MMSR: A Multi-model Super Resolution Framework

Ninghui Yuan, Zhihao Zhu, Xinzhou Wu, and Li Shen(✉)

School of Computer, National University of Defense Technology, Changsha 410073, Hunan, China
lishen@nudt.edu.cn

Abstract. *Single image super-resolution (SISR)*, as an important image processing method, has received great attentions from both industry and academia. Currently, most super-resolution image reconstruction approaches are based on the deep-learning techniques and they usually focus on the design and optimization of different network models. But they usually ignore the differences among image texture features and use the same model to train all the input images, which greatly influence the training efficiency. In this paper, we try to build a framework to improve the training efficiency through specifying an appropriate model for each type of images according to their texture characteristics, and we propose *MMSR*, a multi-model super resolution framework. In this framework, all input images are classified by an approach called *TVAT (Total Variance above the Threshold)*. Experimental results indicate that our MMSR framework brings a 66.7% performance speedup on average without influencing the accuracy of the results HR images. Moreover, MMSR framework exhibits good scalability.

Keywords: Super resolution · Multi-model · General framework · Classification

1 Introduction

Super Resolution (SR) technique is used to recover a super-resolution[1] image from a single (or a series of) low-resolution image(s). This technique has been widely used in the fields including remote sensing, video, medicine and public security, etc. In recent years, with the wide application of deep learning, more and more researches focus on the study of *single image super resolution (SISR)*.

Since SRCNN [1] is proposed by Dong et al., deep convolution neural work has been the basis of other researches of super resolution. This work starts the deep-learning-based super resolution studies. VDSR [4] is also a revolutionary model in the development of super resolution, in which the residual block is firstly proposed and

[1] To distinguish between the output images and the reference images, the output images are called SR (Super-Resolution) images and the reference images are called HR (High-Resolution) images in this paper.

Published by Springer Nature Switzerland AG 2019
X. Tang et al. (Eds.): NPC 2019, LNCS 11783, pp. 197–208, 2019.
https://doi.org/10.1007/978-3-030-30709-7_16

used in a deep network. SRGAN [8], proposed by Christian Ledig et al., makes use of the generative adversarial network (GAN) [13] in super resolution for the first time.

There are still several problems and challenges in current super resolution studies. Firstly, researchers usually use the same model to train and reconstruct all the images, and they do not pay any attention to the differences of images features. For example, some images are smooth while other images have more textures. In general, for images with relatively simple texture features, a simple network model is enough to obtain satisfactory results, with a relatively short time overhead. Therefore, using the same model to train all the images will usually increase the time overheads and waste some computation resources. Secondly, researchers pay all their attentions on the quality of the result SR images, and they usually ignore the training or reconstruction efficiency. In fact, in particular scenarios, the efficiency is significant as well, such as scenarios having high real-time requirements. Thirdly, there is not a satisfactory criterion that can totally fit how the human eyes feel. MSE-based criteria usually make the output images too smooth, their visual results are usually not as good as expected.

In this paper, we focused on how to solve the first two problems. We found that images in the training dataset usually have different texture features. Some images have simple textures and others have complex textures. And we found that for images with different texture features, the most appropriate models are usually different. According to these observations, we proposed a *multi-model super resolution (MMSR)* framework. MMSR can choose a suitable network model for each image for training. MMSR shortens the training time efficiently without decreasing the quality of reconstruction image. We implement a MMSR framework based on SRGAN [8], and experimental results indicate that using *DIV_2K* as the training set, MMSR can reduce 40% training time on average. Moreover, the MMSR framework shows good stability. The main contributions of this paper are as follows:

(1) We proposed *MMSR*, a multi-model super resolution framework. This framework can choose a suitable model according to the texture characteristics of the input images. Therefore, it can improve the training efficiency without influencing the quality of the output SR images.
(2) We proposed *TVAT (Total Variance above the Threshold)*, a method to classify the training images. This approach can be used to describe the complexity of the image texture, and it does not introduce extra computational overheads. Moreover, since points with low pixel variations have almost no effect on the calculation of image texture, they could be removed to improve the accuracy of classification.

The rest of this paper is organized as follows. Section 2 lists some related works. Section 3 introduces our MMSR framework and the image classification method in detail. In Sect. 4, the performance of MMSR is evaluated and experimental results are given. And finally, in Sect. 5, some conclusions are given.

