Super-Resolution

Akino Watanabe	(1004133653)
Yilu Zeng	(1003941265)

Word count: 1998, Penalty: 0%

Introduction

Super-resolution (SR) is the method that upgrades a low resolution (LR) image to a high resolution (HR) image by upscaling and enhancing the details within it [1]. It has various applications that are related to data compression and reconstruction, including satellite image, medical image, and microscopy image processing [2].

In this project, we focused on two types of resolution: the pixel resolution, which is the pixel count in an image, and the spatial resolution, which measures how close lines can be resolved in an image [3]. Even with a high pixel resolution, a low spatial resolution can still make the image to appear blurry, and therefore, it is important to address both.

A hand-coded model that does not involve neural networks (NN), particularly reconstructing by interpolation, could increase the pixel resolution by expanding the image. However, this method is insufficient in enhancing the details (spatial resolution) because of the complexity of and the missing information within the LR images. NN, on the contrary, is capable of working with large, complex data and non-linear relationships. Furthermore, NN models would be able to effectively generate HR images, if appropriately trained to understand what an ‘authentic’ image should look like instead of extracting information just from the given input LR image. Hence, we implemented the Generative Adversarial Network (GAN) that can learn what HR images should look like.

Background & Related Work

In 2017, the first GAN called super-resolution generative adversarial network (SRGAN) was created for SR of 4x upscaling factor [4]. SRGAN uses deep residual network (ResNet), and unlike other published networks that use MSELoss, SRGAN uses a “perceptual loss function” (PLF) based on a pre-trained VGG network to avoid MSELoss’s limitation for capturing high textual details. PLF is a combination of adversarial and content losses. The adversarial loss is computed by the discriminator by classifying the input image as either the original image or generated SR image to ultimately guide the generator to create better HR images. The content loss is calculated based on perceptual similarity instead of pixel-wise similarity to address spatial resolution. SRGAN outperforms previous works on SR, and hence, we referenced this source when exploring GANs and loss functions.

In 2019, a survey [5] was made to assess existing ML models for SR and to provide different ways to assess image quality. This includes peak signal-to-noise ratio (PSNR), which measures image reconstruction quality using the maximum pixel value and mean-square-error. Additionally, the mean opinion score (MOS) mentioned assesses the quality of images based on human ratings. Thus, we used such methods to assess our model's outputs.

Data Source, Labeling and Processing

In this project, we used an existing dataset, DIV2K [6], that contains HR images and their corresponding LR images. The dataset includes 900 images of different pixel sizes with diverse genres, including landscape, fine art, wildlife, food, and fashion. LR images were downscaled by a factor of four, and each HR image has six different 4x downscaled images. We cropped all HR and LR images into the pixel size of 1112x648 and divided each into two 556x648 images. Then we allocated the dataset into 7200 training, 300 test, and 2100 validation images and organized them into six folders (LR/HR_train, LR/HR_validation, LR/HR_test). The software only reads the dataset once before training, thus it was not necessary to improve the data reading speed by converting images into other file types. Since our dataset enlarged after cutting and reformatting, no further data collection was necessary.

