AI6126 Advanced Computer Vision Project 2

Chen Yongquan (G2002341D)
Nanyang Technological University

Contents

Model

The model used is SRResNet as described in [1], with the number of residual blocks extended from 16 blocks deep to 20 blocks deep. This increases the number of parameters from 1,517,571 to 1,812,995, barely under the imposed fairness guideline of 1,821,085 parameters. Depth is preferred over width here, as increasing the number of filters per block didn't seem to improve generalization to the validation set but memorization of the training set instead. So, number of features in the network body remain the same at 64 channels.

Batch-normalization is not used inside the residual blocks in order to retain the range flexibility of the network as described in [2] and [3], which subsequently reduces unpleasant artifacts in the output image and improves training stability and performance.

Pixel-shuffling layers as described in [4] are used after the up-sampling layers in order to reduce checkboard artifacts.

A residual skip connection is also made from the input to the output of the final convolutional layer, with the input being upscaled 4x using bilinear interpolation in order to match the dimensions of the output.

Loss Function

Loss function used is ℓ_1 loss based on empirical results of various training cost functions against the peak signal-to-noise (PSNR) ratio metric in [5].

Dataset Preparation

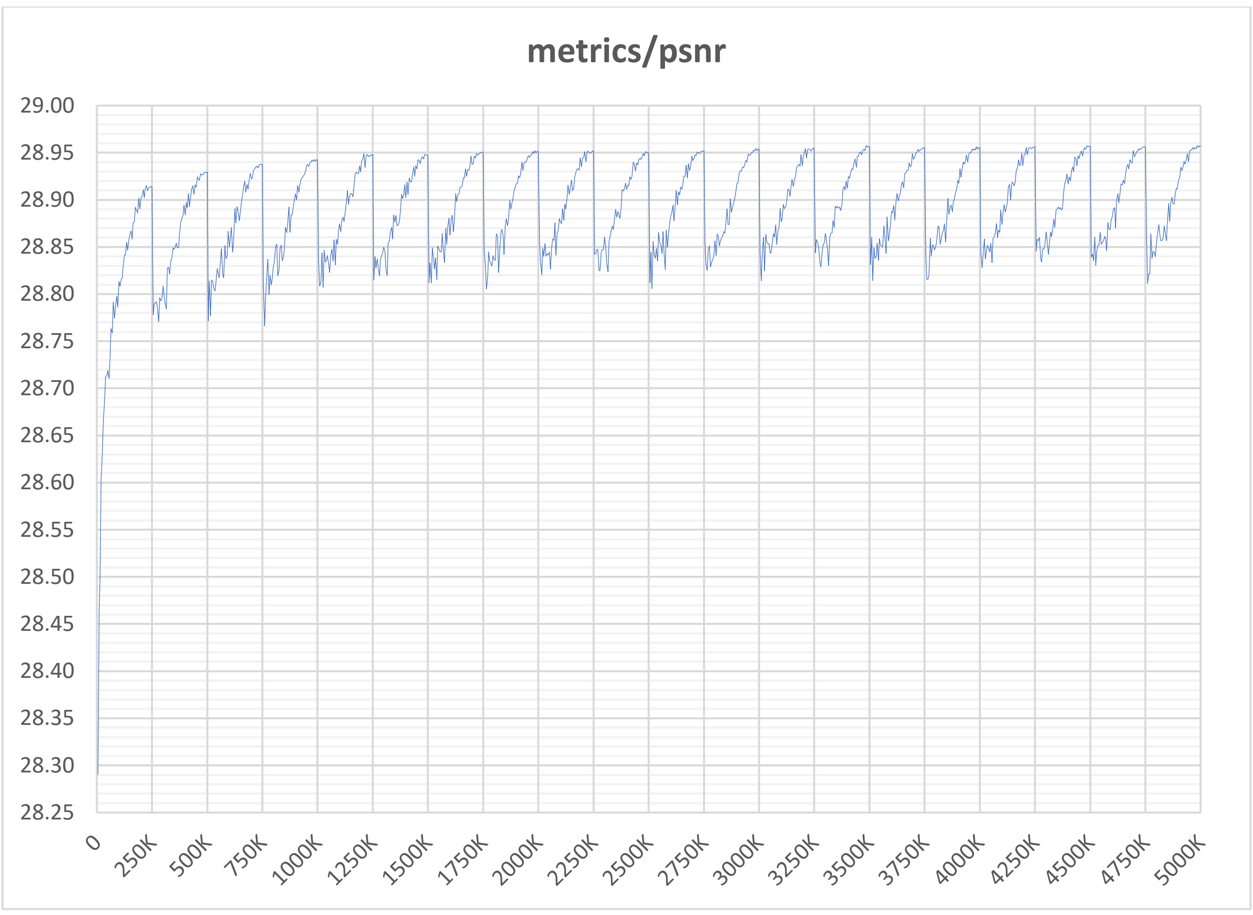
The original training samples in Mini-DIV2k are cropped into 480x480 sub-images in order to speed up IO operations as it is inefficient if large 2K resolution images are read in, but only small patches are randomly cropped out for training every time. 480x480 is a good balance between IO performance and the range of pixels that can be used for random crops.