2 Related Works

In recent years, deep learning has been applied in many areas of image processing and analyzing, including super resolution [1–9]. Reference [1] is a pioneer work that brought super resolution into the deep learning area, in which the authors proposed a simple three-layer convolutional neural network called SRCNN and each layer sequentially deals with feature extraction, non-linear mapping, and reconstruction. The input of SRCNN uses an extra bicubic interpolation to enlarge the resolution of image. But this approach lacks enough high-frequency information and introduces some extra computations. Their later work, FSRCNN [2], removes the bicubic process and adds a deconvolution layer for reconstruction. VDSR [4] is another revolutionary work in the development of super resolution techniques, because the residual blocks are first used in its deep network. Almost all the successive researches on super resolution use residual blocks in their network models. SRGAN [8], proposed by Christian Ledig et al., makes use of (GAN) in super resolution for the first time.

Before the GAN network is used to solve the super resolution problem, the mean square error is often used as a loss function when training the network. Although a high peak signal-to-noise ratio can be obtained in this way, the reconstructed images lose some high-frequency details, which makes people hardly have a good visual experience. Figure 1 [8] describes the whole process of SRGAN, which consists of a generation phase using the Generator Network and an adversary phase using the Discriminator Network. In the last layer of the discriminator network, SRGAN uses perceptual loss to guarantee the quality of the output images. Perceptual loss describes the differences between the generated SR images and the reference HR images. If the perceptual loss of a SR image is larger than the threshold, the SR image will be regenerated.

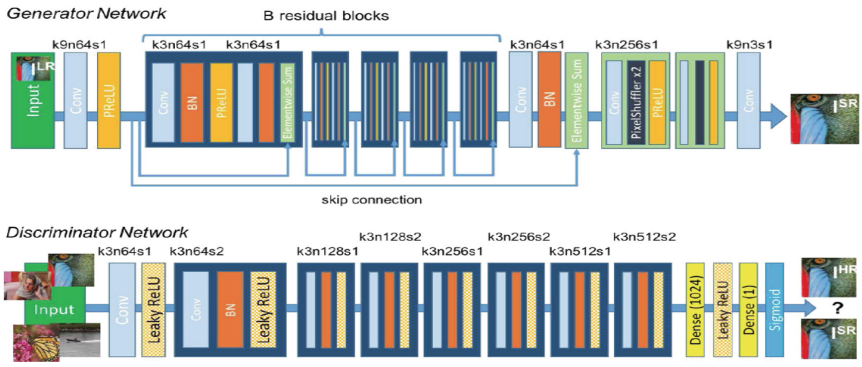

Fig. 1. Architecture of SRGAN with kernel sizes (k), numbers of feature maps (n) and stride (s) specified for each convolutional layer [8].

Most current super resolution approaches using deep learning techniques focus on the optimization of network models as well as the quality of the output SR images.

They did not care much about the efficiency of training, which may cause great waste of computation resources. Therefore, we propose a multi-model SR framework to improve the efficiency of training. Our framework is based on SRGAN, because it is a widely used in current super resolution studies and its reconstruction effect is better than other models.

Another problem concerned by researchers is how to evaluate the quality of output SR images. There are generally two categories of metrics. The first one describes the quality in terms of pixel features, such as MSE (Mean Square Error), PSNR (Peak-Signal to Noise Ratio), SSIM (Structure Similarity), etc. However, under the guidance of such metrics, the texture features of images are usually ignored and the output images tend to be too smooth or too fuzzy. The other one is based on the visual effect of human eyes, such as NIQE (Natural Image Quality Evaluator) [10] and PI (perceptual index). Sometimes the output images are shown and judged by the naked eyes. Obviously, the sharper and the more natural an image is, the better NIQE or PI value it can gets. In recent researches on super resolution, the second category of metrics gradually become the mainstream choice. Therefore, in this paper, we choose PI as the image quality metric. The PI value is calculated using the NIQE method [10].

3 MMSR Framework

In this section, we will first introduce our MMSR framework. The MMSR is composed of a training module and a reconstruction module. The training module trains the models with a train image set and the reconstruction module recovery the LR images to SR images. MMSR has good versatility and different deep learning network models can be integrated into this framework. Then, we will introduce the image classification method, the structure of the multi-model training module and the design of the reconstruction layer in turn.

