B.Tech Project Report

Image Super Resolution Using Generative Adversarial Network

Submitted in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering

Submitted by

Manu Maheshwar P (Roll No 14400028)
Mridul Joshi (Roll No 14400069)
Nabeel Kabeer (Roll No 14400033)
Nithin Chandran P (Roll No 14400036)

Under the guidance of

Dr. Ajeesh Ramanujan

Department of Computer Science and Engineering
College of Engineering, Trivandrum
Kerala
April 2018

CERTIFICATE

This is to certify that the thesis entitled "Image Super Resolution using Generative Adversarial Network" is a bonafide record of the major project done by **Manu Maheshwar P** (Roll No 14400028), **Mridul Joshi** (Roll No 14400069), **Nabeel Kabeer** (Roll No 14400033) and **Nithin Chandran P** (Roll No 14400036) in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering from the University of Kerala for the year 2018.

Dr. Ajeesh Ramanujan
(Guide)
Assistant Professor
Dept. of Computer Science and Engineering

Mrs. Shreelekshmi
Professor and Head of
Dept. of Computer Science and Engineering

Place: Trivandrum
Date: 25-04-2018

Abstract

With the current surge in popularity of image-based applications, improving content quality is vital. Single Image Super-Resolution involves increasing the size of a small image while keeping the attendant drop in quality to a minimum. The task has numerous applications, including in satellite and aerial imaging analysis where the total area of the land on Earth is so large and because resolution is relatively high, satellite databases are huge and image processing (creating useful images from the raw data) is time-consuming. Other applications include medical image processing, compressed image/video enhancement and facial image analysis, text image analysis, sign and number plates reading, and biometrics recognition, to name a few.

Our objective is to take a low-resolution image and produce an estimate of a corresponding high-resolution image. This problem is difficult : multiple high-resolution images can be produced from the same low-resolution image. For instance, suppose we have a 2x2 pixel sub-image containing a small vertical or horizontal bar. Regardless of the orientation of the bar, these 4 pixels will correspond to just one pixel in a picture downscaled 4 times. With real-life images, one needs to overcome an abundance of similar problems, making the task difficult to solve.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image processing is an interesting field with a lot of areas of research. This includes recognizing the contents of an image, manipulating an image, increasing the resolution of an image, etc. Image Super Resolution is an extensive area of research within itself. A lot of research has gone into solving the problem of Single Image Super Resolution which has resulted in methods with better accuracy and speed. Even then, one problem remains largely unsolved. The problem of recovering the finer texture details at large upscaling factors. The choice of the objective function plays a major role in the behaviour of optimization based super resolution methods. The previous works have mainly focussed on minimizing the Mean Squared Error Loss (MSE Loss). Minimization of MSE Loss gives solutions with high signal to noise ratio but are seen lacking in high frequency details. Generative Adversarial Networks can be used for Image Super Resolution to produce photorealistic images for 4X upscaling factors.

The report has been divided into six chapters. The second chapter gives an insight into the background information needed in the understanding of the report.The third chapter handles the literature survey done as part of the project. Several papers were referred during the course of completion of this project. They are spoken about in the third chapter. The fourth chapter essentially deals with the procedure. That is, the several steps involved in completion of the project. The results obtained are summarized and analyzed in the fifth chapter. The report concludes with the sixth chapter.

Chapter 2

Problem Statement

Image super resolution is a problem that has always attracted the attention of software engineers and scientists. As a result there has been many methods addressing the problem including mathematical solutions like Bicubic interpolation and several neural networks solutions. But a central problem remains largely unsolved in all these cases: how to recover the minute texture details while super resolving?

The existing mathematical methods are based on interpolations which results in images with overly smooth edges and least perceptul appeal and the neural networks address the problem by focusing on minimising the Mean Square Error (MSE) loss wherein the resulting image has higher signal to noise ratio but lacks the fidelity expected at the higher resolution.