Training Process

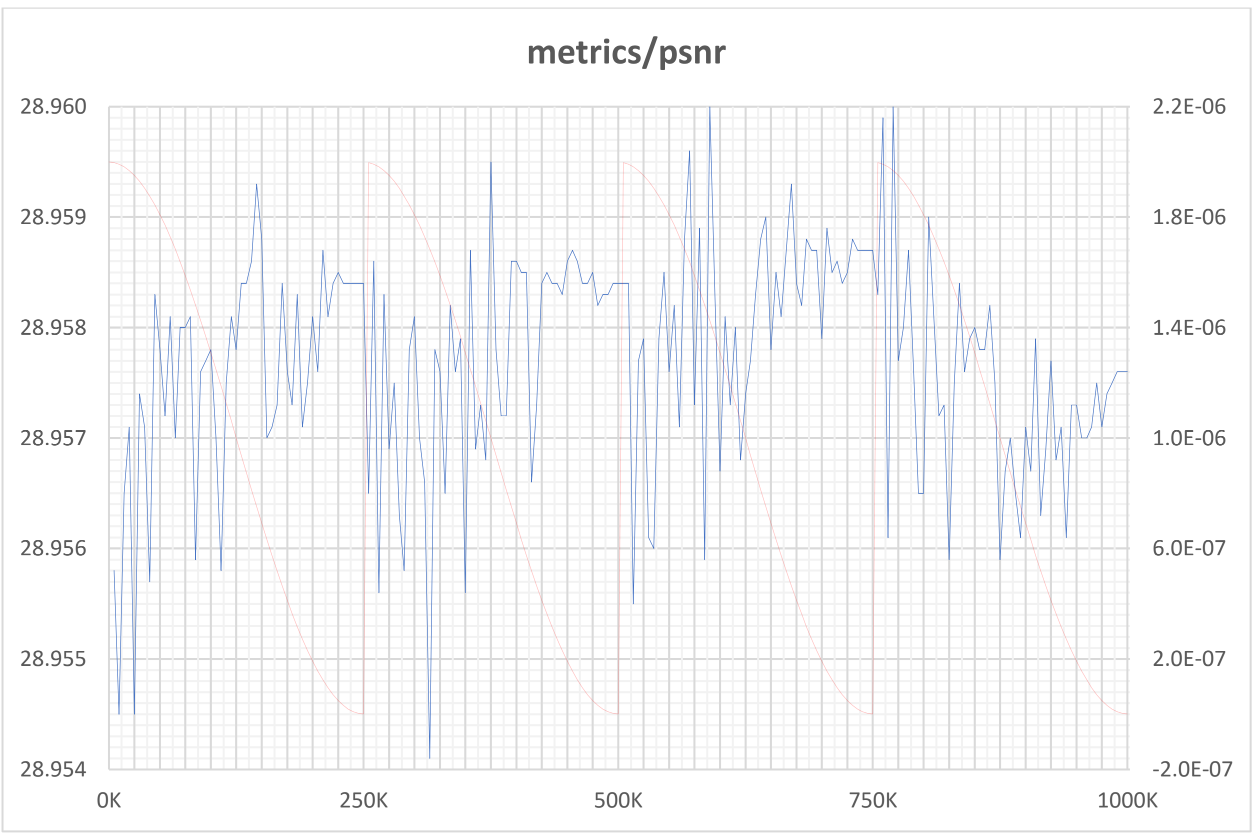
Training is done for 5,000,000 iterations using Adam [6] for optimization, with $\boldsymbol{\beta} = [0.9, 0.99]$. Learning rate is set at 2E-04 initially and then decreased based on a cosine annealing schedule to a minimum of 1E-07, with warm restarts back to the initial learning rate every 250,000 iterations [7]. Validation testing is done every 5000 iterations using PSNR as the metric.

We then select the best model from the above training process and perform finetuning, first with an initial learning rate of 2E-06 and a minimum of 1E-09. We then repeat this finetuning process iteratively, selecting the best model each time and reducing the cosine annealing range.

