Fast and Accurate Image Super-Resolution Using A Combined Loss

Jinchang Xu[1], Yu Zhao[1], Yuan Dong[1], Hongliang Bai[2]
[1]Beijing University of Posts and Telecommunications,
[2]Beijing Faceall Technology Co., Ltd, Beijing China.
{xjc1,space_double7,yuandong}@bupt.edu.cn, hongliang.bai@faceall.cn

Abstract

Recently, several methods for single image super-resolution(SISR) based on deep neural networks have obtained high performance with regard to reconstruction accuracy and computational performance. This paper details the methodology and results of the New Trends in Image Restoration and Enhancement (NTIRE) challenge. The task of this challenge is to restore rich details (high frequencies) in a high resolution image for a single low resolution input image based on a set of prior examples with low and corresponding high resolution images. The challenge has two tracks. We present a super-resolution (SR) method, which uses three losses assigned with different weights to be regarded as optimization target. Meanwhile, the residual blocks are also used for obtaining significant improvement in the evaluation. The final model consists of 9 weight layers with four residual blocks and reconstructs the low resolution image with three color channels simultaneously, which shows better performance on these two tracks and benchmark datasets.

1. Introduction

In recent years, the reconstruction of a high-resolution (HR) image from a given low-resolution(LR) image has been widely investigated by many researchers in digital image processing. This task is called super-resolution(SR), commonly referred to single image super-resolution (SISR), which can be helpful in computer vision applications among various areas such as face recognition[11], medical imaging[24] and surveillance[37]. For super resolution, the LR image data is always considered to be downscaled from the corresponding original HR image with or without noise. In general, the optimization target of SR problems is the minimization of the mean squared error (MSE) between the recovered SR image and the ground truth HR image. The peak signal-to-noise ratio (PSNR) and structural similarity index(SSIM) are two commonly accepted measurements to evaluate and compare SR methods.

A detailed review of SISR methods can be found in [32]. The SISR algorithms can be classified into prediction-based methods[6], classical sparse coding methods[34, 35, 33, 16, 10], edge-based methods[7, 25], and anchored neighborhood regression methods[29, 30, 31].Recently, with the rapaid development of deep learning on image processing, a large number of techniques based on convlutional neual network(CNN) have been successfully applied to SR[4, 5, 14, 15, 23, 18, 13, 20, 22].

In this paper, we present a three-loss super resolution network(TLSR) to address the SR problems as shown in Fig. 1. Our network consists of four residual blocks with a combination of three losses. A low resolution RGB image is considered as an input to our network, and the output of the network is three SR images from different residual blocks. Then we compute the MSE of SR images and the corresponding ground truth separately. Finally, the three losses assigned with different weights act as our optimization target.

2. Related Work

2.1. Image Super Resolution

Prediction-based methods include bilinear interpolation, bicubic interpolation, Lanczos resampling[6] and so on. These filtering approaches utilize the statistical image priors and have a fast speed. However, these methods usually make the texture of image smoother, lacking much realistic texture detail. In order to solve this problem, researches on image statistics suggest that image patches can be well represented as a sparse linear combination of elements from an appropriately chosen over-complete dictionary. Yang *et al.*[34, 35, 33] use the coefficients of sparse representation for each patch of the low-resolution input to generate the high-resolution output. In [10], the authors propose a convolutional sparse coding based on SR method to address the consistency of pixels in overlapped patches issues. Sun *et al.* [25] put forward a gradient field transformation to constrain the gradient fields of the high resolution image based on gradient profile prior. Neighborhood regression methods

