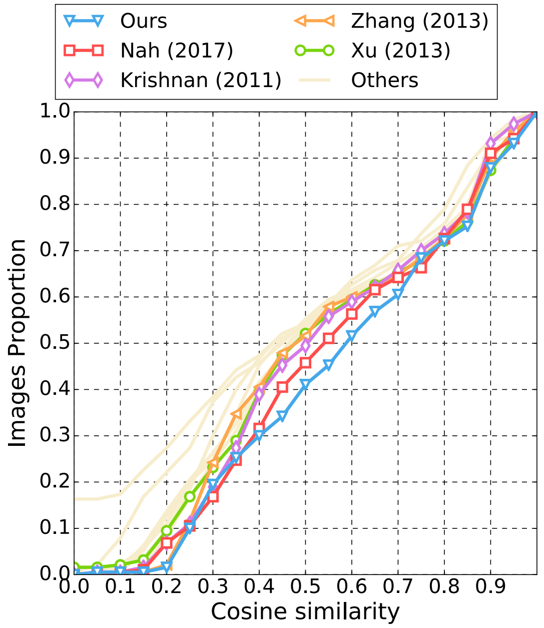


Fig. 22 Cosine distance distribution plots for the real-world blurry video of Sect. 5.6. To avoid cluttering the plot, only the top four methods along with the original blur performance are plotted. The narrower distributions concentrated around one declare closer representation of the ground-truth identity. The legends from top to bottom, left to right declare the ranking of the methods

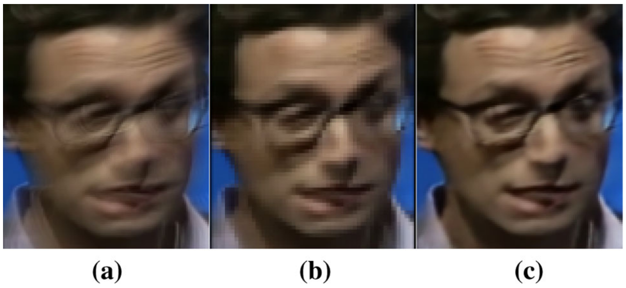


Fig. 24 Failure case of our method. Due to the extensive blur of the original image, the glasses are not correctly deblurred. **a** Blurred, **b** HG, **c** ours

and compared as in the localization experiment (Sect. 5.4). The respective cosine distance distribution plot is depicted in Fig. 22. As it is noticeable in the plot our method is ranked as the one closest to the identity embedding from the sharp frame. An indicative frame is plotted in Fig. 23.

5.7 Discussion

The thorough experimental comparisons above demonstrate that our method is consistently better than the compared methods. The deblurred images are utilized for high level