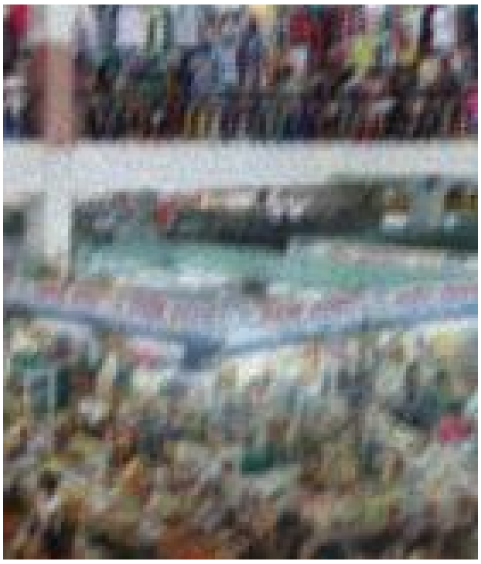


Fig. 1: LR image

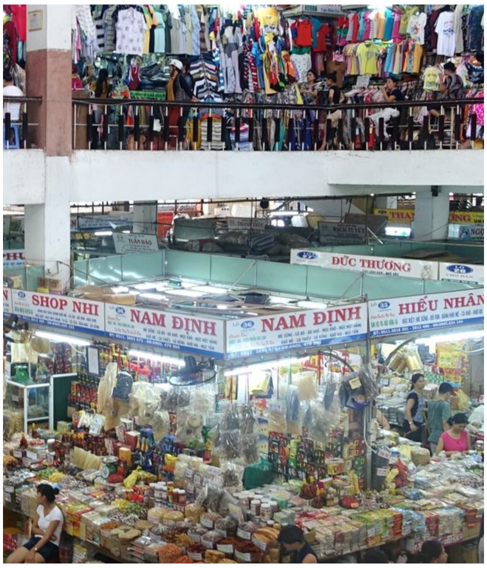


Fig. 2: HR image

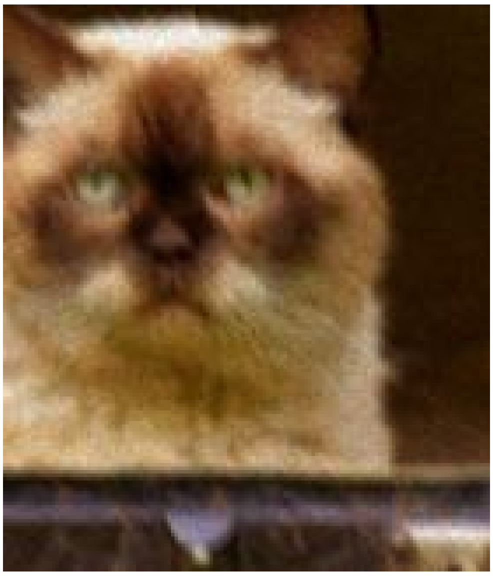


Fig. 3: LR image

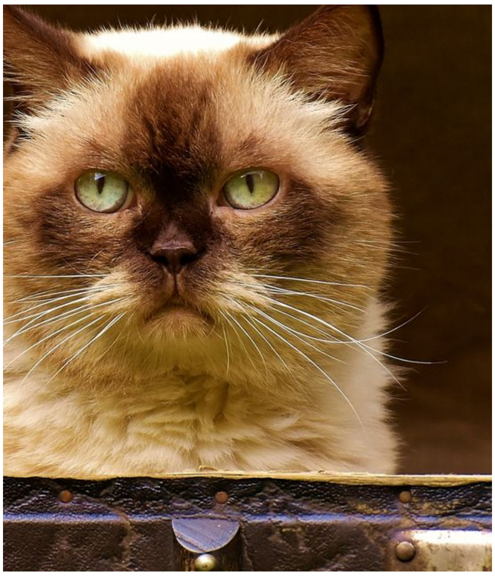
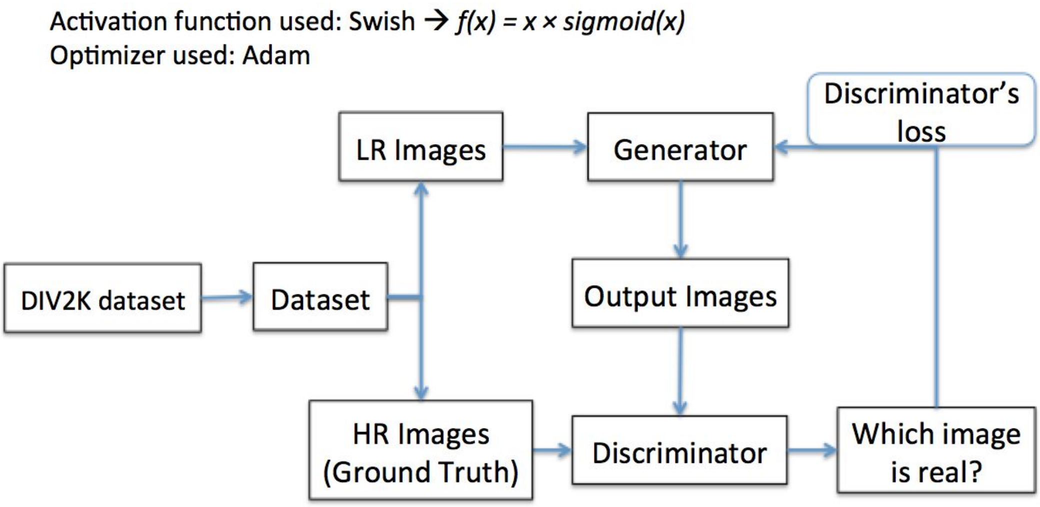


Fig. 4: HR image

Architecture

Our network architecture comprises a convolutional GAN (Fig. 5). The generator (Fig. 6) will take in LR images from the dataset to generate SR images that will be assessed by the discriminator (Fig. 7). The discriminator takes in and distinguishes both the generated image and the real HR image to guide the training of the GAN.



