Machine Learning Engineer Nanodegree

Capstone Project Report

Topic: Super-resolution using Convolutional Neural Networks (SRCNNs)

1. Definition

1.1 Introduction

Image processing techniques for enhancing the quality of images has been a topic of interest in the domain of Computer Vision for a long time. In general, two types of methods have been in practice, which are Spatial Domain Methods and Frequency Domain Methods [1]. Some of the methods listed in this paper include Histogram Equalization, Point Processing operations and other mathematical transformations on pixel data of the image to enhance the quality of gray scale images.

Image 1: Technique of Super-resolution Imaging

Super-resolution imaging (SR) is a class of techniques that enhance the resolution of an imaging system. This operation is performed using a variety of techniques which include performing geometric transformations on the image data, making use of optical or diffraction based methods, etc. Recently, the field of machine learning has given a great impetus to research in this domain with the use of Convolutional Neural Networks (CNNS) and Generative Adversarial Networks (GANs).

The applications of super-resolution imaging are immense and range from enhancing the picture quality of images taken using cameras, phones, etc. to enhancing the images taken by machinery in the healthcare domain. The need for higher quality images and videos has further fuelled the research in this domain to produce videos and images of higher resolution and more clarity.

Where we were once satisfied with 480p quality of videos, today, even 1080p quality seems to be unacceptable and we are headed towards 4k picture quality for videos.

Most conventional methods use concepts like nearest-neighbors and interpolation methods (bi-linear, bi-cubic, etc.) to enhance the picture quality. Although these conventional methods seem to give reasonable results, with the advent of complex deep learning models, it has become easier to train neural network models that learn a mapping between the low resolution images and high resolution images and help transform new low resolution images to their high resolution counterparts.

Image 2: Results from the experiment by ETH Zurich Students on DPED dataset

With the use of deep learning techniques, an image of lower picture quality can be enhanced to a higher quality image as is done by mapping low-quality phone photos into photos captured by a professional DSLR camera, by the students of ETH Zurich with the use of their self-created DPED dataset. [2]

Generative Adversarial Networks (GANs) are a class of neural networks which have gained immense popularity in the past few years. Put most simply, they allow a network to learn to generate data with the same internal structure as other data without giving it direct access to the trained data. If that description sounds a little general, that is because GANs are powerful and flexible tools. To explain this concept in further detail, let us take one of the most common applications of GANs which is image generation. Say you have a bunch of images, such as pictures of cats. A GAN can learn to generate pictures of cats *like* those real ones used for training, but not actually *replicate* any one of the individual images. If given enough cat pictures, it actually learns about the features that define a “CAT” and learns to generate images that meet this standard. With the advancements in the field of Generative Adversarial Networks (GANs), these techniques are also implemented to achieve the goal of super-resolution [3]. However, GANs need high computational power to train and are difficult to train as hyper-parameter tuning plays a major role in the training of such complex networks. Hence, the scope of this project is restricted to use of Convolutional Neural Networks for the task of Super-resolution imaging only.

1.2 Problem Statement

The problem statement is to train a convolutional neural network based model that will act as a mapping between low-resolution images and equivalent high-resolution images. This process is commonly known as Super-Resolution Convolutional Neural Networks (SRCNNs).

A simple introduction is given in the medium article [4]. In order to explain in layman terms, for the preprocessing step, the low resolution image is interpolated to the size of desired high resolution image. After the preprocessing step is done, the neural network is built that consists of three stages – first is to extract patches using convolutional filters, second is to create a non-linear mapping between these filters and new high-resolution filters, and at last the final layer involves reconstructing the image from the new high-resolution filters.

As my capstone project, I intend to train a deep convolutional neural network to achieve the goal of single image super-resolution based on the methodology as described in this paper [5]. I plan to attempt to get results as close to the ones mentioned in the paper as possible. This trained CNN model can be further improved using hyperparameter tuning and scaled for higher dimensions. The main challenge in training these models is the computational complexity and the resources needed to train these heavy CNN models are quite huge. So, here I am trying to establish the concept of super-resolution and will scale it up in future to higher dimension models and maybe video super-resolution too. The scope of this project is limited to implementation of the method stated in the research paper and testing it for multiple datasets.

