Image Super Resolution Using GAN

Under the Supervision of
Dr. Shirshu Varma

Indian Institute of Information Technology, Allahabad

Ankit Kumar (IRM2016002)
Avinash Yadav (ITM2016004)
Neeraj Mishra (IIM2016003)

Introduction

- This project presents the idea to estimate high resolution images from low resolution images.
  - *This estimation is termed as super resolution*
- Advancement in deep learning in recent years is enabling us to solve the problem.
- Generative Adversarial Network (2014) has revolutionised super resolution.
- This project uses Generative Adversarial Network to achieve a good estimation of high resolution images from low resolution images.

Why Super Resolution?

- Lossy image compression is widely used to reduce the size of images to facilitate faster transfers over network and low bandwidth usage.
- But once the image is at target, a good estimation to a high resolution image may be required.
- However, good an upscaling algorithm is, there will always be some amount of high frequency data lost from a downscale upscale function performed on an image.
- Even the best algorithms cannot effectively reconstruct data that does not exist.

Literature Review

- Early prediction models include interpolation-based methods such as bilinear, bicubic, and Lanczos resampling [5].
  - *Though these methods are fast, they oversimplify the problem and produce overly smooth outputs [4].*

- In 2014, Yang et al. [6] have researched all these techniques and arrived at the conclusion that example-based methods yield the best results. These are machine learning approaches which rely on training data to find a complex mapping from low resolution images to high resolution images.

Literature Review

- Dong et al. [7] have shown that convolutional neural networks (CNN) perform even better. These however rely on pixel wise loss functions such as MSE which make the images overly smooth.

- In [9] and [4], the authors have overcome the above problem by employing Generative Adversarial Networks [1] to generate images.

- Ledig et al. [4] propose SRGAN model that uses perceptual loss and adversarial loss to favor outputs residing on the manifold of natural images. The results were very photo-realistic and hence our project aims at performing super resolution using GANs.

What is GAN?

- ❖ A Generative Adversarial Network mainly consists of two components:
  - ➢ *The Generator(G)*
  - ➢ *The Discriminator(D)*

- ❖ Generator
  - ➢ *Given a dataset of similar images produces new similar looking images.*
  - ➢ *Takes in random noise as input*
  - ➢ *Produces corresponding output with help of a deconvolutional neural network.*
  - ➢ *Has initial weights and biases.*
  - ➢ *Parameters adjusted based on validity of output.*
  - ➢ *This requires us to manually provide it the feedback whether the generated image is a plausible output or not.*

- ❖ This is where the discriminator comes in and automates the process.

