Super-Resolution to Improve Classification Accuracy of Low-Resolution Images

Luke Jaffe
Stanford University
jaffe5@stanford.edu

Shiv Sundram
Stanford University
shiv1@stanford.edu

Christian Martinez-Nieves
Stanford University
chris151@stanford.edu

Abstract

Super-resolution is the process of increasing the resolution and quality of an image. Recently, deep-learning based super-resolution methods have been shown to outperform basic interpolation methods in terms of aesthetic value to humans and pixel signal to noise ratio. We investigate whether super-resolution can also be used to enhance the discriminative features of imagery, such that the transformed imagery is more amenable to classification. We test two methods for super-resolution, training them on a variety of different datasets, and then use super-resolution as a pre-processing step to classification training on multiple datasets. We show that super-resolution does not necessarily increase classification accuracy, and thus does make the classes any more separable to a convolutional neural-net classifier.

1. Introduction

The image classification problem has drawn significant attention in recent years, as part of a concerted effort to minimize error and reach or exceed human-like accuracy. Although other approaches for high accuracy image classification exist, convolutional neural networks are currently state-of-the-art for this task [3]. Since these convolutional neural network methods are well-developed, this enables us to focus on more complex image processing problems, such as those present when analyzing low-resolution images.

Although using high quality images for image analysis would be ideal, this is not always possible in practice e.g., attempting to identify relatively small objects in satellite imagery. In these cases, performing transformations to increase image quality may prove useful in the attempt to identify and classify less salient objects in the imagery. Some examples of quality-enhancing image transformations include denoising, colorization, and super-resolution.

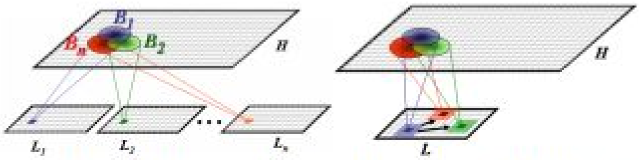
Since efficient image classification methods have already been developed, our main goal is to apply different super-resolution (SR) methods as a pre-processing step when attempting to train and test an image classifier. Further, we conduct experiments to compare image super-resolution methods to see which yield better results.

According to [1], either per-pixel loss or perceptual loss functions can be used to perform super-resolution. When training a super-resolution network with a per-pixel loss function, the goal is to minimize the per-pixel difference between the output and the ground truth image. When using the perceptual loss function, high-resolution images are generated by minimizing the differences between high-level image features of the output and ground truth, which are extracted from a pre-trained convolutional neural network.

Although both methods have potential pitfalls, we hypothesize that using super-resolution as a pre-processing phase will help yield higher classification accuracy for a network being tested with relatively low quality images.

2. Related Work

Image super-resolution methods attempt to recover a high-resolution image from one or more low-resolution input images [5]. Super-resolution methods can be classified in two main families: Multi-image Super Resolution methods, and Single Image Super Resolution methods (SISR). Multi-image Super Resolution methods attempt to use several low-resolution images of the same scene to determine new details in the high-resolution image, where each image imposes a set of linear constraints on the unknown high-resolution intensity values [11, 18]. However, these methods are not always practical, since multiple images of the same scene aren't always available. In addition, these methods tend to be limited to small increases in resolution.



(a) Classical Multi-Image SR (b) Single-Image Multi-Patch SR

Figure 1: (a) Illustrates how Multi-Image SR generally works, (b) Illustrates how Single Image SR generally works. Image from [5].

Single Image Super Resolution, on the other hand, attempts to generate a high-resolution image from a single low-resolution image [20]. Single Image super resolution methods, in turn, can also be divided into the following subcategories:

2.1 Interpolation Based SR

Interpolation based approaches attempt to interpolate the high-resolution image from the low-resolution input and are based on sampling theory. Such approaches tend to not be very effective, since they blur high-frequency details and have noticeable aliasing artifacts along edges [6]. Bicubic Image Interpolation is an example of this kind of SR methods, and is used throughout all of our experiments as a base comparison when upscaling images, since it's fairly common in photo editing software and even printer drivers.