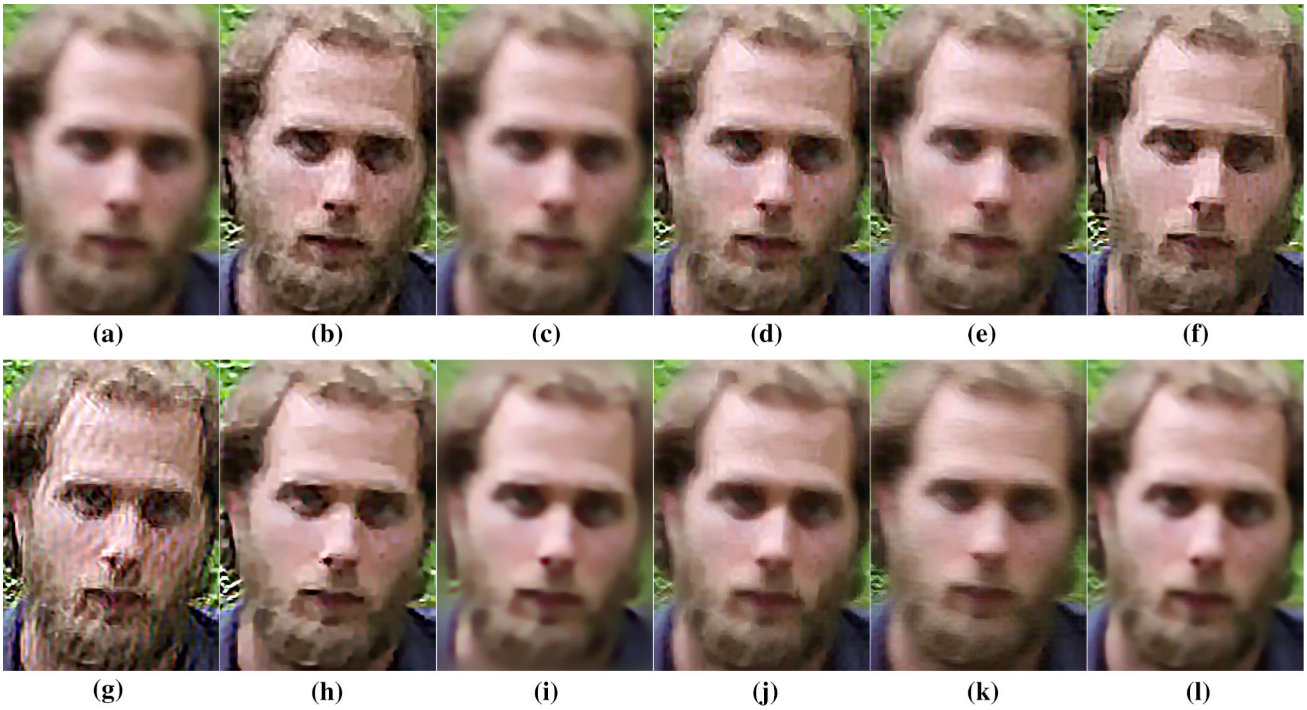


Fig. 23 Deblurring in a real-world blurry image. Even the state-of-the-art methods of Nah et al. (2017), Kupyn et al. (2018) over-smooth the blurred image (please zoom-in for further understanding). On the contrary our method yields an improved outcome. **a** Blurred, **b** Krishnan et al. (2011), **c** Babacan et al. (2012), **d** Zhang et al. (2013), **e** Xu et al. (2013), **f** Pan et al. (2014b), **g** Pan et al. (2014a), **h** Pan et al. (2016), **i** Chakrabarti (2016), **j** Nah et al. (2017), **k** Kupyn et al. (2018), **l** ours

tasks (landmark localization and face verification); our method outperforms the compared methods in these tasks as well. We additionally validate that VLA is an effective way of blurring faces. However, we intend to improve our method in the following two cases. In the rare cases that the first network (HG) fails, the final output is not convincing (see Fig. 24). Another improvement point is the the real-world face deblurring, where most methods do not yield a natural, sharp image. Aside of the motion blur, the additional sources of noise in this video, e.g. compression artifacts, non-Gaussian noise, consist the real-world deblurring still challenging.

6 Conclusion

In this work, we introduce a method for performing motion deblurring of faces. We introduce a two-step architecture, where in the first step a strong discriminative network restores the low-frequency details; in the second step the high-frequency details are restored. To train this model, we devise a new way of simulating motion blur by averaging a variable number of frames. The frames originate from videos in the $2MF^2$ dataset that we collect for this task. We test our system in a thorough experimentation, using both quality metrics typically used in deblurring but also more established quantitative tasks in face analysis, i.e. landmark localization and face verification. In both tasks and in all the conducted experiments our method performs favourably to the compared methods setting the new state-of-the-art for face deblurring.

References

Arjovsky. M., Chintala. S., & Bottou, L. (2017). Wasserstein gan. arXiv:1701.07875.

Babacan, S. D., Molina, R., Do, M. N., & Katsaggelos, A. K. (2012). Bayesian blind deconvolution with general sparse image priors. In *Proceedings of European conference on computer vision (ECCV)* (pp. 341–355).

Bansal, A., Castillo, C., Ranjan, R., & Chellappa, R. (2017). The dos and donts for CNN-based face verification. In *IEEE proceedings of international conference on computer vision (ICCV)*.

Bulat, A., & Tzimiropoulos, G. (2017). How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *IEEE proceedings of international conference on computer vision (ICCV)*.

Cao, Q., Lin, L., Shi, Y., Liang, X., & Li, G. (2017). Attention-aware face hallucination via deep reinforcement learning. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Chakrabarti, A. (2016). A neural approach to blind motion deblurring. In *Proceedings of European conference on computer vision (ECCV)* (pp. 221–235). Springer.

Chen, W., Fu, Z., Yang, D., & Deng, J. (2016). Single-image depth perception in the wild. In *Advances in neural information processing systems* (pp. 730–738).

Cho, C. M., & Don, H. S. (1991). Blur identification and image restoration using a multilayer neural network. In *1991 IEEE international joint conference on neural networks*, 1991 (pp. 2558–2563).

Chrysos, G. G., & Zafeiriou, S. (2017a). Deep face deblurring. In *CVPR Workshops* (pp. 2015–2024).

Chrysos, G. G., Antonakos, E., Snape, P., Asthana, A., & Zafeiriou, S. (2018). A comprehensive performance evaluation of deformable face tracking "in-the-wild". *International Journal of Computer Vision*, *126*(2–4), 198–232.