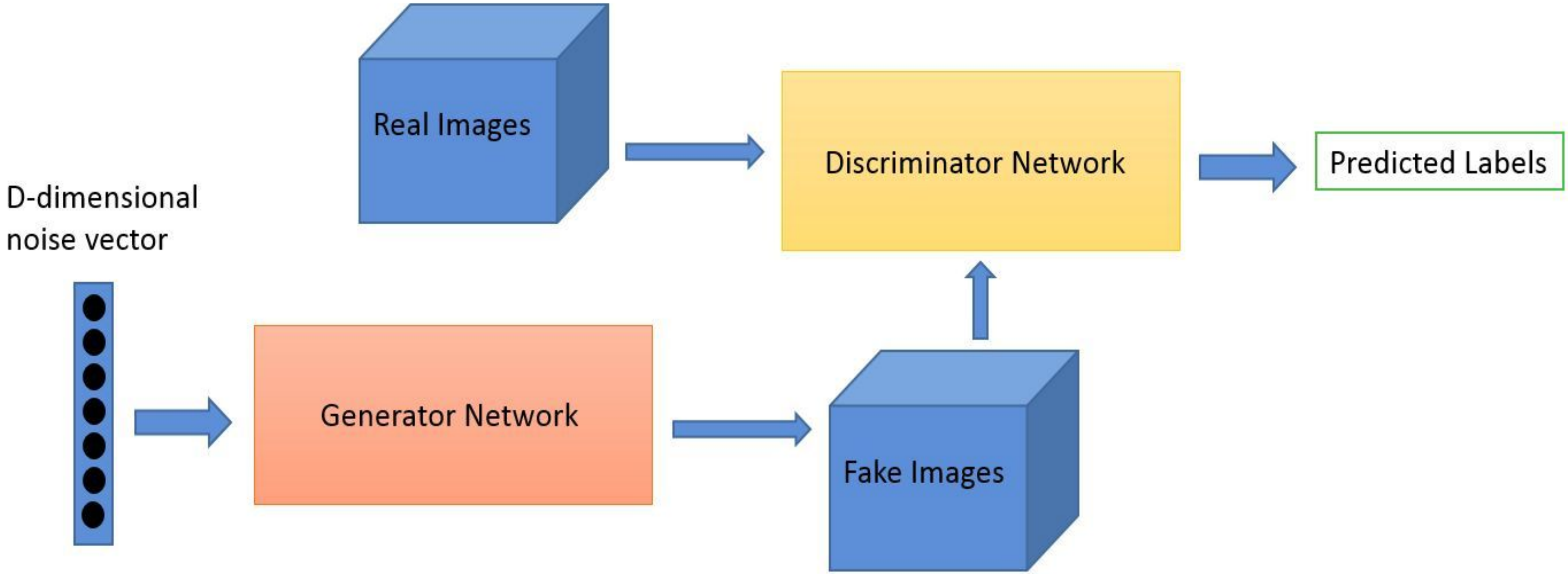
What is GAN?

- Discriminator
  - *Takes in inputs from both the original dataset as well as the generated images*
  - *Tries to distinguish the real images from the generated images.*

- We initially train the discriminator (D) upto some accuracy and then alternate between generated images and real
  images. The objective of the generator (G) is to maximize the error rate of D.

- The key idea is to backpropagate gradients from the results of D's classification to G, so that G gets better at producing fabricated images that can fool D and D gets better at flagging generated images

What is GAN?



Real Images
Discriminator Network
Predicted Labels
D-dimensional noise vector
Generator Network
Fake Images

GAN For Single Image Super Resolution

- The discriminator converts the 128x128x3 image(3 implies the RGB channels of the image) into a 16x16x1 matrix which is used to determine if the image is real or generated.

