# Images Synthesis And Image Degradations Using Generative Adversarial Networks

[1]Patel Tarunkumar k, [2]Dhaval A. Parikh , [1]ME Student , [2]Associate Professor

Lalbhai Dalpatbhai College of Engineering Ahmedabad,
Gujarat Technological University, Ahmedabad, India

***Abstract :*** Generating images from natural language is one of the primary applications of recent conditional generative models. Besides testing our ability to model conditional, highly dimensional distributions, text to image synthesis has many exciting and practical applications such as photo editing or computer-aided content creation. Recent progress has been made using Generative Adversarial Networks (GANs). This material starts with a gentle introduction to these topics and discusses the existent state of the art models so here vast majority of prior work for this problem focus on how to increase the resolution of low-resolution images which are artificially generated by simple bilinear down-sampling (or in a few cases by blurring followed by down-sampling) and Decrease Loss Function.

***Keywords* – Image Synthesis , Super-resolution, Generative Adversarial Networks, GANs.**

## I. INTRODUCTION

This paper is on enhancing the resolution and quality of low-resolution, noisy, blurry, and corrupted by artefacts images. We collectively refer to all these tasks as image super-resolution and image synthesis. This is a challenging problem with a multitude of applications from image enhancement and editing to image recognition and object detection to name a few.

Our main focus is on the problem of super-resolving real-world low-resolution images for a specific object category.

## II. MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

## III. MACHINE LEARNING METHODS

Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

## IV. ADVERSARIAL MACHINE LEARNING

Adversarial machine learning is a research field that lies at the intersection of machine learning and computer security. It aims to enable the safe adoption of machine learning techniques in adversarial settings, such as spam filtering, malware detection, and biometric recognition.

Machine learning techniques were originally designed for stationary environments in which the training and test data are assumed to be generated from the same (although possibly unknown) distribution. However, In the presence of intelligent and adaptive adversaries this working hypothesis is likely to be violated to at least some degree (depending on the adversary). In fact, a malicious adversary can carefully manipulate the input data, exploiting specific vulnerabilities of learning algorithms to compromise the whole system security.

Examples include attacks in spam filtering, where spam messages are obfuscated through misspelling of "bad" words or insertion of "good" words, attacks in computer security such as obfuscating malware code within network packets or to mislead signature detection, attacks in biometric recognition where fake biometric traits may be exploited to impersonate a legitimate user or to compromise users template galleries that adapt to updated over time.

**V. Generative Adversarial Networks**

To understand GANs, you should know how generative algorithms work, and for that, contrasting them with discriminative algorithms is instructive. Discriminative algorithms try to classify input data; that is, given the features of a data instance, they predict a label or category to which that data belongs.

For example, given all the words in an email, a discriminative algorithm could predict whether the message is spam or not spam. spam is one of the labels, and the bag of words gathered from the email are the features that constitute the input data. When this problem is expressed mathematically, the label is called y and the features are called x. The formulation p(y|x) is used to mean “the probability of y given x”, which in this case would translate to “the probability that an email is spam given the words it contains”.

So discriminative algorithms map features to labels. They are concerned solely with that correlation. One way to think about generative algorithms is that they do the opposite. Instead of predicting a label given certain features, they attempt to predict features given a certain label.

The question a generative algorithm tries to answer is: Assuming this email is spam, how likely are these features? While discriminative models care about the relation between y and x, generative models care about “how you get x.” They allow you to capture p(x|y), the probability of x given y, or the probability of features given a class. (That said, generative algorithms can also be used as classifiers. It just so happens that they can do more than categorize input data.)

