

Global-Local Face Upsampling Network

Oncel Tuzel Yuichi Taguchi John R. Hershey

Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA

Abstract. Face hallucination, which is the task of generating a high-resolution face image from a low-resolution input image, is a well-studied problem that is useful in widespread application areas. Face hallucination is particularly challenging when the input face resolution is very low (e.g., 10×12 pixels) and/or the image is captured in an uncontrolled setting with large pose and illumination variations. In this paper, we revisit the algorithm introduced in [1] and present a deep interpretation of this framework that achieves state-of-the-art under such challenging scenarios. In our deep network architecture the global and local constraints that define a face can be efficiently modeled and learned end-to-end using training data. Conceptually our network design can be partitioned into two sub-networks: the first one implements the holistic face reconstruction according to global constraints, and the second one enhances face-specific details and enforces local patch statistics. We optimize the deep network using a new loss function for super-resolution that combines reconstruction error with a learned face quality measure in adversarial setting, producing improved visual results. We conduct extensive experiments in both controlled and uncontrolled setups and show that our algorithm improves the state of the art both numerically and visually.

1 Introduction

Face has been one of the main targets for image enhancement tasks. In particular, upsampling a low-resolution face image to a high-resolution one has been an important problem, called by its own name of face hallucination [2].

The problem is formulated as follows. Given a low-resolution $N_L = n \times m$ face image I_L , our goal is to obtain a photorealistic high-resolution $N_H = (dn) \times (dm)$ face image I_H whose down-sampled version is equal to I_L , where d is the upsampling factor. The relation can be written as

$$\mathbf{x}_L = \mathbf{K} \mathbf{x}_H, \quad (1)$$

where \mathbf{x}_L and \mathbf{x}_H are the low and high-resolution images stacked into column vectors and \mathbf{K} is an $N_L \times N_H$ sparse matrix implementing low-pass filtering and down-sampling. To invert this largely (d^2 -times) under-determined linear system and recover the high-resolution image, additional constraints are needed.

We approximate the solution of this linear inverse problem using a deep neural network where facial constraints are explicitly modeled and learned using training data. Our considerations for the proposed face upsampling network

are inspired by the face hallucination work of Liu et al. [1]. Similar to [1], we utilize the following three constraints to regularize the under-determined problem. (1) Global constraint: The reconstructed high-resolution face image should satisfy holistic constraints such as shape, pose, and symmetry, and should include detailed characteristic facial features such as eyes and nose. (2) Local constraint: Statistics of the reconstructed local image regions should match that of high-resolution face image patches (e.g., smooth regions with sharp boundaries), and should include face-specific details. (3) Data constraint: The reconstruction should be consistent with the observed low-resolution image and satisfy Eq. (1).

Liu et al. [1] used a two-step approach according to these constraints. First a global face reconstruction is acquired using an eigenface model, which is a linear projection operation. In the second step details of the reconstructed face are enhanced by non-parametric patch transfer from a training set where consistency across neighboring patches is enforced through a Markov random field. This method produces high-quality results when the face images are near frontal, well aligned, and lighting conditions are controlled. However, when these assumptions are violated, the simple linear eigenface model fails to produce satisfactory global reconstruction. In addition, the patch transfer does not scale well with large datasets due to the nearest-neighbor (NN) search.

In this paper, we present a deep network architecture that resembles Liu et al.’s framework [1] but solves the aforementioned problems for accurate and efficient face hallucination. Our network consists of the two sub-networks: the first one implements holistic face reconstruction according to the global constraints, and the second one enhances face-specific details and enforces local patch statistics. However, they are learned jointly using a large amount of training data, providing an optimized structure for upsampling. Moreover, the feed-forward operation provides computational efficiency in the test time. In extensive experiments using two benchmark datasets captured under controlled and uncontrolled setups, we show that our algorithm outperforms the state-of-the-art algorithms.

1.1 Contributions

Our main contributions can be summarized as follows: (1) We present a deep interpretation of the global-local face hallucination framework [1]. (2) We design a deep network architecture that replaces the original two-step approach with an end-to-end learning and feed-forward operation, improving both the accuracy and speed. (3) We learn the deep network by minimizing a combination of reconstruction error and a learned face quality loss in adversarial setting, which produces high resolution images with improved visual quality. (4) We conduct extensive comparisons with the state-of-the-art algorithms and demonstrate that our algorithm outperforms them both qualitatively and quantitatively.

1.2 Related Work

Face hallucination is the single-image super-resolution (SR) problem specific to face images. Single-image SR algorithms developed for generic images share

the same formulation in (1). To invert the under-determined system, local constraints are enforced as priors based on image statistics [3,4,5] and exemplar patches [6,7,8]. Global constraints are typically not available for the generic SR problem, which limits the plausible upsampling factor. Yang et al.’s recent study [9] showed that $4\times$ upsampling results in the lower bound of the human perceptual scores.

Liu et al. [1] used a global constraint for face hallucination based on eigenfaces [10], and proposed a two-step approach where the initial global reconstruction is improved by local non-parametric patch transfer [6]. As described above, the simple eigenface model has a difficulty when the datasets include large pose and illumination variations, and their local refinement process is computationally expensive due to the NN patch search. Ma et al. [11] assumed that training and test images are precisely aligned and searched the NN patches of a target pixel in the test image only at the specific pixel location in the training images. Using the location-specific patches provides global constraints implicitly, as long as the images are well-aligned. Yang et al. [12] partitioned a face image into three groups of facial components, contours, and smooth regions based on a facial landmark detection [13]. They used the NN search for each of the facial components with the training images, while for contours and smooth regions edge-based statistics and NN patch search were used. The result was generated by integrating gradient maps from the three groups and imposing them on the high-resolution image. Their method relies on the facial landmark detection and thus the result degrades for low-resolution input images where the landmark localization is typically inaccurate.