Chapter 3

Background Information

3.1 Image Super Resolution

Images are very important sources of information. Images have applications in a wide variety of fields. Image capturing equipments have not been able to achieve the capacity to produce high quality images under all circumstances. Recent upsurge in the popularity of image based applications has necessitated the need for content quality. Image Super Resolution (ISR) aims to increase the size of an image while keeping the accompanying loss in quality to minimum.

3.2 Generative Adversarial Network

Generative Adversarial Networks (GANs) are a class of machine learning algorithms consisting of two neural networks contesting with each other in a zero-sum game framework. GANs were introduced by Ian Goodfellow in his paper Generative Adversarial Nets published in 2014. The concept is that given enough computing power, two neural networks can contest with each other and learn through plain old backpropagation. One network generates while the other evaluates. The generative network learns to map data to a particular data distribution of interest, while the discriminative network discriminates between instances from the true data distribution and candidates produced by the generator. The generative network's training objective is to increase the error rate of the discriminative network . The training procedure needs a dataset of high resolution images. The discriminator is trained by giving images until it reaches a certain level of accuracy. The generator is given as input low resolution images. It produces super resolved versions of its input. The samples generated by the generator are evaluated by the discriminator. Backpropagation is applied to both the networks, where the the discriminator becomes more skilled at flagging the images produced by the generator as fake (due to it not being as good as the original high resolution image) whereas generator becomes more skilled at producing super resolved images that are as close to reality as possible. Thus, this method can be used to generate images that look authentic to human observers, having many realistic characteristics. It is the first framework capable of inferring photo realistic natural images for 4X upscaling factors. The usual convolutional neural networks tries to improve its performance by minimizing the Mean Squared Error Loss(MSE Loss). The MSE Loss is a pixel concept and overly relies on the pixel space. Another loss function that captures the perceptual properties of the image is the VGG Loss Network. As the name suggests, it is a neural network that it used to compute the loss, which is to be minimized. The VGG network is a pre trained network that extracts the features from an image. It takes in the features of the super resolved image of the generator and compares it with the features extracted from the original high resolution image. The use of VGG loss pushes the GAN to produce images of photo-realistic quality.

The presented method and algorithm was first presented in a paper by Christian Ledig, Lucas Theis, Ferenc Huszr and others. It was presented to the world at the Conference on Computer Vision and Pattern Recognition on July 21 2017. The paper presented the method for producing photo realistic super resolved images using Generative Adversarial Networks.

Chapter 4

Literature Survey

ISR is a topic that has captivated scientists for a long time. Earliest methods for Single Image Super Resolution (SISR) were prediction based methods like Bicubic Interpolation and Lanczos filtering. These methods though fast, most often oversimplify the ISR problem and yield solutions with smooth textures. Better approaches establish a complex mapping between low and high resolution information and usually rely on training data. William T Freeman presented a method for ISR that are based on example pairs. Recently convolutional neural network (CNN) based ISR algorithms have shown excellent performance. In Deep networks for image super-resolution with sparse prior by Z.Wang and others, the authors encode a sparse representation prior into their feed-forward network architecture based on the learned iterative shrinkage and thresholding algorithm (LISTA) . In Image super-resolution using deep convolutional networks,C.Dong and C.C Loy used bicubic interpolation to upscale an input image and trained a three layer deep fully convolutional network end-to-end to achieve state- of-the-art SR performance. Subsequently, it was shown that enabling the network to learn the upscaling filters directly can further increase performance both in terms of accuracy and speed. Deeper network architectures have been shown to increase performance for SISR. In Deeply-recursive convolutional network for image super-resolution J.Kim, J.K Lee and K.M Lee formulate a recursive CNN and present state-of-the-art results.

The concept of Generative Adversarial Network is explained in detail in the paper by Ian Goodfellow and others titled "Generative Adversarial Nets". The application of Generative Adversarial Nets to the problem of Single Image Super Resolution is discussed in The architecture for this project was derived from a paper, "Photo-Realistic Single Image Super Resolution Using a Generative Adversarial Network" by Christian Ledig, Lucas Theis and others.The architecture for the generator network and the discriminator network was also obtained from this paper. The architecture involves a very important layer called the Batch Normalization layer . In ”Accelerating Deep Network Training by Reducing Internal Covariate Shift” by Sergey Ioffe and Christian Szegedy, normalizing layer inputs for solving the problem of internal co-variate shift observed during the training of Deep Neural Networks is discussed in length.

