SRGANs

Table of Contents

* Business Problem
* Previous work on the problem
* Deep Learning Approach Solution
* Understanding Dataset
* Architecture of Model
* Results
* References

**Business Problem**

Image Resolution is highly dependent on the hardware limitations and the equipment used. We cant compare a satellite image with an image captured by a street CCTV Camera. Increasing image resolution by pushing the hardware limitation is a challenge. Image enhancement techniques are used in various domains like satellite imaging, surveillance systems, super zoom telescopes and cameras.

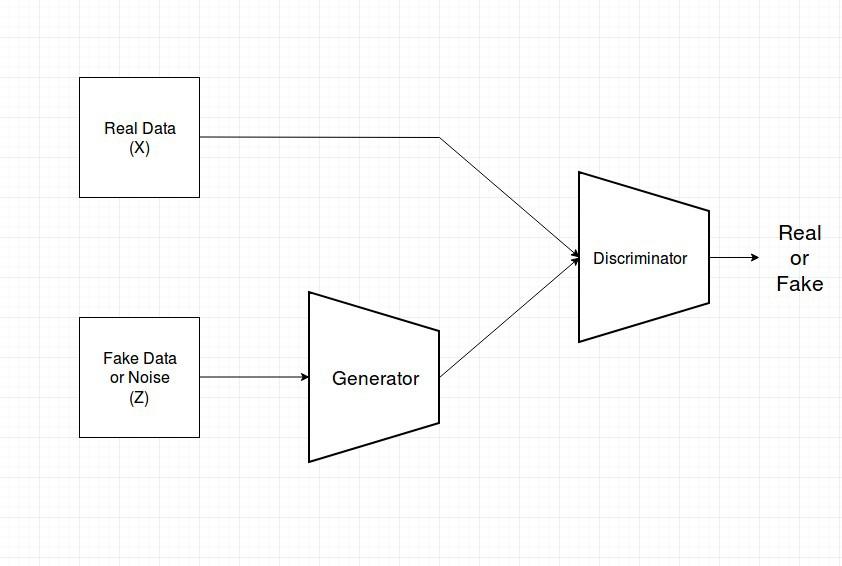
**Previous work on the problem**

There are several image enhancement techniques which are used for a long time. The easiest technique which is used is interpolations. Interpolations mean filling the pixels by interpolating the pixel values, Interpolations could be done by simple averaging or using more complex methods like Linear or cubic interpolations.

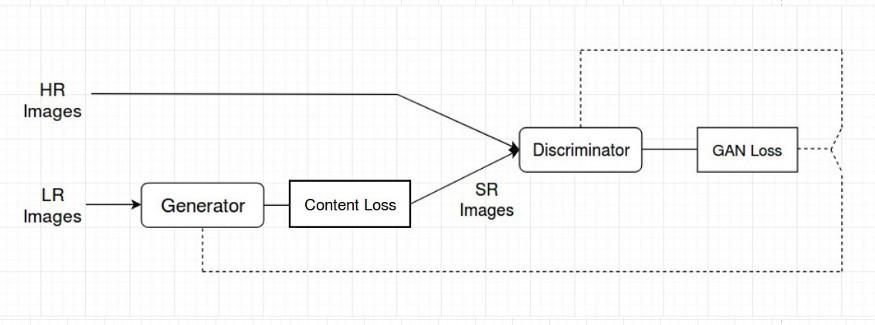
Another technique that is used especially by Satellite imaging is Wavelet transformation. Here the image is converted into a frequency domain using Discrete wavelet transformation where the image is broken into small wavelets and is enhanced

**Deep Learning Approach Solution**

With the advancement of deep learning algorithms especially Generative Adversarial Network(GAN) in 2014 by Ian Goodfellow, a wide domain of possibilities were opened in Image processing. With the help of GANs we can enhance the image resolution by using Super-Resolution Generative Adversarial Network (SRGANs).



GANs basically have two components, one is Generators which will generate data from Fake input. The aim of the generator is to generated data as close as possible to the real data. The other component Discriminator, the discriminator will try to distinguish between the generated output by the generator and the real output.



For SRGANs we are converting a low-resolution input image to a super-resolution image. For this purpose, we require to train the network on images. We want the generator to produce a super-resolution image from a low-resolution image. We would give input to the generator out a low-resolution image and the output would be a super-resolution image. Now the discriminator acts as a classifier that would classify between the generated image and the high-resolution image.