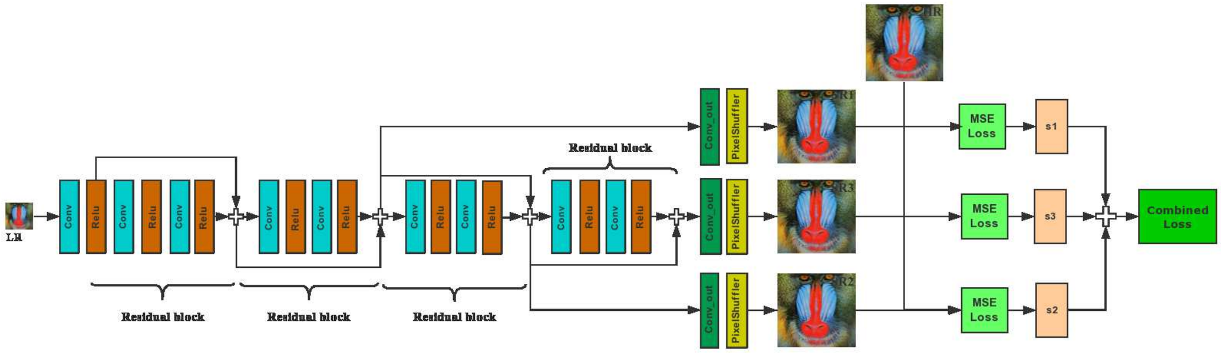


Figure 1: Architecture of the TLSR Network.

upsample a LR image patch by finding similar LR training patches in a low dimensional manifold and combining their corresponding HR patches for reconstruction [29, 30, 31].

2.2. Deep Learning Networks

In recent years, deep learning networks have reached an extreme high accuarcy in various computer vision fields including image classification[17, 28, 12], image segmentation[19], object detection[9, 21, 8], and face recognition[26, 27]. Due to its outstanding performance, methods concerning deep learning networks have been proposed in super resolution. Dong *et al*.[4] firstly propose an approach, termed SRCNN, to learn a mapping from LR to HR in an end-to-end manner by investiagting the connection between sparse-coding-based methods with CNN. Motivated by the fact that SRCNN achieves state-of-the-art performance with only three layers in the SR problem, many researchers attempted to prepare deeper models to recover the LR to HR. Kim *et al*.[14] present a highly accurate SR method based on a very deep convolutional network with twenty weight layers. Also, in [15], the authors formulate a recursive CNN and achieve an awesome results. However, all of the above methods increase the resolution of LR image using interpolation methods such as bicubic interpolation before transformed into convolution neural network. Then the input image is much larger than the original LR image, raising the computational complexity for neural networks. In order to eliminate this problem, Shi *et al*.[23] introduce an efficient sub-pixel convolution layer learning an array of upscaling filters, to upscale the final LR feature maps into the HR output. Dong *et al*.[5] introduce a deconvolution layer at the end of the network and learn an end-to-end mapping between the original LR and HR images with no pre-processing. As He *et al*. [12] demonstrate that residual blocks of convolution neual network show a higher performance, the authors of [18] bring up SRGAN, a generative adversarial network (GAN) for image super resolution using a perceptual loss function which consists of an adversarial loss and a content loss.In [20], the authors propose a pixel recursive model to resolve the SR problem.

3. Method

As shown in Fig. 1, our deep convolutional network for SR image reconstruction consists of convolution layers, rectified linear units(relu), residual blocks, sub-pixel convolution layers and a loss network. Firstly, we use a convolution layer to extract feature from the original LR image. Different from most SR methods in which only Y channel splited from YCbCr images that are converted from the corresponding RGB ones has been used, we utilize all the channels information of the LR image. Thus, the input image can be considered as a tensor with the size of C×H×W, where C refers to the colour channels and H, W refer to the height and width of the image respectively. The fisrt *conv* layer has 64 fiters of spatial size 5×5 and the padding number is set to 2 in order not to change the size of image. Except the fisrt *conv* layer, other *conv* layers have 64 filters of the spatial size 3×3 with the padding number (1,1). After each convolution layer, the *relu* is followed as the activation function. The *conv_out* layers are in the same type: C×r×r filters of the spatial size 3×3, where r is referred as to the upscale factor.

3.1. Residual network architecture

Inspired by [12], He *et al*. use residual connections to train very deep networks for image classification. In our network, we use four residual blocks, each of which contains two convolution and relu layers. Different from Kim *et al*.[14], we replace the original LR image with the output of effect several convolutional neural networks. By doing this, we can enhance each image after convolutional network instead of only improving the final output image. We denote the input LR image as I_{LR}, the output of residual block as O_{rb}, the output of *conv_out* layer as I_{out}, the SR image obtaining from the network as I_{SR} and the ground truth high resolution image as I_{HR}. The mapping between functions of convolution layer can be described as follows:

$$g^1(x) = max(W_1 * x + b_1, 0) \tag{1}$$

$$g^n(x) = max(W_n * g^{n-1}(x) + b_n, 0) \tag{2}$$

where W_n is the weight value of the *conv* layer, b_n is the bias of the *conv* layer, and *n* is the number of the *conv* layers. In our network, W_1 is in the shape of 3×64×5×5 while the W_n (*n* = 2. . . 9) is in the same size of 64×64×3×3, b_n (*n* = 1. . . 9) is a vector of length 64.The ReLU activation function is *max*().