In the past several years, the success of deep learning methods has revolutionized the computer vision field from image classification [14,15] and object detection [16] to face recognition [17], segmentation [18], and video event detection [19]. These methods have also been replacing highly optimized hand-designed algorithms in low-level vision tasks such as image denoising [20,21,22], image enhancement [23,24], and SR [25,26,27]. Dong et al. [25] proposed super-resolution convolutional neural network (SRCNN) for generic SR. They interpreted it as a deep network version of the conventional sparse coding methods [8]. SRCNN provides the state-of-the-art performance for generic SR, but not for face-specific SR as we compare in experiments. More recently, Wang et al. [26] proposed an improved deep model for generic SR that also takes into account self similarities. Zhou et al. [27] presented bi-channel convolutional neural network (BCCNN) for face-specific SR. They used a convolutional neural network architecture whose output was blended with the bicubic upsampled image using a weighting factor which is also predicted from the network. The last layer of this network linearly combines high-resolution basis images, which corresponds to a global face reconstruction and smooths out the person-specific details.

Basic building blocks of our algorithm are well known neural network architectures such as encoder [28,29,30], convolutional [31], and deconvolutional [32,33] neural nets. Our architectural design enables effective learning of global and local

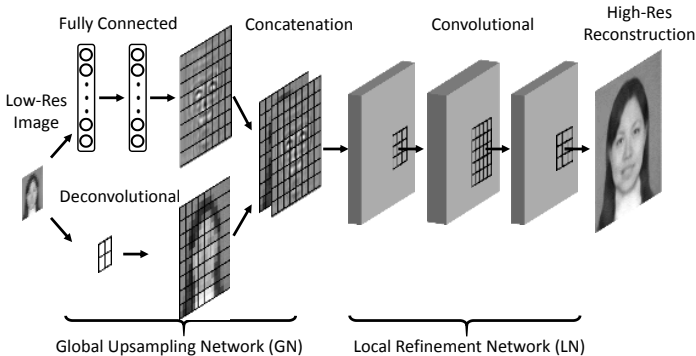


Fig. 1. Our global-local face upsampling network (GLN).

constraints that are important for face upsampling task using these well-known building blocks.

Recently, generative adversarial networks (GANs) [34] have been proposed as an alternative to learn deep generative models. In GAN framework, a generative network learns to generate samples from a given data distribution, while simultaneously a discriminative network learns to identify the samples that are generated from this network. Since then, GANs have been successfully used for image [34,35], scene [36], and sequence synthesis [37] tasks. In this paper, we use GAN framework to learn a discriminative network which evaluates face quality, while at the same time optimizing the face super-resolution network according to the learned quality measure.

2 Method Overview

Figure 1 shows an overview of our global-local face upsampling network (GLN). Our network consists of two sub-networks, referred to as Global Upsampling Network (GN) and Local Enhancement Network (LN), which model the global and local constraints for face hallucination. The operations performed by the two sub-networks are conceptually similar to Liu et al.’s global reconstruction followed by local enhancement [1]. However, by jointly modeling and learning global and local constraints using a deep architecture, our method provides higher accuracy while being more efficient in the test time due to the feed-forward processing, as detailed below.

Holistic face reconstruction according to global constraints is achieved using GN, which is a two stream neural network running in parallel. The first stream implements a simple interpolation-based upsampling of the low-resolution face using a deconvolutional network, producing a smooth image without details. The second stream produces the high frequency characteristic facial details, such as eyes and nose, using a fully connected neural network. Hidden layers of this encoder network [28,29] build a global representation of high-resolution face

images that can be inferred from the low-resolution input. Compared to the linear eigenface model of [1], the multi-layer nonlinear embedding and reconstruction used in our framework enables more effective encoding of characteristic facial features, in addition to variations such as alignment, face pose, and illumination. We concatenate the two streams generated by GN to be processed by LN.

The local constraints are modeled in LN using a fully convolutional neural network, which implements a shift-invariant nonlinear filter. This network enhances the face-specific local details by fusing the smooth and detail layers produced by the GN. Even though the convolutional filters are relatively small (5×5 or 7×7), by stacking many filters, the receptive field of the network becomes quite large (43 pixels in 8 layer network). The large receptive field enables resolving the ambiguity (e.g., eye region vs. mouth region) and the deep architecture has enough capacity to apply necessary filtering operation to a given region. Compared to the non-parametric detail transfer of [1], our structure is very efficient and produces higher quality enhancement.

Although we do not explicitly model the data constraint within the network, by training the network using a large amount of training data, the network learns to produce high-resolution face images that are consistent with the low-resolution images according to Eq. (1). One could enforce the data constraint by using the back-projection (BP) algorithm [38] in a postprocessing step, which is a common approach used in many upsampling schemes. However, in our experiments, we found that the results directly obtained from our network and those obtained after the BP postprocessing were indistinguishable both qualitatively and quantitatively. Thus we did not use such a postprocessing.

3 Global-Local Upsampling Network (GLN)

This section presents the details of our deep network architecture that is used to upsample the low-resolution face images. Our network structure is designed for very low-resolution input face images. We consider two upsampling factors: (1) $4\times$ upsampling where 32×32 input face image is mapped to 128×128 resolution; (2) $8\times$ upsampling where 16×16 input face image is mapped to 128×128 resolution. We have two different network configurations for the $4\times$ and $8\times$ cases. We assume that the low-resolution face images are roughly aligned.

3.1 Global Upsampling Network (GN)

The structure of the GN is summarized in Table 1. GN is a two stream network running in parallel. The image upsampling stream maps the input face image to a high-resolution face image using linear interpolation. In our network, we implemented the image upsampling stream using a deconvolution layer [32,33]. We initialize the interpolation weights using bilinear matrix but allow the weights to change during training.

The global detail generation stream is implemented as a fully connected encoder network with 3 hidden layers. We use rectified linear unit (ReLU) af-

Table 1. Global upsampling network architecture. First and second columns are the image upsampling and global detail generation streams for $4\times$ image upsampling. Third and fourth columns are the corresponding streams for the $8\times$ image upsampling.