Activation function used: Swish → *f(x) = x × sigmoid(x)*
Optimizer used: Adam

Fig. 5: GAN model

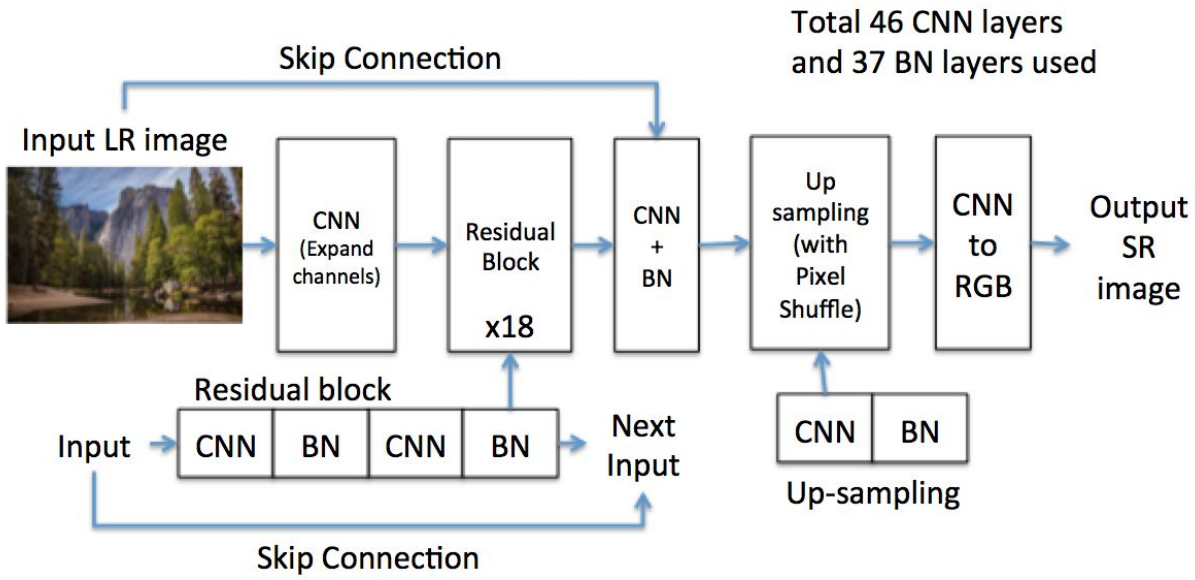


Fig. 6: Generator

As shown above, the generator's first CNN layer increases the number of channels from 3 to 20, so that they can retrieve more features. Then the feature maps travel through 18 residual blocks, each consisting of 2 CNN and 2 batch normalization (BN) layers, with no changes in channel size. These blocks act as ResNet with the skip-connections in each block [4], which helps training the deep convolutional network. The generator then re-configures the size by up-sampling by a factor of 4 on both dimensions. The last convolutional layer resizes the images to have 3 RGB channels using pixel shuffling [7], which converts depth (kernels) to height and width of proper image size.

