Super resolution with Generative Adversarial Networks

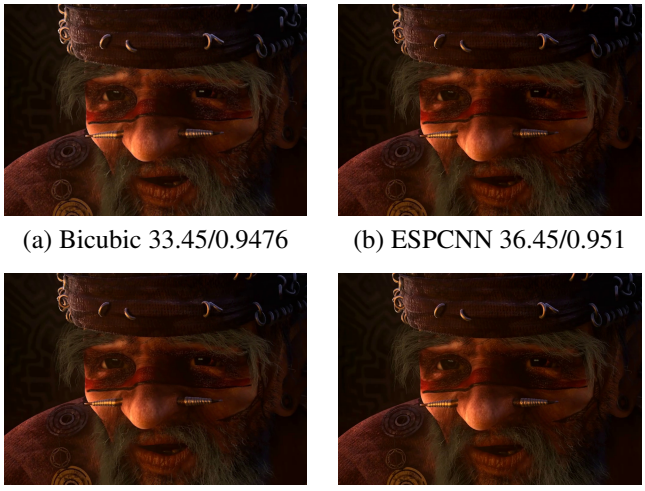
Boris Kovalenko
SUID 06201315
kboris@stanford.edu

1. Abstract

Single image super-resolution is an approach for improving imaging system. Recently, models based on convolutional neural networks, specifically designed for this task, became a topic of research and shown great results. Models based on convolutional neural networks outperform other approaches in terms of image quality metrics such as peak signal to noise ratio (PSNR) and structural similarity (SSIM). The perceptual image quality of resulting super-resolved image is principally dependant on choice of a loss function, which is optimized during model training. Recent work is largely based on optimizing mean squared reconstruction error. Such loss does improve wieldy used image quality metrics. But this reconstruction metrics like PSNR and SSIM may not capture fine details in the image and give high scores to images with unsatisfying quality. Most recently model based on generative adversarial networks (GAN) was proposed for the task. In GANs settings adversarial loss, pushes the reconstructed image to natural image manifold using a discriminator model. The adversarial loss is combined with reconstruction loss to limit model "fantasy". The discriminator model main task is to differentiate between low-resolution images, which were super-resolved and high-resolution images. Extensive quality testing including metrics with a use of human opinion had shown that this approach generates more pleasant images. The goal of this project is to implement super-resolution model based on GAN and test quality of image reconstruction.

2. Introduction

Recently task of super-resolution received substantial attention from researchers within machine learning community [10, 15, 14, 2]. This task has a broad range of applications in medicine, cosmology, computer graphics, etc. The task of super-resolution is highly challenging and ill-posed. The main challenge for upscaling algorithms is to reconstruct image texture detail. Commonly, mean squared error loss (MSE) is chosen for optimization. This loss is measured between pixels of high-resolution original and upscaled low-resolution counterpart. It's convenient to optimize MSE since it's also optimized PSNR metric, which is commonly used to measure image quality. However, PSNR and other popular metrics SSIM have the main drawback: its value is not always well correlated with human judgment. In other words, for human, the reconstructed image with high metrics values doesn't necessarily look better than the image with lower metrics values. An example of this is shown in Figure 1, where the image with highest PSNR is not the best in terms of quality. And since MSE is mean the difference between pixels, it's common for models which trained to minimize MSE to produce over-smoothed images.



(a) Bicubic 33.45/0.9476 (b) ESPCNN 36.45/0.951

(c) GAN 35.2/0.94 (d) Original image

Figure 1. Figure 1: From left to right: bicubic interpolation, efficient sub-pixel convolutional network optimized for MSE, deep residual generative adversarial network optimized for a composite loss. Corresponding PSNR and SSIM are shown in brackets. Image is a frame 3498 from Sintel CGI movie (x2 upscaling)

In this project, we focus on a model described in [10], which is super resolution generative adversarial network (SRGAN). As a generator in this model, we employ deep residual network (ResNet) with skip-connection. We focus on composite loss, which consists of content loss (we use

MSE for this part) and adversarial loss, which is produced during a min-max game between generator and discriminator models.