2.2 Reconstruction/Edge Based SR

Reconstruction based methods produce high-resolution images by enforcing prior knowledge on the upsampled image, such as the assumption that edges are smooth along their contours. The appearance of the upsampled images also has to be consistent with the low-resolution version using back-projection [6]. The performance of such approaches depends on the prior used and its compatibility with the given image. According to [7], the use of smooth contour priors help to reconstruct the unknown pixels by interpolating along the contours of strong spatial edges. This leads to reconstructed edges that are both sharp and smooth along their contours [13]. An example of this is proposed by [7], where the proposed "Fast Edge-directed SISR" method uses an edge-directed interpolation operator as its main component, improving speed and stability, but it is limited to just a 2x scaling factor.

2.3 Statistical/Learning Based SR

Learning based methods estimate high-resolution details from a large training set of high-resolution images that encode the relationship between high and low-resolution images [6]. These methods are effective in generating missing details in high-resolution images based on similarities between the low-resolution image and the high-resolution training set. However, the effectiveness of such methods are also limited by the similarities between the training dataset and the test images. As discussed in [6], the author attempts to tackle super-resolution by combining both edge based and learning based techniques to produce suitable results, though the results are highly dependent on the gathered training data set, and tend to be computationally expensive [21].

While many deep-learning-based methods also exist for generating higher-resolution images from lower resolution inputs [4, 16, 17], these methods can produce significantly different results; as stated by [1, 20], super-resolution is inherently ill posed, meaning that for a single low-resolution image, multiple high-res images could be considered a valid upscaling. This is especially apparent in the recent work by Dahl et al [8], which extends the PixelCNN network architecture [15] and uses extremely low-resolution (8x8) face images to produce realistic pictures of faces that often look nothing like the ground truth. Other methods like LAPGAN [9] also use low-resolution/frequency seeds to generate realistic using images by feeding the input through a series of Generative Adverserial Networks [10].

While PixelCNN and LAPGAN use ultra low-resolution, almost incomprehensible images inputs (random noise, in the case of LAPGAN), to create higher resolution realistic images, we - like Johnson et al [1] and Shi et al [2] - wish to limit our input domain to *comprehensible* low-resolution images, where it is still possible to deduce the content of the image. We are thus interested in the class of methods whose inputs should have a reasonable resolution and classification accuracy to begin with. Other super-resolution methods include SRCNN [4], a three layer convolutional network that minimizes PSNR, and SRGAN [14], which like LAPGAN, uses Generative Adversarial Networks to generate the images.

The perceptual loss network we use for super-resolution can also perform style transfer, producing results similar to that of the Gatys et al Neural Style algorithm but in real-time [19].

As will be discussed throughout this paper, our approach falls under learning based SR, which uses features learned from convolutional neural networks to perform upsampling.

3. Methods

The goal of the project is to determine whether super-resolution can be used as a pre-processing step to improve image classification accuracy. Experiments are carried out using general image classification datasets, like CIFAR-10 and STL-10, as well as domain specific datasets like Food-101.

Thus, our framework consists of a two-stage process: the first stage consists of a super-resolution network that upscales each image in a dataset by x3 or x4, where the inputs are of size 32x32. When using the Johnson et al method, histogram matching is also used as a post processing step to make the colors match that of the inputs. This is necessary because differences in final pixel values are not used in perceptual loss functions. The specific architecture of the Johnson et al network is shown in figure 4, and details of the Shi architecture are shown in figure 3.

The second stage is the classification network, which is composed of a standard VGG net with a few layers added simply to account for the larger image sizes (128x128 of 96x96 as opposed to 32x32). A 16-layer VGG network was used for classifying the original 32x32 data, and a network with 4 additional layers (three basic blocks and one 3x3 stride 3 max pool) was built from that to fit the 96x96 upsampled data. To classify the 128x128 images, we used 8 additional layers on top of the VGG-16 (three basic blocks and one 2x2 stride 2 max pool, twice).