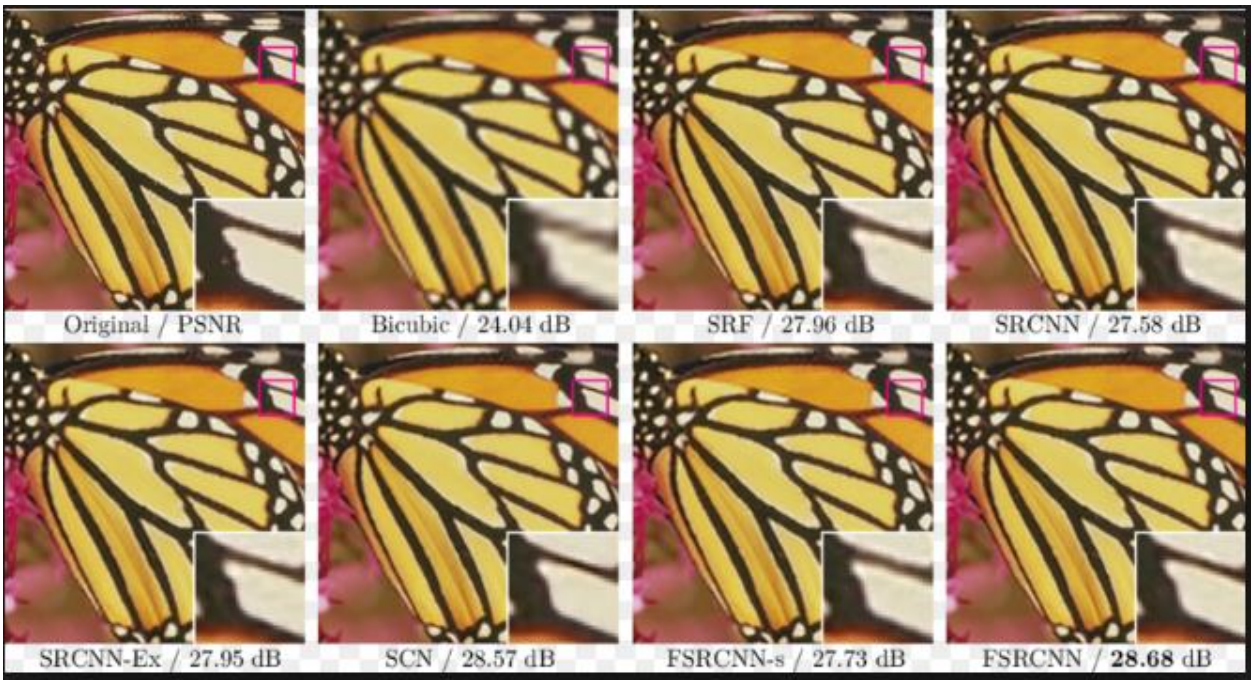

Image 3: Results mentioned in the reference paper
(We can observe that SRCNN performs better than simple bi-cubic interpolation)

1.3 Metrics

The primary metric of evaluation is the PSNR (Peak signal-to-noise ratio) which is the implementation of the mean squared error in the image processing domain. Higher the value of PSNR, lower is the mean-squared error of the model.

Another more suitable metric of evaluation is the structural similarity (SSIM) index which helps predict the perceived quality of pictures. Higher PSNR means more noise is removed but being a least squares result, it is slightly biased towards over smoothed results, that is along with the noise, some key features and textures of the image might be removed only to generate a high PSNR score. SSIM, on the other hand, has quality reconstruction metric that considers the similarity of the edges (high frequency content) between the lower and higher resolution images. To have a good SSIM measure, an algorithm needs to remove the noise while also preserving the edges of the objects. SSIM is a "better quality measure", but it is more complicated to compute and hence, PSNR is the preferred evaluation metric.

$$
\begin{aligned}
PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\
&= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\
&= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE)
\end{aligned}
$$

Image 4: Peak Signal to Noise Ratio (PSNR) metric

The PSNR is a widely-used metric for quantitatively evaluating image restoration quality and hence, will be used to evaluate our model.

2. Analysis

As mentioned in the reference paper, I have chosen to use the famous T91 (91- image dataset) as an initial study to try and replicate the results mentioned in the paper which will help me gain an intuition about the working of the algorithm. Similar, to the paper, I use Set5 as the validation set for the SRCNN model along with two other datasets, Set14 and Urban100. All the mentioned datasets are openly available here. [6] The primary reason for choosing a smaller dataset for proof of concept is the computational constraints. However, as mentioned in the paper, concept can be scaled for larger datasets.

2.1 Data Preparation

The T91 images dataset contains 91 high resolution images. For the first part of the project, we will be extracting multiple images of size 33X33 pixels using these high resolution images. With

a stride of 8, I have extracted 65895 images of 33X33 pixels which will form the training set for the SRCNN model.

Some of the images from this generated training set are shown below:

Image 5: Sub-images generated from T91 dataset (for SRCNN training)

Other datasets are Set5, Set14 and Urban100 datasets which are used for the validation of the SRCNN model. The Manga109 dataset is used for training the SRCNN for the second half of the project where we create a SRCNN model to scale 32X32 pixels size image by a factor of 2.