$4\times$ GN		$8\times$ GN	
deconv4	fc-512	deconv8	fc-256
	fc-256		fc-256
	fc-512		fc-256
	fc-(128×128)		fc-(128×128)
concatenation		concatenation	

Table 2. Local refinement network architecture. The first number after conv. indicates the kernel size whereas the second number is the number of filters. We analyzed three convolution architectures with 4, 6, and 8 layers. The same architectures are used for both $4\times$ and $8\times$ upsampling tasks.

4 Layer LN (LN4)	6 Layer LN (LN6)	8 Layer LN (LN8)
conv5-16	conv5-16	conv5-16
conv7-64	conv7-32	conv7-32
conv5-16	conv7-64	conv7-64
	conv7-32	conv7-64
	conv5-16	conv7-64
		conv7-32
		conv5-16
conv5-1	conv5-1	conv5-1

ter every linear map except for the last layer which generates the 128×128 -dimensional upsampled global detail. In our encoder network the code layer is 256-dimensional, both for $4\times$ and $8\times$ upsampling networks. This is mainly dictated by the limited amount of training data where larger latent spaces for the global detail tend to overfit. Finally we concatenate the outputs of the image upsampling stream and the global detail generation stream, and form a $2\times 128\times 128$ tensor to be processed by the LN.

Figure 2 shows a typical output of the $8\times$ GN. Even though we allow the image upsampling stream’s weights to change during training, the weights tend not to change much and the network implements a smooth upsampling (Figure 2(a)). The output of the global detail generation stream (Figure 2(b)) encodes high frequency details and more difficult to interpret. The pattern is more visible around the characteristic facial features such as eyes, nose and mouth.

3.2 Local Refinement Network (LN)

The structure of the LN is summarized in Table 2. These structures are identical for $4\times$ and $8\times$ upsampling tasks. We analyzed three fully convolutional neural network architectures with different numbers of layers (LN4, LN6, and LN8). Before each convolution operation the image is padded with the ceiling of the half filter size so that the output image dimension is same as the input dimension. After every convolutional layer we apply ReLU except the last layer which

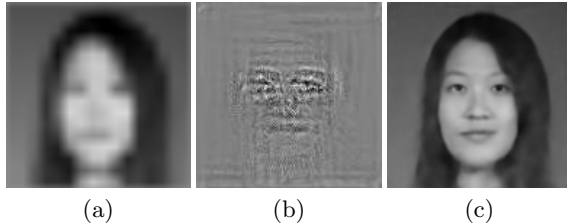


Fig. 2. Example output of the (a) image upsampling and (b) global detail generation streams of the $8\times$ GN. (c) Final upsampling result after LN.

constructs the final upsampled image. We do not perform pooling and the stride is 1, therefore this network learns a very large shift-invariant nonlinear filter. As shown in Figure 2(c), the LN enhances the face specific local details by fusing the smooth and detail layers produced by the GN (see eyes and nose). In addition, the reconstructed image’s local statistics match that of high-resolution face image patch statistics (e.g., smooth cheek region and sharp face boundaries).

3.3 Training

We conducted two stage training procedure. In the first stage, we train the network by minimizing a reconstruction error criteria. During the (optional) second stage, we fine-tune the network by minimizing a weighted combination of reconstruction error and a learned face quality loss function.

Training for reconstruction: We minimize the mean-squared loss between the ground truth high-resolution images and the reconstructed images to learn the network parameters. Let $\{(\mathbf{x}_L^i, \mathbf{x}_H^i)\}_{i=1,\dots,n}$ be the set of n low-resolution and high-resolution training image pairs. The loss is given by

$$L_{MS} = \frac{1}{n} \sum_{i=1}^n \|G(\mathbf{x}_L^i) - \mathbf{x}_H^i\|^2, \quad (2)$$

where $G(\cdot)$ is the GLN function. The network is trained using stochastic gradient descent (SGD) on mini-batches of 5 images, with fixed learning rate schedule 10^{-8} , and momentum 0.9.

Adversarial fine-tuning: Mean-squared loss function prefers blurry reconstructions at the regions with high ambiguity, such as edges. Here we complement this loss function with a learned loss function, which is tuned to measure the quality of reconstruction by discriminating reconstructed images from the true high resolution images. We use a variant of the generative adversarial network framework proposed by Goodfellow et al. [34] to learn the discriminative loss function in conjunction with the GLN parameters.

In our framework, the discriminative network $D(\cdot)$ detects the images reconstructed by the GLN. It is trained to maximize the output probability when the input is reconstructed by the GLN and minimize the output probability when

the input is a true high resolution face image using the loss function:

$$L_D = -\frac{1}{2n} \sum_{i=1}^n (\log(1 - D(\mathbf{x}_H^i)) + \log(D(G(\mathbf{x}_L^i)))). \quad (3)$$

The GLN is trained both to minimize the reconstruction error L_{MS} , and to confuse the discriminative network by minimizing the output probability of the discriminative network on the input reconstructed images. This is achieved by using the combined mean-squared and adversarial loss function:

$$L_G = L_{MS} - \lambda \frac{1}{n} \sum_{i=1}^n \log(1 - D(G(\mathbf{x}_L^i))), \quad (4)$$

where λ is a weighting factor between the two loss terms.

Similar to [34], we alternate between minimizing L_D with respect to parameters of the discriminative network, while keeping the GLN parameters fixed, and minimizing L_G with respect to parameters of the GLN, while keeping discriminative network parameters fixed. We used 10 SGD iterations for discriminative network and 50 SGD iterations for the GLN during alternations. We switched 10000 times between optimizing discriminative network and the GLN.

The discriminative network was implemented as a convolutional neural network with four layers: (1) conv5-16, ReLU, MaxPool 2x2, (2) conv5-16, ReLU, MaxPool 2x2, (3) fc-50, ReLU, (4) fc-2. We started adversarial fine-tuning using the network trained for reconstruction only. The weighting factor λ was set such that the initial adversarial loss was equal to 1/10 of the mean-squared loss. We used open source CAFFE library [39] to implement the networks.