Chrysos, G. G., Antonakos, E., Zafeiriou, S., & Snape, P. (2015). Offline deformable face tracking in arbitrary videos. In *IEEE proceedings of international conference on computer vision, 300 videos in the wild (300-VW): Facial landmark tracking in-the-wild challenge & workshop (ICCV-W)* (pp. 1–9).

Chrysos, G. G., & Zafeiriou, S. (2017b). Deep face deblurring. In *IEEE proceedings of international conference on computer vision and pattern recognition workshops (CVPR'W)*.

Chung, J. S., & Zisserman, A. (2016). Lip reading in the wild. In *Asian conference on computer vision (ACCV)* (pp. 87–103).

Denton, E. L., Chintala, S., Fergus, R., et al. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems* (pp. 1486–1494).

Ding, C., & Tao, D. (2018). Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 1002–1014.

Dosovitskiy, A., & Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems* (pp. 658–666).

Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., et al. (2016) Adversarially learned inference. arXiv:1606.00704.

Gong, D., Yang, J., Liu, L., Zhang, Y., Reid, I., Shen, C., et al. (2017). From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014) Generative adversarial nets. In *Advances in neural information processing systems*.

Gopalan, R., Taheri, S., Turaga, P., & Chellappa, R. (2012). A blur-robust descriptor with applications to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *34*(6), 1220–1226.

Gross, R., Matthews, I., Cohn, J., Kanade, T., & Baker, S. (2010). Multi-pie. *Image and Vision Computing*, *28*(5), 807–813.

Grossberg, M. D., & Nayar, S. K. (2003). Determining the camera response from images: What is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *25*(11), 1455–1467.

Guo, Y., Zhang, L., Hu, Y., He, X., & Gao, J. (2016). Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Proceedings of European conference on computer vision (ECCV)* (pp. 87–102).

Hacohen, Y., Shechtman, E., & Lischinski, D. (2013). Deblurring by example using dense correspondence. In *IEEE proceedings of international conference on computer vision (ICCV)* (pp. 2384–2391).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Hradiš, M., Kotera, J., Zemcík, P., & Šroubek, F. (2015). Convolutional neural networks for direct text deblurring. In *Proceedings of British machine vision conference (BMVC)*.

Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., & Belongie, S. (2017). Stacked generative adversarial networks. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (Vol. 2, p. 4).

Huang, Y., Yao, H., Zhao, S., & Zhang, Y. (2015). Efficient face image deblurring via robust face salient landmark detection. In *Pacific rim conference on multimedia* (pp. 13–22).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Kim, T. H., Nah, S., & Lee, K. M. (2018). Dynamic video deblurring using a locally adaptive linear blur model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(10), 2374–2387.

Krishnan, D., Tay, T., & Fergus, R. (2011). Blind deconvolution using a normalized sparsity measure. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 233–240).

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., & Matas, J. (2018). Deblurgan: Blind motion deblurring using conditional adversarial networks. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Lai, W. S., Huang, J. B., Hu, Z., Ahuja, N., & Yang, M. H. (2016). A comparative study for single image blind deblurring. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Lee, C. H., Zhang, K., Lee, H. C., Cheng, C. W., & Hsu, W. (2018). Attribute augmented convolutional neural network for face hallucination. In *IEEE proceedings of international conference on computer vision and pattern recognition workshops (CVPR'W)* (pp. 721–729).

Levin, A., Weiss, Y., Durand, F., & Freeman, W. T. (2009). Understanding and evaluating blind deconvolution algorithms. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 1964–1971).

Liao, S., Jain, A. K., & Li, S. Z. (2016). A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *38*(2), 211–223.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Smolley, S. P. (2017). Least squares generative adversarial networks. In *IEEE proceedings of international conference on computer vision (ICCV)* (pp. 2813–2821). IEEE

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv:1411.1784.

Nah, S., Kim, T. H., & Lee, K. M. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *Proceedings of European conference on computer vision (ECCV)* (pp. 483–499).

Nguyen, D. T., Cho, S. R., Pham, T. D., & Park, K. R. (2015). Human age estimation method robust to camera sensor and/or face movement. *Sensors*, *15*(9), 21898–21930.

Nishiyama, M., Hadid, A., Takeshima, H., Shotton, J., Kozakaya, T., & Yamaguchi, O. (2011). Facial deblur inference using subspace analysis for recognition of blurred faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *33*(4), 838–845.

Noroozi, M., Chandramouli, P., & Favaro, P. (2017). Motion deblurring in the wild. In *German conference on pattern recognition* (pp. 65–77).