Before we scale an image from low to high resolution, we need to alter the pixel size of image which is done by performing some kind of mathematical transformation like in our case; we perform bi-cubic interpolation as the preprocessing step. For example, if we have a 32X32 pixel RGB image and we need to enhance it to 64X64 pixel RGB image, the first step is to resize the

image from 32 to 64 pixels using the resize feature of the OpenCV library by performing a bi-cubic interpolation operation. This results in an image which is of the size matching our high-resolution image output but the quality of image is not impressive. As the image is generated using interpolation, we notice that most of its key features like sharp lines and ends seem to have become blurred due to smoothing operation performed during the process of scaling.

The Manga109 dataset was chosen because the images in it are of better quality with sharp edges, well-defined bright objects in images and also of larger pixel sizes as needed for extracting sub-images of 64X64 pixels for the second half of the project. Some of the extracted images are shown below:

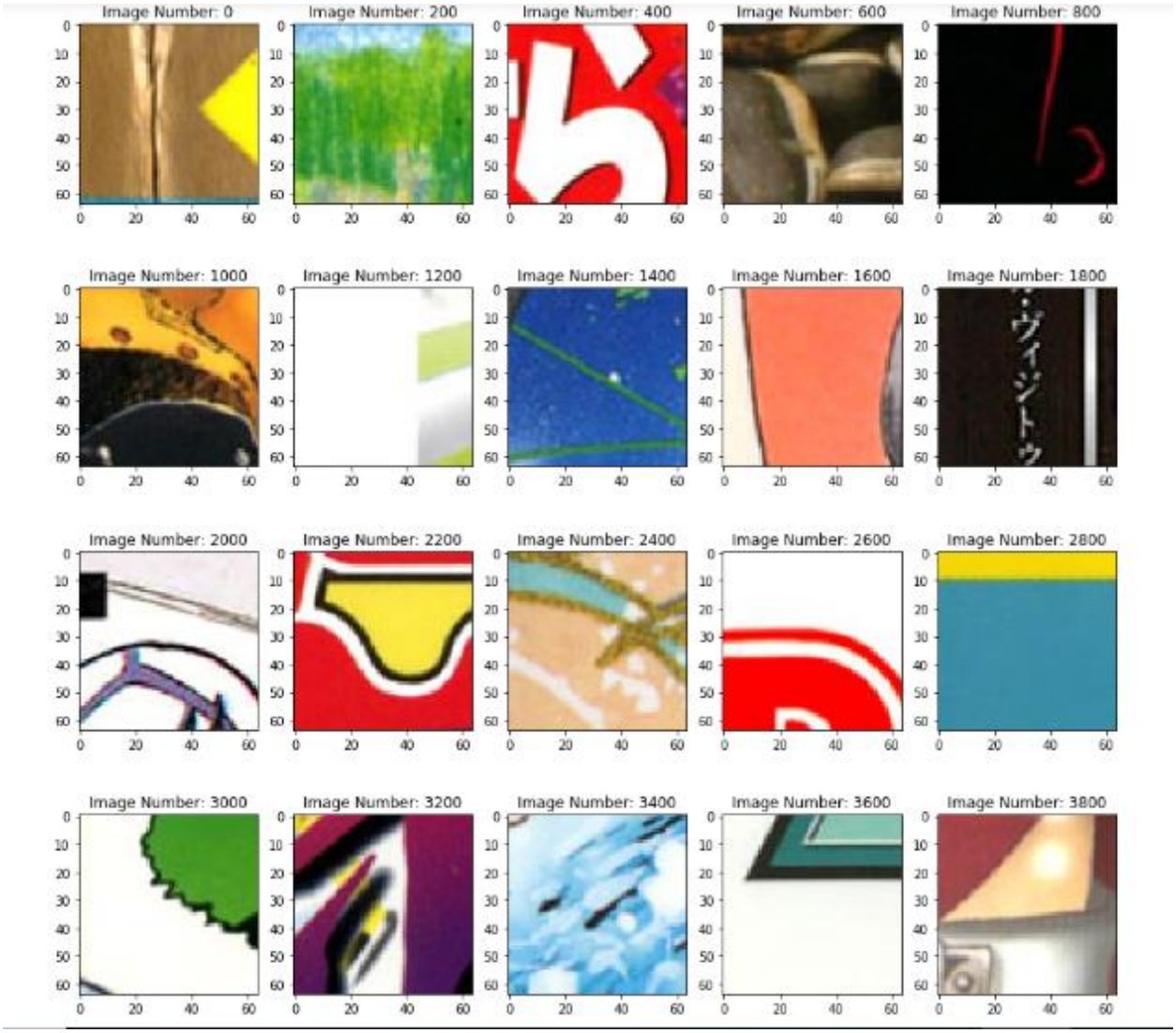

Image 6: Training images from Manga109 dataset of size 64X64 pixels

For the first half of the project, we downscale the training images by a scale factor of 3, use these images of 11X11 pixels which are our low resolution images. These images are interpolated to 33X33 pixels using bi-cubic interpolation as preprocessing step. The original 33X33 pixels images are the labels for the process of supervised learning using CNN model.