4 Experiments

We designed two sets of experiments under controlled and uncontrolled (in the wild) settings. The controlled setting was conducted using Face Recognition Grand Challenge (FRGC) dataset [40], where frontal face images were taken in a studio setting under two lighting conditions with only two facial expressions (smiling and neutral). We used a total of 22,149 images, where 20,000 images were used for training and 2,149 for testing. We used a variant of the supervised descent face alignment algorithm [41] to detect facial landmarks. We then applied similarity transformation to the input face images (translation, rotation and uniform scaling) to approximately align detected eye and mouth center locations to a set of fixed points.

The uncontrolled setting was conducted using an aligned version of the Labeled Faces in the Wild dataset [42], called Labeled Faces in the Wild-a (LFW-a) [43]. This dataset is intended for studying unconstrained face recognition problem and includes face images with various illumination, poses, and expressions. It contains 13,233 face images from 1,680 people, where we used 12,000 for training and 1,233 for testing. In both settings we kept the identities of the

people in the training and testing sets disjoint. Note that the alignment is quite noisy for both settings, particularly for the LFW-a dataset.

We evaluated two upsampling factors of $4\times$ and $8\times$. Faces occupied approximately 20×24 pixel area in the $4\times$ upsampling case and 10×12 pixel area in the $8\times$ upsampling case. The low-resolution images were generated using the procedure described in [9], filtering the high-resolution images with a Gaussian blur kernel σ followed by down-sampling. We used $\sigma = 1.2$ for $4\times$ down-sampling as suggested, and $\sigma = 2.4$ for $8\times$ down-sampling.

4.1 Comparisons

We compared our method with the state-of-the-art generic and face-specific SR algorithms. As the generic SR algorithms, we used Kim and Kwon’s algorithm (KK) [5] and the SRCNN algorithm [25], which are among the top performers in the evaluations reported in [9,25]. We used the implementations and pre-trained models (the 9-5-5 network for SRCNN) available on the authors’ websites¹. Since their algorithms were trained only up to $4\times$ upsampling factor, we performed $4\times$ upsampling followed by $2\times$ upsampling to generate $8\times$ upsampling results. As the face-specific SR algorithms, we used (1) Liu et al.’s algorithm (LSF) [1]; (2) Ma et al.’s (MZQ) [11]; (3) Yang et al.’s (YLY) [12]; and (4) Zhou et al.’s BCCNN [27]. We used Yang et al.’s implementations [12] for LSF, MZQ, and YLY, and our own implementation for BCCNN. All the face-specific SR algorithms were trained using the same sets of training images as ours. Our results presented in this section, both qualitatively and quantitatively, were obtained by minimizing the reconstruction error only. Note that after adversarial fine-tuning, the visual quality improves while quantitative results change marginally.

Qualitative Results: Figures 3 and 4 respectively show $4\times$ and $8\times$ upsampling results. Note that our input resolution is low (32×32 and 16×16 pixels), as it can be observed in the NN interpolation results. The bicubic interpolation results are blurry as expected. The generic SR algorithms (KK and SRCNN) sharpen the images by recovering some high-frequency components, but they do not reproduce face-specific features. The face-specific SR algorithms recover such facial features. In particular, we observed that MZQ and BCCNN produce visually pleasing results in our settings where the input resolution is low. However, since MZQ assumes precise alignments between the test and training images, the results degrade for the LFW-a dataset including larger alignment errors. Moreover, finding similar patches is hard for the case of $8\times$ upsampling, leading to inaccurate hallucination results for MZQ. BCCNN’s network structure mainly performs global reconstruction by computing a weighted average of several high-resolution basis images and the bicubic upsampled image in the last layer of the network. This structure is similar to the GN-Only stream of our network, which is detailed in Section 4.2, and provides high resolution images with global details (e.g., symmetry and characteristic facial features) while suffering

¹ We also retrained the SRCNN using our datasets. These models had almost identical PSNR (0.02dB better for both $4\times$ and $8\times$ upsampling) to the pre-trained models.



Fig. 3. Qualitative comparisons of $4\times$ upsampling results for FRGC (top 3 rows) and LFW-a (bottom 3 rows) datasets. YLY does not produce results when the face landmarks cannot be detected in the low-resolution input. KK and SRCNN produced visually similar results, so the results obtained with KK are omitted.

from loss of person-specific details. YLY, which relies on the facial landmark detection [13], cannot produce results when the landmark detection fails; even if the detection is successful, the landmark localization accuracy is typically low for the low-resolution input, resulting in the facial features recovered at incorrect locations. We note that the resolution of our input images is lower than the images used in their original work. Our GLN produces globally consistent and locally sharp images, which are the closest to the ground truth (GT).

Quantitative Results: In addition to the standard peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), we computed the information fidelity criterion (IFC) [44], weighted PSNR (WPSNR), and noise quality mea-