In a paper titled ”Perceptual Losses for Real-Time Style Transfer and Super-Resolution” by Justin Johnson, Alexandre Alahi, and Li Fei-Fei the VGG Loss function is explained in detail with reference to its application for the purpose of style transfer in images. In ”Deep Face Recognition”, authors Omkar Parkhi and others give insight into the process of increasing the size of the dataset efficiently to enable deep neural networks to learn better. The need for batch normalization is debated in ”Enhanced Deep Residual Networks for Single Image Super-Resolution” by Bee Lim and others. The paper uses a network architecture without the Batch Normalization layer and achieves comparable results at a much faster rate.

Chapter 5

Materials and Methodology

5.1 Algorithms

5.1.1 Generative Adversarial Network

Generative adversarial networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two neural networks contesting with each other in a zero-sum game framework. They were introduced by Ian Goodfellow et al. in 2014. This technique can generate photographs that look at least superficially authentic to human observers, having many realistic characteristics.

The generative model is pitted against an adversary: a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. The generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistiguishable from the genuine articles.

The discriminator network is a standard convolutional network that can categorize the images fed to it, a binomial classifier labeling images as real or fake. The generator is an inverse convolutional network, in a sense: While a standard convolutional classifier takes an image and downsamples it to produce a probability, the generator takes a vector of random noise and upsamples it to an image. The first throws away data through downsampling techniques like maxpooling, and the second generates new data as shown.

The entire process can be summerized as a game theoretic framework where two player D and G play the following two-player minimax game with

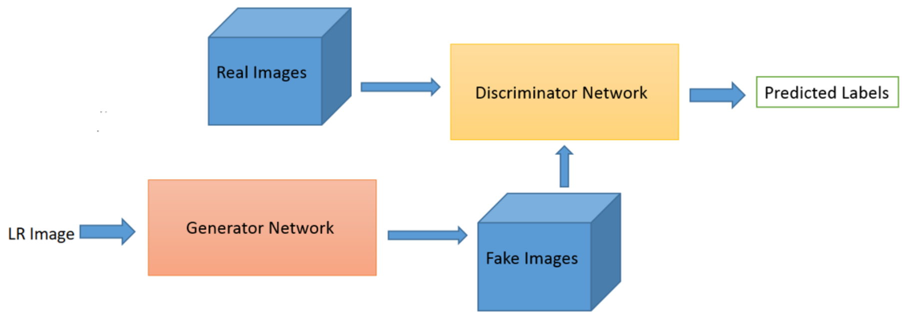

Figure 5.1: GAN workflow diagram

value function V(G,D):

$$\min_{G} \max_{D} V(D,G) = E_{x \sim p_{data}(x)}[log D(x)] + E_{z \sim p_z(z)}[log(1 - D(G(z)))] \quad (5.1)$$

The essence of the equation is that the generator tries to decrease the entropy of the system while the discriminator tries to increase the entropy of the system.Decreasing the entropy of the system allows to increase the predictability of the system.Hence the generator becoming more accurate.

The steps taken by the GAN are:

- The generator takes in low-resolution image and returns a high-resolution-image
- This generated image is fed into the discriminator alongside a stream of high-resolution images taken from the actual dataset.
- The discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a prediction of authenticity and 0 representing fake.

