Self-supervised methods for low-level vision

Antoine Cadiou
antoine.cadiou@ens-paris-saclay.fr

Megi Dervishi
megi.dervishi@ens.fr

Abstract

*Since for certain fields (*i.e. *such as biomedicine) the acquisition of noisy and clean images is difficult, there comes the need for self-supervised methods for image restoration tasks. The objective of this project will be to explore such methods to perform image manipulation tasks such as de-noising, super-resolution and deblurring.*

1. Introduction

In this project we will focus on exploring Topic B: Self-supervised learning methods for low level vision. In particular we will explore the performance of the *Noise2Void* (N2V) [2] and *Noise2Noise* (N2N) [3] methods, as well as implementing super-resolution using N2V/N2N and de-blurring using N2N methods. The reproduction of results was shared among the authors, the super-resolution was done by Antoine Cadiou and de-blurring by Megi Dervishi.

2. Model Architectures

2.1. Baseline: RDUNet

The base-model is the Residual Dense U-Net presented in the paper [1]. The model achieves state-of-the-art results in de-noising tasks and will serve as a reference for our experiments. The model aims to learn a latent space describing the noisy input image, and then to reconstruct this input without the noise. RDUnet is a fully-supervised model i.e. we need both the noisy and the clean images to train. However, for certain fields such as biomedicine or cosmology, the acquisition of clean/noisy image pairs is very difficult.

2.2. Noise2Noise

Noise2Noise (N2N)[3] is a generalization of the previous model which works around the clean/noisy image pairs requirement. The authors show that by training the U-Net on pairs of noisy images the model will learn to predict the expected value of the corrupted images. Hence assuming the noise is zero-mean, N2N will be able to predict the clean image. Nevertheless, even if this method is asking for less costly data to train on, it is still difficult to acquire pairs of noisy/noisy images.

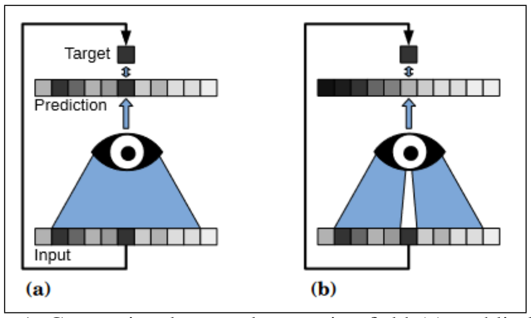


Figure 1. Conventional network receptive field (a) vs blind-spot receptive field (b).

2.3. Noise2Void

The advantage of N2V in comparison to the previous denoising methods/models is that we avoid the acquisition of pairs of noisy images or clean/noisy images. The algorithm only needs the individual corrupted image assuming the noise is zero-mean and independent between pixels [2]. The principle of the algorithm is as follows (*c.f.* Fig 1). In the conventional network the prediction for an individual pixel depends an a square patch of input pixels, known as a pixel's receptive field (pixels under blue cone). If we train such a network using the same noisy image as input and as target, the network will degenerate and simply learn the identity. To solve this issue, N2V proposed a blind-spot network were the receptive field of each pixel excludes the pixel itself, preventing it from learning the identity. In that way, blind-spot networks can learn to remove pixel-wise independent noise when they are trained on the same noisy images as input and target.

3. Experiments

Setup. We experiment upon the official code implementations of N2V [1] and N2N. [2] The dataset used is BSD300. RDUNet model is trained for 21 epochs with batch size 16. The optimizer is AdamW with learning rate 1e-4 and

[1]Official implementation N2V github.com/juglab/n2v
[2]Official implementation N2N https://github.com/NVlabs/noise2noise

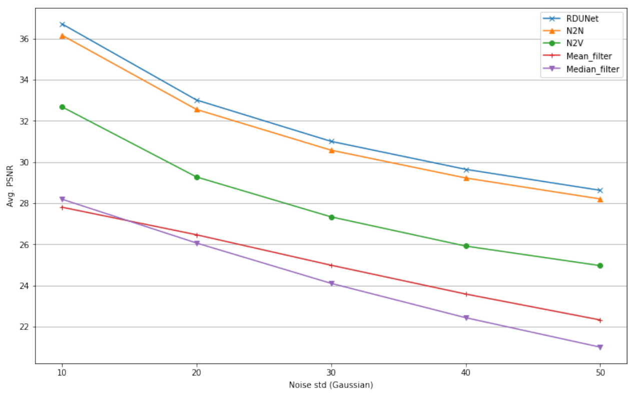


Figure 2. Average PSNR scores for each models and with a variation on std of a Gaussian noise. We have trained/tested our models on the BSD68 dataset[4].

