# Fourier and Edge Loss for GAN-Based Single Image Super Resolution

Jason Plawinski[1,a)] Michael Waechter[1,b)] Yasuyuki Matsushita[1,c)]

## Abstract

Recent breakthroughs with neural networks led to great progress of image super resolution methods. However, it remains difficult to produce fine images or texture details. In this paper we discuss two loss functions that can be used in conjunction with the commonly used mean squared error (MSE) loss: a frequency domain loss (FFT loss), preventing the image from being over-smoothed by the MSE loss, and an edge loss, ensuring the reconstruction of sharp edges. We show how these losses can be used either in a standard supervised training setup or in a generative adversarial network (GAN) and test the advantages and limits of the different losses on multiple datasets of natural images and textures.

## 1. Introduction

Single image super resolution (SISR) is the task of upscaling a single low-resolution image by a fixed scaling factor. It is useful in image transmission where a low-resolution image is transmitted and then upscaled on the device. Further it is useful in the context of image restoration, texturing 3D models from low-resolution images, and countless more applications from academia and industry.

SISR methods have been mostly based on interpolation or image patch matching but recently the convolutional neural network (CNN) revolution led to impressive improvements. Fully supervised training in SISR boils down to simply downsampling high-resolution images. Convolutions are well-suited for upscaling textures because they mostly deal with local information. A common choice for a loss function in SISR is the mean squared error (MSE) which, however leads to overly smooth results. The network averages over all possible solutions and the output is therefore blurry. Further, MSE does not account for human perception and the visual pleasingness of results. An alternative to using MSE is using another training method – generative adversarial network (GANs) – where the network learns to recreate believable images. It can be seen as pushing the network to select one possible solution instead of averaging over all solutions.

This paper discusses different loss functions for super resolution. First, we present a frequency domain loss (a fast Fourier transform loss, or FFT loss, hereafter). It compares the frequency spectra of an upscaled result and its ground truth. The intuition is that introducing a control over the frequencies enables limiting the smoothing induced by the MSE less. We further discuss an edge loss. Using edges for super resolution was even used before the rise of neural networks. We show that, if not too heavily weighted, our edge loss can improve MSE results. Finally, we will also compare and test our loss function in a GAN setting.

## 2. Related Work

The annual competition on SISR [1] is mostly geared towards improving architectures and leaves less space to improving perceptual quality because the competition focuses on minimizing MSE loss.

A pioneering work in super resolution is the SR-Resnet [2]. This network introduced upscaling modules using "pixel shuffling" based on fractional convolutions [3]. This module has now mostly replaced transposed convolutions and deconvolutions

[1] Osaka University
[a)] jason.plawinski@ist.osaka-u.ac.jp
[b)] waechter.michael@ist.osaka-u.ac.jp
[c)] yasumat@ist.osaka-u.ac.jp

for upscaling even in fields other than SR. An improvement made since the introduction of SRResnet is that now instance normalization is favored over batch normalization [4].

To eliminate the need for an explicit loss function and paired training data, Goodfellow *et al.* [5] introduced the GAN architecture where two networks compete against each other: The generator is trained to generate good results for the task at hand and the discriminator tries to tell apart results from the training data and results generated by the generator. Since their introduction, GANs have been improved practically and theoretically and some training tips and architectural changes have facilitated convergence [6]. GANs have been shown to perform reasonably well in texture synthesis tasks [7] and super resolution [2]. Some improvements specific to textures such as patch GANs [7], [8], content loss and texture loss can be added to improve the result quality.

## 3. Proposed Method

In the following we first introduce our new loss functions before describing the architecture of a conventional, fully supervised network as well as an adversarial network based on SRGAN [2].

**FFT loss:** The FFT loss enforces high-frequency information to reduce the smoothing from the MSE loss. Let $I_m$ be the Fourier transform of the initial image $i_m$. Let $M_i$ further be the magnitude of $I_m$:

$$M_i = f\left(\|I_m\|_2^2\right).$$

Where the function $f$ should be positive and increasing on $\mathbb{R}_+$. It dampens the loss's gradient in cases where the image has very high energy. Without this term, the training would be too sensitive to uniform images. We choose $f(x) = \log(1+x)$ for 4× upscaling or $f(x) = \sqrt{x+\varepsilon}$ for 8× upscaling (which means working with the complex modulus).

The Fourier loss for the ground truth (GT) and the super resolution image (SR) is then defined as

$$L_{\mathrm{FFT}}(M_{\mathrm{SR}}, M_{\mathrm{GT}}) = |M_{\mathrm{SR}} - M_{\mathrm{GT}}|.$$

The forward and backward passes for FFT in neural networks are natively implemented in deep learning frameworks and run quickly on GPUs. A theoretical advantage of this loss is that a network based purely on MSE and convolutions can have a small receptive field whereas the Fourier transform incorporates global information. This means that MSE and FFT loss work somewhat independently and FFT loss is something that a layer in a conventional network could not easily learn by itself.