3.1 Framework Overview

The first part of MMSR is the training module, which is shown in Fig. 2. It consists of two stages, a classification stage and a multi-model training stage. The classification stage divides the images into different categories according to their texture features. The multi-model training stage chooses an appropriate network model for each category of images and the classified images will enter the corresponding module for training. Using the classification module to classify the images can make the training process more targeted, and also improve the efficiency of training. The main difference among these network models is mainly that they use different parameters, such as the number of residual blocks in generator and the number of layers in discriminator.

The second part of MMSR is the reconstruction framework, which is shown in Fig. 3. It consists of four main parts, i.e. the segmentation layer, the classification module, the multi-model training module and the reconstruction module. The classification module and the multi-model training module are the same as those in Fig. 2. The segmentation layer is used to divide the input LR images to be reconstructed into a group of fragments. Then, these fragments enter the classification module and are

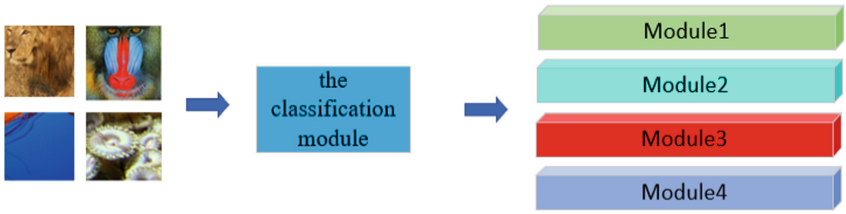

Fig. 2. The architecture of training module.

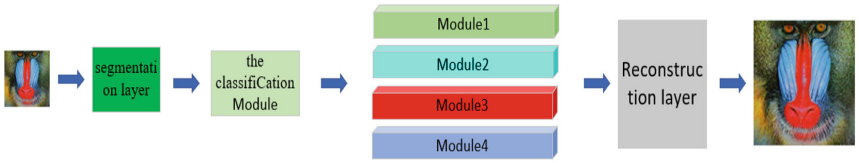

Fig. 3. The architecture of reconstruction module.

classified according to their texture features. After the classification (and network model assignment), these fragments will enter the corresponding modules for reconstruction. And finally, the reconstructed fragments are assembled by the reconstruction layer into a complete SR image.

3.2 Image Classification

In general, different images have different texture features. At present, most of the deep-learning-based SR approaches do not take the influence of the characteristics of the image on the training or the reconstruction process into account. This paper proposes a classification method to divide the images into several categories based on their texture features with low time overheads.

3.2.1 Total Variance Above the Threshold

We tested some images in order to observe the training methods of different fragments and the features hidden in them. We found that for most images, the more complex the texture of an image is, the longer the training time it requires. Therefore, we try to propose a suitable method to describe the texture feature of an image. The simplest way to describe the texture complexity is usually based on the variance of the whole image. However, we have found that this method does not work well, because the variance of some images with relatively uniform texture is large, although the variance of each point is relatively small. We found the training of these images does not require a very deep network model, but the method to assign the network models to image categories requires some training. Therefore, we consider using an innovative method to describe the variance of the pixel variation between each pixel and its 8 neighbor pixels, as shown in Fig. 4. In this paper, it is called the variance of single pixel (VSP). All VSPs in an image larger than a threshold are added together to obtain the total variance above the threshold (TVAT). The threshold is chosen through tests. We set the threshold to all

integers in 0–25 to test the classification effect. We find that when the threshold is set to 5, we can get the best effect, so we choose to set the threshold to 5.

The VSP value of the i-th pixel can be calculated as follows.

$$VSP_i = \sum_{j=1}^{8} ((Ri, Gi, Bi) - (Rj, Gj, Bj))^2 \tag{1}$$

The TVAT value of the whole fragment can be calculated as follows.

$$TVAT = \left(\sum_{i=1}^{n} VSP_i * judge_{index_i}\right) / (cols * rows) \tag{2}$$

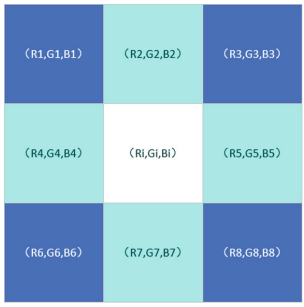

Fig. 4. The VSP of a pixel in the 3 × 3 Neighborhood.