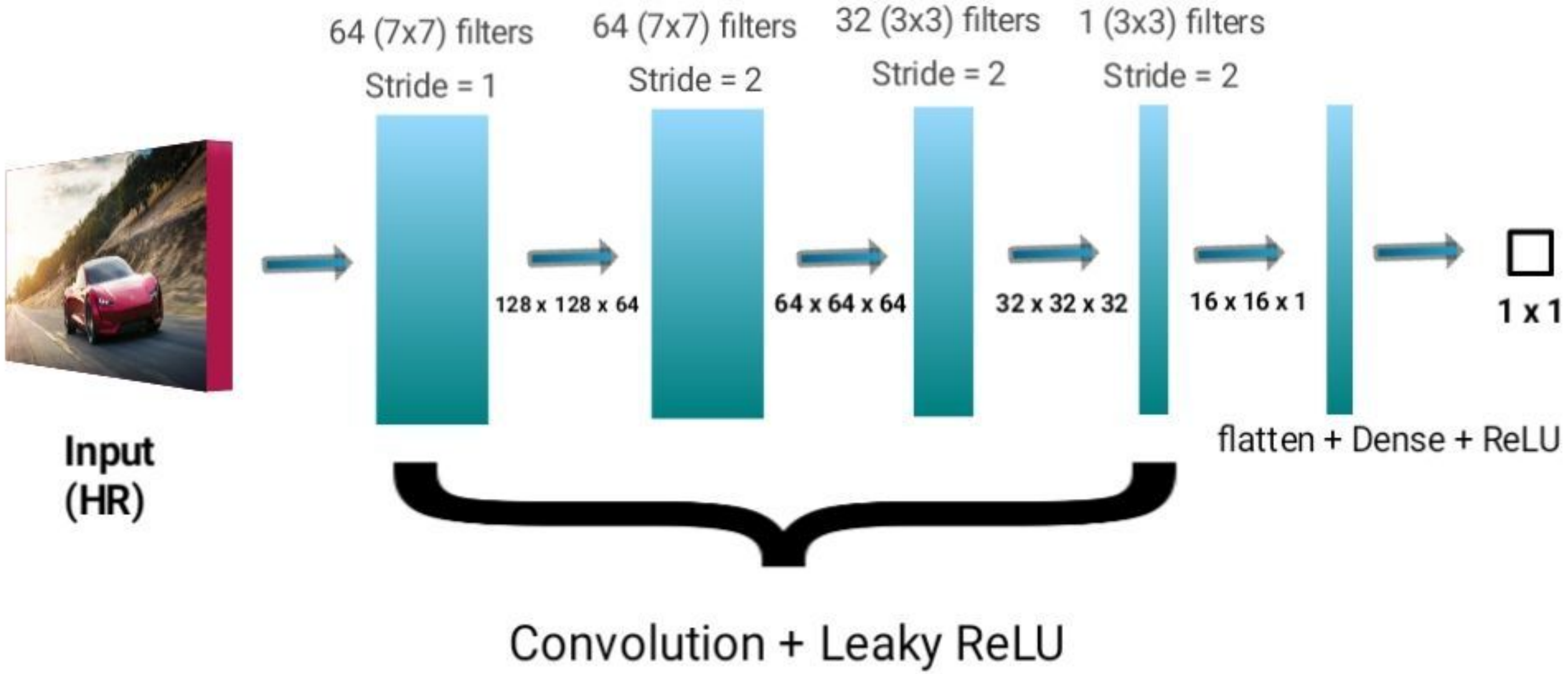
- Leaky ReLU is used as activation function for all hidden layers except last.

- Last layer is a fully connected dense layer.

- The leaky ReLU activation function is defined as max {x, threshold * x } .

- For our model we set the threshold as 0.01 it is simply a design choice.