The fisrt residual block in our network can be expressed as below:

$$O_{rb1} = g^3(I_{LR}) + g^1(I_{LR}) \tag{3}$$

where O_{rb1} is the output of the first residual block. In the left of formula (3), we use a *three-conv-relu* layer to learn feature maps from the LR image like SRCNN[4] or ESPCN[23]. Instead of using these feature maps to reconstruct the I_{SR} simply, we add another three residual blocks to recover the final I_{SR}.

The other three residual blocks are listed as follows:

$$O_{rb2} = g^2(O_{rb1}) + O_{rb1} \tag{4}$$

$$O_{rb3} = g^2(O_{rb2}) + O_{rb2} \tag{5}$$

$$O_{rb4} = g^2(O_{rb3}) + O_{rb3} \tag{6}$$

where O_{rb2}, O_{rb3}, O_{rb4} are the output of the residual blocks respectively. As they are in the size of 64×H×W, the *conv_out* layer is used for converting the 64 channels into (rC)^2 channels. Thus, we can get three tensors I_{out1}, I_{out1}, I_{out1} separately, the size of which is (rc)^2×H×W. Finally, we use sub-pixel convolution layer proposed by Shi *et al.*[23] to reconstruct the I_{SR} from the outputs of the *conv_out* layer.

3.2. Loss network architecture

While most methods[4, 5, 23, 14] about SR use the mean square error (MSE) as the cost function, for this pixel-wise loss function can obtain a high PSNR. It can be calculated as:

$$l_{SR} = \frac{1}{r^2CHW}\sum_{k=1}^{C}\sum_{i=1}^{rH}\sum_{j=1}^{rW}(I_{HR}(i,j,k) - I_{SR}(i,j,k)) \tag{7}$$

Then we adopt a combined loss function with three MSE losses as our final optimization objective to train our network in order to find optimal parameters of our model. It can be represented as:

$$l_{SR}^{final} = s_1 \times l_{SR1} + s_2 \times l_{SR2} + s_3 \times l_{SR3} \tag{8}$$

where l_{SR}^{final} is the final loss function to minimize, l_{SRi} (i∈{1,2,3}) denotes the MSE between the super resolution image I_{SRi} (i∈{1,2,3}) and the high resolution image I_{HR}, and s_i (i∈{1,2,3}) stands for the assigned wight value of each l_{SRi}.

4. Training and testing sets

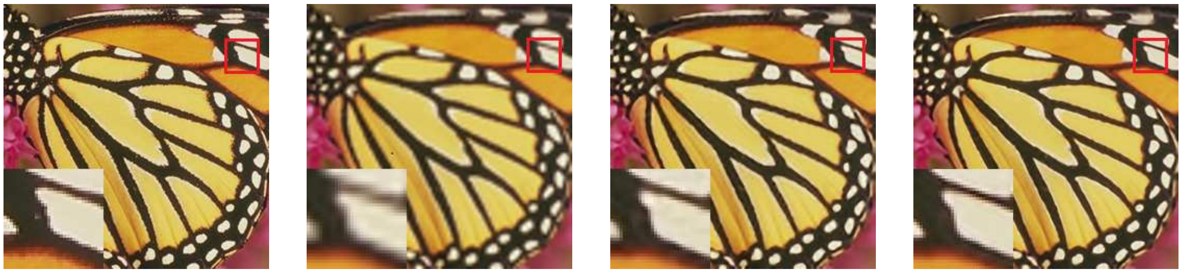
In this section, we describe the data used for training and testing our method. Meanwhile, training details and parameters are given. The NTIRE challenge is divided into two tracks. **Track 1**: "Classic bicubic" follows the classic / standard settings from the single-image super-resolution literature, that is, the degradation operators are the downscaling with bicubic interpolation (imresize Matlab function) of the ground truth high resolution image. **Track 2**: "Unknown" assumes that the degradation operators are unknown for the participants under explicit form (such as blur kernel, decimation, downscaling strategy). The large training set of examples of low and corresponding high resolution images are intended for modeling the low to high image resolution mapping relation.