3. Related work

Most basic way to approach super-resolution problem is filtering approach, using linear or bicubic, or Lanczos [4] filters. These algorithms are based on the idea of pixel neighborhood similarity, they are very fast. Their main problem is that they over-simplify the problem and produce over-smoothed images.

Machine learning based approaches are require training data. As training data usually pairs of low and high-resolution images are used. Recent popular approaches can be classified into edge preservation [17], image-statistic based [19, 7] and patch based [20, 12, 18].

One of the most popular approaches is sparse coding. This approach assumes, that images can be sparsely represented by a dictionary of atoms in some transformed domain [5, 13]. The dictionary is learned during the training process. The main drawback is that optimization algorithm used for training are computationally expensive.

Image super-resolution with the use of neural networks became popular recently. Neural network based solutions require big datasets, like ImageNet in order to infer mapping between low-resolution and high-resolution images. Most notable models are [2, 15, 8, 9]. In this papers, different architectures of neural networks and losses for optimizations are proposed. In architectures of convolutional neural networks, geared for super-resolution, a usually first step of image processing is computationally cheap image upscaling. Commonly, in this step bicubic filtering is used. Behind this approach is an idea, that neural network will be able to learn a way to "fix" bicubic filtering and produce a higher quality super-resolved image. In [15] different, yet similar model proposed. This model utilizes novel sub-pixel layer, such approach computationally more efficient than previously proposed models. The main idea behind this architecture is that image filtering better than bicubic filtering can be learned during the training process and there is no need to work in high-resolution space, all work can be done efficiently in low-resolution space.

Minimizing MSE makes the final solution to be an average of plausible solutions. This means that final solution will be over-smoothed and may not be perceptually satisfying [8, 1, 3]. Illustration of this problem presented in figure 2.

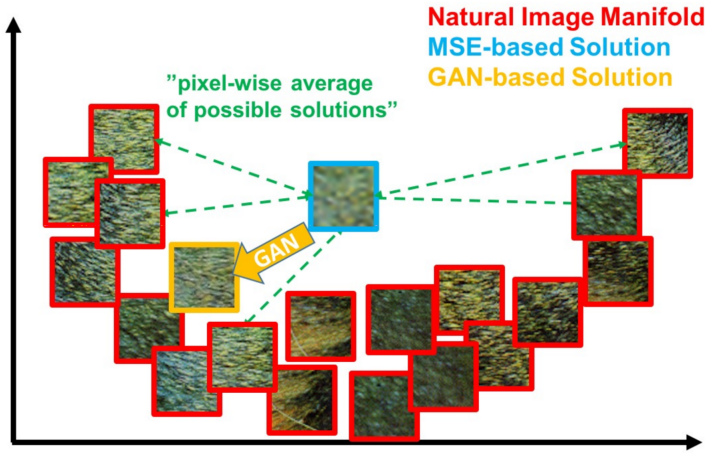


Figure 2. Multiple potential solutions are averaged to produce MSE smooth reconstruction.

Described problem can be tackled with generative adversarial networks. GAN's are successfully used for image generation[6], style transfer[11], inpainting[3], etc. Augmentation of pixel-wise MSE loss with an adversarial loss from discriminator makes generated images visually superior to images generated by models, trained to optimize only MSE.

4. Approach

In the problem of single image super-resolution main task is to estimate I^{HR} from its low-resolution counterpart I^{LR}. During training pairs of images (I^{LR}, I^{HR}) are used by backpropagation algorithm to optimize network weights. Training dataset usually obtained from high-resolution images, by applying a Gaussian filter to I^{HR}, followed by downsampling operation with factor k. Image I^{LR} described by a tensor of size HxWxC, image I^{HR} described by the tensor of size kHxkWxC, where C - the number of color channels.

Generator function G estimates I^{HR} from I^{LR}. This function is parametrised by θ_G, where $\theta_G = W_{1:L}, b_{1:L}$. This denotes weights and biases of L - layer deep neural network. The parameters are obtained during backpropagation. To achieve good reconstruction of I^{HR} we need specific loss function l^{SR}, which can be minimised using backpropagation. We can summarise our objective:

$$\hat{\theta}_G = \underset{\theta_G}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} l^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR})$$

where I_n^{LR}, I_n^{HR} - training images from a dataset of size N. l^{SR} is composite loss and has two parts: content loss and adversarial loss.