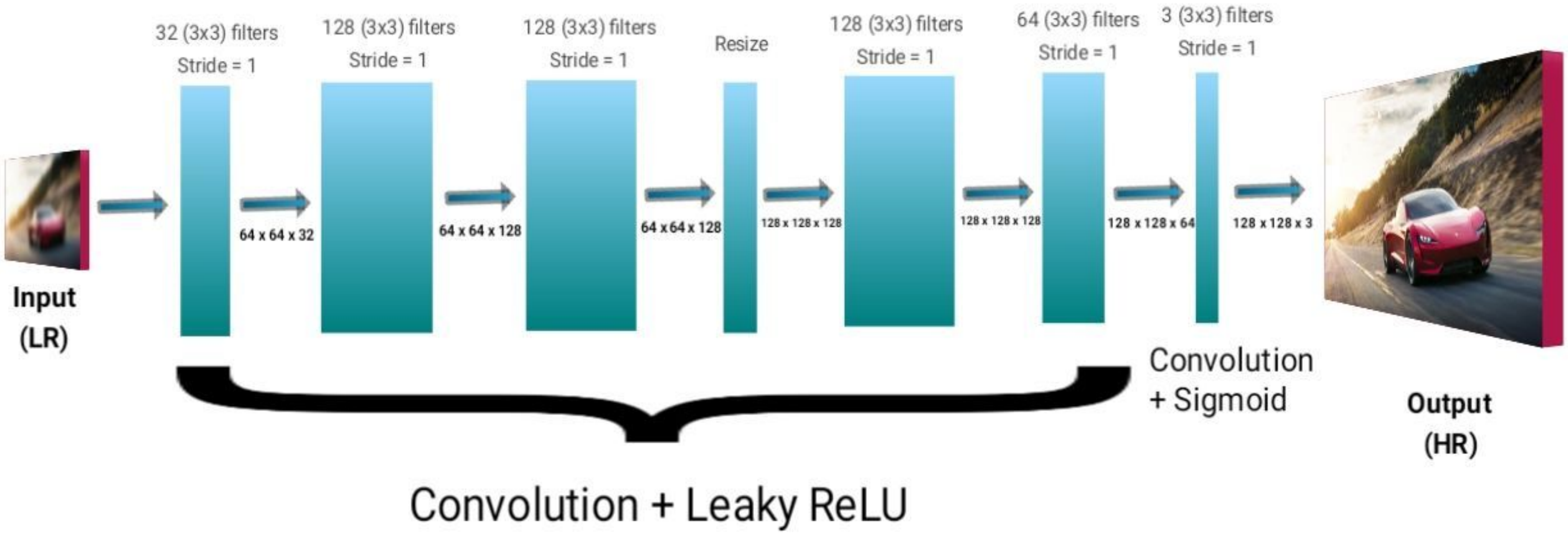
Discriminator Architecture



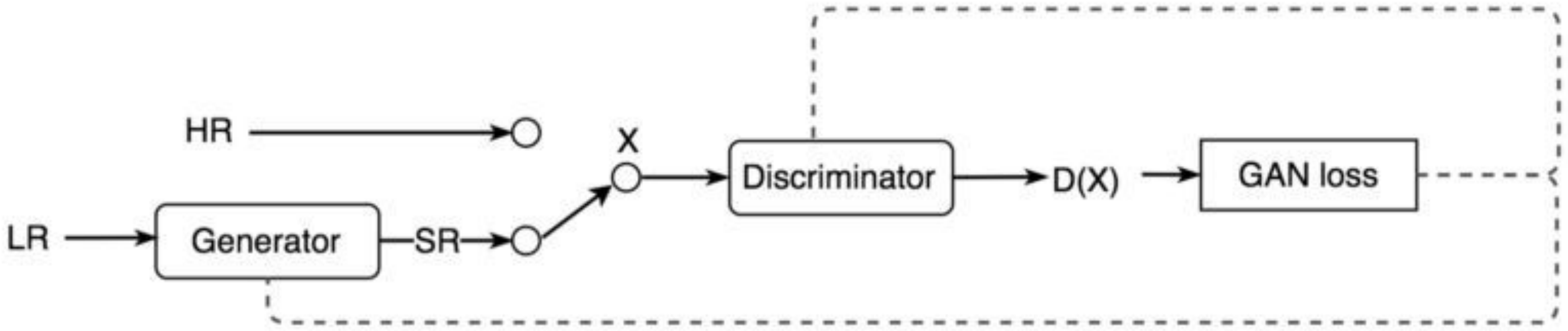
GAN For Single Image Super Resolution

- ❖ The generator model on the other hand converts the low resolution 64x64x3 images to 128x128x3.
- ❖ 3 convolution layers are applied. These apply important details of the image.
- ❖ The resulting image is then resized.
- ❖ 3 more convolution layers are applied to add the sharper details like edges and sharp changes in the colors to the resized image.
- ❖ Leaky ReLU is used as activation function for all hidden layers except last which uses sigmoid

Generator Architecture



Complete Architecture



LR = Low Resolution Image
SR = Super Resolution Output of Generator
HR = Original High Resolution Image

Table 1: Generator

Layer	Filters	Kernel Size	Activation Function
Convolutional Layer 1	32	3x3	Leaky Rectified Linear Unit
Convolutional Layer 2	128	3x3	Leaky Rectified Linear Unit
Convolutional Layer 3	128	3x3	Leaky Rectified Linear Unit
Resize Image			
Convolutional Layer 4	128	3x3	Leaky Rectified Linear Unit
Convolutional Layer 5	64	3x3	Leaky Rectified Linear Unit
Convolutional Layer 6	3	3x3	Sigmoid Activation Layer