2.2 Convolutional Neural Network Model

For the first case, I have chosen the same model mentioned in the paper which consists of three layers for super-resolution. The three layers are as follows:

1] Patch extraction: Patches from the scaled low-resolution image are extracted which capture the key features of the image and form the set of feature maps used in a convolutional network training model.

2) Non-linear mapping: This step non-linearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch.

3) Reconstruction: The final step aggregates the high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be like the original high-resolution image X and the goal of the model is to minimize the error (difference between the obtained image and the original high-resolution image).

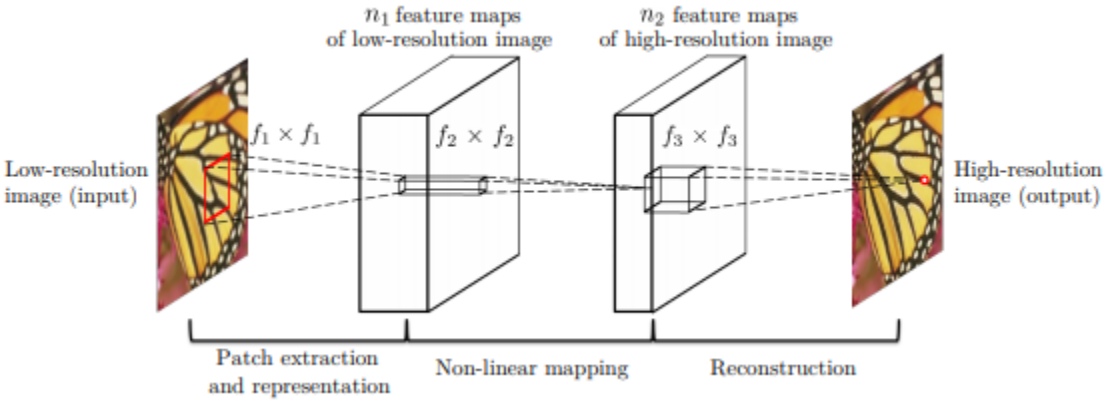

Image 7: The three stages of SRCNN model

The three layers of Convolutional2D are generated using a Sequential model built using the Keras library on Tensorflow backend. The hyperparameters for the first part are chosen from the base case mentioned in the paper while later I have tested it for different values to generate a CNN model that will create a mapping specifically for conversion from 32X32 pixels image to image of 64X64 pixels (scale factor 2).

2.3 Benchmark Model

The mentioned paper states that the SRCNN model trained rigorously for several days on the T91 dataset resulting in an average PSNR score of 32.39 dB while the T91+ImageNet Dataset give an average PSNR value of 32.52 dB

To get these results, the amount of training required is quite huge (extending to several days of training on powerful GPUs) and so even after generating more images than those mentioned in the paper; the model is not able to perform to the stated state-of-the-art methods. So, for comparison we check whether we have desirable improvement over the simple bi-cubic interpolation step. So, in our case, the PSNR value obtained for a bi-cubic interpolation image happens to be our baseline score and this becomes our benchmark model.

3 Methodology

As mentioned earlier, bi-cubic interpolation is the only preprocessing step before passing the images to the CNN model. The model has three layers with ReLu activation function in each layer. The three layers have filters of sizes 9X9, 1X1 and 5X5 respectively and the number of filters is 64, 32 and 3 respectively. The reason for selecting 3 filters for the last layer is that the number of channels for the images are 3 namely red, blue, green (RGB). This was in accordance with the network mentioned in the research paper. Using a batch size of 128 and validation split of 0.2 (20% data), the model is trained for 50 epochs.

For the second part of the project, I have selected the Manga109 dataset and extracted 64X64 high-resolution images. The images would be then downscaled to 32X32 pixels, interpolated using bi-cubic interpolation to 64X64 pixels and then training similar to first half is carried out. After several rounds of refinement, a network of three layers is chosen, where the first layer has 9X9 size filters, second layer has 3X3 size filters and the last layer has 8X8 size filters. The number of filters in each of the layers is changed to 128, 64, and 3 layers respectively. Using a batch size of 128 and validation split of 0.2 (20% data), the model is trained for 30 epochs.