Algorithm 1: Minibatch stochastic gradient descent training fro generative adversarial nets. The number of steps to apply to the discriminator ,k ,is a hyperparameter.

for *number of training iterations* **do**
  for *k steps do* **do**
    Sample minibatch of m low-resolution images $\{z^1, \ldots, z^m\}$ from prior $p_g(z)$
    Sample minibatch of m examples of high resolution images $\{x^1, \ldots, x^m\}$ from data generating distribution $p_data(x)$
    Update the discriminator by ascending its stocastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [logD(x^i) + log(1 - D(G(z^i)))] \tag{5.2}$$

  Sample minibatch of m low resolution image samples $\{z^1 \ldots z^m\}$ from prior $p_g(z)$
  Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^i))) \tag{5.3}$$

The gradient-based updates can use any standard gradient-based learning rule.We used adam-optimizer in our experiments.

5.1.2 Batch Normalization

Training Deep Neural Networks is complicated by the fact that the distribution of each layers inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. We refer to this phenomenon as internal covariate shift, and address the problem by normalizing layer inputs. It accomplishes this via a normalization step that fixes the means and variances of layer inputs. Batch Normalization also has a beneficial effect on the gradient flow through the network, by reducing the dependence of gradients on the scale of the parameters or of their initial values. This allows us to use much higher learning rates without the risk of divergence

Consider a mini-batch B of size m. Since the normalization is applied to each activation independently, let us focus on a particular activation $x^{(k)}$ and omit k for clarity. We have m values of this activation in the mini-batch,

$$B = \{x_{1...m}\} \tag{5.4}$$

Let the normalized values be $\hat{x_{1...m}}$, and their linear transformations be $y_{1...m}$. The BN Transform in Algorithm is shown below.

Algorithm 2: Batch Normalization Transform applied to activation x over a mini-batch

---

Input: Values of x over a mini-batch: B = { $x_{1...m}$}; Parameters to be learned: γ, β

Output: { $y_i = BN_{\gamma,\beta}(x_i)$}

$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$`// mini-batch mean`

$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$ `// mini-batch variance`

$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ `// normalize`

$y_i \leftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$ `// scale and shift`

5.1.3 Pixel Shuffler

The previous works on image super-resolution using deep learning where based on initially interpolating the low-resolution image $\mathbf{I}_{LR}$ to high-resolution image by means of bicubic interpolation and the interpolated image is fed as input into the network for extracting features.Thinking from first principles, if it where possible to give the low resolution image as input the network could have learned more features within the same complexity as learning less features with the bicubically high-resolved image.

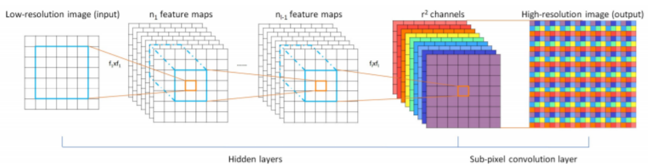

Figure 5.2: Pixel shuffler operation

The Pixel shuffler allows to reshape the input tensor of size $H \times W \times Cr^2$into that of size $rH \times rW \times C$ using the below equation

$$\mathbf{I}^{SR} = f^L(\mathbf{I}^{LR}) = PS(W_L * f^{L-1}(\mathbf{I}^{LR}) + b_L) \tag{5.5}$$

where f is the activation function used in each layer and W_L is the kernal applied on the last layer L, * represents the convolution operator and PS is a periodic suffling operator that does the rearrangement described as

$$PS(T)_{x,y,c} = T_{\lfloor \frac{x}{r} \rfloor, \lfloor \frac{y}{r} \rfloor, C*r*mod(y,r)+C*mod(x,r)+c} \tag{5.6}$$

where x,y,c are the co-ordinates of the upscaled image.

5.2 Loss Function

Pixel-wise loss functions such as MSE struggle to handle the uncertainty inherent in recovering lost high-frequency details such as texture. Minimizing the MSE encourages finding pixel-wise averages of plausible solutions which are typically overly-smooth and thus have poor perceptual quality. We illustrate the problem of minimizing MSE in Figure 4.3 where multiple potential solutions with high texture details are averaged to create a smooth reconstruction.