Table 2: Discriminator

Layer	Filters	Kernel Size	Activation Function
Convolutional Layer 1	64	7x7	Leaky Rectified Linear Unit
Convolutional Layer 2	64	7x7	Leaky Rectified Linear Unit
Convolutional Layer 3	32	3x3	Leaky Rectified Linear Unit
Convolutional Layer 4	1	3x3	Leaky Rectified Linear Unit
Densely Connected Layer	-	-	Rectified Linear Unit

Why Does This Work?

- In CNNs, kernels (filters) are the main reason for working of the architecture.
- For the classification tasks,the filters are trained to detect the key features of the input datasets.
- But in generative task the convolution with these filters adds more features
- We can not predict what exactly each filter is going to learn.
- The main difference is that the Generative Adversarial Networks use a better loss function than simple CNNs.
- It includes apart from content loss, adversarial loss that comes from the adversary of the generator which is discriminator.
- This simply implies that the generator does not directly learn from the images but also from what the discriminator thinks about the generated output.
- This makes the generator generate images which are not only closes to the original images in terms of pixel wise difference but also tries to generate images which are acceptable by the discriminator network.

Loss Functions

- Content Loss
  - *Absolute value of the pixel wise differences between the generated image and the original high resolution image.*
  - $Content\ loss = mean(abs(g - highres))$
  - *g=Generated Image , highres=High Resolution Image*
- Adversarial Loss
  - $Adversarial\ Loss = 0.1 \times mean(z * -log(sigmoid(x)) + (1 - z) * -log(1 - sigmoid(x)))$
  - $x = d_{fake}$ and $z = d_{label}$
  - d_{fake}=*Output of discriminator*
  - d_{label}=*Considers if the high resolution image is valid or invalid(0 or 1)*
  - *0.1 is a value of choice.*
  - $z * -log(sigmoid(x)) + (1 - z) * -log(1 - sigmoid(x))$= *Cross Entropy with Sigmoid*