L1Loss. In comparisons N2N was trained for 300k iterations on batch size of 4 and Adam optimizer with learning rate 3e-4. Finally N2V was trained for 100 epochs on 64 batch sizes and Adam optimizer with learning rate 1e-4. Both N2N and N2V are trained with MSE loss. Furthermore, the models are trained with Gaussian noise images of different variances.

Results. We have re-produced both the qualitative (*c.f*. Fig 2) and quantitative results (*c.f*. Fig 3) of the paper [2]. As expected RDUNet has the best performance for all variances. Although it is very closely followed by N2N. N2V performs better than standard denoising techniques such as Mean and Median filtering however its PSNR is considerably worse than N2N and RDUNet as expected. Nonetheless, qualitativaly (*c.f*. Fig 2) the denoised images are all close to the ground truth as perceived by the human eye.

4. Extensions

4.1. Super-resolution

Model. The idea is to map super-resolution to a denoising or inpainting task. In other words, given a low-resolution picture we can enlarge the resolution by adding missing pixels for every known pixels. Following this idea, we have thought of a way to apply Noise2Void to degraded (downsampled + upsampled) images.

Fig 4 shows the way we have processed our data to create a new upsampled dataset to give for the training. Given an image we have first downsampled it by a factor of 2 and then upsampled it using bicubic linear interpolation. To this new image we have added a Gaussian Noise and given it to our models.

Results. The N2N model performs much better than the N2V model. We have calculated the Average PSNR between the images predicted by our models and the ground truths (the results are visible in the Table 4.1). Furthermore, we have calculated the PSNR score between the ground truth and the upsampled image (by bicubic interpolation). Our model has a slightly worse perfomance than a bicubic interpolation. However this is to be expected since the N2N/N2V models try to predict the expected value of the corrupted image and with our pipeline that value is the bicubic interpolation.

	N2N	N2V	Bicubic interpolation
Avg. PSNR	27.88	25.82	28.41

Table 1. PSNR scores for Super-resolution

4.2. De-blurring

Deblurring is important in retrieving the lost information on images due to different factors such as motion or focus. A blurred image can be modeled using the equation [5]

$$B = x * K + N \tag{1}$$

where x is the clean sharp image, K a blurring kernel and N additive noise. The idea behind the following new architecture is to transform the deblurring problem into a denoising one, and hence make use of the N2N method. Considering that deblurring is a novel architecture the following sections show the different implementations that were created to try and make the model work. [3]

4.2.1 De-blurring to additive noise

Model. As described in section 2.2, the N2N model takes an image with an additive noise. By passing the image x through the architecture shown in Figure 5 we obtain:

$$\mathcal{F}^{-1}(\log(\mathcal{F}(x))) + \mathcal{F}^{-1}(\log(\mathcal{F}(K_i))) \tag{2}$$

We can hence consider the first part of the sum as a new image and the second part as the noise.

Assumptions. We assume that the initial image x has no additive noise i.e. $N = 0$ (eq. 1). It is important to transform back to the primal domain before giving the input to the N2N model as done in section 3.4 of N2N paper [3]. This is because the model is able to better learn pixel-correlation in the real space rather than that fourier space. The last assumption is that the kernels must be zero-mean as the N2N model converges to the expectancy of the image input.

Zero-mean Kernel. To ensure such assumption, we first build an $(n \times n)$ matrix where each entry is sampled from a log-normal distribution and n is the dimension of the kernel. We apply $\mathcal{F}^{-1}$ to the matrix to obtain our kernel K. Due to the log-normal distribution properties, the obtained kernel is a Gaussian kernel with zero-mean. A concern of

[3]De-blurring implementation github.com/MegiDervishi/deblur_n2n

using such kernel to train our model is that it isn't a realistic kernel in the real world, however we argue that since a blurring kernel is a sample of K then the model should still be able to work.

Experiments. The initial images are from the BSD300 dataset transformed into black and white pictures to optimize training. We use (5×5) kernel with $\mu = 0$ and $\sigma \in [0, 1]$ or $\sigma \in [0, 50]$ for training and $\sigma = 0.5$ or $\sigma = 25$ for validation/testing. We train for 100k iterations on a GTX-1060 GPU which takes around 10 hours.