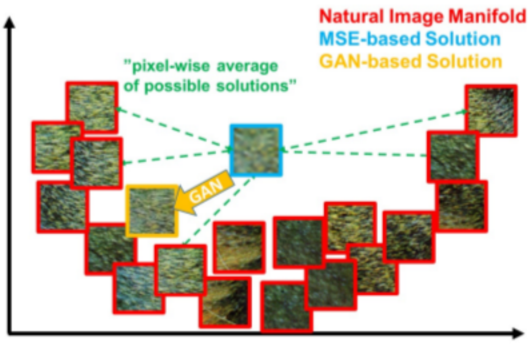

Figure 5.3: Illustration of patches from the natural image manifold (red) and super-resolved patches obtained with MSE (blue) and GAN (orange).

5.2.1 Perceptual Loss Function

The definition of the perceptual loss function l^{SR} is critical for the performance of the generator network. We use the perceptual loss as the weighted sum of a content loss l_X^{SR} and an adversarial loss component as:

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR} \tag{5.7}$$

$l_X^{SR} \rightarrow$ Content Loss

$10^{-3} l_{Gen}^{SR} \rightarrow$ Adversarial Loss

Content Loss

The pixel-wise MSE loss is calculated as:

$$l_{MSE}^{SR} = \frac{1}{r^2 W H} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2 \tag{5.8}$$

This is the most widely used optimization target for image SR on which many state-of-the-art approaches rely. However, while achieving particularly high PSNR,solutions of MSE optimization problems often lack high-frequency content which results in perceptually unsatisfying solutions with overly smooth textures.

Instead of relying on pixel-wise losses we use a loss function that is closer to perceptual similarity. We define the VGG loss based on the ReLU activation layers of the pre-trained 19 layer VGG network. With $\phi_{i,j}$ we indicate the feature map obtained by the j^{th} convolution (after activation) before the i^{th} maxpooling layer within the VGG19 network, which we consider given. We then define the VGG loss as the euclidean distance between the feature representations of a reconstructed image $G_{\theta_G}(I^{LR})$ and the reference image I^{HR}:

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \tag{5.9}$$

Here $W_{i,j}$ and $H_{i,j}$ describe the dimensions of the respective feature maps within the VGG network.

Adversarial Loss

In addition to the content losses described so far, we have to add the generative component of our GAN to the perceptual loss. This encourages our network to favour solutions that reside on the manifold of natural images, by trying to fool the discriminator network. The generative loss l^{SR}_{Gen} is defined based on the probabilities of the discriminator $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ over all training samples as:

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -logD_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{5.10}$$

Here, $D_{\theta_D}(G_{\theta_G}(I^{LR}))$ is the probability that the reconstructed image $G_{\theta_G}(I^{LR})$ is a natural HR image. For better gradient behaviour we minimize $-logD_{\theta_D}(G_{\theta_G}(I^{LR}))$ instead of $log[1 - D_{\theta_D}(G_{\theta_G}(I^{LR}))]$.

5.3 Architecture

5.3.1 Generator

The ulimate aim of a Generative Adversarial Network is to train the Generator network to estimate an HR image for every LR image given as input.

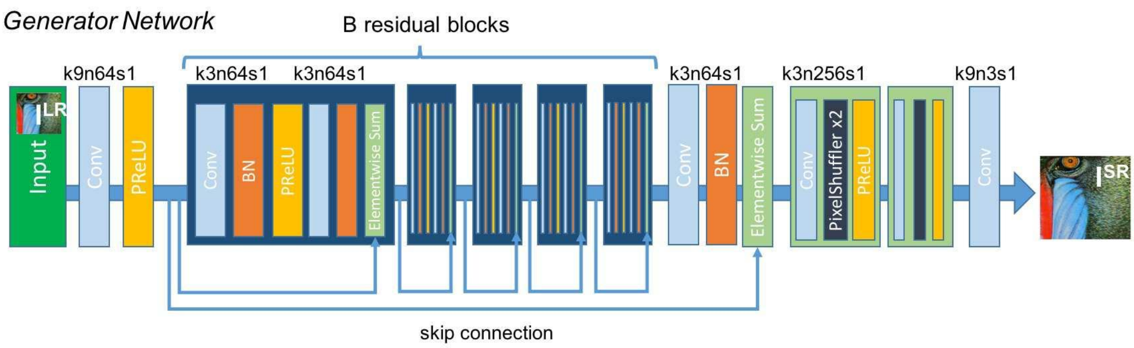

Figure 5.4: Generator Network Architecture [3]