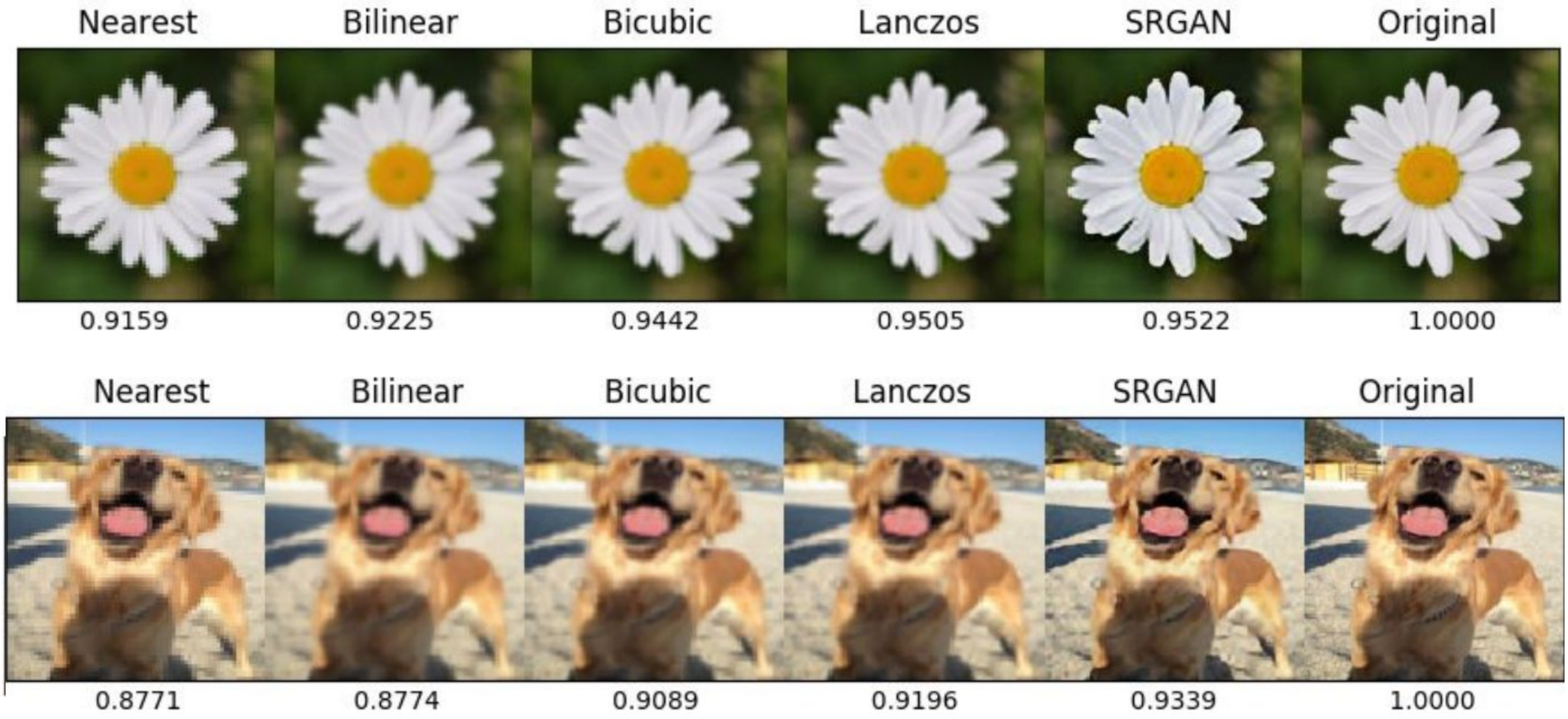
Loss Functions

- Generator Total Loss
  - *Generator loss = Adversarial loss + Content loss*
  - $0.1 \times mean(z * -\log(sigmoid(x)) + (1-z) * -\log(1-sigmoid(x))) + mean(abs(g - highres))$
  - $0.1 \times mean(z * \log(1+e^{-x}) + (1-z) * (-\log(e^{-x}) + \log(1+e^{-x})) + mean(abs(g - highres))$
  - $0.1 \times mean(z * \log(1+e^{-x}) + (1-z) * (x + \log(1+e^{-x})) + mean(abs(g - highres))$
  - $0.1 \times mean((1-z) * x + \log(1+e^{-x})) + mean(abs(g - highres))$

- Discriminator Total Loss
  - *Total loss*$=d_{LossFake}+d_{LossReal}$
  - $d_{LossReal} = mean(d_{LabelReal} * -\log(sigmoid(d_{real}))+(1-d_{LabelReal})*-\log(1-sigmoid(d_{real})))$
  - $d_{LossFake} = mean(d_{LabelFake} * -\log(sigmoid(d_{fake}))+(1-d_{LabelFake})*-\log(1-sigmoid(d_{fake})))$
  - d_{real}*=output of the discriminator when the input is a real(High resolution image)*
  - d_{fake}*=output of the discriminator when the input was a generated image*
  - $d_{LabelReal} = 1,\ d_{LabelFake} = 0$

Results

- The results obtained are photo-realistic and are better than interpolation techniques such as nearest neighbour,bilinear, bicubic and lanczos. SSIM is used for similarity metrics



Conclusions

- In this project, we have proposed a Super Resolution GAN which produces photo-realistic images for 2xscaling and our model performs better than current approaches

- We confirmed the superior perceptual performance of GANs for the task of Super Resolution using similarity Metrics like SSIM.

- GANs are very powerful and have a lot of potential. We can improve image quality of higher scaling factors by tweaking the model and training on more powerful GPUs

Thank you