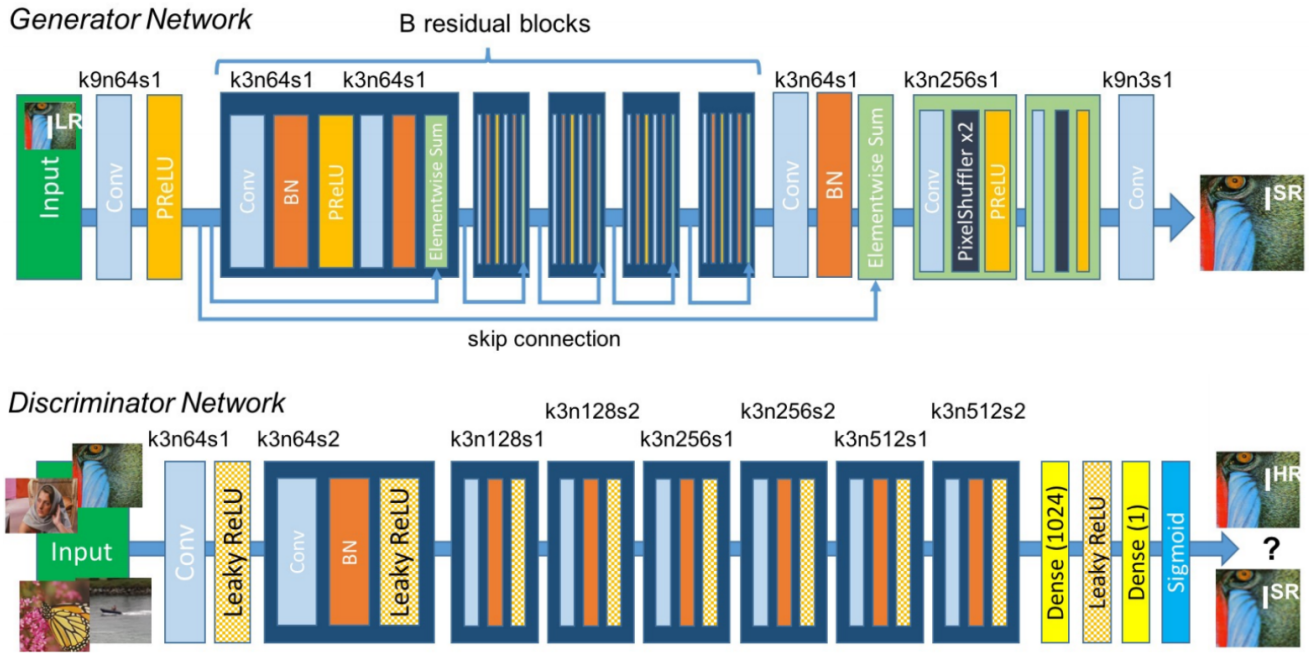


Figure 3. Architectures for generative network G_{θ_G} and for discriminator network D_{θ_D}

Discriminator network D_{θ_D} is optimised in alternating manner, with G_{θ_G}. This approach is used to solve adversarial min-max problem:

$$\min_{\theta_G} \max_{\theta_D} E_{I^{HR} \sim p_{train}(I^{HR})}[log D_{\theta_D}(I^{HR})] +$$

$$E_{I^{LR} \sim p_{train}(I^{LR})}[log(1 - D_{\theta_D}(G_{\theta_G}(I^{HR})))]$$

The main goal is to train generator such that it is able to fool discriminator model by producing super-resolved images indistinguishable from real high-resolution images. The main idea of the discriminator is to learn to differentiate super-resolved images from real high-resolution images. This approach makes super-resolved images more similar to real images.

Generator network consists of B residual blocks, architecture is depicted in figure 3. The single block consists of two convolutional layers with 64 kernels of size 3x3. Feature maps are followed by batch normalization and ReLu non-linearity. Image resolution is increased with the use of single or two sub-pixel layers.