Another way to think about it is to distinguish discriminative from generative like this:

- Discriminative models learn the boundary between classes
- Generative models model the distribution of individual classes

**V.1 GAN Work**

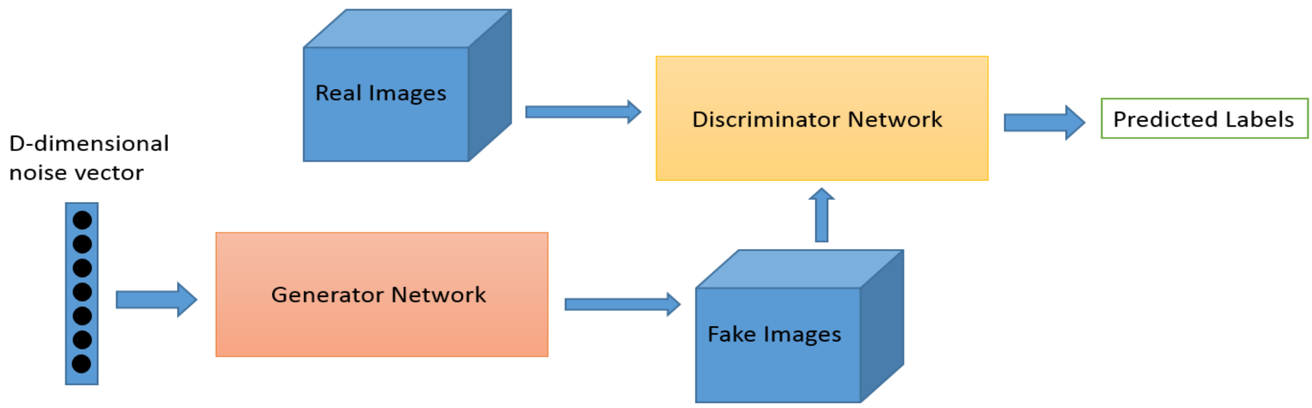



Fig 1 The generator is in a feedback loop with the discriminator and generator

Neural network, called the generator, generates new data instances, while the other, the discriminator, evaluates them for authenticity; i.e. the discriminator decides whether each instance of data it reviews belongs to the actual training dataset or not.

Let’s say we’re trying to do something more banal than mimic the Mona Lisa. We’re going to generate hand-written numerals like those found in the MNIST dataset, which is taken from the real world. The goal of the discriminator, when shown an instance from the true MNIST dataset, is to recognize them as authentic.

Meanwhile, the generator is creating new images that it passes to the discriminator. It does so in the hopes that they, too, will be deemed authentic, even though they are fake. The goal of the generator is to generate passable hand-written digits, to lie without being caught. The goal of the discriminator is to identify images coming from the generator as fake.