Results & Discussion. Using the original model we see that even though the loss (*c.f*. Fig 7, 3.2.1) converges, the average PSNR (*c.f*. Fig 6, 3.2.1) is very noisy and does not increase. The network has indeed a lot of trouble predicting the de-blurred image. So much so that the validation images are unrecognizable. One explanation for why it might be so difficult for the model to produce acceptable results is that any error made by the model will exponentially impact the resulting image as can be seen in the architecture depicted in Figure 5. Furthermore, we notice that the model is able to give acceptable predictions for the high frequency components but the DC component is almost completely lost. These reasoning are what lead to the following improved models.

4.2.2 Removal of DC components

Model. Before inverting back to Fourier space (step d.) we remove the DC component and save it in memory. We later add such component after step g.

Experiments. We keep the same assumptions and experimental setup as before, except during training we do not save the DC-component in memory because it would be too expensive, but only save it and add it during validation for visualization purposes. We also train for 20k iterations.

Results & Discussion. We see a significant improvement in the obtained images both qualitatively (*c.f*. Fig 6, 3.2.2) and quantitatively (*c.f*. Fig 7, 3.2.2). Indeed, the PSNR steadily increases during training and reaches a plateau which is much higher than any previously obtained value. Furthermore, the model generalizes well and can predict reasonable images also on the testing/validation set. When looking at the predicted images, we see that the model restored much of the original images, however there is still heavy noise persisting in the final output. We can also see that the PSNR is plateau-ing around 29 which is still far from the actual PSNR (reached for Gaussian de-noising using N2N) for the testing set at 32.48 [3]. This is probably due to the fact that we still haven't solved the exponential error problem. Hence the following model.

4.2.3 Multiplicative noise in Fourier space

Model. As discussed above the exponential magnifies any error made by N2N, hence the idea is to remove the $\log$ and exponential step from previous model. This means that the noise will now be multiplicative instead of additive. Theoretically this should still work since the expectancy is a linear operator.

N2N with multiplicative $\mathcal{N}$ noise. To test this fact, we tried corrupting images by multiplying point-wise $\mathcal{F}(x)$ with random points in the complex plane sampled using a 2D Gaussian law, as shown in Figure 6, 3.2.3. The model is trained for 20k epochs and reaches a PSNR of 28.89. The qualitative results are indeed very satisfactory which shows that N2N is able to perform de-noising for multiplicative noise in the frequency domain.

Hence we validate such model on images that are blurred with a Dirac-delta mean kernel where each entry is taken according to Gaussian distribution. The results are acceptable however not yet satisfactory. The model did indeed remove noise however blurriness is still present.

Convolve2Convolve. As we saw in the previous model, N2N is able to remove multiplicative noise in the Fourier domain. Hence one could expect that N2N is able to do de-convolution since this operation is equivalent to denoising in the Fourier domain. Therefore we take 2 Dirac-delta mean kernels whose entries are taken according to a Gaussian distribution and convolve them with the original image to create an input-target pair and give it to N2N.

However, if we try this (*c.f*. Fig 6 3.2.4) the results are not at all satisfactory and at this stage it is still not clear from where the problem originates.

5. Conclusion

In conclusion in this project we have re-produced the results of *Noise2Void* paper [2] which are in agreement with the papers. We then extend these methods to perform super-resolution and deblurring with reasonable success for novel architectures. However we believe that we are pretty close to finding a fully-functional deblurring (N2N) method but it would require more time.

References

[1] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa E. Alarcón. A residual dense u-net neural network for image denoising. *IEEE Access*, 9:31742–31754, 2021. 1

[2] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. *arXiv:1803.04189v3*, 2019. 1, 2, 3

[3] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv:1803.04189v3*, 2018. 1, 2, 3

[4] Stefan Roth and Michael Black. Fields of experts. *International Journal of Computer Vision*, 82:205–229, 04 2009. 2

[5] Siddhant Sahu, Manoj Kumar Lenka, and Pankaj Kumar Sa. Blind deblurring using deep learning: A survey, 2019. 2