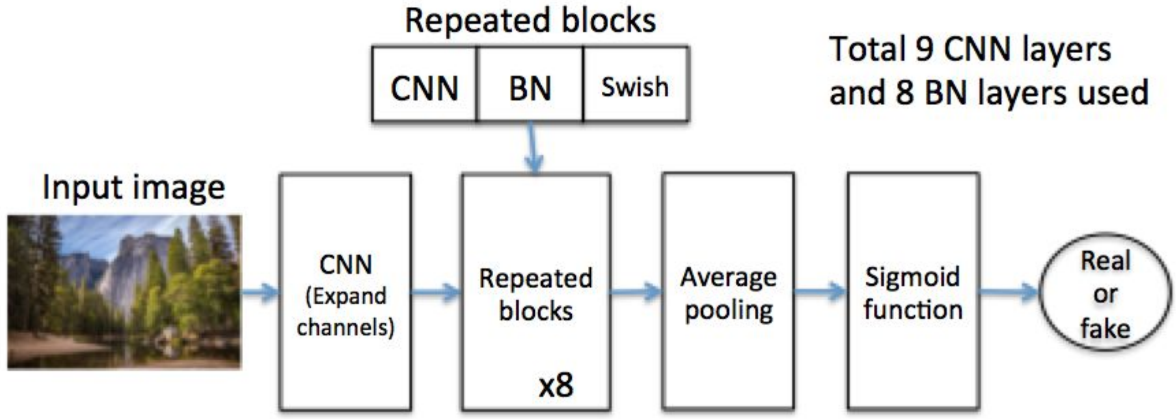


Fig. 7: Discriminator

The discriminator operates a binary classification between 'real' and 'fake/generated' on each input image. The first CNN layer expands the input image's channels as the generator did. Then, eight repeated blocks, each consisting of one convolutional and one BN layers, were used to expand the channel size by a factor of 2 every even number of layers. The final output channel size is one, and the average pooling and the last sigmoid function make the output a probability with '0' meaning that the input is recognized as an image generated by the generator and '1' signifying that the input is considered as a real HR image. Another way to structure the repeated blocks was to make each of the first four repeated blocks to expand the channel size by a factor of 2 and each of the last four repeated blocks to decrease the channel size by a factor of 2. Ultimately, the first way of structuring the blocks was quite effective for GAN, so this was used.

As specified in the "Background and Related work" section, the generator uses a combination of the adversarial (BCE-loss from the discriminator) and the content (MSE loss) losses [4]. The discriminator uses a BCE-loss function twice in each training loop: once between the outputs of the discriminator (probabilities) on real HR images against 1s and once between the probabilities on generated images against 0s. The discriminator's loss measures how well the generator can create realistic SR images.

For both networks, Adam optimizer, swish, and LeakyReLU (with alpha=0.01) activation functions (Fig. 5) were used. The swish function performed slightly better than LeakyReLU. Additionally, swish is continuous and differentiable at x=0 and is effective for deeper networks [8]; hence, swish was chosen to be used longer during training.

Baseline Model

We chose a baseline model that does not use NN to judge the difficulty of the project and how well the GAN performs. The baseline model takes in an LR image and creates an empty image twice its height and width. The new image has alternating rows and columns filled with the original image's pixels. Then empty pixels are filled using the average of the adjacent neighboring pixels. This function is called twice overall to increase the pixel resolution by 4x in each dimension. However, this method does not address the spatial resolution, so it cannot enhance details.

Qualitative Results

To qualitatively assess the models, Mean-Opinion-Score (MOS) [9] based on human ratings were used on LR and HR images and images generated from the baseline model and the GAN. MOS is calculated by taking the arithmetic average of the ratings. The scoring is out of 5 with 1 having the worst perceptual quality. Although MOS may include biases, MOS is a valid assessment to use for this project because of the project's goal to create an ML model that can increase both pixel and perceptual resolutions.

The following compares the generator's and the baseline model's qualitative performances. Values in parentheses are PSNR values for future reference.

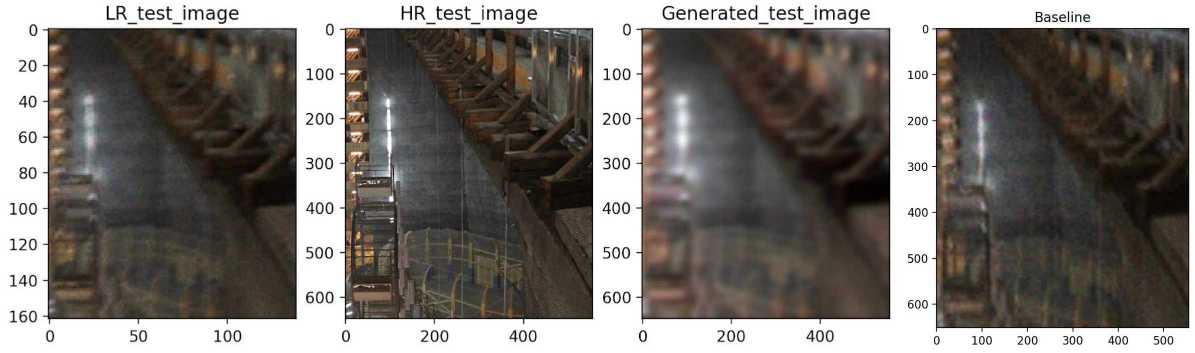


Fig. 8 Input LR, Expected HR, Output of the baseline model (16.1), Output of the generator (15.5)