The architecture of discriminator network is depicted in figure 3. As non-linearities we can use ReLu or LeakyReLu. The model has 8 layers, with kernels of size 3x3. Layer depth increasing from beginning to an end with the x2 factor, from 64 to 512, similarly to VGG architecture [16].

Perceptual loss is formed as weighted sum of content loss and adversarial loss:

$$l^{SR} = l^{SR}_{Content} + 10^{-3} l^{SR}_{Adversarial}$$

Content loss is pixel-wise difference between super-resolved and original image:

$$l^{SR}_{Content} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2$$

This loss will allow us to achieve high PSNR levels. To improve perceptual quality of upscaled images, we will add adversarial loss:

$$l^{SR}_{Adversarial} = \sum_{n=1}^{N} -log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

This loss pushes upscaled image closer to manifold of natural images. Here $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ is probability, that upscaled image is a natural image. Instead of $log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))$ we will minimise $-log D_{\theta_D}(G_{\theta_G}(I^{LR}))$ as discussed in [6] it improves gradient behavior.

Also additional loss, which was tested is least squares GAN (LS GAN):

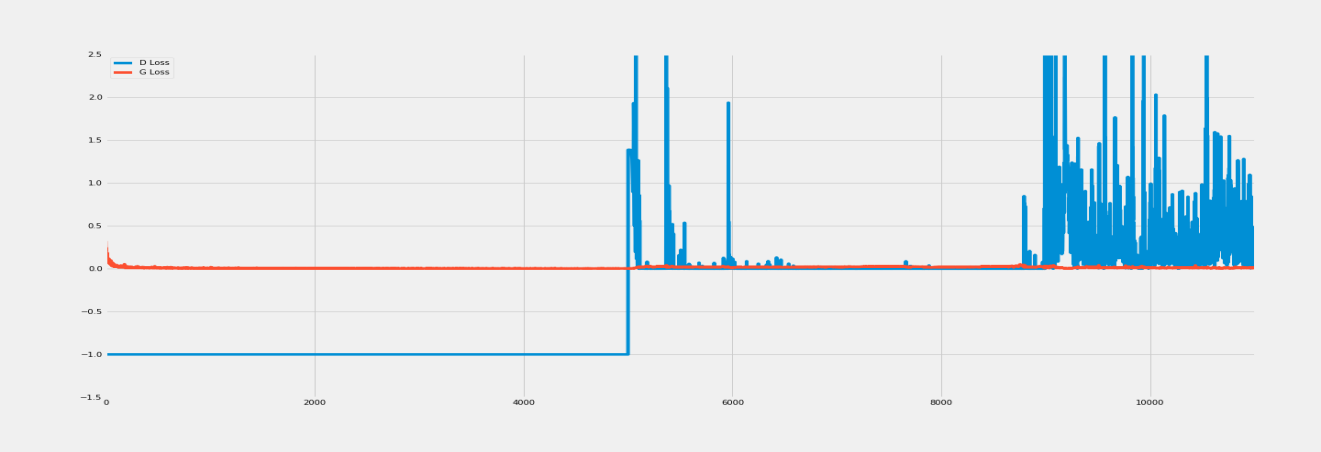


Figure 4. Discriminator and generator loss for simple GAN loss

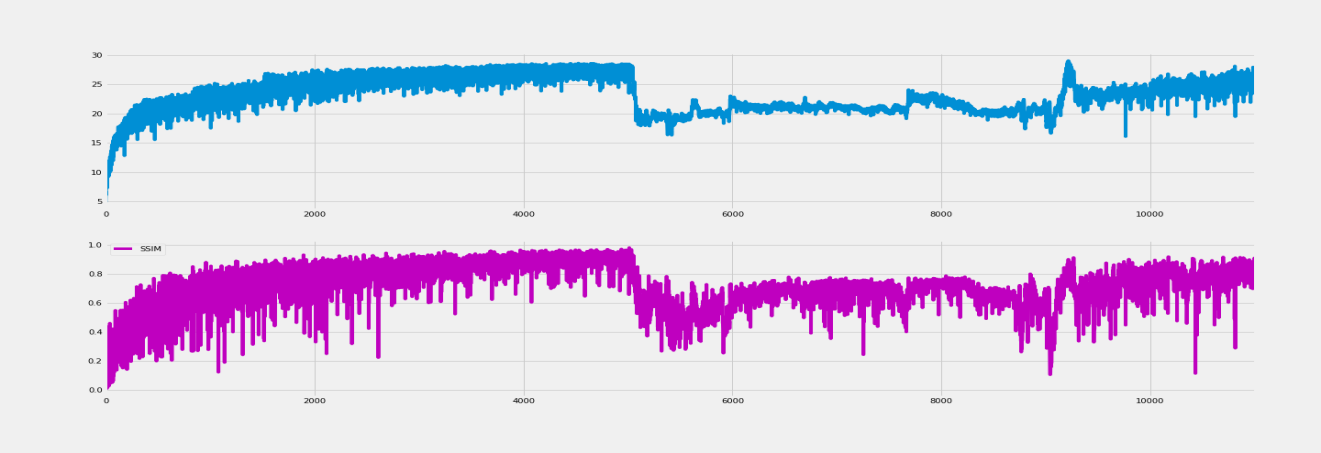


Figure 5. PSNR and SSIM metrics for simple GAN loss

$$l^{SR}_{Adversarial} = \frac{1}{2}\sum_{n=1}^{N}(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))^2$$

5. Experiments

5.1. Data and metrics

As training dataset, we used a sample of 50000 images from ImageNet. During training these images were converted into a YCrCb format, normalized and sliced into tiles of size 64x64. To create pairs of (I^{LR}, I^{HR}) for training, different downscaling functions could be applied to downscale original image. Gaussian filter with downsampling simulates low-resolution imaging system, while max(X), where X is a block of 4 pixels is a just simple approach to creating training set. During experiments we tried both and established that model does well in both cases. Choice of the downscaling function should be dictated by domain of model deployment. Following results are obtained for the Gaussian filter with downsampling as a downscaling function.

For model performance testing we chose BSD100 dataset. All experiments were performed with factor x2. For reference, we compare our GAN based model with bicubic and ESPCNN models.

As metrics we used two most popular image reconstruction metrics: PSNR and SSIM. PSRN metric can be calculated using MSE:

$$PSNR = 20\log_{10}(MAX_I) - 10\log_{10} MSE$$

with MAX_I - a max level of intensity, usually 255.

SSIM metric is calculated on various windows of an image. The measure between two windows x and y y of common size N x N is:

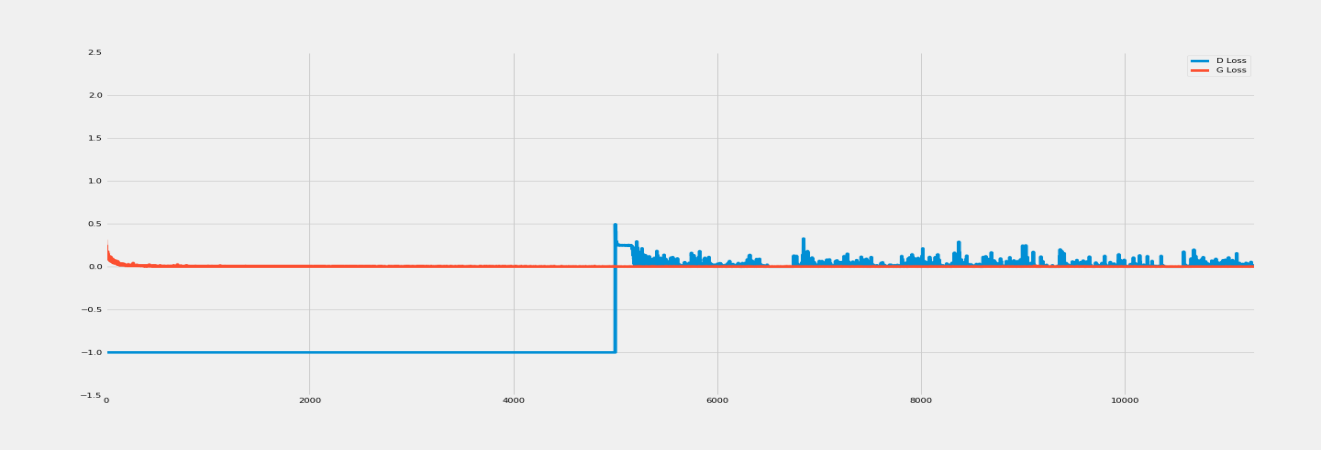


Figure 6. Discriminator and generator loss for LS GAN loss

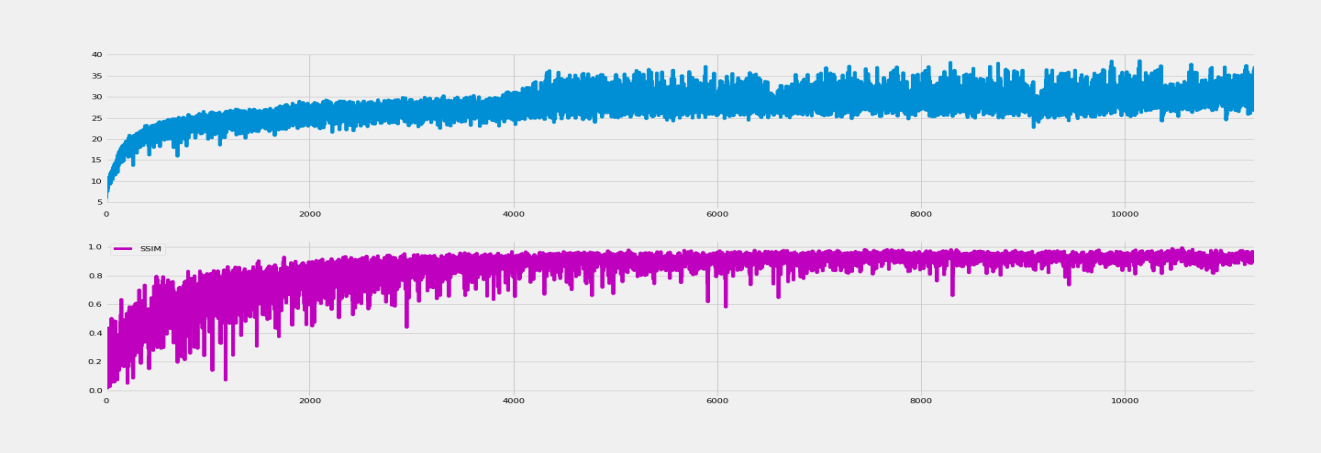


Figure 7. PSNR and SSIM metrics for simple LS GAN loss

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with c_1 and c_2 - two variables to stabilize the division with weak denominator. μ_x, μ_y - average value for window. σ_x^2, σ_y^2 - variance value for window. σ_{xy} - covariance for windows x and y.

5.2. Training details

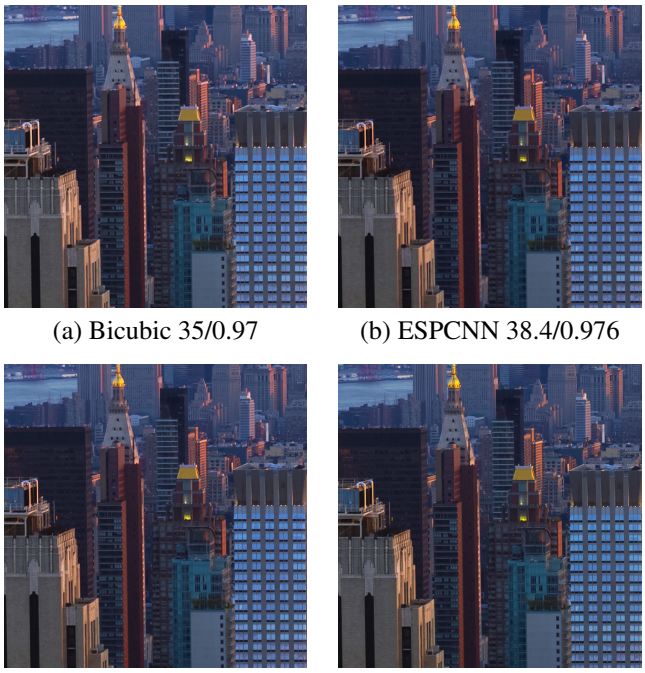
All networks were trained on Nvidia P40 GPU. As a framework to implement model we used Tensorflow 1.1. Our generator network has 5 residual blocks. For optimization, we use Adam for both generator and discriminator. Learning rate for both models was set to 10^{-4}. Batch size was set to 256 tiles, weights updates are alternated. The generator has warmup period when only MSE loss is optimized. This period is active in first 5000 steps after 5000 steps are done, discriminator becomes active.