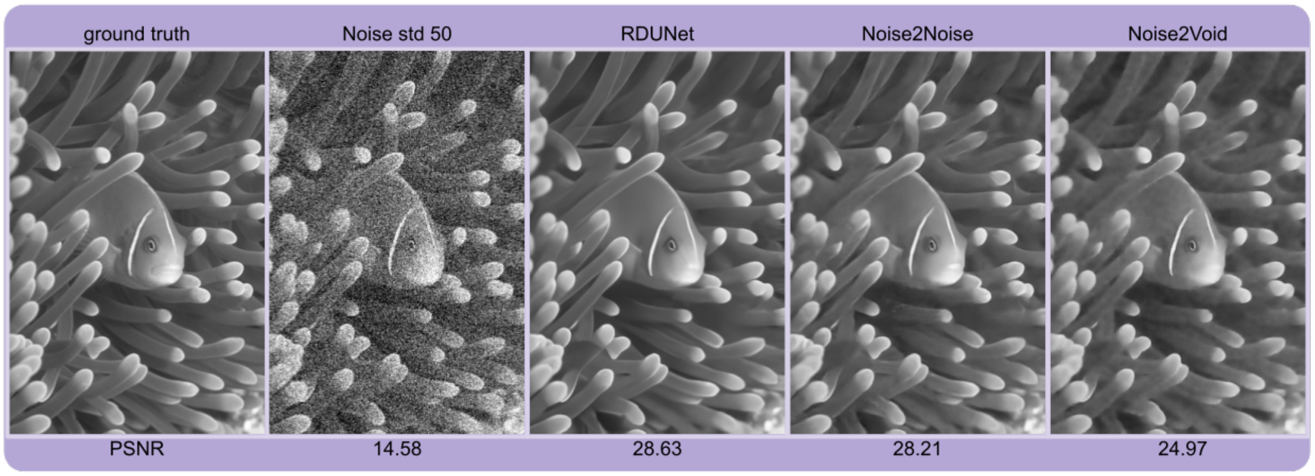


Figure 3. Qualitative results for denoising an image using different models.

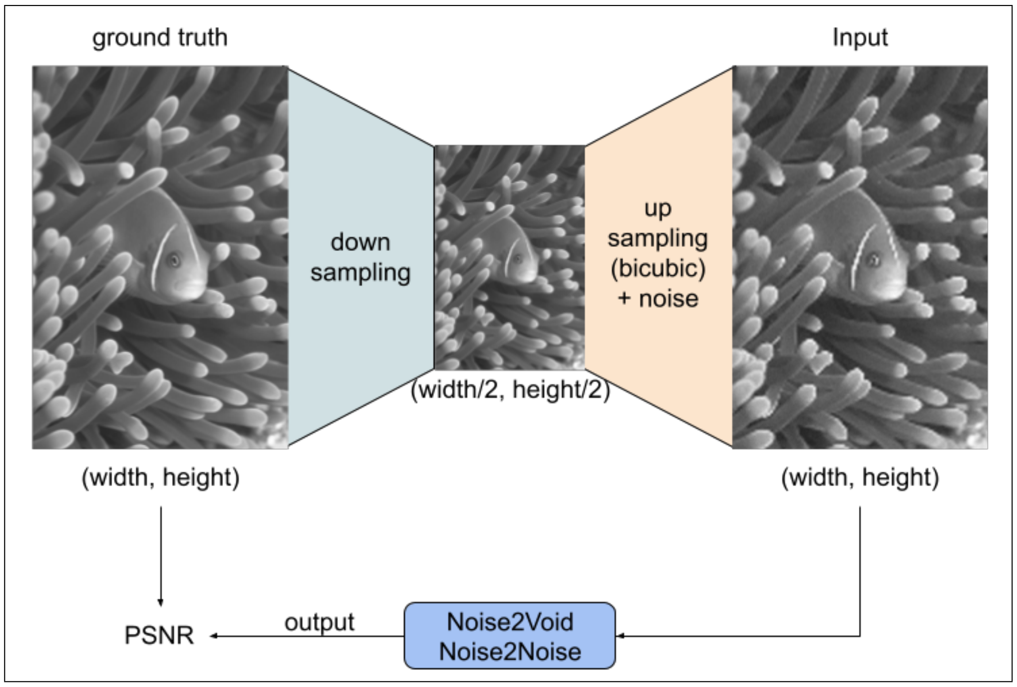


Figure 4. Data Processing pipeline for super-resolution.