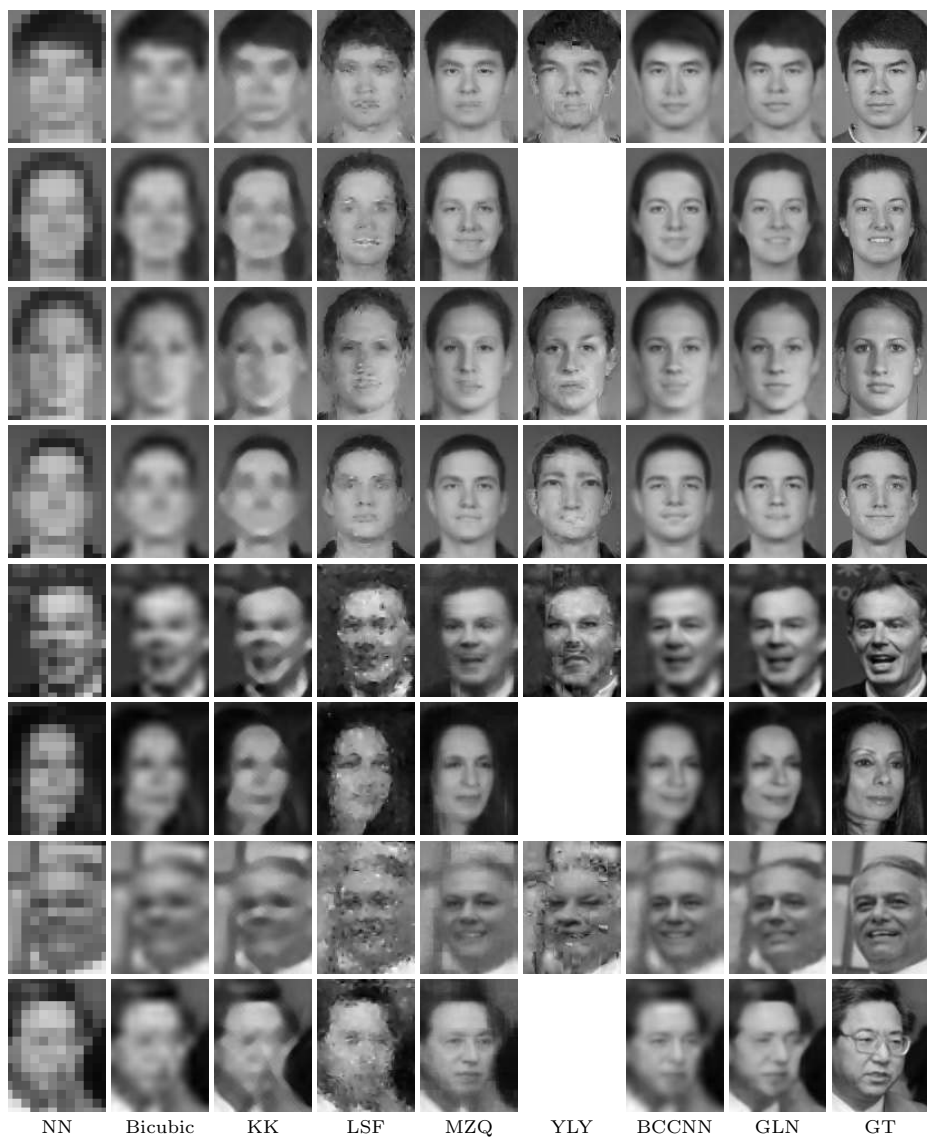


Fig. 4. Qualitative comparisons of $8\times$ upsampling results for FRGC (top 4 rows) and LFW-a (bottom 4 rows) datasets. YLY does not produce results when the face landmarks cannot be detected in the low-resolution input. KK and SRCNN produced visually similar results, so the results obtained with SRCNN are omitted.

Table 3. Quantitative comparisons for FRGC 4× and 8× upsampling.

Method	FRGC 4×					FRGC 8×				
	PSNR	SSIM	IFC	WPSNR	NQM	PSNR	SSIM	IFC	WPSNR	NQM
NN	25.39	0.694	1.25	35.17	6.94	21.95	0.466	0.39	28.79	3.49
Bicubic	27.39	0.797	1.84	36.04	9.09	23.71	0.617	0.84	30.27	5.29
KK [5]	28.06	0.825	2.10	37.34	9.72	24.11	0.631	0.89	30.85	5.74
SRCNN [25]	28.19	0.829	2.10	37.47	9.91	24.18	0.641	0.91	30.89	5.84
LSF [1]	25.64	0.723	1.28	33.56	7.42	23.21	0.614	0.76	30.35	4.90
MZQ [11]	29.22	0.857	2.22	38.67	11.21	25.81	0.752	1.34	33.17	7.67
YLY [12]	26.80	0.791	1.79	35.67	8.55	22.57	0.602	0.83	29.45	4.21
BCCNN [27]	28.67	0.842	2.13	37.98	10.63	25.70	0.749	1.34	32.90	7.60
GLN	30.34	0.884	2.66	40.94	12.37	26.75	0.787	1.56	34.37	8.60

Table 4. Quantitative comparisons for LFW-a 4× and 8× upsampling.

Method	LFW-a 4×					LFW-a 8×				
	PSNR	SSIM	IFC	WPSNR	NQM	PSNR	SSIM	IFC	WPSNR	NQM
NN	24.16	0.687	1.23	33.55	7.99	20.56	0.441	0.38	27.22	4.43
Bicubic	26.62	0.796	1.84	34.61	10.72	22.16	0.575	0.77	28.38	6.13
KK [5]	27.53	0.826	2.07	36.04	11.55	22.75	0.603	0.84	29.14	6.70
SRCNN [25]	27.55	0.827	2.03	36.12	11.55	22.74	0.607	0.83	29.08	6.66
LSF [1]	22.98	0.601	0.80	29.95	7.01	20.02	0.434	0.41	26.44	4.01
MZQ [11]	26.36	0.784	1.61	34.10	10.69	22.64	0.621	0.83	29.11	6.89
YLY [12]	25.52	0.750	1.54	33.62	9.51	20.80	0.500	0.59	27.30	4.82
BCCNN [27]	26.63	0.800	1.81	34.57	10.90	22.72	0.627	0.90	29.08	6.89
GLN	28.82	0.863	2.35	37.80	13.01	24.07	0.688	1.12	30.75	8.19

sure (NQM) [45] for quantitative evaluations as suggested in [9]². Tables 3 and 4 show the results for different datasets and upsampling factors, demonstrating that our method provides the best performance in all the different metrics.