4.1. Datasets for Training and Testing

For the images used for training SR, there are many different training image datasets such as 91 images, 291 images and random images from ImageNet dataset. As for our method, we use the DIV2K(DIVerse 2K) dataset[1] for training and testing, which has 1000 RGB images in total. The images are no larger than 2048 pixels on horizontal or vertical direction with a large diversity of contents. The DIV2K dataset is divided into three parts: 800 images for training, 100 images for validation and another 100 images for testing. The low resolution images can be categorized of two groups:

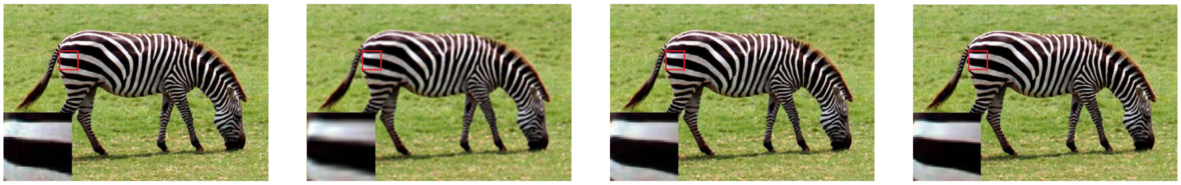
- obtained by using the Matlab function "imresize" with bicubic interpolation and the desired downscaling factors: 2, 3, and 4 from the ground truth high resolution RGB image.
- obtained by using unknown degradation operators with the desired downscaling factors: 2, 3, and 4 from the ground truth high resolution RGB image.

Track1, in the experiments of recovering the LR images belonging to the first group, apart from the 800 low resolution DIV2K images for training, we also adopt data augmentation as in [4]. The high resolution images are augmentated by being downscaled with the factor 0.6, 0,7, 0.8 and 0.9; and being rotated with the degree of 90, 180 and



(a) GroundTruth / PSNR (b) Bicubic / 24.04dB (c) SRCNN / 27.95dB (d) TLSR / 29.56dB

Figure 2: The butterfly image from Set5 with an upscaling factor 3



(a) GroundTruth / PSNR (b) Bicubic / 26.63dB (c) SRCNN / 29.29dB (d) TLSR / 29.70dB

Figure 3: The zebra image from Set14 with an upscaling factor 3

270. After this augmentation, we will have 16000 high resolution images for training. As for getting the low resolution images, we also use the Matlab function ”imresize” with bicubic interpolation from the corresponding high resolution images. So, we have 16000 pairs of LR images and HR images in total to train our network.

Track 2, in the experiments of recovering the LR images which belong to the second group, due to unknown degradation operators to obtain the LR images, we just use the given 800 pairs as training images.

We mainly use the 100 LR images of validation data via validation sever to validate our methods, for the groundtruth can't be obtained by us. And in the end, we run our methods on the 100 testing data and get the results from the challenge organizers.

4.2. Implementation details

To prepare the training data, we have attempted two strategies. The first one is to crop the LR training images into a set of 64×64-pixel sub-images with a stride 64. At the same time, the corresponding HR sub-images in the shape of (r×64)×(r×64) are also cropped from the ground truth images with a stride r×64. The second one is to crop the LR training images from the center of the images with a large scale. For ×2, ×3, ×4, the size of LR/HR sub-images are set to be $324^2/(324\times2)^2$, $128^2/(128\times3)^2$ and $96^2/(96\times4)^2$ respectively. As we find using the large scale images to train our network can achieve better performance than using the small scale images, so all of our models are trained based on the second strategy.