**Edge loss:** The edge loss preserves edges, which are usually destroyed in the downsampling process because of pixel re-quantization and loss of context. Forcing the network to focus on edges is useful because it is an important part of what is lost but can be retrieved easily. We compute the edge loss as $\ell_2$ loss over the Sobel filtered input image.

### 3.1 Network Architecture

Here we first describe a supervised network trained on pairs of low-res and high-res images, which can be seen as a generator network. We then introduce a discriminator network which, in conjunction with the generator, can be used in adversarial training.

**Supervised network:** The network (shown in the top of Fig. 1) consists of residual blocks, upscaling modules and skip connections. The residual blocks consist of 3 × 3 convolutions, leaky ReLUs and instance normalization (IN). The upscaling module, also referred to as fractional convolution, is a pixel shuffler that rearranges channels to upscale the image [3]. The skip connection is used to transfer all the low-level edge information back before the upscaling.

**Adversarial network:** For our adversarial network the generator stays the same as above. The discriminator (shown in the bottom of Fig. 1) is an encoder with strided convolutions and ReLU activations. Instead of outputting a single value, the network is following a patch GAN architecture. The last convolutional layer is not followed by a fully connected layer since this would be inefficient and memory intensive. This makes the discriminator fully convolutional and it can therefore be seen as multiple discriminators acting independently on smaller patches [7], [8]. This makes sense in the context of texture super resolution because it is a local problem where, in contrast to general image synthesis, checking the entire image for coherence (*e.g.* checking the relative position of objects) is