**Architecture of Model**

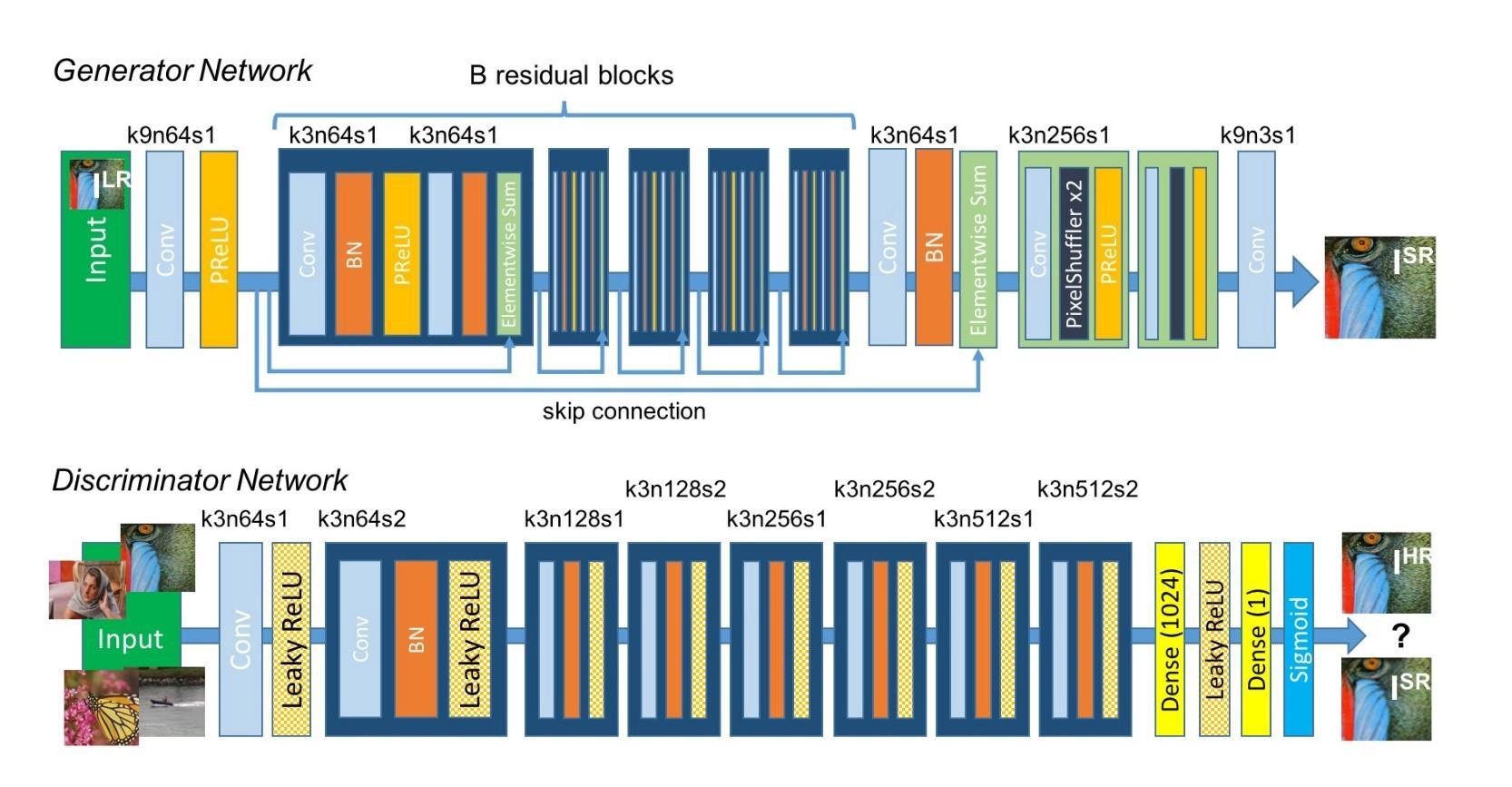
As described earlier, the SRGAN’s architecture consists of two parts. Below is the image of

**Understanding Dataset**

We have used COCO 2017 Dataset for training the model. Since we just need images, with COCO dataset we get a variety of images. Now we need a low-resolution image and a high-resolution image for training. So we resized all the images using opencv in python to 200,200 pixels, this serves as a high-resolution image. Now again for the same images we resize them to 50,50 pixels, this serves as low-resolution images and we are upscaling the images bu 4 times.

In COCO 2017 dataset we have 18 gb of training data and 6 gb of validation data. <https://cocodataset.org/#download>

architecture



Few things to note about Architecture here is:

* The residual learning framework eases the training of these networks, and enables them to be substantially deeper, leading to improved performance. We are using 16 residual blocks in the Generator block of the neural network with upsampling network twice. The generator also implements skip connections similar to ResNet Architecture.
* k3n64s1 this means kernel 3, channels 64 and strides 1.
* To compare two images and compare them we used pixel to pixel comparison using the mean squared error function. However, we can also use the Perpetual loss function using VGG19 architecture. IN this loss function we use don’t use pixel to pixel comparison but the feature comparison after passing the image through the VGG19 network.
* For the discriminator, we are using the standard cross-entropy loss function as we classify the images as real or fake.
* The activation function which is used is PRelu(Parameterized Relu).
* The optimizer for loss minimization is Adam for both Generator and Discriminator as they give stable results as compared to other optimizers

The overall loss function here consists of two parts, One of them is Content loss given by the generator using VGG19 Architecture. The other one consists of the loss by Discriminator given by Cross-entropy function. The desired results are achieved when the discriminator gives similar results when high-resolution image and generated super-resolution image and the generator losses is low. This means that discriminator is not able to distinguish between the real image and generated image because the generated image is close enough to the real image.

**Results**



Low resolution input image



Super-resolution image generated by Generator



High resolution image

**References**

Adversarial Network :[*https://arxiv.org/pdf/1609.04802.pdf*](https://arxiv.org/pdf/1609.04802.pdf)

Perceptual loss: <https://arxiv.org/pdf/1603.08155.pdf>

Deep Residual Learning for Image Recognition : <https://arxiv.org/pdf/1512.03385.pdf>

GANs Original Paper: <https://arxiv.org/abs/1406.2661>

SRGANS Paper: <https://arxiv.org/abs/1609.04802>