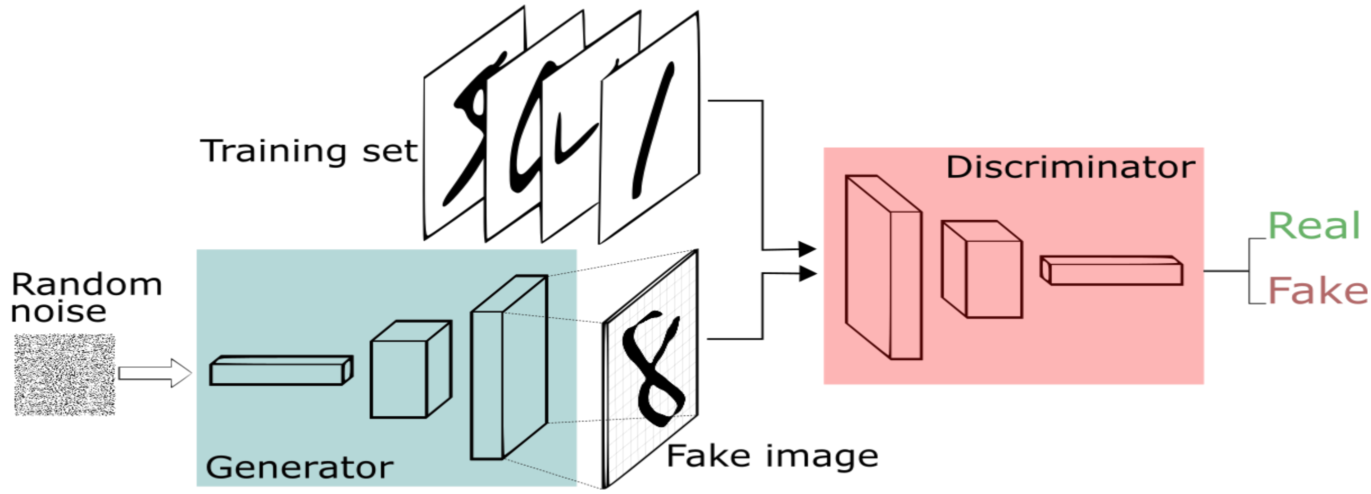



Fig 2 GANs Model

**VI. Closely related work**

There is a very long list of image sythesis and super-resolution papers and a detailed review of the topic is out of the scope of this section. Here in, we focus on related recent work based on Convolutional Neural Networks (CNNs).

The standard approach to super-resolution using CNNs is to use a fully supervised approach where a low-resolution (LR) image is processed by a network comprising convolutional and up sampling layers in order to produce a higher solution (HR) image which is then matched against the original HR image using an appropriate loss function. We call this paired setting as it uses pairs of LR and corresponding HR images for training.

Early attempts based on the aforementioned setting [6, 7] use various Lp losses between the generated and the ground truth HR images for training the networks which however result in blurry super-resolved images. A notable improvement is the so-called perceptual loss which applies an L2 loss over feature maps calculated using another pre-trained network. More advanced deep architectures for super-resolution including recursive, laplacian and dense networks have been recently proposed in [10–12].

More recently, and following the introduction of GANs , the authors of proposed a super-resolution approach which, on top of pixel- and/or feature based losses, it also uses a discriminator to differentiate between the generated and the original HR images which is found to produce more photo-realistic results. Notably, [14], which is an improved version of won the first place in the challenge of . More recently, proposed a patch-based texture loss which is found to improve the reconstruction quality. Different from the aforementioned4 Adrian Bulat*, Jing Yang*, Georgios Tzimiropoulos methods is which does not use a GAN but proposes a pixel recursive super resolution method which is based on PixelCNNs From the aforementioned works, our method has similar objectives to those of which also targets the case of real-world image super-resolution. However, the methodology used in and the one proposed in this paper are completely different. While proposes to capitalize on internal image statistics to do real world super-resolution, our method proposes to use unpaired LR and HR images to learn the image degradation process, and then use it to learn super-resolution

### *VII. Method*

1.Overall architecture

Given a LR facial image of size 16 × 16, our system uses a super-resolution network, which we call Low-to-High, to super resolve it into a HR image of 64 × 64. This Low-to-High network is trained with paired LR and HR facial images. we propose to learn both degrading and downsampling a HR facial image using another network which we call High-to-Low. Notably, High-to-Low is trained using unpaired data from 2 completely different and disjoint datasets. The first of these datasets contains HR facial images from a number of face alignment datasets. The second dataset contains blurry and low quality LR facial images from Widerface.