Here *cols* and *rows* represent the number of columns and rows of the fragment respectively, and $judge_{index_i}$ is a step function, which is calculated as follows:

$$judge\_index_i = \begin{cases} 0, & \text{VSP}_i < threshold \\ 1, & \text{VSP}_i \geq \text{threshold} \end{cases} \tag{3}$$

3.2.2 TVAT Values

In this work, TVAT values are used to guide the image classification. In Table 1, *X-Y* means the GAN model has *X* residual blocks in generator and *Y* layers in discriminator. For example, *16-8* means that the GAN model has 16 residual blocks in generator and 8 layers in discriminator. We can find that for most images, the larger the TVAT value is, the more complicated an image is. However, when the depth of the network increases, the results do not always get better.

We randomly selected 80 image fragments from DIV_2K image set to test the recovery quality of these images in different models. We calculated the TVAT of images and found the following observations, as shown in Fig. 5: when the value of TVAT is relatively small (i.e. between 0 and 2), a 4-6 model can get the best performance. When the TVAT value is between 2 and 4, the 2-2 and 16-8 models have better performance. When the value of TVAT is large than 4, the 4-2 and 16-8 models perform best. Therefore, in this paper we classify images according to their TVAT values.

Table 1. The relationship between TVAT and the number of residual blocks in generator and the number of layers in discriminator in GAN.

Image number	TVAT	Perceptual index					
		2-2	4-2	4-6	8-6	16-8	Best
1	0.22316	10.7709	10.7093	8.9474	10.6772	13.4531	4-6
2	0.14125	10.7286	10.2412	9.8015	10.6265	11.0304	4-6
3	5.84019	6.8456	6.7848	6.9019	6.8478	6.4810	16-8
4	0.36391	14.9585	10.4451	14.3677	12.9602	11.1268	4-2
5	1.46280	7.0466	6.9128	7.6555	7.4745	7.7626	4-2
6	4.54511	5.8232	5.7170	6.0661	5.4763	5.3578	16-8
7	4.15810	6.4053	6.4270	6.8263	6.4317	6.6969	2-2
8	2.50022	6.8408	6.9172	6.8703	6.7201	7.0015	8-6
9	0.05513	12.8878	9.0682	14.9559	12.1362	14.2192	4-2
10	3.30724	6.6211	6.2182	6.9568	6.1581	6.1914	16-8
11	3.01382	7.2778	6.9914	6.6588	6.6947	6.5318	16-8
12	0.06423	11.7144	10.3977	10.3883	11.1998	11.4946	4-6
13	4.39549	6.6559	6.5765	6.7881	6.6619	5.8498	16-8
14	0.27429	11.2172	10.5885	9.8657	12.9558	10.6933	4-6
15	0.32317	9.6611	9.0914	8.961	9.6730	9.5313	4-6
16	5.95909	7.7423	7.5606	7.6745	7.7127	7.1994	16-8

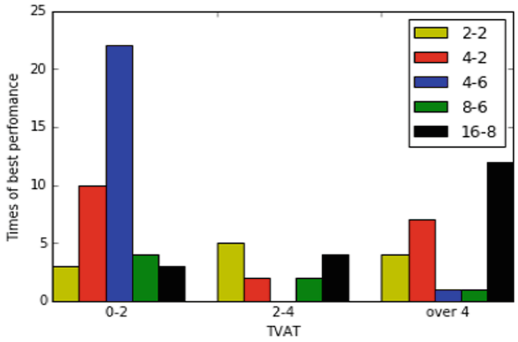

Fig. 5. Different network models are suitable for different TVAT values

3.3 Multi-model Training Module

After image classification, we need to use different models to train each class of images, as shown in Fig. 6. We deploy different training models on different GPU nodes. These models may be completely different kinds of deep learning network models, or the same kind of models with different depths. This paper chooses the second way because no matter for simple texture images or complex images, the recovery quality of SRGAN is better than previous works.

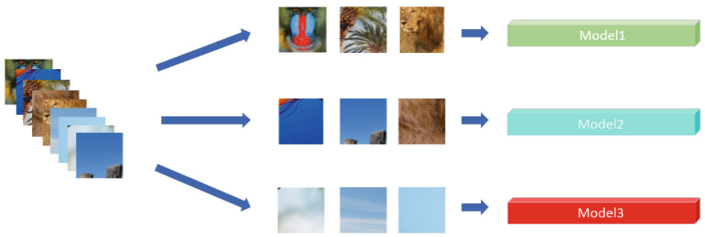

Fig. 6. Image fragments are input into different models based on their texture features.