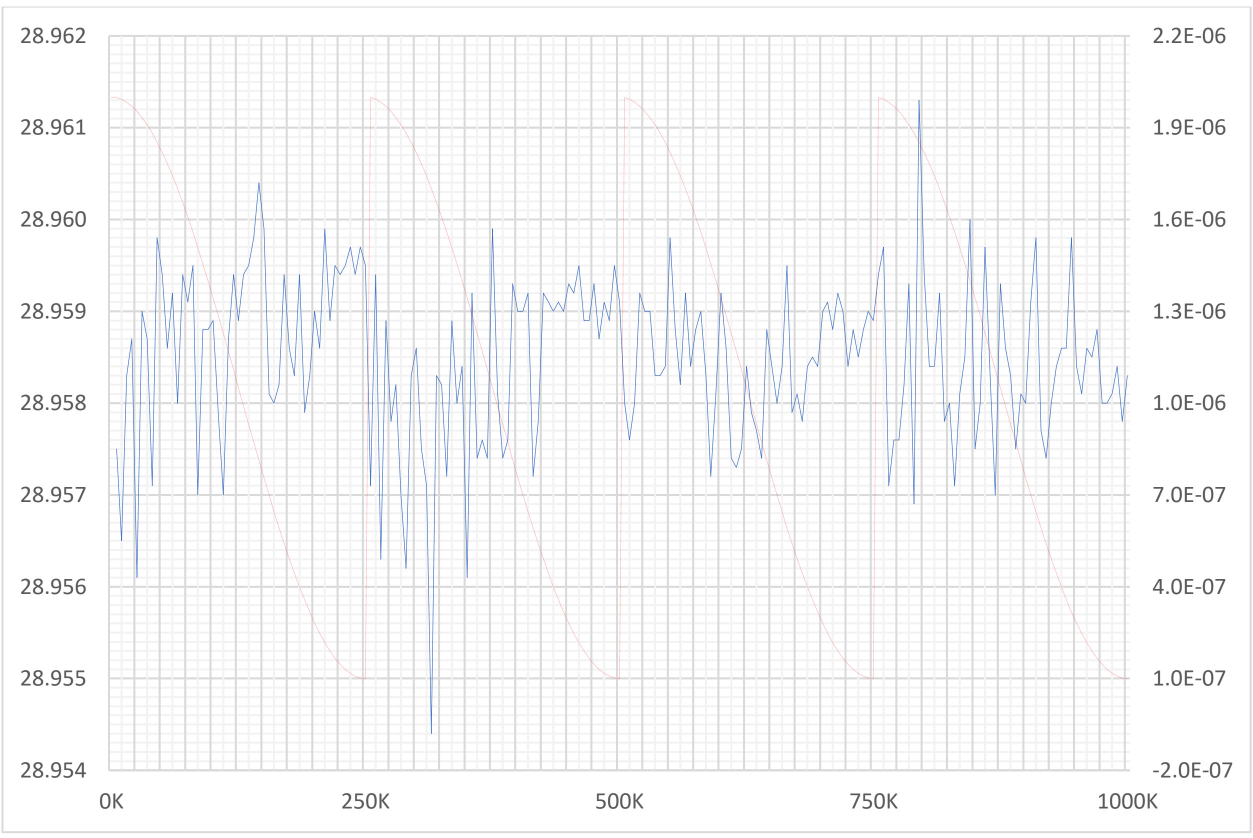
Training Curve



The chart above shows the PSNR of the validation set over the first 5,000,000 iterations. Best PSNR was obtained at iteration 4,985,000, with a PSNR of 28.9577.



The chart above shows the PSNR obtained during the finetuning iterations using the model from iteration 4,985,000 as a pretrained network. The orange curve is the learning rate and is charted using the secondary y-axis on the right. Best PSNR obtained was at iteration (5,)590,000 and (5,)770,000, with a PSNR of 28.96. Average learning rate for the respective 5000 iterations were 1.51E-06 and 1.98E-06. We see that training slows down and PSNR kind of plateaus after learning rate drops below 6.0E-07, barring some instable spikes and drops.



Based on the training observation above, we perform another 1,000,000 training iterations with the same initial learning rate but a minimum of 1.0E-07, using iteration (5,)770,000 as the starting point. Best PSNR was at iteration (6,)795,000, giving a PSNR of 28.9608 on the validation set.

We performed a final set of finetuning iterations with a minimum learning rate of 1.0E-06 but PSNR failed to break the 28.96 threshold, with the best PSNR missing it by 0.0001. So, it seems for cosine annealing scheduling it would be better to leave some room at the lower end, in order for the annealing to settle down to a better optimum.

Since the final set of iterations did not yield a better performing model, we used iteration (6,)795,000 as the submission model.

References

[1] 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," 2016.

[2] B. Lim, S. Son, H. Kim, S. Nah and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," 2017.

[3] S. Nah, T. H. Kim and K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring," 2016.

[4] W. Shi, J. Caballero, F. Huszár, 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.

[5] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Neural Networks for Image Processing," 2015.

[6] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014.

[7] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," 2016.