The Genertor Network G, is a very deep Network as illustrated in the figure. The input layer of the Generator network is a part of a convolutional neural network with a 9×9 kernal size and has 64 feature maps. The convolution neural network uses Parameterised Rectified Linear Unit(PReLU) as activation function.

The approximating linear components of the input using nonlinear activation function is too expensive. Hence Residual neural networks, which can asymptotically approximate complicated functions, are used. The working of residual neural networks can be said as, they explicitly approximate $F(x)-x$ rather than $F(x)$, where x is its input. There are 16 blocks of Residual networks. Each having a convolution neural network, Batch Normalisation BN, activation function PReLU, which is followed by another convolutional nueral network and batch normalisation layers. The last layer of the residual block is an Elementwise sum layer which adds does the elementwise sum of output and input of the residual block. Both the convolution layers have 3×3 kernals convolting with stride 1 producing 64 feature maps. The Entire stack of 16 residul blocks are followed by a convolutional network(k3n64s1), Batch normalisation layer and an Elementwise Sum layer.

The resolution of the Image is increased by two blocks with a Convolution layer(k3n256s1), a Pixel shuffler($\times 2$) each. And is followed by the final layer being a convolutional network with 9×9 kernal and 3 feature maps.

5.3.2 Discriminator

The dicriminator network D, is trained to discriminate real High Resolution image from generated Super Resolved image. The architecture of the Discriminator network is shown in the image.

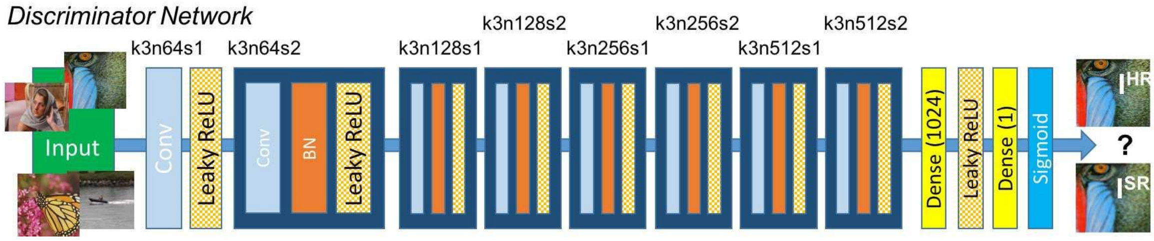

Figure 5.5: Discriminator Network Architecture [3]

The input is received by a convolution neural network with a 3×3 kernal and 64 feature maps convoluting with stride 1. It uses an activation function Leaky ReLU ($\alpha = 0.2$). Maxpooling is avoided throughout the network.

Discriminator consists of eight such convolutional layers with an increase number of $3{\times}3$ filter kernels, becoming twice from 64 to 512. The architecture closely resembles that of the VGG networks. The layers convolute with stride 2 and 1 for alternate layers.

The stack of convolution layers with 512 feature map output is followed by two dense layers to serialise the feature maps. Finally a sigmoid activation function is applied to obtain the probability of the image for being SR or HR.