3.4 Reconstruction Layer

As shown in Fig. 7, the reconstruction layer is used to recovery fragments into a complete SR image. Since different fragments will enter different models for training after an image is segmented, the recovery time of a set of images maybe different. After all the fragments reach the reconstruction module, the reconstruction module combines them into a complete SR image.

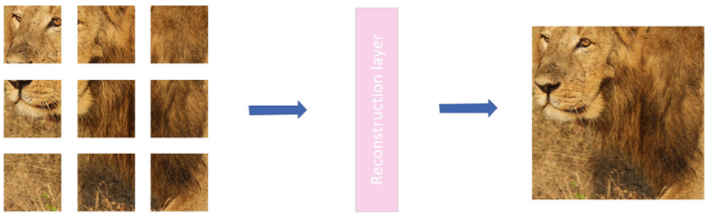

Fig. 7. The recovered fragments are combined into a complete SR image.

At the same time, some edge effects may be generated during the process of assembling the fragments into a complete image. As shown in Fig. 8, some overlapped image fragments are combined in the reconstruction module and the overlapping reduces edge effects. In our framework, the size of the overlap part can be adjusted according to users' requirements to ensure that the edge effects can be eliminated as possible.

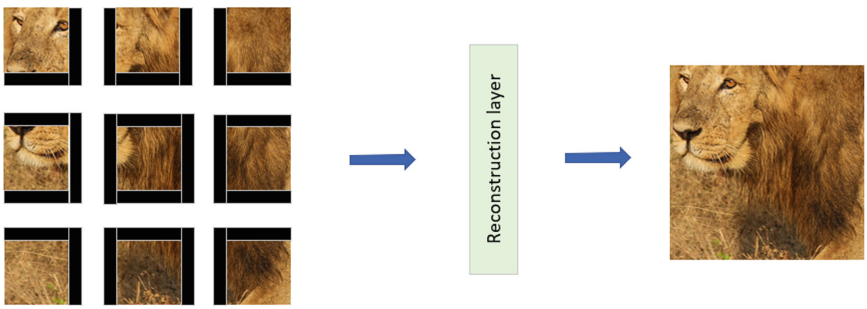

Fig. 8. The overlapped fragments are combined into a complete SR image.

4 Experiment Results

4.1 Environment Setup

We construct a cluster which consists of 4 CPU-GPU heterogeneous nodes to evaluate the performance and scalability of our MMSR framework. The main system parameters of each node are listed in Table 2.

Table 2. System parameters of each computation node.

HW/SW module	Description
CPU	Intel® Xeon® E5-2660 v3 @2.6 GHz x 2
GPU	NVIDIA Tesla K80 x 2
Memory	64 GB
OS	Linux CentOS 7.4
Development Environment	Anaconda 3, Pytorch 1.0

In this work, *DIV_2K* image set is used as both train set and test set. As a widely used image quality metric, the *perceptual index (PI)* value is used by us to compare different SR frameworks or models. The PI value can be calculated using following formula:

$$Preceptual\,index = 12((10 - Ma) + NIQE) \tag{4}$$

In formula (4), *NIQE* (Natural Image Quality Evaluator) is based on the construction of a "quality aware" collection of statistical features based on a simple and successful space domain natural scene statistic (NSS) model. And Ma is an effective and efficient metric to assess the quality of super-resolution images based on human perception, it uses three types of low-level statistical features in both spatial and frequency domains to quantify super-resolved artifacts, and learn a two-stage regression model to predict the quality scores of super-resolution images. Figure 9 shows that a lower perceptual index indicates better perceptual quality. We can see that mathematically that distortion and perceptual quality are at odds with each other [10–12].

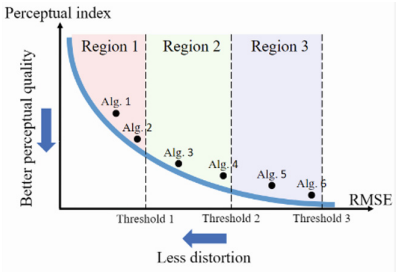

Fig. 9. The relationship between perceptual quality and distortion of images

The advantage of PI value is that different from the traditional image quality metrics. It can better match the senses of the human eye. Moreover, the GAN model itself tries to improve the sensory level of SR images. Therefore, using PI value as the metric can directly reflect the advantages of our MMSR framework.