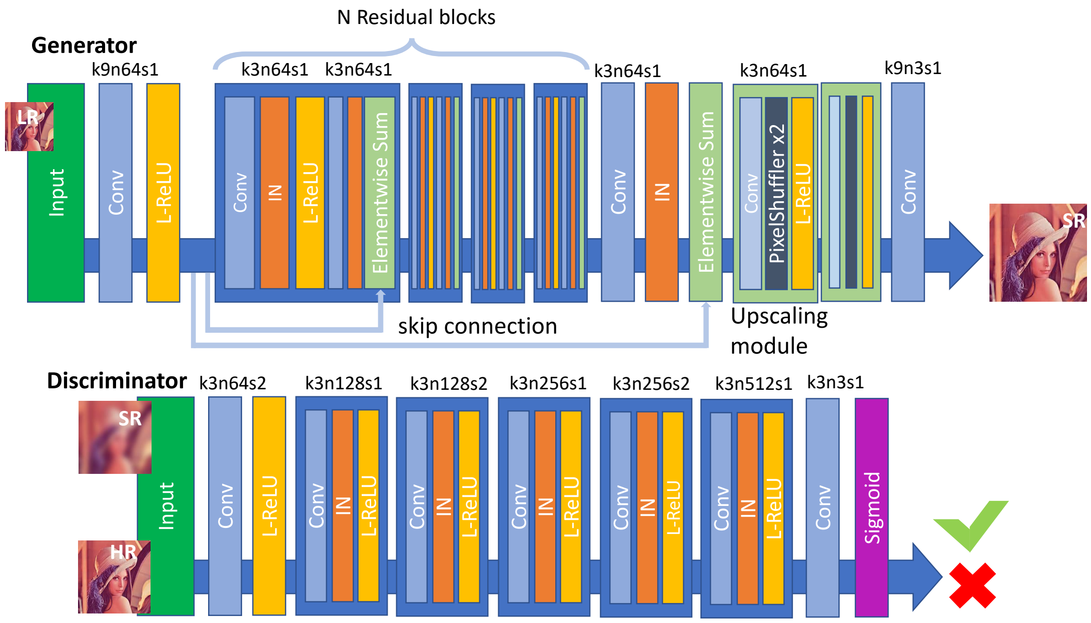

Fig. 1: Architecture of the generator and the discriminator network

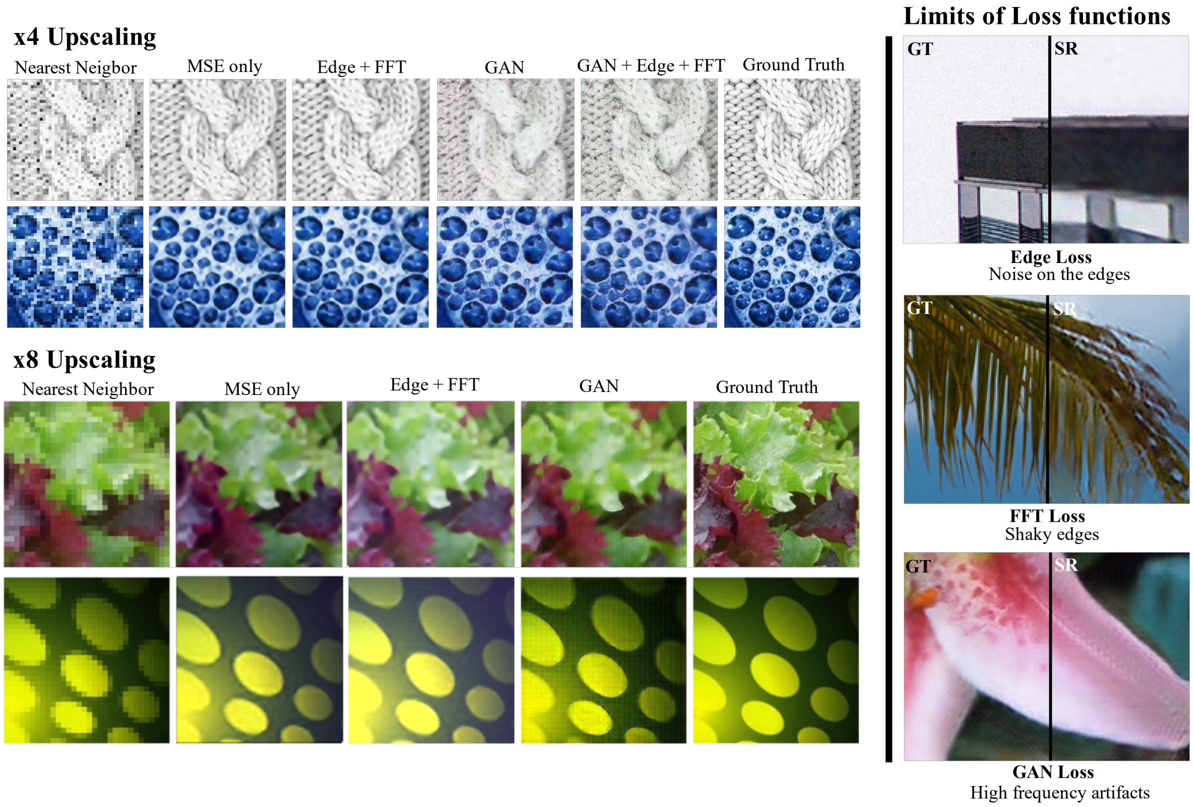

Fig. 2: Results of the network on different low resolution images from the texture dataset after training

not vital.

**Training:** The training images were downsampled with a factor of 4× or 8× and linear interpolation, randomly cropped to 256×256 and randomly flipped horizontally. The non-GAN networks were trained from scratch on around $10^6$ patches with ADAM and a learning rate of $10^{-4}$ turned down to $10^{-5}$ at the mid-point of training. The GAN networks were trained on $1.5 \times 10^6$ images and the generator was initialized with an MSE pretrained network.

## 4. Experiments

The goal of our experiments is to see the performance of the different loss functions on various data sets. A yet unsolved research problem is an objective metric for judging the perceptual quality of upscaling results (or image processing results in general). We therefore go a slightly different way: We tune most parameters quite extremely in order to easily observe meaningful differences with the human eye.

The tests were done on two datasets containing images with various properties: Div2k [1] (a dataset of diverse natural images with a 2k resolution (800 images for training)) and textures of various sizes larger than $256 \times 256$ [9] (4699 images for training).

On both the 8× and 4× upscaling the networks are able to retrieved a good part of the lost information (left side of Fig. 2). The edge loss makes a noticeable difference on the quality and sharpness of the results compared to the simple MSE. The FFT loss is useful in cases with really high frequencies but may generate artifacts on uniform regions. This can be seen as when then network is tuned to have more than 90% of its loss originate from the FFT loss (right side of Fig. 2). This can be due to the fact that, no windowing is introduced. Indeed, applying FFT to a squared image leads to artifacts where some frequencies are attributed with too low weights. Another reason is that applying a log weighting in to the modulus might be a too strong dampening.

In the case of GANs, using additional losses leads to less noisy and higher quality. Improvements on GAN stability are hard point out because not enough experiments have been undertaken to make a statistical argument.

Using a GAN for 8× upscaling has been difficult. Since a lot of local information is lost while downscaling, having the network converge without resorting to high frequency noise (bottom left of Fig. 2) means relying on extremely deep networks and long training times. In this work, the discriminator used is most likely too shallow to deal with 8× upscaling in a healthy manner.

## 5. Conclusion

In super resolution, adding our loss functions to established CNN architectures can indeed lead to better results. Using Fourier domain information can reduce over smoothing, though, some improvements are still required for the Fourier loss to work homogeneously over the frequency spectrum. The edge loss is an efficient yet simple way to enhance training efficiency and result visual quality. Finally, applying this in an adversarial context can lead to perceptually pleasing results given enough resources. This algorithm is a promising step in the direction of applying super resolution to textures, for example in the case of image based 3D reconstruction.

**Acknowledgement** This work is supported by the New Energy and Industrial Technology Development Organization (NEDO).

### References

[1] Participants of the NTIRE 2017 Challenge on Single Image Super-Resolution, "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *CVPR Workshops*, 2017.

[2] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.

[3] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *CVPR*, 2016.

[4] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *ICML*, 2016.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[6] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *NIPS*, 2015.

[7] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *ECCV*, 2016.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, 2017.

[9] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *CVPR*, 2014.