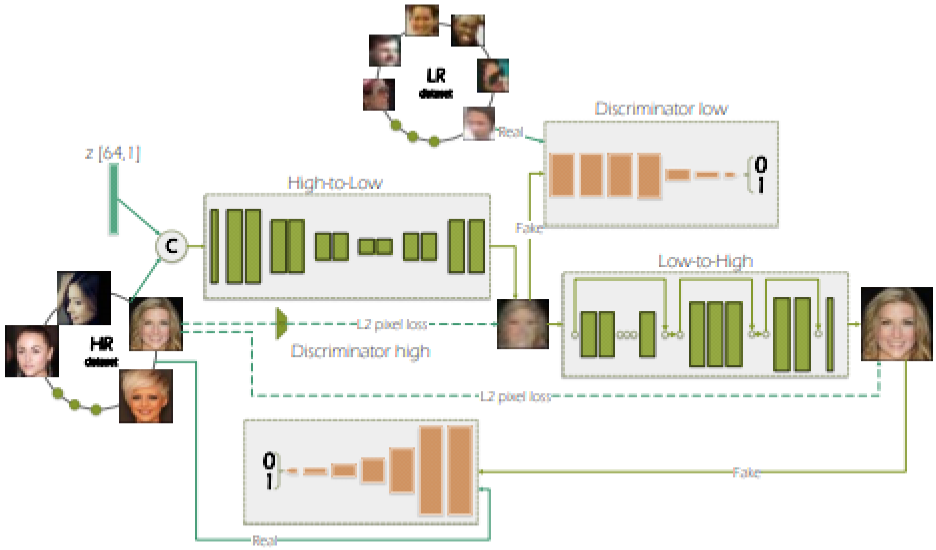



Fig. 2: Overall proposed architecture and training.

### VIII.Dataset

- Caption data in Torch format
  - ➢ Birds and Flowers
  - ➢ HR and LR
- Image data
  - ➢ Birds and Flowers
  - ➢ HR and LR
- Text encoder
  - ➢ Birds and Flowers Descriptions
  - ➢ HR and LR Description.

### IX. Loss functions

We trained both High-to-Low and Low-to-High using a weighted combination of a GAN loss and an L2 pixel. As mentioned earlier, a second fundamental difference between this paper and previous work is how these losses are combined. While recent works on image super-resolution also use such a combination (in many cases there is also a feature loss), in these works, the L2 pixel loss dominates with the GAN loss playing a refinement role for making the images look sharper and more realistic (as the L2 pixel loss is known to generate blurry images).

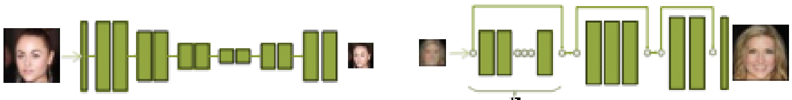
For each network, we used a loss defined as:

$$l = \alpha l_{pixel} + \beta l_{GAN} ,$$

where α and β are the corresponding weights and $\beta l_{GAN} > \alpha l_{pixel}$ in general.

### X. Training

To crop all facial images in a consistent manner, we ran the face detector [38] on all datasets. To further increase the diversity, we augmented the data during training by applying random image flipping, scaling, rotation and color jittering. In order to train the Low-to-High network, we generated on-the-fly LR images, each time providing as input a different random noise vector to High-to-Low, sampled from a normal distribution, in order to simulate a large variety of image degradation types. Both the High-to-Low and Low-to-High networks were trained for 200 epochs (about 570,000 generator updates), with an update ratio 5:1 between the discriminator and the generator. In the end, we fine-tuned them together for another 2,000 generator updates. The learning rate was kept to 1e−4 for the entire duration of the training process. We used $\alpha = 1$ and $\beta = 0.05$ in Eq. 1. All of our models were trained using PyTorch [39] and optimized with Adam [40] ($\beta 1 = 0$ and $\beta 2 = 0.9$).



(a) High-to-Low generator. (b) Low-to-High generator.

Fig. 3: The generator architecture used for the (a) High-to-Low and (b) Low-to-High networks

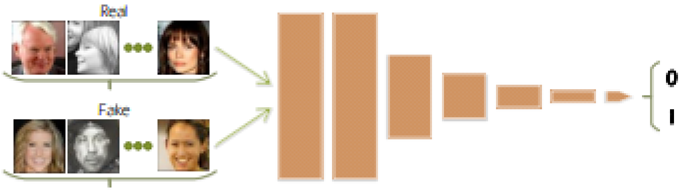



Fig. 4: The discriminator architecture used for both High-to-Low and Low-to-High networks.