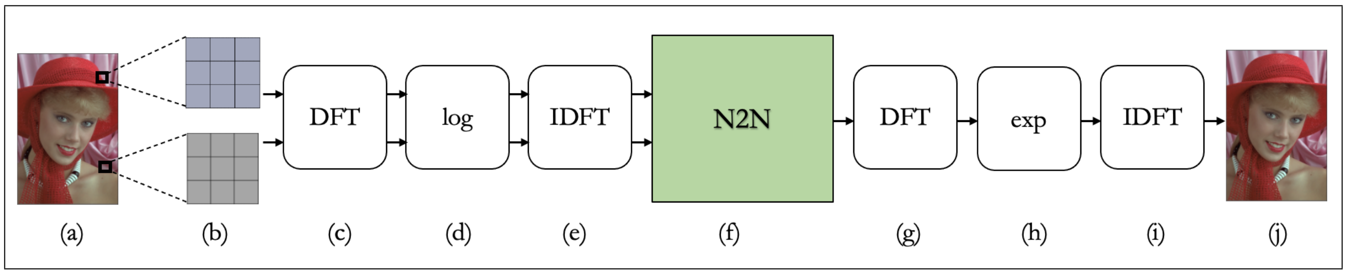
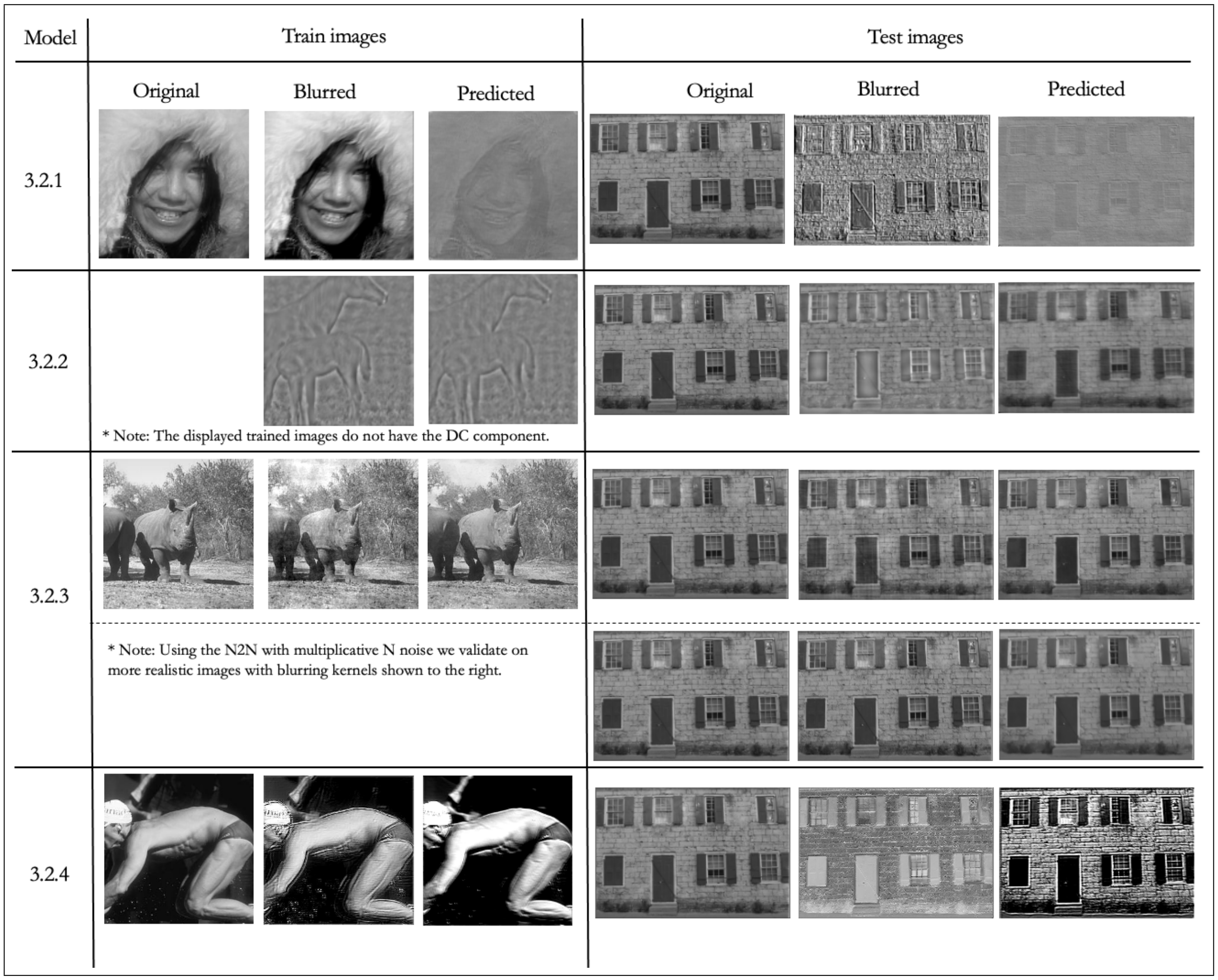


Figure 5. De-blurring model architecture from step a.(input image) to step j. (predicted image).