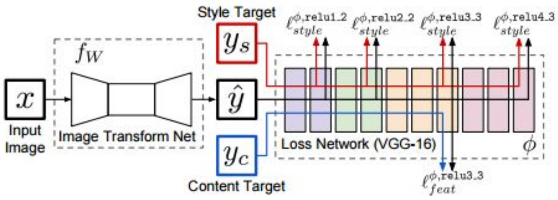

Style Target
$\ell_{style}^{\phi,relu1\_2}$
$\ell_{style}^{\phi,relu2\_2}$
$\ell_{style}^{\phi,relu3\_3}$
$\ell_{style}^{\phi,relu4\_3}$
f_W
y_s
x
$\hat{y}$
Input Image
Image Transform Net
y_c
Loss Network (VGG-16)
ϕ
Content Target
$\ell_{feat}^{\phi,relu3\_3}$

Figure 2: System overview of Johnson's et al [1] implementation of a perceptual loss network. The super-resolution network does not use the style-layers (used for style transfer), only the content relu3_3 layer. Images from [1].

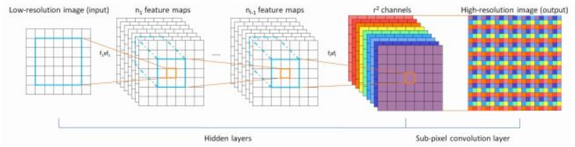
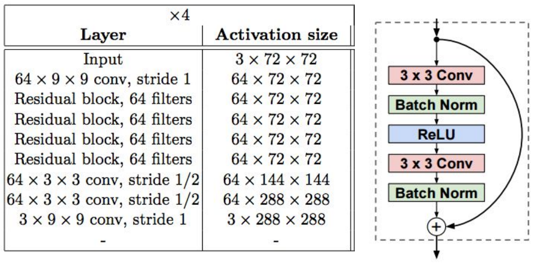


Figure 3: Shi et al [2] efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates feature maps from LR space and builds SR image in a single step. Image from [2].



×4	
Layer	**Activation size**
Input	3 × 72 × 72
64 × 9 × 9 conv, stride 1	64 × 72 × 72
Residual block, 64 filters	64 × 72 × 72
Residual block, 64 filters	64 × 72 × 72
Residual block, 64 filters	64 × 72 × 72
Residual block, 64 filters	64 × 72 × 72
64 × 3 × 3 conv, stride 1/2	64 × 144 × 144
64 × 3 × 3 conv, stride 1/2	64 × 288 × 288
3 × 9 × 9 conv, stride 1	3 × 288 × 288
-	-

Figure 4: Left: architecture overview of the Johnson et al [1] super-resolution network. Right: sub-architecture of the residual block layer mentioned in the left. Images from [1].

4. Experiments

A variety of super-resolution and classification experiments were conducted involving both general and domain specific datasets. In the following sections, we discuss what was done in detail, including the results obtained using super-resolution CNNs implementing both pixel loss and perceptual loss functions.

4.1 External Code

Our perceptual loss super-resolution network was based on a fast style transfer network from the implementation in [22]. We added the functionality necessary to perform super-resolution, with the exception of the histogram matching post-processing script, which we adapted from [23].

The pixel loss based super-resolution implementation was based on [24]. Finally, the VGG classifier code was adapted from [25].