### XI. Results

In this section, we evaluate the performance of our system, and compare it with that of a few interesting variants and related state-of-the-art methods. Our main results are on the 3,000 images of our LR test set which contains images from the Widerface dataset.

| No | Iteration(x10000) | Function Loss | Time(Min) |
|---|---|---|---|
| 1 | 0 | 1.5 | 0 |
| 2 | 10 | 0.7365 | 20-25 |
| 3 | 20 | 0.6106 | 30-35 |
| 4 | 30 | 0.5036 | 45-50 |
| 5 | 40 | 0.4662 | 70-80 |
| 6 | 50 | 0.4510 | 90-100 |
| 7 | 60 | 0.4470 | 120-150 |

Table1. GAN Training

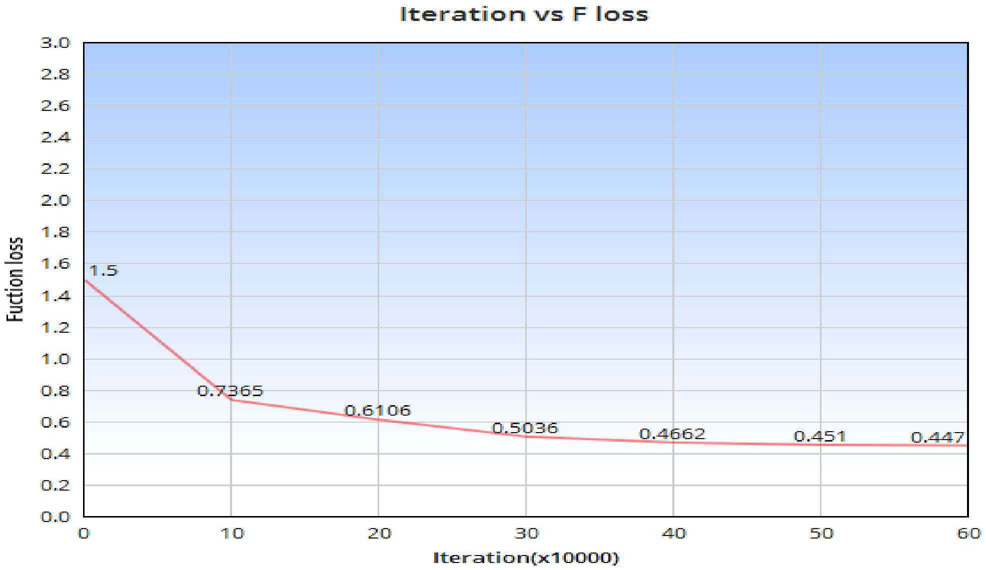



Fig 5. Resultant graph GAN (0-600K)