For weight initialization, we use the the orthogonal matrixs and for the assigned weight value of our loss structure, we set s_1, s_2 and s_3 to 0.5, 0.5 and 1.0 separately. All networks are trained on a NVIDIA GTX-1080 GPU. The relu is chosen as activation function for the final model while Adaptive Moment Estimation(Adam) is utilized as optimizing method. The training batches are set to 64 and the training stops after 500 epochs. Initial learning rate is set to 0.001 while the final learning rate is set to 0.0001. We use the PSNR and SSIM as the performance metric to evaluate our models.

5. Image super-resolution results

For benchmark, we follow the 5-3-3 ESPCN [23] framework. Several experiments have been done before we decide on the best model. Also, in order to demonstrate the performance of our proposed method, we compare it with the existing state-of-art SR methods on several benchmark datasets such as Set5[2], Set14[36].

5.1. Y channel *vs* RGB channels

Many methods on SR have evaluated their methods only on y channels, for the reason of human is more sensitive to it. Based on this, initially , we convert the RGB image into YCbCr image and then split the YCbCr image into Y, Cb, Cr channels separately. And we reconstruct the Y channel by using ESPCN network while the Cb channel and Cr channel are reconstructed using the Matlab “imresize” function with bicubic interpolation. Comparing to the above method, we do another experiment. This way, we don't convert the RGB image into YCbCr image. We use the RGB image as an input to ESPCN network directly to obtain the SR image. The PSNR and SSIM of using three channels to reconstruct the LR image to SR image are much more better than just using Y channel. The results can be seen in Tab. 1.

Dataset	Scale	Y PSNR/SSIM	RGB PSNR/SSIM
DIV2K	2	32.25/0.9156	33.33/0.9259

Table 1: The PSNR and SSIM of using three channels to reconstruct the LR image to SR image are much more better than just using Y channel. Where the DIV2K denotes for the bicubic downscaling validation dataset.

5.2. One loss *vs* Combined loss

Meanwhile, in Tab. 2, we compare the network with a simple loss and a combined loss trained using same images and validate on the the DIV2K dataset. For simple loss, we set s_1, s_2 and s_3 to 0.0, 0.0 and 1.0 separately while for the combined loss, we design the weight value of s_1, s_2 and s_3 to 0.5, 0.5 and 1.0. The three SR images I_{SRi} (i$\in\{1,2,3\}$) are obtained via 5,7,9 *conv* layers. As we believe that the image restored from the deeper layers has richer details. So we assign the last MSE loss l_{SR3} with highest weight value than another two. The other parameter settings of s_i (i$\in\{1,2,3\}$) are investiagted in future work. From the results we can see, a combined loss improve 0.01 in PSNR and 0.0003 in SSIM.

Dataset	Scale	One loss PSNR/SSIM	Combined loss PSNR/SSIM
DIV2K	2	33.76/0.9295	33.77/0.9298

Table 2: The results of using a combined loss to train network is better than using a simple loss.

5.3. ESPCN *vs* Ours

Due to the fact that using three channels to reconstruct low resolution image can get a sharp gain in PSNR and SSIM, we use the RGB images as our input to networks and compare our method with ESPCN on the DIV2K Dataset. Both networks are trained with the same images. The results for ×2, ×3, ×4 are shown in Tab. 3

Dataset	Scale	ESPCN PSNR/SSIM	TLSR PSNR/SSIM
DIV2K	2	33.33/0.9259	33.77/0.9298
DIV2K	3	29.31/0.8409	30.12/0.8590
DIV2K	4	27.60/0.7805	28.19/0.7996

Table 3: The results of bicubic interpolation, ESPCN and TLSR on DIV2K validation dataset for diffrent scale: ×2, ×3, ×4. Where ESPCN denotes for the ESPCN 5-3-3 network[23] trained on the 16000 pairs of images of DIV2K dataset.

5.4. Evaluation on DIV2K testing dataset

By using the residual blocks structure with a combination of three loss, we test our methods on DIV2K testing data. As we mentioned above, for the bicubic downscaling test, we train our models using the data augmentation with 16000 pair of images. But for the unknown downscaling test, we only have the 800 pairs of images, so we fintune our model on these training images to get the best results. All the results are shown in Tab. 4. Also, we compute the time between the input of the netowrk and the output of the network in Tab. 5, and the time for recovering a LR image to SR image is obout 0.009 seconds via the NVIDIA GTX-1080 without I/O times.