Run Time: Table 5 compares the average run time for processing a single test image for FRGC 4× upsampling. LSF and YLY are computationally expensive because of the NN search for each of the training images, requiring the run time linear to the number of training images. MZQ’s run time also increases linearly with the number of training images, but it is faster than LSF and YLY since the NN patches are searched only at the specific pixel location. The other algorithms have the run time independent of the number of training images. The algorithms based on feed-forward neural networks, including ours, achieve efficient processing in the test time³. The run time of our algorithm was measured using Intel i7 CPU (single core implementation) and NVIDIA 780 GTX GPU.

² The multi-scale SSIM (MSSSIM) [46] with the default number of scales was not applicable due to the small output resolution (128 × 128 pixels).

³ The run time of SRCNN was measured using the Matlab implementation available on the authors’ website [25]. It would be the fastest with a C++/GPU implementation since their network has fewer arithmetic operations than BCCNN’s and ours.

Table 5. Average run time for FRGC 4× upsampling.

Method	KK	SRCNN	LSF	MZQ	YLY	BCCNN	GLN
Run Time (seconds)	3.9	1.7	940	1.1	2900	0.005 (GPU) / 0.04 (CPU)	0.023 (GPU) / 0.91 (CPU)



(a) GN-Only (b) LN-Only (c) GLN (d) GT

Fig. 5. Qualitative comparison of sub-modules of our network. First and second row show 4× and 8× upsampling results respectively. See text for details.

4.2 Analysis of the Network Architecture

Here we analyze the sub-modules of our network. Figure 5 shows 4× (top row) and 8× (bottom row) upsampling results using the sub-modules—global upsampling network and local refinement network. These networks are slightly different from the original networks such that (1) we train global detail generation stream (fully connected) of GN to directly produce the high-resolution image, which we call GN-Only; (2) we train LN8 network using only bilinear upsampled low-resolution image as the input, which we call LN-Only. As shown in Figure 5(a), GN-Only produces high quality global details such as symmetry and characteristic facial features, while smoothing out uncommon features (e.g., details on the cheeks and highlights) and producing high frequency artifacts that are not consistent with face patch statistics. Figure 5(b) shows an example of LN-Only result where global details such as characteristic features and symmetry are not preserved (especially for 8×), but the local patch statistics are consistent with face patch statistics (e.g., sharp edges) and local details are preserved. The results of 4× upsampling is significantly better than 8× upsampling using LN-Only where resolving patch level ambiguities is easier. The GLN successfully utilizes both global and local cues and produces significantly higher quality results than both sub-modules (Figure 5(c)).

Table 6 shows quantitative comparisons using the sub-modules of our network. These results are consistent with the qualitative comparisons and show that LN-Only produces high quality 4× upsampling and the results degrade at 8×. We also compare the variants of GLN with different numbers of convolutional layers as shown in Table 2. The results show that the improvement is significant from 4 to 6 convolutional layers but it starts saturating afterwards.

Table 6. Results using the sub-modules of our network and different configurations (PSNR / SSIM).

Method	FRGC 4×	FRGC 8×
GN-Only	27.36 / 0.801	25.78 / 0.752
LN-Only	29.92 / 0.874	25.97 / 0.743
GLN4	29.59 / 0.867	26.35 / 0.765
GLN6	29.94 / 0.875	26.61 / 0.775
GLN8	30.34 / 0.884	26.75 / 0.787

**Fig. 6.** Qualitative comparisons of FRGC 8× (left 3 columns) and 4× (right 3 columns) upsampling results using the reconstruction cost only (top row) and after the adversarial fine-tuning (bottom row).

4.3 Adversarial Fine-Tuning

Figure 6 compares the results of training the GLN using reconstruction cost only (top row) with fine-tuning the GLN using the combined adversarial loss function (bottom row), as explained in Section 3.3. The adversarial fine-tuning further improves the visual quality of the generated high-resolution face images where the images are sharper and have more characteristic details. However, this step marginally reduces the PSNR score— $0.01dB$ and $0.25dB$ for 4× and 8× respectively. This is expected since the additional adversarial loss does not use the identity of the faces but only evaluates the quality of the generated face images.

We analyze the effect of the weighting factor, λ , between the mean-squared loss and the adversarial loss, equation (4), on the super-resolution result. The 4× upsampling results are shown in Figure 7, and 8× upsampling results are shown in Figure 8. The weighting factors $\lambda = 10^3$ (for 4× upsampling) and $\lambda = 4 * 10^3$ (for 8× upsampling) correspond to the results presented in Figure 6. We also present results for $\lambda = 2 * 10^3$ and $\lambda = 8 * 10^3$ for 4× and 8× upsampling respectively. The weighting factor $\lambda = 0$ corresponds to training the network using only the mean-squared loss.

The more we weight the adversarial loss (larger λ values), the reconstructed images become sharper and they include more facial details. However, with larger λ we also observe some high frequency artifacts.

4.4 Color Face Upsampling

Human vision is not sensitive to chrominance channels (u, v). Therefore, a common procedure for handling color images is to process only the luminance channel (Y) and add bicubic-upsampled chrominance channels to the result. We used the same procedure for obtaining color upsampling results shown in Figures 9 and 10.

4.5 Failure Cases

Our method does not have major failure modes since it does not rely on very precise alignment. The algorithm produces less satisfactory results when there are big variations in pose and facial expression, and/or occlusion. Several less satisfactory results of our algorithm are shown in Figures 11 and 12.

5 Conclusion

We proposed a face hallucination algorithm that produces high quality images even when the input face resolution is very low and the image is captured in an uncontrolled setting. The key element of our algorithm is a deep learning architecture that jointly learns global and local constraints of the high resolution faces. We conducted extensive comparisons with the state-of-the-art algorithms and showed improved performance.