5.3. Training process

To make sense of what is going on during training we monitored discriminator and generator loss, as well as image quality metrics like PSNR, SSIM. On figure 4 and 6 we can see plots of discriminator and generator loss, during training for simple GAN loss and for LS GAN loss. On figures 5 and 7 we can see plots of images quality metrics, measured on validation set. It seems that LS GAN provides better convergence properties and less noisy.

5.4. Model performance

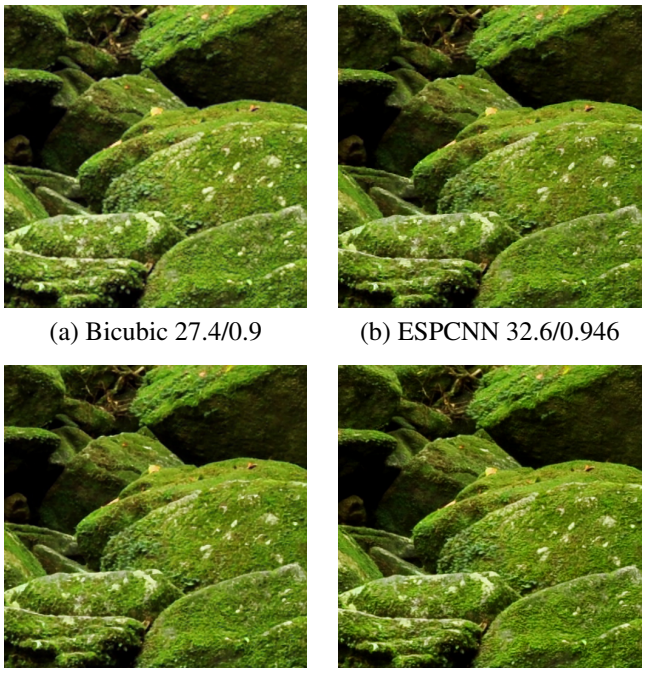
Model performance is measured on BSD100 dataset.