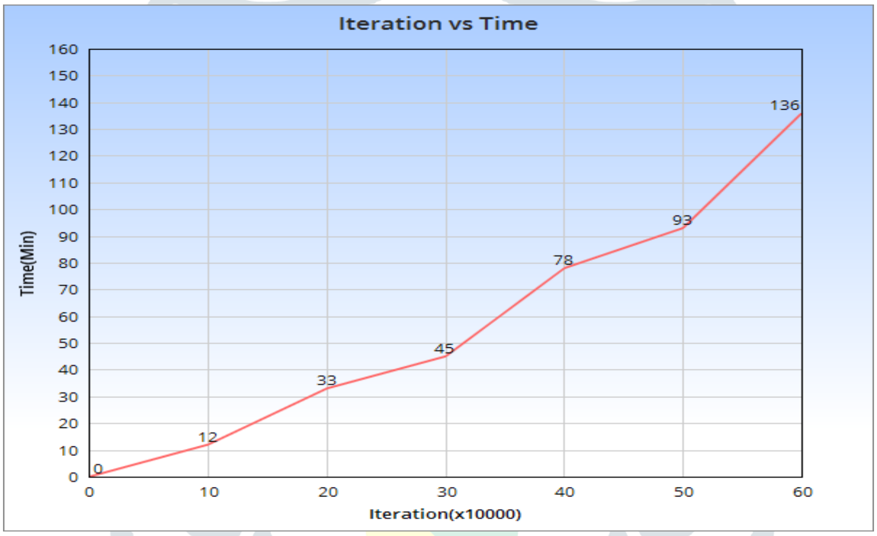



Fig 5. Iteration vs Time

| No | Method | FID | PSNR |
|---|---|---|---|
| 1 | SRGAN | 104.80 | 23.19 |
| 2 | CycleGan | 19.01 | 16.10 |
| 3 | Low-to-High | 85.59 | 23.50 |
| 4 | High-to-Low+Low-to-High | 87.91 | 23.22 |
| 5 | Ours | 15.20 | 18.40 |
| | | | |

Table2. (a) FID-based performance on our real-world LR test set. Lower is better. (b) PSNR results on LS3D-W (the input LR images are bilinearly downsampled images).

## XII. Failure cases

By no-means we claim that the proposed method solves the real-world image and face super-resolution problem. We show several failure cases of our method .We can group failures into two groups: the first one contains cases of complete failures where the produced image does not resemble a face. For many of these cases, we note that the input does not resemble a face either. In total, we found that the percentage of fail cases in our test set is about 10%.

## XIII. Conclusions

We presented a method for image and face super-resolution which does not assume as input artificially generated LR images but aims to produce good results when applied to real-world, LR, low quality images. To this end, we proposed a two-stage process which firstly uses a High-to-Low network to learn how to downgrade high-resolution images requiring only unpaired high- and lower solution images and uses the output of this network to train a Low-to-High network for image super-resolution. We showed that our pipeline can be used to effectively increase the quality of real-world LR images. We reported large improvement over baselines and prior work.

## XIV. References


1.Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

2.Image Super-Resolution Using Deep Convolutional Networks(Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaoou Tang, Fellow, IEEE)

3.StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

4.Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks(Jun-Yan Zhu∗ Taesung Park∗ Phillip Isola Alexei A. Efros Berkeley AI Research (BAIR) laboratory, UC Berkeley)

5.Generative Adversarial Text to Image Synthesis(Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee)

6. M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784,2014

7.A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," arXiv preprint arXiv:1710.07035, 2017.

8.L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in Advances in Neural Information Processing Systems 28, 2015, pp. 262–270.

9F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in ICLR, 2016.

10.S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text-to-image synthesis," in Proceedings of The 33rd International Conference on Machine Learning, 2016.

11. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR. (2017)

12. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: ICCV. (2017)

13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) 2672–2680

14. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR-W. (2017)

15. Sajjadi, M.S., Scholkopf, B., Hirsch, M.: Enhancenet: Single image super-resolution through automated texture synthesis. In: ICCV. (2017)

16. Dahl, R., Norouzi, M., Shlens, J.: Pixel recursive super resolution. In: ICCV. (2017)

17. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. arXiv (2016)

18. Yu, X., Porikli, F.: Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders. In: CVPR. (2017)

19. Cao, Q., Lin, L., Shi, Y., Liang, X., Li, G.: Attention-aware face hallucination viadeep reinforcement learning. In: CVPR. (2017)

20. Huang, H., He, R., Sun, Z., Tan, T.: Wavelet-srnet: A wavelet-based cnn for multiscale face super resolution. In: ICCV. (2017)

21. Zhu, S., Liu, S., Loy, C.C., Tang, X.: Deep cascaded bi-network for face hallucination. In: ECCV. (2016)

22. Yu, X., Porikli, F.: Ultra-resolving face images by discriminative generative networks. In: ECCV. (2016)

23. Yang, C.Y., Liu, S., Yang, M.H.: Structured face hallucination. In: CVPR. (2013)