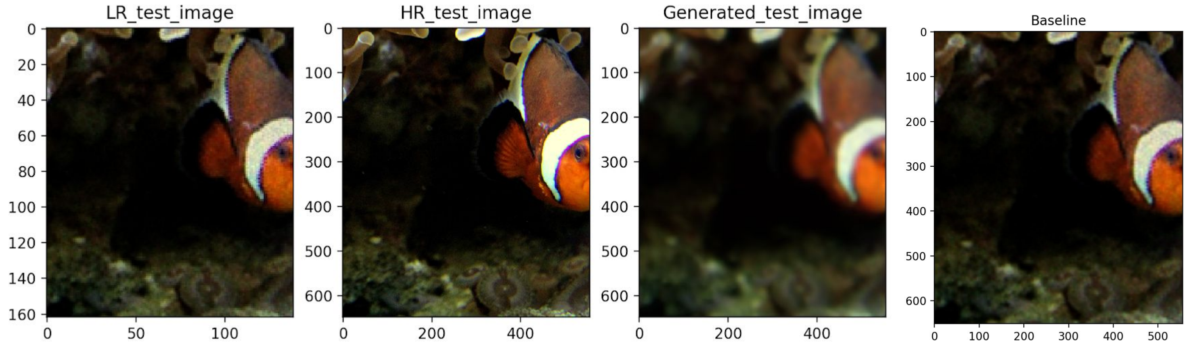


Fig. 9 Input LR, Expected HR, Output of the generator (19.1), Output of the baseline (20.5)

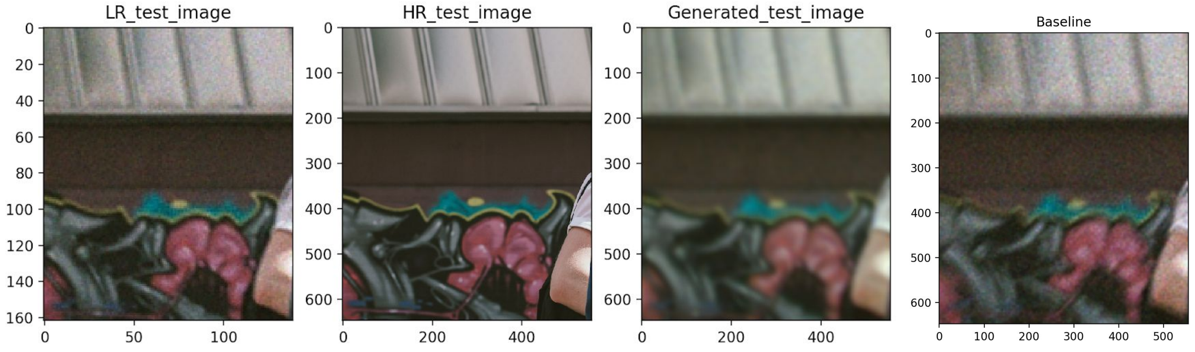


Fig. 10 Input LR, Expected HR, Output of the generator (22.8), Output of the baseline model (22.0)

As shown above, the GAN performs similarly to the baseline model. Although GAN increases the pixel resolution by 4x in each dimension, it is not improving the spatial resolution or details. Rather, it makes images appear more smooth. The MOS for this set of output is 3.4 for GAN and 3.6 for baseline.

Quantitative Results

The generator's loss and PSNR (peak signal-to-noise ratio) values calculated between the generated and the original HR images were used to quantitatively measure the GAN's performance.

Firstly, the changes in the generator's loss are essential in observing whether the generator is learning to create realistic HR images or not. The model was trained for over 5 epochs and a batch size of 16 with a learning rate of around 0.0001.