(a) Bicubic 35/0.97 (b) ESPCNN 38.4/0.976

(c) GAN 35.1/0.975 (d) Original image

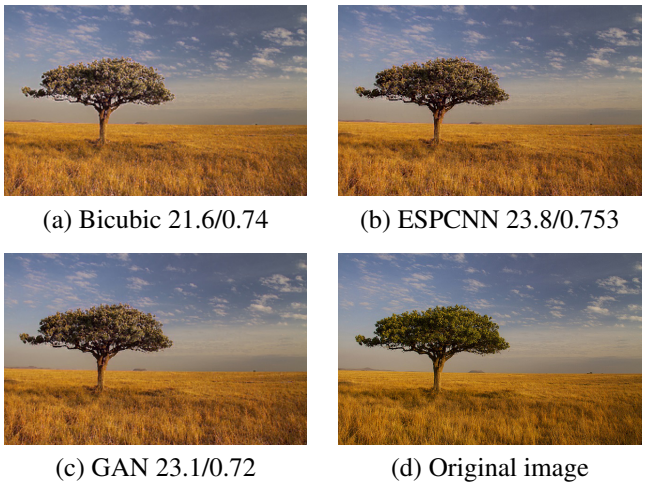
Figure 8. Comparison of upscaled images



(a) Bicubic 27.4/0.9 (b) ESPCNN 32.6/0.946

(c) GAN 32.5/0.93 (d) Original image

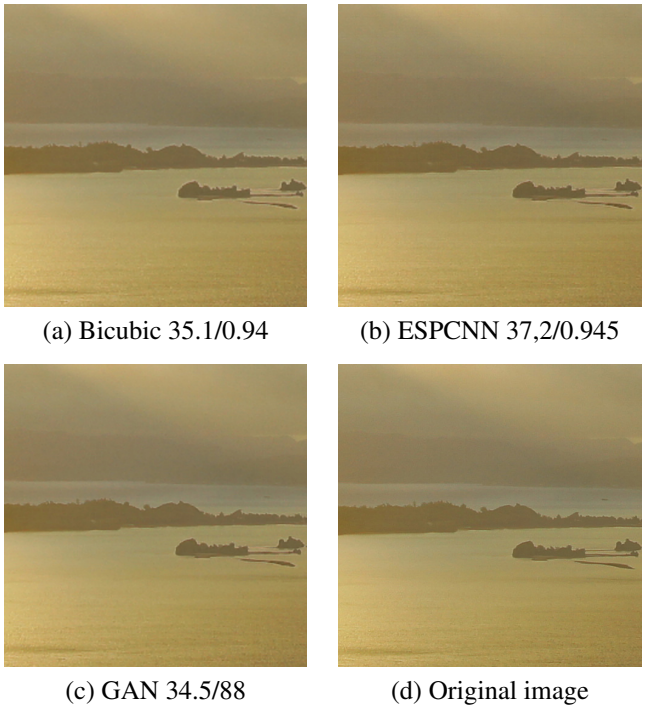
Figure 9. Comparison of upscaled images



(a) Bicubic 21.6/0.74 (b) ESPCNN 23.8/0.753

(c) GAN 23.1/0.72 (d) Original image

Figure 10. Comparison of upscaled images



(a) Bicubic 35.1/0.94 (b) ESPCNN 37,2/0.945

(c) GAN 34.5/88 (d) Original image

Figure 11. Comparison of upscaled images

Model	PSNR	SSIM
Bicubic	25.94	0.6606
ESPCNN	26.71	0.7312
GAN	25.66	0.658

Table 1. Average values of image quality metrics for BSD100

6. Conclusion

In this project we implemented generative adversarial network to solve task of super-resolution. We gained practi-

cal experience with model training and GANs hyperparameters tuning. We obtained reasonably good results in terms of image reconstruction metrics. We established that GANs model "babysitting" is a challenging task, because GANs training process is somewhat unstable and unpredictable.

References

[1] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. *CoRR*, abs/1511.05666, 2015.

[2] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks, 2014.

[3] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. *CoRR*, abs/1602.02644, 2016.

[4] C. E. Duchon. Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology*, 18:1016–1022, Aug. 1979.

[5] M. Elad. *Sparse and redundant representations : from theory to applications in signal and image processing*. Springer, New York, 2010.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[7] H. He and W. C. Siu. Single image super-resolution using gaussian process regression. In *CVPR 2011*, pages 449–456, June 2011.

[8] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[9] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. *CoRR*, abs/1511.04491, 2015.

[10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2016.

[11] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. *CoRR*, abs/1601.04589, 2016.

[12] X. Li. Multi-scale dictionary for single image super-resolution. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 1114–1121, Washington, DC, USA, 2012. IEEE Computer Society.

[13] S. G. Mallat. *A wavelet tour of signal processing : the Sparse way*. Elsevier /Academic Press, Amsterdam Boston, 2009.

[14] Y. Romano, J. Isidoro, and P. Milanfar. Raisr: Rapid and accurate image super resolution, 2016.

[15] W. Shi, J. Caballero, F. Huszr, 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, 2016.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[17] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *Trans. Img. Proc.*, 20(6):1529–1542, June 2011.

[18] S. Wang, L. Zhang, Y. Liang, and Q. Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2216–2223, June 2012.

[19] J. Yang, Z. Lin, and S. Cohen. Fast image super-resolution based on in-place example regression. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 1059–1066, Washington, DC, USA, 2013. IEEE Computer Society.

[20] Y. Zhu, Y. Zhang, and A. L. Yuille. Single image super-resolution using deformable patches. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2917–2924, June 2014.