Chapter 6

Results and analysis

6.1 PSNR SCORE

The Peak-signal-to-noise ratio (PSNR) score is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.PSNR score is used for Image super-resolution works as an accuracy measurment even though they can't model human perceptual similarity

PSNR score is easily defined via the mean squared error(MSE).Given the high resolution image I^{HR} and super resolved image I^{SR},MSE is defined as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I^{SR} - I^{HR}]^2 \tag{6.1}$$

The PSNR (in dB) is defined as:

$$PSNR = 20log_{10}(MAX_I) - 10log_{10}(MSE) \tag{6.2}$$

Here, MAX_i is the maximum possible pixel value of the image.In our case $MAX_I = 255$

The accuracy of the network where tested against two bechmarks 'Set5' and 'Set14' and the corresponding average psnr score for each of them where calculated

Dataset	**SRRESNET**	**SRGAN**
Set5	33.096	32.84
Set14	31.95	31.90

6.2 Result Visualization

The Images from standard benchmark where downsampled using bicubic interpolation and where upscaled using both trained SRResNet Model(consisting only the generator) and SRGAN Model.Some samples of the results are shown in figure: 5.1

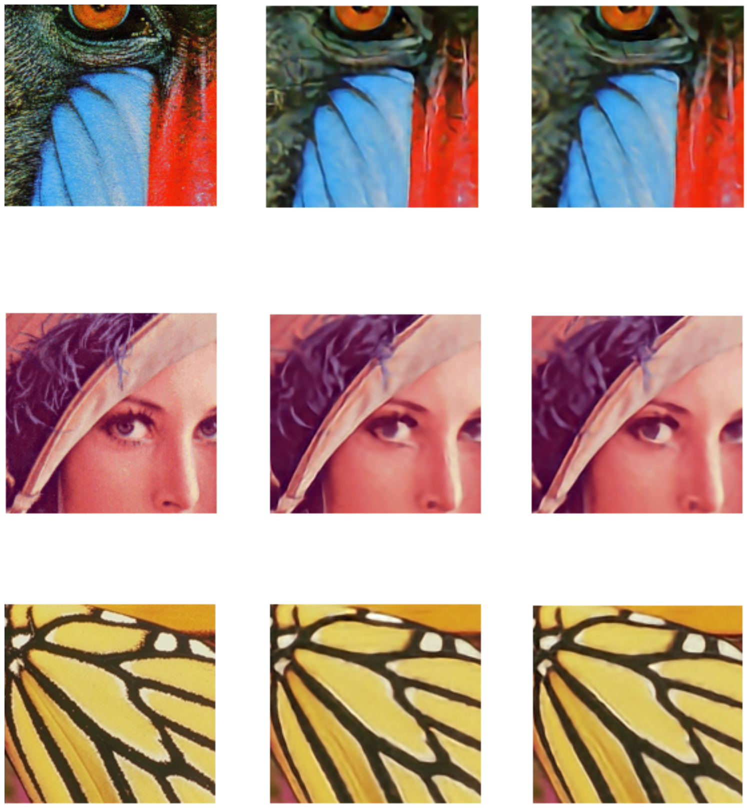

Figure 6.1: Comparision of sample Ground Truth Image, SRGAN Image and SRRESNET Image