4 Results

After training for 50 epochs on the T91 images generated training set that consists of 65895 images of size 33X33 pixels with a scaling factor of 3, a reasonable PSNR score of 28.65 dB is obtained. For the validation process, when validating on Set5 dataset, we get average PSNR of 31.06 dB for 2488 images. Set14 Dataset with 19877 generated images gives 27.987 dB and the Urban100 Dataset (25550 images) gives 25.83 dB as validation PSNR score.

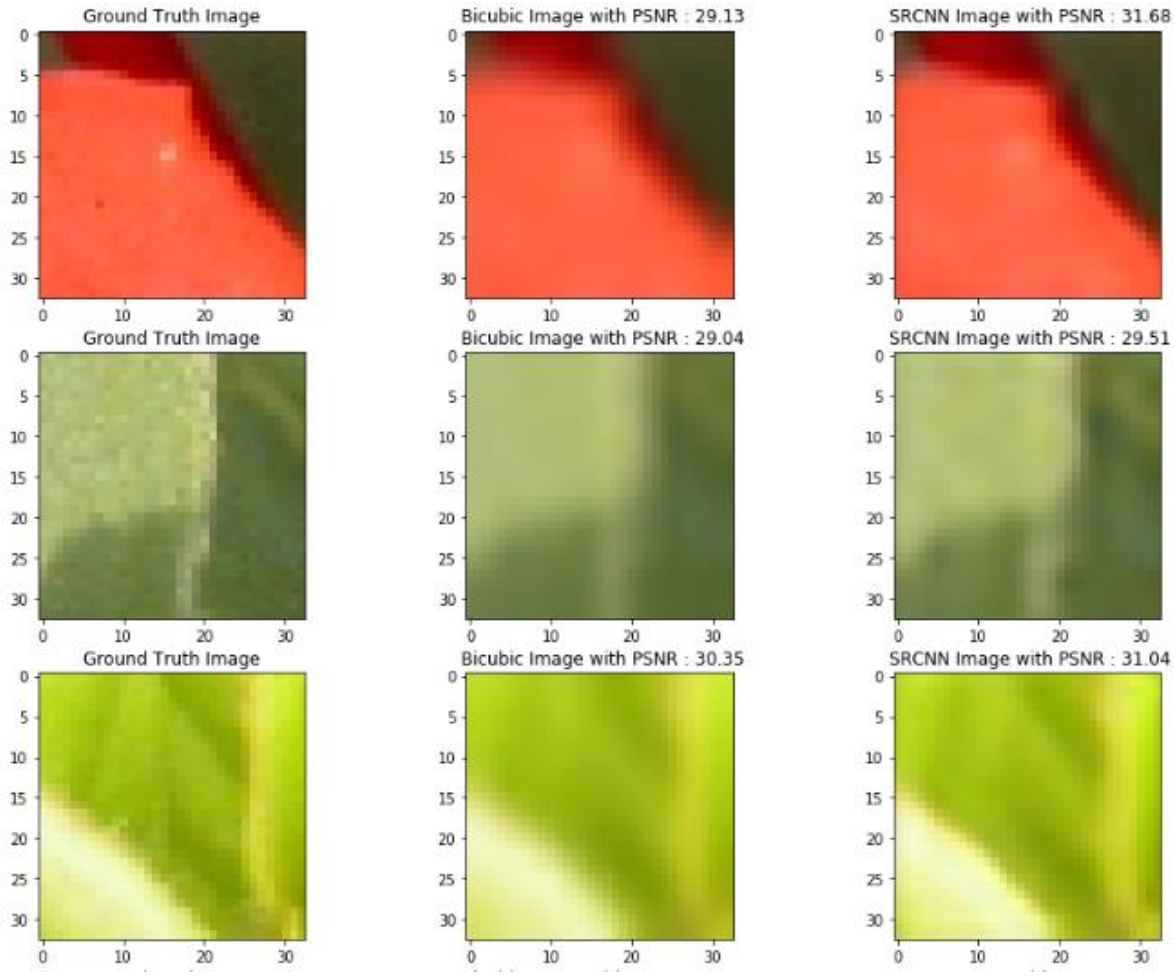

Image 8.1: Sample Results from first trained model

We compare individual images with their bi-cubic interpolated counterparts to observe that we have an improved resolution after passing them through our trained SRCNN model. In most of the images, we can observe from the PSNR values that our SRCNN model has brought an improvement in the image over the bi-cubic interpolation step. The purpose of this exercise was to prove that the SRCNN model is a feasible and useful process to upscale the images and aid in the process of super-resolution.