References

1. Liu, C., Shum, H.Y., Freeman, W.T.: Face hallucination: Theory and practice. *Int'l J. Computer Vision* **75**(1) (2007) 115–134
2. Baker, S., Kanade, T.: Hallucinating faces. In: *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*. (2000) 83–88
3. Fattal, R.: Image upsampling via imposed edge statistics. *ACM Trans. Graphics* **26**(3) (2007) 95
4. Sun, J., Sun, J., Xu, Z., Shum, H.Y.: Image super-resolution using gradient profile prior. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2008) 1–8
5. Kim, K.I., Kwon, Y.: Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(6) (2010) 1127–1133
6. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. *IEEE Computer Graphics and Applications* **22**(2) (2002) 56–65
7. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*. (2009) 349–356

8. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. *IEEE Trans. Image Proc.* **19**(11) (2010) 2861–2873
9. Yang, C.Y., Ma, C., Yang, M.H.: Single-image super-resolution: A benchmark. In: *Proc. European Conf. Computer Vision (ECCV)*. (2014) 372–386
10. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* **3**(1) (1991) 71–86
11. Ma, X., Zhang, J., Qi, C.: Hallucinating face by position-patch. *Pattern Recognition* **43**(6) (2010) 2224–2236
12. Yang, C.Y., Liu, S., Yang, M.H.: Structured face hallucination. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2013) 1099–1106
13. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2012) 2879–2886
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. (2012) 1097–1105
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
16. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2014) 580–587
17. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2014) 1701–1708
18. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8) (2013) 1915–1929
19. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: *Advances in Neural Information Processing Systems*. (2014) 568–576
20. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with BM3D? In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2012) 2392–2399
21. Agostinelli, F., Anderson, M.R., Lee, H.: Adaptive multi-column deep neural networks with application to robust image denoising. In: *Advances in Neural Information Processing Systems*. (2013) 1493–1501
22. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: *Advances in Neural Information Processing Systems*. (2009) 769–776
23. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: *Advances in Neural Information Processing Systems*. (2012) 341–349
24. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: *Advances in Neural Information Processing Systems*. (2014) 1790–1798
25. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. *arXiv preprint arXiv:1501.00092* (2014)
26. Wang, Z., Yang, Y., Wang, Z., Chang, S., Han, W., Yang, J., Huang, T.S.: Self-tuned deep super resolution. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. (2015)
27. Zhou, E., Fan, H., Cao, Z., Jiang, Y., Yin, Q.: Learning face hallucination in the wild. In: *Proc. AAAI Conf. Artificial Intelligence*. (2015)
28. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786) (2006) 504–507

29. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proc. Int'l Conf. Machine Learning (ICML). (2008) 1096–1103
30. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proc. Int'l Conf. Machine Learning (ICML). (2011) 689–696
31. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11) (1998) 2278–2324
32. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). (2010) 2528–2535
33. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). (2015)
34. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc. Neural Information Processing Systems (NIPS). (2014) 2672–2680
35. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Proc. Neural Information Processing Systems (NIPS). (2015) 1486–1494
36. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
37. Lotter, W., Kreiman, G., Cox, D.: Unsupervised learning of visual structure using predictive generative networks. arXiv preprint arXiv:1511.06380 (2015)
38. Irani, M., Peleg, S.: Improving resolution by image registration. *CVGIP: Graphical models and image processing* **53**(3) (1991) 231–239
39. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
40. Phillips, P.J., Flynn, P.J., Scruggs, T., Bowyer, K.W., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). Volume 1. (2005) 947–954
41. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR). (2013) 532–539
42. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, ECCV Workshop on Faces in Real-life Images (2008)
43. Wolf, L., Hassner, T., Taigman, Y.: Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(10) (2011) 1978–1990
44. Sheikh, H.R., Bovik, A.C., De Veciana, G.: An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Trans. Image Proc.* **14**(12) (2005) 2117–2128
45. Damera-Venkata, N., Kite, T.D., Geisler, W.S., Evans, B.L., Bovik, A.C.: Image quality assessment based on a degradation model. *IEEE Trans. Image Proc.* **9**(4) (2000) 636–650
46. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multi-scale structural similarity for image quality assessment. In: Proc. Asilomar Conf. Signals, Systems and Computers. Volume 2. (2003) 1398–1402