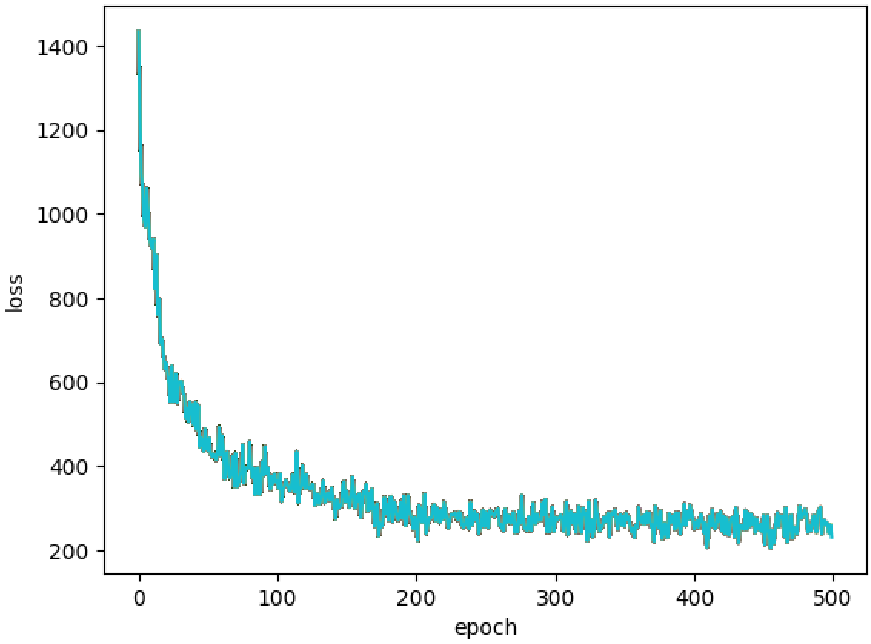

Figure 6.2: Plot of loss against number of Epochs for SRResNet

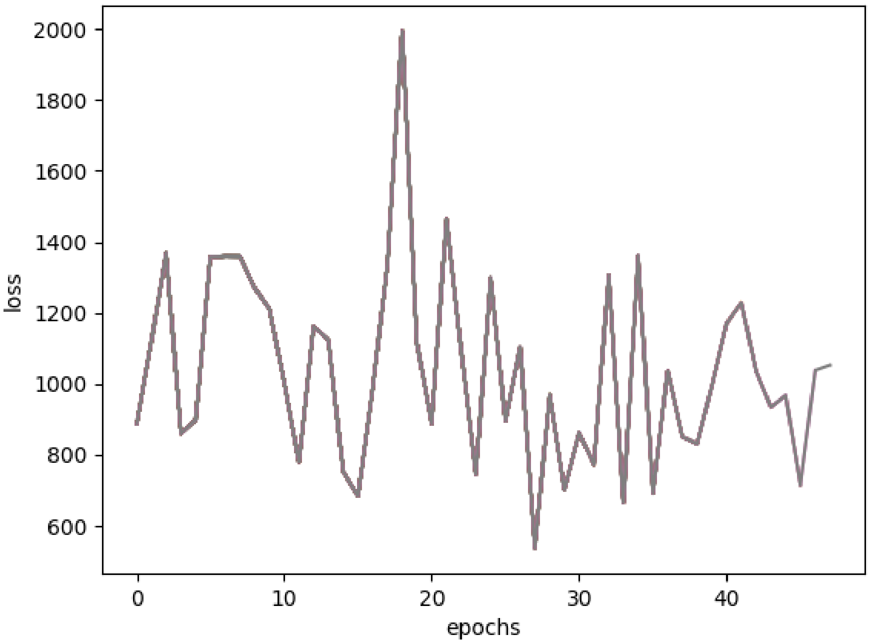

Figure 6.3: Plot of loss against number of Epochs for SRGAN

Chapter 7

Conclusion

It is clear from the results obtained that the super resolved image obtained from Generative Adversarial Network is close to reality. It has a higher PSNR score than the older methods like Bicubic Interpolation, but the score is lower compared to that obtained with SRResNet. But the image obtained from GAN is perceptually closer to reality and thus would entail a higher Mean Opinion Score (MOS). The output obtained is constrained by the effectiveness of the training procedure. Training for the generator along was done with NVIDIA Jetson TX1. But training the Generative Adversarial Network (GAN) as a whole required a machine of higher computing power. The GAN was trained using an NVIDIA Quadro M5000 GPU. Another important parameter in the training procedure is the dataset. The richer the dataset, the network learns with different types of images and thus will be able to super resolve any kind of input image.

Bibliography

[1] I.Goodfellow,J.Pouget-Abadie,M.Mirza,B.Xu,D.Warde-Farley,S.Ozair,A.Courville,and Y.Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems (NIPS), pages 26722680, 2014. 3, 4, 6

[2] Wenzhe Shi, Jose Caballero, Ferenc Huszr, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, Zehan Wang.Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

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

[4] Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

[5] C.Dong,C.C.Loy,K.He,and X.Tang. Learning a deep convolutional network for image super-resolution. In European Conference on Computer Vision (ECCV), pages 184199. Springer, 2014. 3, 6, 8

[6] C.Dong,C.C.Loy,K.He,and X.Tang. Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2):295307, 2016.