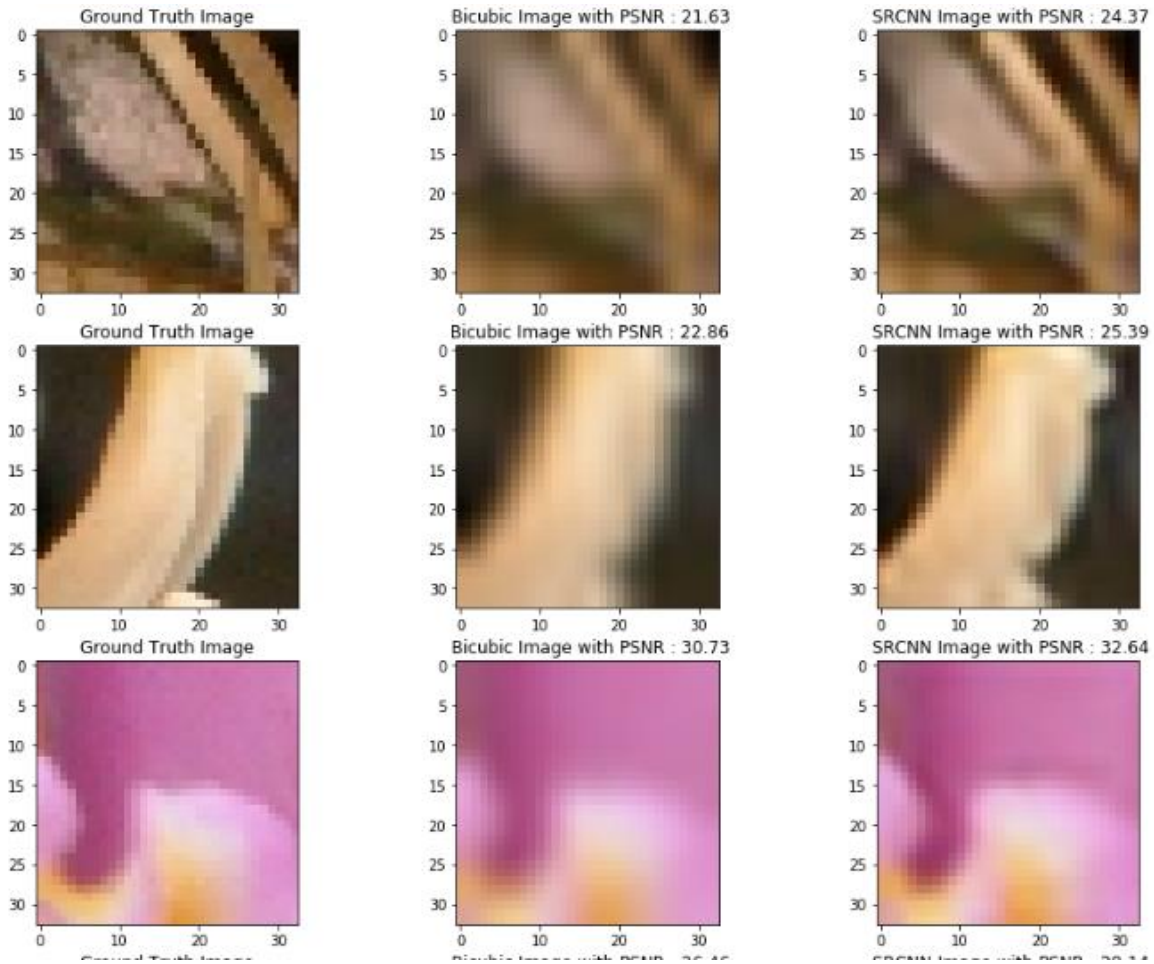

Image 8.2: Sample Results from first trained model

In the second part of the project, the described model is trained on 41802 images of 64X64 pixels generated from the Manga109 dataset that has bright colors and sharp-edged images with a scaling factor of 2. After training for 30 epochs, we get a training PSNR score of 31.94 dB on training dataset. For validation, the PSNR scores are: Set5 (441 images) gives 32.73 dB, Set14 (17459 images) gives 29.85 dB and Urban100 (18221 images) gives 28.45 dB