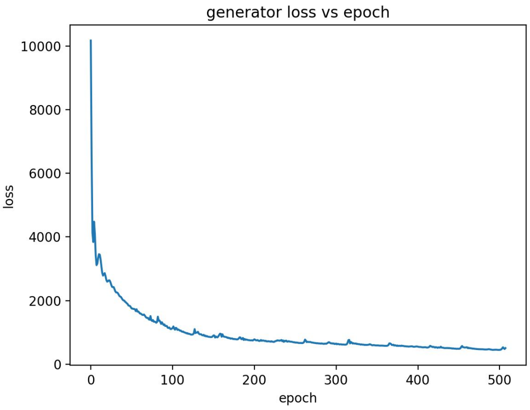


Fig. 11: Trained on Swish

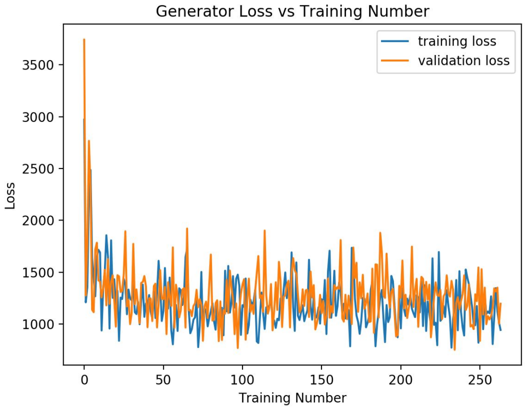


Fig 12. Training Loss and Validation Loss

Both training and validation losses of the generator decreased dramatically at the earlier training stage, which is correlated to the improving resolution of the images as shown in Figs. 8-10.

Secondly, PSNR measures the differences between the two images using mean-squared-error, and it is calculated using the following equations with R=1. The higher the PSNR, the better the quality of the reconstructed images [10]. For comparison, extremely HR images tend to have PSNR=60-80 and highly compressed images tend to have PSNR=30-50.

$$PSNR = 10 \log_{10}\left(\frac{R^2}{MSE}\right) \qquad MSE = \frac{\sum_{M,N}[I_1(m,n) - I_2(m,n)]^2}{M * N}$$

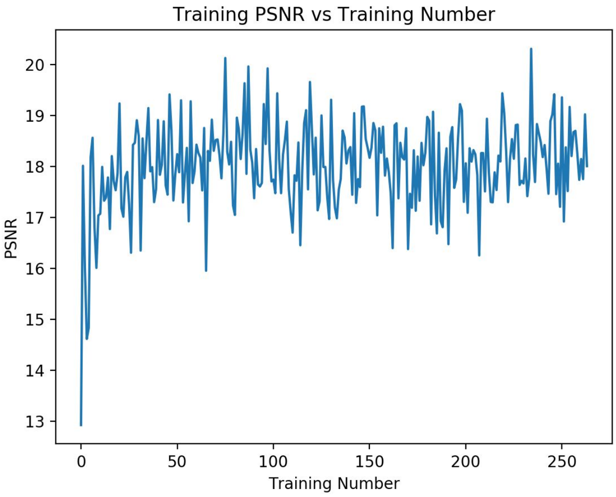


Fig. 13 PSNR value for training with Leaky ReLU

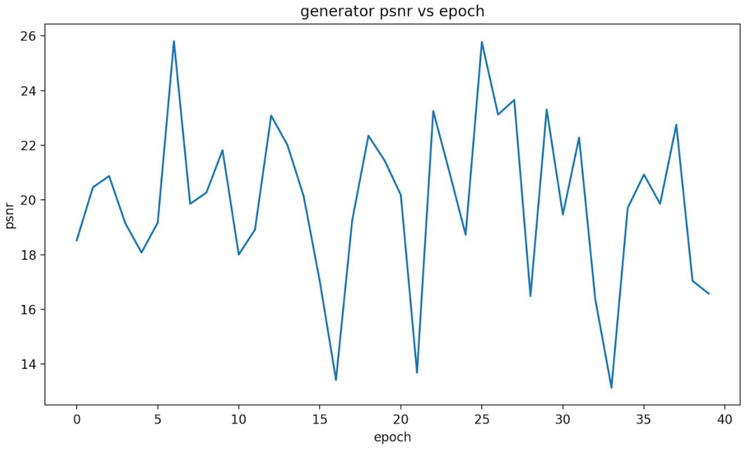


Fig. 14 PSNR values for last 40 epochs with swish

As mentioned in the architecture, we have implemented both the LeakyReLU and the swish. From Figs. 13-14, the swish function performed slightly better in terms of PSNR of average 20, whereas LeakyReLU had PSNR of average 18.

Discussion and Learnings

Overall, from decreasing loss curves and increasing PSNR graphs, the GAN seems to be training and improving well. However, the PSNR values on the test images shown above suggest that there is still space for improvement. We considered the following.