4.2 Datasets

- CIFAR-10: Consists of 60,000 32x32 images with 10 basic classes (airplane, bird, car, cat, deer, dog, horse, frog, ship, truck), and 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images. This dataset was used both with and without pre-processing, in order to test the effects of super-resolution on classification performance.
- STL-10: Has 500 training images and 800 test images, with 10 classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). This dataset also contains 100,000 unlabeled images, useful for unsupervised learning. All images are of size 96x96.
- IMAGENET: Contains around 15 million labeled high-resolution images with approximately 22,000 categories. Both the image distribution per category, as well as the image size varies. Images are split about 50/50 between train and test sets.
- Microsoft COCO 2014: Contains ~80,000 labeled high-resolution training images with 80 categories. Image resolution varies per image, and image distribution is ~80,000 for train set, ~40,000 for validation set, and ~40,000 for test set (based on MS COCO paper revised in 2014).
- Food-101: Consists of 101,000 images with 101 food categories, each class having 1,000 images. Each category in turn has 750 training images, as well as 250 test images. All images are rescaled to have a maximum side length of 512 pixels, and all images contain some amount of noise.

4.3 Pixel Loss

We conducted our first experiments on the CIFAR-10 dataset. For the first experiment, two approaches for upsampling were employed as a pre-processing step to classification. The first approach was that of Shi et al. [2] that uses a standard per-pixel loss function to train a set of upscaling filters. We will refer to this as the pixel loss method. The second approach was a standard bicubic interpolation, mainly to be used as a control.

In this first experiment, we trained the pixel loss method on the unlabeled partition of the STL-10 dataset. This partition has 100k images, a good size set for our learning problem. We trained the Shi network for 100 epochs, with a learning rate of 1e-3 using an Adam optimizer. The network was trained to upscale by a factor of three, since STL-10 consists of 96x96 images, and CIFAR-10 consists of 32x32 images. We hypothesize that using a scaling factor which scales the target imagery to match the training imagery is a good strategy. In addition, random crops and random flips were applied to the training data for all trials.

We trained and tested a standard VGG-net classifier on CIFAR-10 for the original data, the data upsampled with bicubic interpolation, and the data upsampled with the trained pixel loss network. We wanted to keep the networks used on each set as similar as possible to reduce experimental variables.

4.4 Perceptual Loss

For our second experiment, we used the method of Johnson [1] et al, which leverages a perceptual loss objective. This objective calculates loss by penalizing differences in higher level features, which in this case are determined by a VGG loss network pre-trained for classification.

We then compared the performance of the classifier on data pre-processed with this method to the performance of our results from the first experiment. The pre-trained loss network was trained on ImageNet [24], while the super-resolution network was trained on Microsoft COCO.

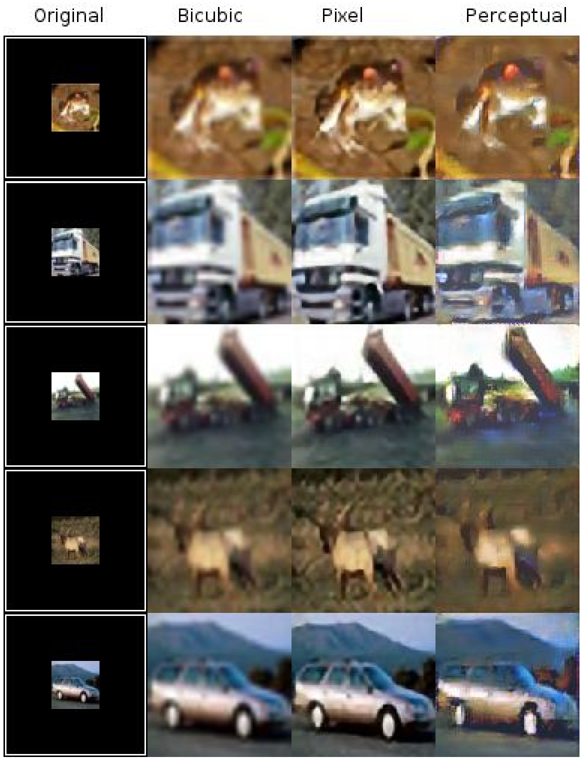


Figure 5: Original CIFAR-10 images are juxtaposed against versions transformed with upsampling methods.