Nowozin, S., Cseke, B., & Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems* (pp. 271–279).

Odena, A. (2016). Semi-supervised learning with generative adversarial networks. In *International conference on machine learning workshops*.

Pan, J., Hu, Z., Su, Z., & Yang, M. H. (2014a). Deblurring face images with exemplars. In *Proceedings of European conference on computer vision (ECCV)* (pp. 47–62).

Pan, J., Hu, Z., Su, Z., & Yang, M. H. (2014b). Deblurring text images via l0-regularized intensity and gradient prior. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 2901–2908).

Pan, J., Sun, D., Pfister, H., & Yang, M. H. (2016). Blind image deblurring using dark channel prior. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)*.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 2536–2544).

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434.

Ramakrishnan, S., Pachori, S., Gangopadhyay, A., & Raman, S. (2017). Deep generative filter for motion deblurring. In *IEEE proceedings of international conference on computer vision (ICCV)*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241).

Rothe, R., Timofte, R., & Van Gool, L. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, *126*(2–4), 144–157.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234–2242).

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 815–823).

Schuler, C. J., Hirsch, M., Harmeling, S., & Schölkopf, B. (2016). Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *38*(7), 1439–1451.

Shen, J., Zafeiriou, S., Chrysos, G., Kossaifi, J., Tzimiropoulos, G., & Pantic, M. (2015). The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *IEEE proceedings of international conference on computer vision, 300 videos in the wild (300-VW): Facial landmark tracking in-the-wild challenge & workshop (ICCV-W)*.

Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., & Wang, O. (2017). Deep video deblurring for hand-held cameras. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 1279–1288).

Sun, J., Cao, W., Xu, Z., & Ponce, J. (2015). Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 769–777).

Tai, Y. W., Chen, X., Kim, S., Kim, S. J., Li, F., Yang, J., et al. (2013). Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, *35*(10), 2498–2512.

Tekalp, A., Kaufman, H., & Woods, J. (1986). Identification of image and blur parameters for the restoration of noncausal blurs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *34*(4), 963–972.

Tran, L., Yin, X., & Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (Vol. 3, p. 7).

Tulyakov, S., Liu, M. Y., Yang, X., & Kautz, J. (2017). Mocogan: Decomposing motion and content for video generation. arXiv:1707.04993.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions in Image Processing (TIP)*, *13*(4), 600–612.

Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., et al. (2017) Iarpa janus benchmark-b face dataset. In *IEEE proceedings of international conference on computer vision and pattern recognition workshops (CVPR'W)*.

Wieschollek, P., Hirsch, M., Schölkopf, B., & Lensch, H. P. (2017). Learning blind motion deblurring. In *IEEE proceedings of international conference on computer vision (ICCV)* (Vol. 4).

Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 529–534).

Xu, L., Zheng, S., & Jia, J. (2013). Unnatural l0 sparse representation for natural image deblurring. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 1107–1114).

Xu, X., Sun, D., Pan, J., Zhang, Y., Pfister, H., & Yang, M. H. (2017). Learning to super-resolve blurry face and text images. In *IEEE proceedings of international conference on computer vision (ICCV)* (pp. 251–260).

Yang, J., et al. (2017). Stacked hourglass network for robust facial landmark localisation. In *IEEE proceedings of international conference on computer vision and pattern recognition workshops (CVPR'W)*.

Yoo, D., Kim, N., Park, S., Paek, A. S., & Kweon, I. S. (2016). Pixel-level domain transfer. In *Proceedings of European conference on computer vision (ECCV)* (pp. 517–532).

Zafeiriou, S., Trigeorgis, G., Chrysos, G., Deng, J., & Shen, J. (2017). The menpo facial landmark localisation challenge: A step towards the solution. In *IEEE proceedings of international conference on computer vision and pattern recognition workshops (CVPR'W)*.

Zhang, H., Wipf, D., & Zhang, Y. (2013). Multi-image blind deblurring using a coupled adaptive sparse prior. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 1051–1058).

Zhu, J. Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., et al. (2017). Toward multimodal image-to-image translation. In *Advances in neural information processing systems* (pp. 465–476).

Zhu, S., Li, C., Loy, C. C., & Tang, X. (2015). Face alignment by coarse-to-fine shape searching. In *IEEE proceedings of international conference on computer vision and pattern recognition (CVPR)* (pp. 4998–5006).

Zhu, S., Liu, S., Loy, C. C., & Tang, X. (2016). Deep cascaded bi-network for face hallucination. In *Proceedings of European conference on computer vision (ECCV)* (pp. 614–630).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.