Firstly, as shown in Fig. 12, although the loss quickly decreases at the beginning, it fluctuates afterward. Additionally, the PSNR value is increasing at the start of the training, but it later stops increasing and starts to fluctuate. These possibly suggest that the model is stuck at a local minimum potentially due to the small learning rate used during training; hence, these could be improved using adaptive-learning-rate. In addition, more residual blocks in the generator to create a deeper network could help to produce better outputs.

Secondly, contrary to our prediction, the generator trained quite fast at the beginning to learn various features of the images, including basic color schemes, lines, and shapes of the objects. Nevertheless, GANs are notably hard to train, especially because of the simultaneous training of two networks. In the future, the output SR images can be further improved by training the generator separately to produce more realistic SR images before we start training the discriminator.

Lastly, if multiple GPUs and storage spaces can be allowed, the deeper network can be trained on more images to further enhance the resolution.

Ethical Framework

All the images in the DIV2K dataset were collected from the Internet, and it has a comment: "If any of the images belong to you and you would like it removed, please kindly inform us, we will remove it from our dataset immediately" [7]. Hence, the dataset can violate respect for autonomy, not only because of the copyright of images, but also because the dataset consists of many images of human faces.

The GAN, after properly trained well, can be used for various applications as mentioned before. For example, capturing greater details on medical images for surgical purposes can greatly enhance the quality of medical care people could receive due to the precision it can provide. Moreover, satellite imaging using SR can be also used beneficially, especially in deepening our understanding of natural disasters and weather, which could help us make emergency response quicker and more effectively.

References

[1] C. Thomas, “Deep learning based super resolution, without using a GAN,” Towards Data Science. [Online]. Available: https://towardsdatascience.com/deep-learning-based-super-resolution-without-using-a-gan-11c9bb5b6cd5. [Accessed October 25, 2019].

[2] A. Singh, J. S. Sidhu, “Super Resolution Applications in Modern Digital Image Processing,” International Journal of Computer Applications, Vol. 150, No. 2, September, 2016. [Online]. Available: https://pdfs.semanticscholar.org/d576/d9b9f941537953fd833629f8476235c7db28.pdf. [Accessed October 25, 2019].

[3] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C. Liang, M. Levoy, P. Milanfar, “Handheld Multi-Frame Super-Resolution,” Google Research, May, 2019. [Online]. Available: https://arxiv.org/pdf/1905.03277.pdf. [Accessed October 25, 2019].

[4] C. Ledig, L. THeis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” May, 2017. [Online]. Available: https://arxiv.org/pdf/1609.04802.pdf. [Accessed October 25, 2019].

[5] Z. Wang, J. Chen, S. Hoi, Fellow, IEEE, “Deep Learning for Image Super-Resolution: A Survey,” Feb, 2019. [Online]. Available: https://arxiv.org/pdf/1902.06068.pdf. [Accessed November 30, 2019].

[6] R. Timofte, E. Agustsson, S. Gu, J. Wu, A. Ignatov, L. Van Gool, “DIV2K dataset: DIVerse 2K resolution high quality images as used for the challenges @ NTIRE (CVPR 2017 and CVPR 2018) and @ PIRM (ECCV 2018)”. [Online]. Available: https://data.vision.ee.ethz.ch/cvl/DIV2K/. [Accessed October 25, 2019].

[7] V. Sinha, “Super Resolution GAN(SRGAN)”. [Online]. Available: https://medium.com/analytics-vidhya/super-resolution-gan-srgan-5e10438aec0c. [Accessed November 25, 2019].

[8] P. Ramachandran, B. Zoph, Q. V. Le, “Searching for Activation Functions,” Google Brain. [Online]. Available: https://arxiv.org/pdf/1710.05941.pdf. [Accessed November 15, 2019].

[9] Z. Wang, J. Chen, S. Hoi, Fellow, IEEE, “Deep Learning for Image Super-Resolution: A Survey,” Feb, 2019. [Online]. Available: https://arxiv.org/pdf/1902.06068.pdf. [Accessed November 30, 2019].

[10] “PSNR,” MathWorks. [Online]. Available: https://www.mathworks.com/help/vision/ref/psnr.html. [Accessed November 15, 2019].

Permissions:

- Permission to post video: yes
- Permission to post final report: yes
- Permission to post source code: yes