In figure 5, we show the result of applying the pixel loss method, trained on STL-10, to some exemplars of CIFAR-10. We also show the result of the perceptual loss method, trained on COCO and applied to CIFAR-10. The images upsampled with the pixel loss method are of the highest aesthetic quality.

In figures 6 and 7 we show the results of training and testing per epoch for the original, bicubic upsampled, pixel loss upsampled, and perceptual loss upsampled CIFAR-10 data. The networks are able to overfit the training data for all methods, but we see distinctions in the test data performance. In particular, the perceptual loss method performs worst, while the bicubic and pixel loss methods perform nearly identically.

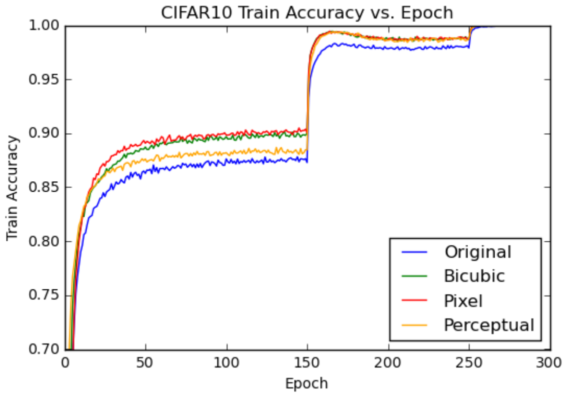


Figure 6: Train accuracy v. epoch for original, bicubic upsampled, pixel loss super-resolution, and perceptual loss super-resolution CIFAR-10 data.

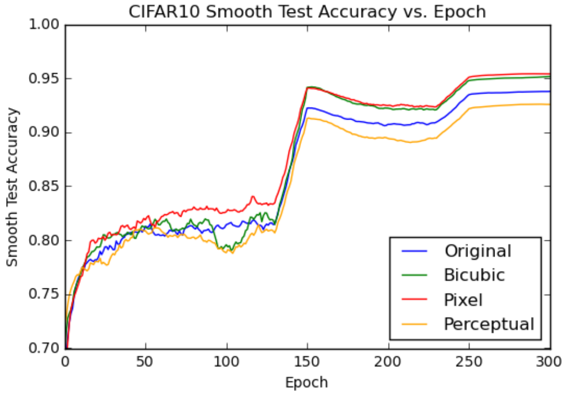


Figure 7: Test accuracy v. epoch for original, bicubic upsampled, pixel loss super-resolution, and perceptual loss super-resolution CIFAR-10 data. Accuracies are moving averages over 20 preceding points.

4.5 Domain Restriction

In our final experiment, we restricted the domain of the super-resolution network to better match the data used for classification. Specifically, we split the Food-101 dataset into three partitions: 50%

for training a super-resolution network, 25% for classifier training data, and 25% for classifier testing data. All partitions contained an equal number of samples from each class.

We hypothesized that upscaling the classifier partitions with the super-resolution network trained on data from the same set was more likely to add discriminating features, and potentially improve classification performance. We were careful not to use any of the same images for training the super-resolution network and classifier, as this would trivialize the problem.

The Food-101 dataset is large enough to allow for sufficient data to train the super-resolution network and the classifier, but still contains enough variability to make the experiment interesting i.e. the experiment could apply to real computer vision problems.

We used both the pixel and perceptual loss methods for this experiment, since the first experiment offered insufficient evidence that either might be ineffective in this case. For the pixel loss method, we trained the Shi network for 100 iterations over the "super" partition of the Food-101 data (50.5k images). For the perceptual loss method, we trained the Johnson network for 150 epochs over the same data partition. These networks were trained to upscale by a factor of four. Examples of super-resolved images using these methods, in addition to the naive bicubic interpolation and the ground truth are shown in Figure 8. Here, ground truth means the original imagery at 128x128 resolution.