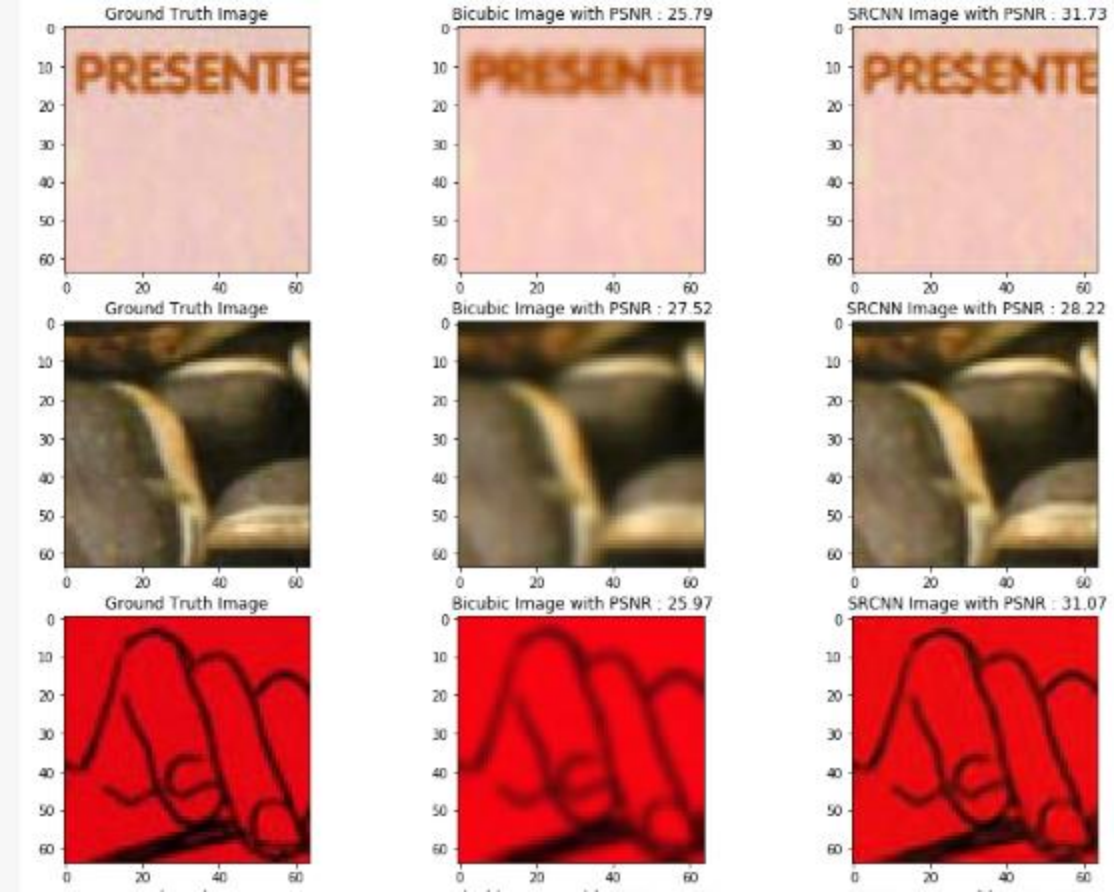

Image 9.1: Sample Results from second trained model

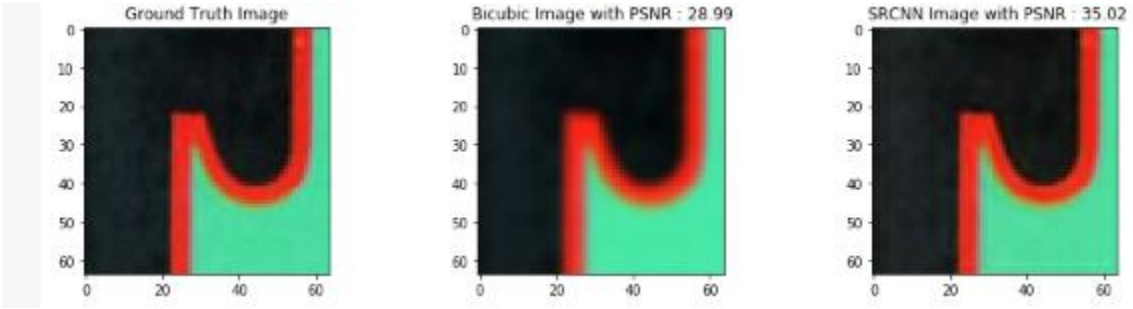

Image 9.2: Sample Result with great improvement in PSNR score (around +6dB)

We test this trained model on other images to understand how well the model scales up for larger images. An attempt is made to try and apply this model to large images by taking patches from the image of sizes 32X32 pixels and scaling them by a factor of 2 but the resulting images tend to have local super resolution effect and the effect in one frame is independent of the effect in the others resulting into lack of improvement in image resolution at later stages.