4.2 Experiment Details

Since we need to classify the images into different parametric models for learning, we tested the training under different parameters, using the SRGAN model. Firstly, we input different texture complexity images into different models for training. We noticed that different types of images have different training effects under different model. In other words, in a limited training time, the training effect and the depth of the model are not necessarily positively correlated.

We choose the Python language to implement the framework. The classification module can divide the image into suitable block. In this experiment, we divide images into three types according to the TVAT value, and the images are sent to different GPU nodes to train. Finally, the whole image is merged into SR image. Our training time is shortened compared with 16-8 SRGAN [8] (i.e. standard SRGAN). The reconstruction effect of model trained by MMSR will not reduce obviously for most of the pictures, and the reconstruction effect of some pictures even increase. The training time of these three methods are listed in Table 3.

Table 3. Training time of different methods.

Method	Training time (s)	Average time (s)
SRGAN (one GPU node)	16415.600 17680.502 17417.912	17171.338
MMSR (one GPU node)	10683.279 10592.480 10598.944	10624.901
MMSR (three GPU nodes)	5921.761 5872.322 5890.451	5894.845

We trained all networks on PyTorch [14–16], which is an open source Python machine learning library based on Torch, used in the field of artificial intelligence. It can be seen the acceleration ratio of MMSR is about 1.62 on one GPU node. And when we use three GPU nodes for acceleration, the acceleration ratio of MMSR is about 2.9.

Figure 10 compares the reconstruction quality of MMSR with other methods. The smaller the value of PI is, the better the visual perception of the result image is, so we can find that the effect of the bicubic method is relatively poor, and the reconstruction effect of MMSR is not much different from that of SRGAN, and even achieves better results for some images (such as 4, 9 and 13).

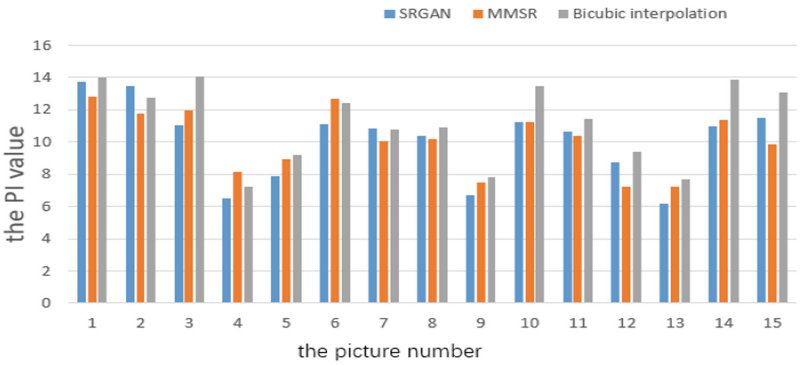

Fig. 10. The comparison of reconstruction quality of different approaches.

5 Conclusions

This paper proposes MMSR, a general multi-model framework for super-resolution image reconstruction. The highlight of our work is to build a general-purpose framework to improve the training or reconstruction efficiency of SR. To implement this framework, we propose a classification method based on experiments (TVAT) to classify the training set. This classification method can divide images into several categories according to their texture characteristics, and we input the images into the most suitable model to train. Experimental results show that the proposed framework is efficient to train the models and do not have too much impact on the training effect. Moreover, because we can use different models in MMSR, our framework has wide applicability.

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_13
2. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 391–407. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_25
3. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR (2016)
4. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **3**(6), 8 (2016)
5. Mao, X.-J., Shen, C., Yang, Y.-B.: Image restoration using convolutional auto-encoders with symmetric skip connections. In: The Annual Conference on Neural Information Processing Systems (NIPS), August 2016
6. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

7. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: IEEE International Conference on Computer Vision (ICCV) (2017)
8. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
9. Wang, X., Yu, K., Dong, C., Change Loy, C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
10. The Pirm Challenge on Perceptual Super Resolution. https://www.pirm2018.org/PIRM-SR.html
11. Blau, Y., Michaeli, T.: The Perception-distortion tradeoff. In: ECCV 2018
12. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
13. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS), pp. 2672–2680, March 2014
14. Yegulalp, S.: Facebook brings GPU-powered machine learning to Python. InfoWorld, 19 January 2017
15. Lorica, B.: Why AI and machine learning researchers are beginning to embrace PyTorch. O'Reilly Media, 3 August 2017
16. Ketkar, N.: Deep Learning with Python, pp. 195–208. Apress, Berkeley (2017)