Both super-resolution networks were applied to the entire train and test partitions to generate upsampled imagery for the classification experiment. The classification experiment parameters were similar to those used in the first experiments. The network used to train the downsampled "original" imagery (32x32) was identical to the one used for the original CIFAR-10 imagery. The network used to train the upsampled/ground truth imagery (128x128) was similar, but had additional conv layers and max-pooling layers to appropriately downscale the feature layers. The training schemes were identical to those used in the first experiments, with 350 training epochs, and a manually adjusted learning rate.

The results of these classification experiments are shown in Figures 9 and 10. Most importantly, we see that the original imagery performed worst, and the ground truth imagery performed best for test accuracy. We assume that the best possible result of an upsampling algorithm would produce the ground truth image, and that the ground truth images are optimal for the classification task. While it is possible that an upsampling algorithm could produce images better for classification than the ground truth, our super-resolution methods are unlikely to produce this type of result.

The pixel loss and bicubic methods performed nearly identically. This suggests that neither method adds any discriminative information. On the other hand, the perceptual loss method performed markedly worse. This suggests the type of features it added are actually worse than the original image content for discrimination. Figure 8 shows that all upsampled (and ground truth) sets were easily overfit during training.

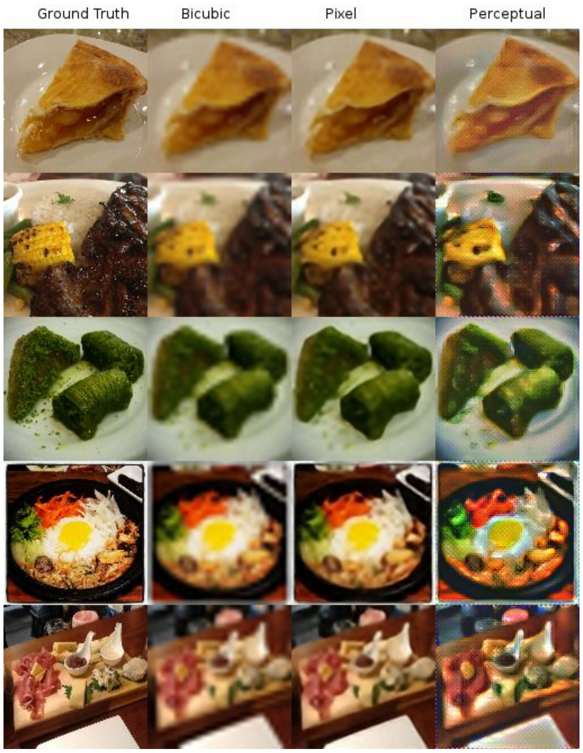


Figure 8: Ground truth Food-101 images are juxtaposed against bicubic upsampled (middle) and upscale filtered (right).

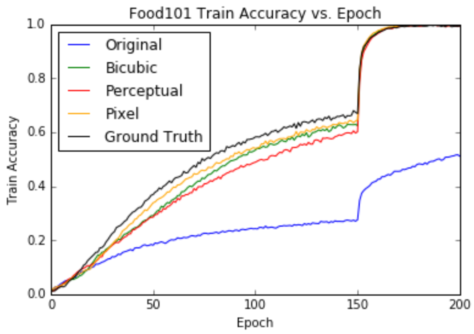


Figure 9: Train accuracy v. epoch for original, bicubic upsampled, and upscale filtered (Super) data.

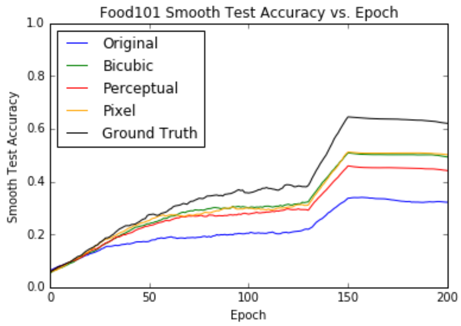


Figure 10: Test accuracy v. epoch for original, bicubic upsampled, and upscale filtered (Super) data. Accuracies are moving averages over 20 preceding points.