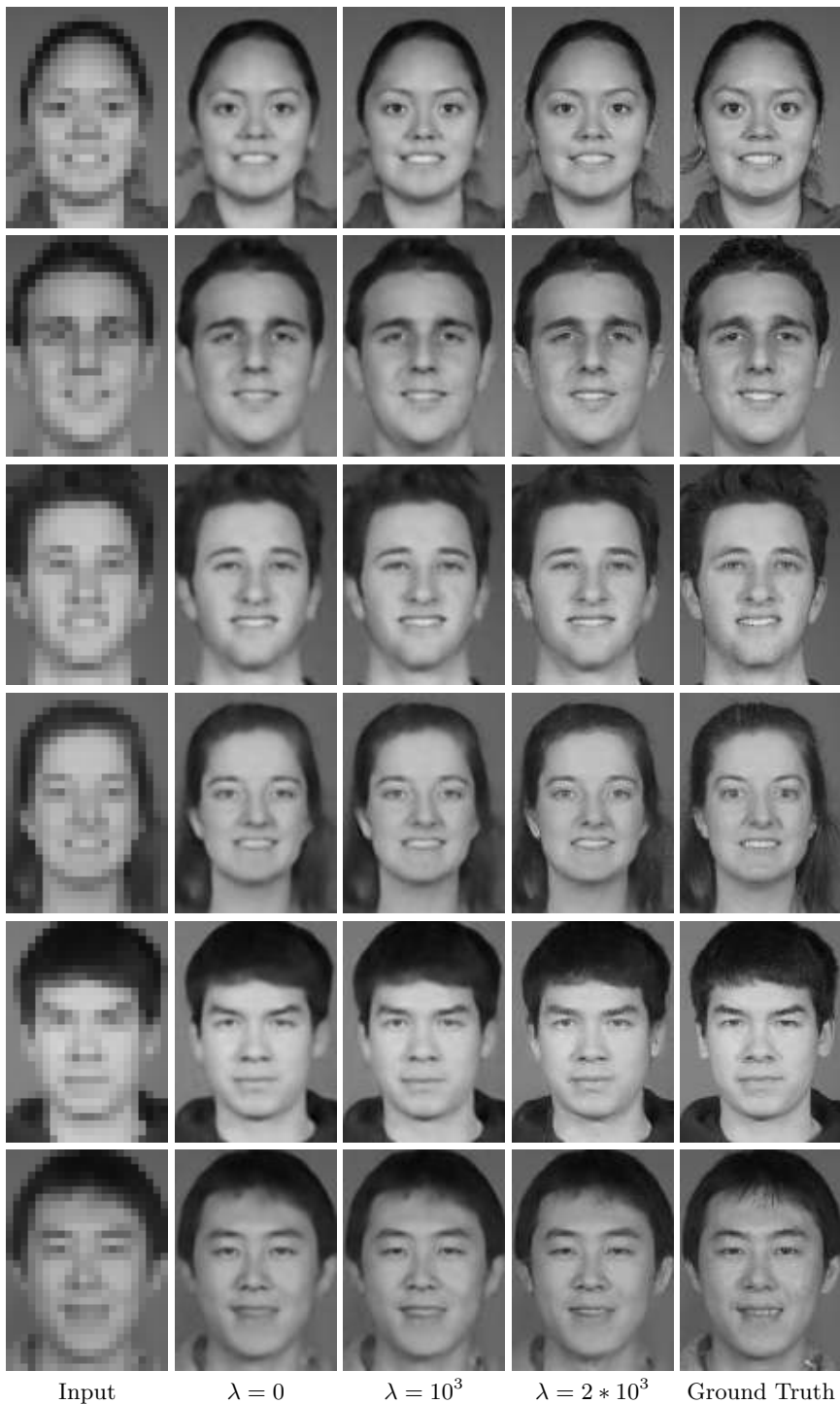


Fig. 7. $4\times$ upsampling results for various weighting factors used in the adversarial loss.

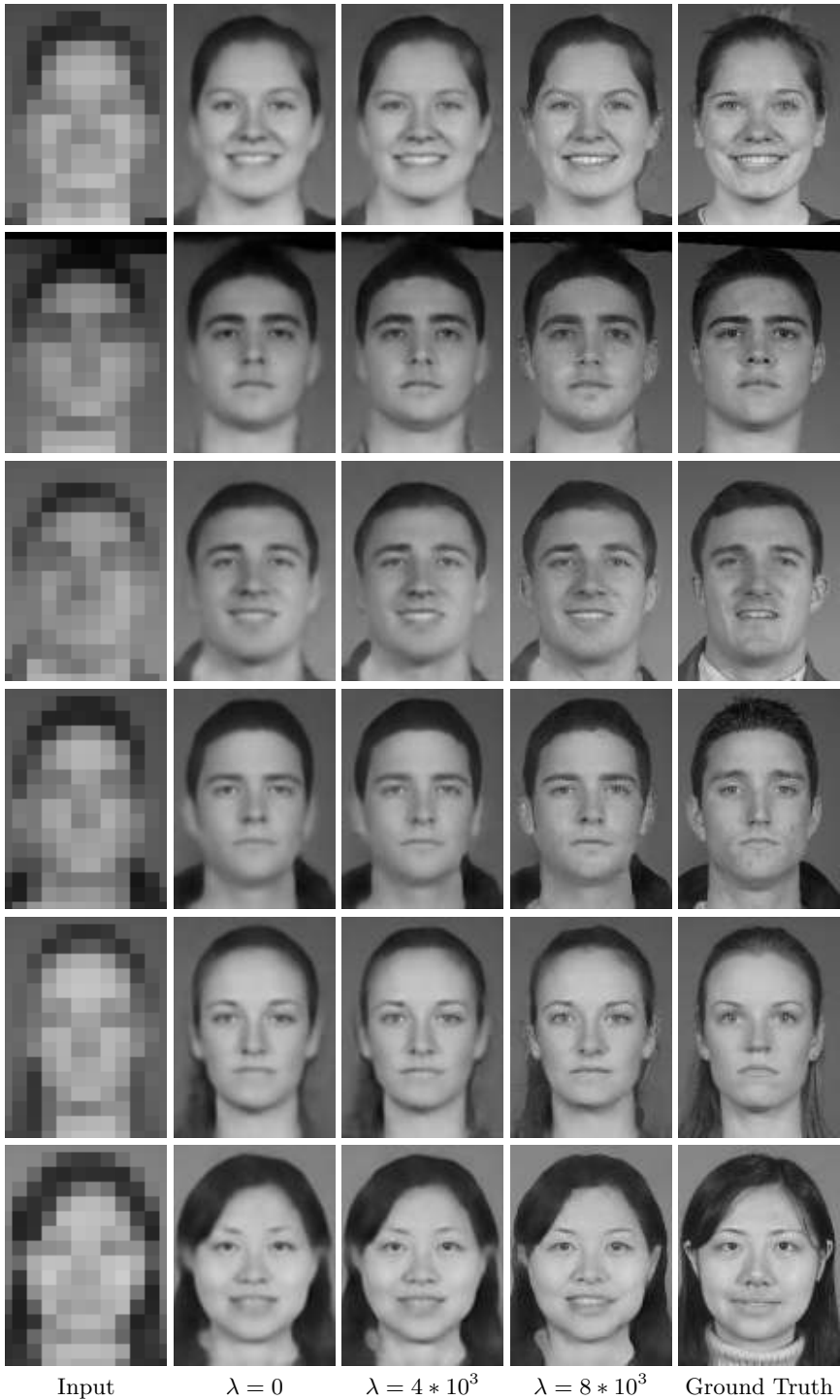


Fig. 8. $8\times$ upsampling results for various weighting factors used in the adversarial loss.



Fig. 9. $4\times$ color upsampling results.



Fig. 10. $8\times$ color upsampling results.



Fig. 11. $4\times$ upsampling failure examples.



Fig. 12. $8\times$ upsampling failure examples.