Challenge	Scale	Bicubic PSNR/SSIM	TLSR PSNR/SSIM
Track 1	3	28.22/0.822	30.07/0.869
Track 1	4	26.65/0.761	27.99/0.805
Track 2	3	25.81/0.736	29.87/0.862
Track 2	4	21.84/0.583	26.84/0.762

Table 4: The results on the DIV2K test of our method and bicubic interpolation method. Where the DIV2K(bicubic) stands for the bicubic downscaling test and the DIV2K(unknown) denotes for the unknown downscaling test.

Challenge	Scale	TIME
Track 1	3	0.008474
Track 1	4	0.009891
Track 2	3	0.009039
Track 2	4	0.01001

Table 5: The time denotes for the image through the TLSR network excludes I/O times.

Dataset	Scale	Bicubic	SRCNN	TNRD	ESPCN	TLSR
Set5	3	30.39	32.75	33.17	33.13	33.60
Set14	3	27.54	29.30	29.46	29.49	29.66
Set5	4	28.42	30.49	30.85	30.90	31.05
Set14	4	26.00	27.50	27.68	27.73	27.81

Table 6: The results of different methods evaluated on benchmark datasets. Where SRCNN denotes for the SRCNN 9-5-5 model trained on ImageNet[4], TNRD stands for the Trainable Nonlinear Reaction Diffusion Model[3], ESPCN stands for ImageNet model with tanh activation[23], TLSR stands for our model trained on DIV2K dataset.

5.5. Comparision on benchmark datasets with the state-of-art methods

We also evaluate our methods on benchmark datasets like Set5[2] and Set14[36] for ×3, ×4 super resolution. Comparing with these existing methods such as SRCNN 9-5-5 model trained on ImageNet[4], Trainable Nonlinear Reaction Diffusion Model[3] and ESPCN with tanh activation[23]. The results can be shown in Tab. 6. There are also some visual results to better assess the performance of this method shown in Fig. 2 and 3.

6. Conclusion

In this work, we have presented a super resolution method using three losses assigned with different weights to be regarded as optimization target. We have demonstrated that our method outperforms the existing method on the benchmark datasets. Also, with simplicity and robustness of our network, our approach can be applicable to other image restoration problems such as denoising and deblurring.

Acknowledgement

The research is sponsored by by Beijing Faceall Technology Co.,Ltd.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 3

[2] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 4, 6

[3] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 6

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014. 1, 2, 3, 6

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016. 1, 2, 3

[6] C. E. Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, 18(8):1016–1022, 1979. 1

[7] G. Freedman and R. Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*, 30(2):12, 2011. 1

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[10] S. Gu, W. Zuo, Q. Xie, D. Meng, X. Feng, and L. Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015. 1

[11] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE transactions on image processing*, 12(5):597–606, 2003. 1

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 1

[14] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. 1, 2, 3

[15] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1645, 2016. 1, 2

[16] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE transactions on pattern analysis and machine intelligence*, 32(6):1127–1133, 2010. 1

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[18] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016. 1, 2

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[20] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 1, 2

[21] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2015. 2

[22] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 1

[23] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 1, 2, 3, 4, 5, 6

[24] W. Shi, J. Caballero, C. Ledig, X. Zhuang, W. Bai, K. Bhatia, A. M. S. M. de Marvao, T. Dawes, D. ORegan, and D. Rueckert. Cardiac image super-resolution with global correspondence using multi-atlas patchmatch. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 9–16. Springer, 2013. 1

[25] J. Sun, Z. Xu, and H.-Y. Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *IEEE Transactions on Image Processing*, 20(6):1529–1542, 2011. 1

[26] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014. 2

[27] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015. 2

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[29] R. Timofte, V. De Smet, and L. Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1920–1927, 2013. 1, 2

[30] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014. 1, 2

[31] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016. 1, 2

[32] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-image super-resolution: A benchmark. In *European Conference on Computer Vision*, pages 372–386. Springer, 2014. 1

[33] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing*, 21(8):3467–3478, 2012. 1

[34] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[35] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 1

[36] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 4, 6

[37] L. Zhang, H. Zhang, H. Shen, and P. Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010. 1