* Note: The displayed trained images do not have the DC component.

* Note: Using the N2N with multiplicative N noise we validate on more realistic images with blurring kernels shown to the right.

Figure 6. Qualitative results of models described in section 3.2.

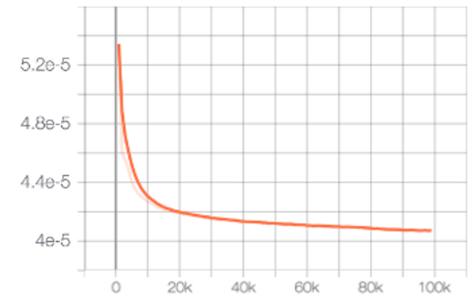
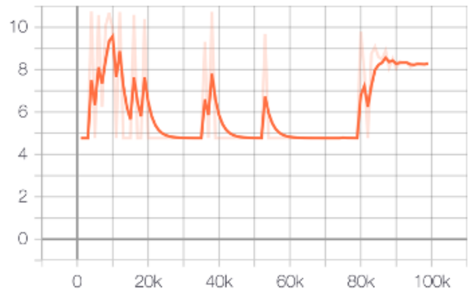
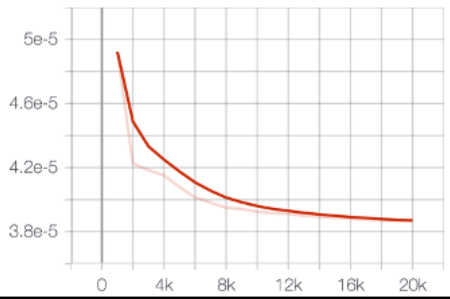
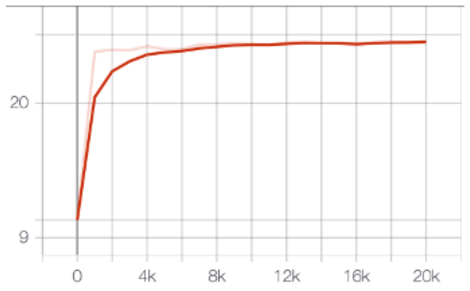
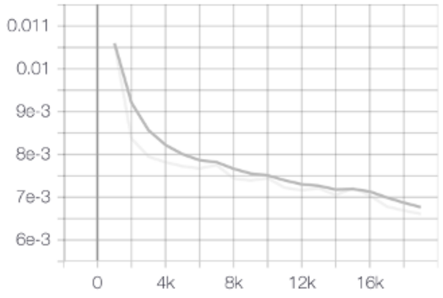
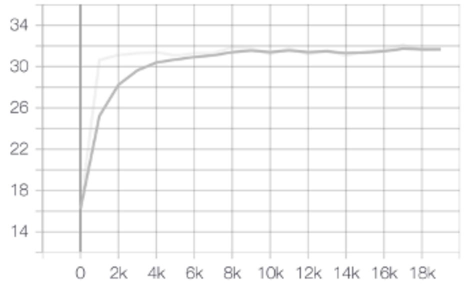
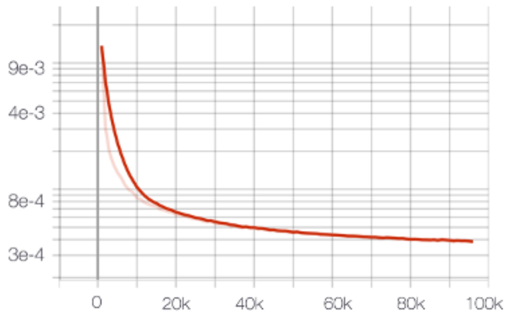
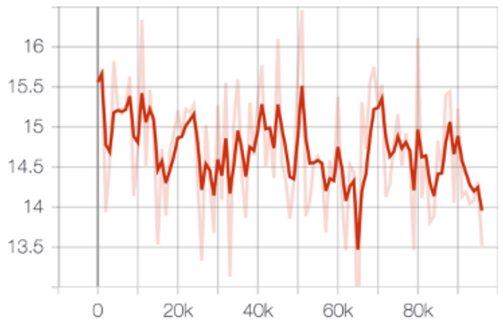
Model	Loss	Average PSNR
3.2.1		
3.2.2		
3.2.3		
3.2.4		

Figure 7. The loss and average PSNR results of models described in section 3.2.