5. Conclusion/Future Work

Our experiments show that our original hypothesis was disproved, at least for the methods and scenarios that we used. Specifically, these super-resolution methods do not improve the discriminative capacity of the imagery used for classification. This shows that the aesthetic quality these methods sought to improve did not correspond to bringing the imagery any closer to the ground truth in the feature space. We suspect that incorporating this discriminative goal into the objective would be crucial to making super-resolution effective as a preprocessor for classification.

In the future, we would like to try some of the other super-resolution methods highlighted in section 2. We think a type of LAPGAN could be promising, because LAPGANs train a discriminator in the process of generating fooling samples. This discriminator could encourage generated samples to be effective for training in the classification context. If we could modify the input of LAPGAN to take downsampled images instead of a random seed, it could be effective for super-resolution focused on class separation instead of aesthetic quality.

References

[1] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." *European Conference on Computer Vision*. Springer International Publishing, 2016.

[2] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[4] Dong, Chao, et al. "Learning a deep convolutional network for image super-resolution." *European Conference on Computer Vision*. Springer International Publishing, 2014.

[5] Glasner, Daniel, Shai Bagon, and Michal Irani. "Super-resolution from a single image." *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009.

[6] Tai, Yu-Wing, et al. "Super resolution using edge prior and single image detail synthesis." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.

[7] Rouf, Mushfiqur, et al. "Fast edge-directed single-image super-resolution." *Electronic Imaging* 2016.15 (2016): 1-8.

[8] Dahl, Ryan, Mohammad Norouzi, and Jonathon Shlens. "Pixel recursive super resolution." *arXiv preprint arXiv:1702.00783* (2017).

[9] Denton, Emily L., Soumith Chintala, and Rob Fergus. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks." *Advances in neural information processing systems*. 2015.

[10] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems. 2014*.

[11] Borman, Sean, and Robert L. Stevenson. "Super-resolution from image sequences-a review." *Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on*. IEEE, 1998.

[12] Bruna, Joan, Pablo Sprechmann, and Yann LeCun. "Super-resolution with deep convolutional sufficient statistics." *arXiv preprint arXiv:1511.05666*(2015).

[13] Vanam, Rahul, Yan Ye, and Serhad Doken. "Joint edge-directed interpolation and adaptive sharpening filter." *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013.

[14] Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).

[15] Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks." *arXiv preprint arXiv:1601.06759* (2016).

[16] Dong, Chao, et al. "Image super-resolution using deep convolutional networks." *IEEE transactions on pattern analysis and machine intelligence*38.2 (2016): 295-307.

[17] Cui, Zhen, et al. "Deep network cascade for image super-resolution."*European Conference on Computer Vision*. Springer International Publishing, 2014.

[18] Li, Xuelong, et al. "A multi-frame image super-resolution method." *Signal Processing* 90.2 (2010): 405-414.

[19] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).

[20] Kim, Kwang In, and Younghee Kwon. "Single-image super-resolution using sparse regression and natural image prior." *IEEE transactions on pattern analysis and machine intelligence* 32.6 (2010): 1127-1133.

[21] Jagtap, Mr SH, M. M. Patil, and S. D. Ruikar. "Single Image Super-Resolution."

[22] Tejani, Shafeen. “Fast-Style-Transfer.” GitHub, 12 Jan. 2017, github.com/ShafeenTejani/fast-style-transfer. Accessed 12 June 2017.

[23] Johnson, Justin. “histogram_matching.Py.” GitHub, gist.github.com/jcjohnson/e01e4fcf7b7dfa9e0dbee6c53d3120b6. Accessed 12 June 2017.

[24] Tejani, Alykhan. “Pytorch/Examples.” GitHub, 28 Mar. 2017, github.com/pytorch/examples/tree/master/super_resolution. Accessed 12 June 2017.
[25] Liu, Kuang. “Liu/Pytorch-Cifar.” GitHub, 7 June 2017, github.com/kuangliu/pytorch-cifar. Accessed 12 June 2017.