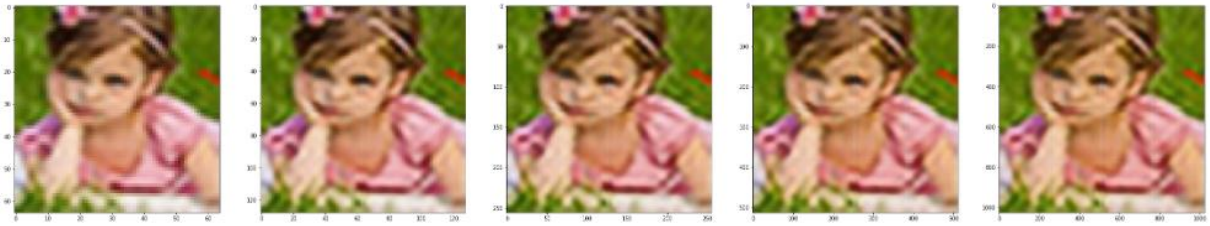

Image 10: Attempt at enhancing the image patch-wise by 2X in each stage

(Not much improvement is observed at later stages)

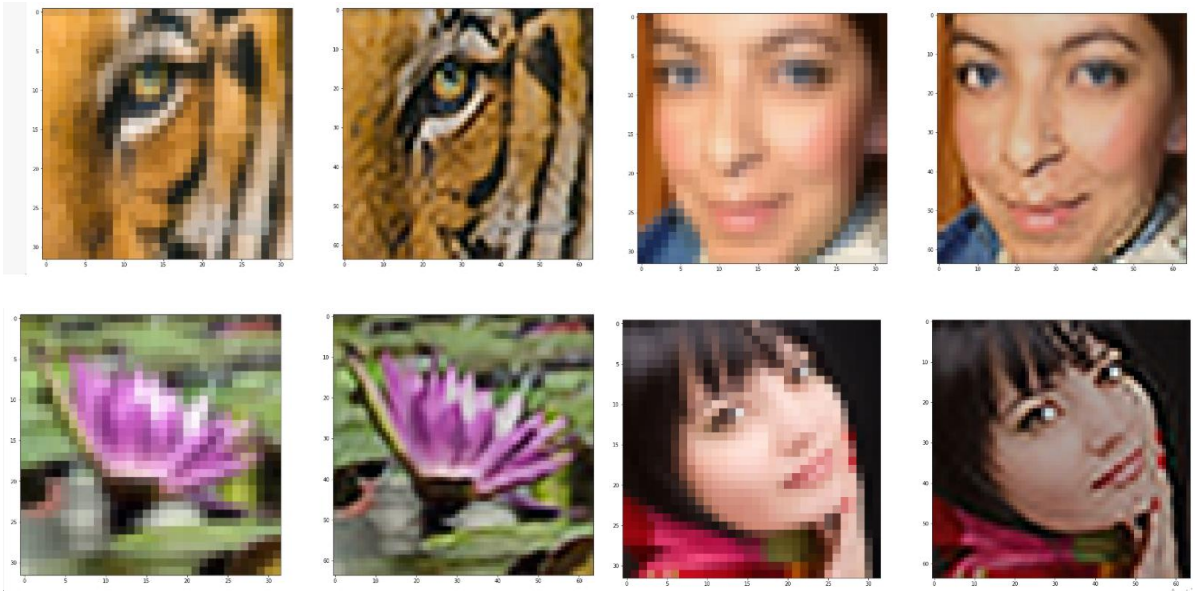

Image 11: Testing the trained model on 32X32 pixel images

Left: low-resolution input (32X32) and Right: high-resolution output (64X64)

5 Conclusions and Future Work

In most of the images, we can observe from the PSNR values that our SRCNN model has brought an improvement in the image over the bi-cubic interpolation step. The purpose of this project is to establish the process of SRCNN model to upscale the images and aid in the process of super-resolution. In future, I would love to use this concept, train better models and extend this concept of SRCNN to handle large images. The major challenge in training larger models is the availability of computational resources required for such processes. I would love to utilize the trained SRCNN models to the domain of videos at a later point of time.

THANK YOU

References

[1] https://arxiv.org/ftp/arxiv/papers/1003/1003.4053.pdf

A Comprehensive Review of Image Enhancement Techniques, Raman Maini and Himanshu Aggarwal
JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010

[2] http://people.ee.ethz.ch/~ihnatova/#dataset

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool.
"DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks",
in IEEE International Conference on Computer Vision (ICCV), 2017

[3] https://arxiv.org/pdf/1609.04802v5.pdf

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
CVPR 2017 • Christian Ledig • Lucas Theis • Ferenc Huszar • Jose Caballero • Andrew Cunningham • Alejandro Acosta • Andrew Aitken • Alykhan Tejani • Johannes Totz • Zehan Wang • Wenzhe Shi

[4] https://medium.com/coinmonks/review-srcnn-super-resolution-3cb3a4f67a7c

Review: SRCNN (Super Resolution)

[5] https://arxiv.org/pdf/1501.00092.pdf

Image Super-Resolution Using Deep Convolutional Networks
Chao Dong; Chen Change Loy; Kaiming He; Xiaoou Tang

[6] http://vllab.ucmerced.edu/wlai24/LapSRN/

91-image dataset and other validation datasets