Image Super Resolution

Vivaan Jain
24/B03/061

Approach and Thought Process

After coming across a variety of models,ESRGAN was the one that stood out for me because of its architecture,the results it produced,and the new perspective of perceptual quality it introduced in the field of super resolution. My proposal contains 2 solutions one with better perceptual quality and the other one with better structural similarity in comparison to HR images.

So what I had observed after looking at the dataset was , all the LR images were not in order and aligned properly with their corresponding HR images, there was not a fixed size within both the LR and HR images.Then,the next thing that caught my eye was the size of the dataset given was 80 pairs of LR and HR images , which was too small .So,using only transfer learning and training the model on the whole dataset might not yield fruitful results.

Firstly,we import the github repo which has the original model and install its dependencies and then preprocess the data through these following steps:

1) Sorting all the LR images such that they are matched with their corresponding HR images.The data given was random so it is important that the images are properly aligned.
2) Normalizing and resizing LR and HR images to 64x64 and 256x256.Normalizing the pixel intensities is important because it helps in stabilizing the training because if the there is an imbalance in the weights it might hamper the learning of some neurons due to unstable gradients.Resizing is done as all the images are not in the same orientation some are either portrait or in landscape so in order to make it easy to visualize the setup it is crucial to resize the images to a preferred size.

Secondly, fine tuning the original ESRGAN's weights
This is done by just making the weights of RRDB learnable and freezing all the other weights present in other components of the architecture.
Fine tuning is crucial for such a dataset which comprises such small training examples because training a complete model with this type of dataset might not lead to good

results and may even distort the original weights. Just making a few weights learnable ensures that the ESRGAN model is able to understand what is going on in the dataset rather than blindly evaluating the test dataset.

Lastly,we train our model with 5 epochs with batch size 1 that is through stochastic gradient descent using Adam optimizer with learning rate=10^-4,beta1=0.9,beta2=0.999 and filtering out those parameters which are freezed so that no gradient is calculated for them with our training dataset by optimizing MSE Loss.
I utilized the T4-GPU for my runtime to traverse through the training as this task could not have been done on the CPU alone due to its complexity.
Apart from this training, I used the same ESRGAN model but with a different setting. The model now ran for 10 epochs and optimized the perceptual loss.

Blockers
One major obstacle that I found in my path during training my model was the computational resources that the model took.Since the model is actually trained for large datasets so it not easy to train such a large model that is why I had to restrict my number of epochs to 5 and had to use a smaller batch size in order to execute my training.With better computational resources this task could have been executed in an efficient manner and hopefully better results by undertaking more number of epochs and a larger batch size for optimization.
Moreover,due to continuous usage of the GPU which is allotted for a certain duration for free it became difficult to run all the commands again and again in Colab(the IDE that I used) making it difficult to progress further into the task such as training the model multiple times by tuning hyperparameters or evaluating other models on the same dataset through finetuning.

Learnings
Through multiple readings through different literatures I came across a wide variety of models that have been deployed, tested to produce remarkable results in the field of super resolution.This involves Bicubic interpolation technique which upscales low resolution images to higher dimension same as that of the corresponding high resolution images and then tries to map low resolution dictionary to high resolution dictionary.This technique was then downplayed through the use of convolutional neural networks to enhance image quality. Many models incorporated deep CNN’s to gather and extract information from the low resolution images through multiple layers and aggregated feature maps to produce high resolution images.Models such as SRCNN worked on the similar phenomenon and were able to generate good results.Though the performance of these models were good but still the textures and the perceptual quality in the high resolution images lagged behind. Employance of GAN’s to this space

completely revolutionized this field and produced remarkable results making the generated images almost similar to the ground truth HR images. In the proposed framework, 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. SRGAN also introduced a new optimizing metric which was Perceptual loss as pixel-wise loss functions such as MSE struggle to handle the uncertainty inherent in recovering lost high-frequency details such as texture: minimizing MSE encourages finding pixel-wise averages of plausible solutions which are typically overly-smooth and thus have poor perceptual quality.Perceptual loss along with the GAN framework makes the setup suitable for upscaling tasks with remarkable results.
Apart from the employance of CNN's as the basic building block for SR related tasks, uprise of vision transformers such as Swin have completely changed our perspective on how one can approach SR tasks. It incorporates the usage of transformers to account for long range dependencies by taking the LR images in form of numerous patches,and then enriching these patches with contextual information from other patches using attention blocks.The main issue with these transformer based architectures is that the transformer network is not able to account for inductive bias and translation equivariance which is possessed by CNN's.

Model Architecture

ESRGAN does is improves and builds upon the work of SRGAN and provides even better results in terms of finer edge and perceptual quality of images.
It incorporates the usage of RRDB blocks which propose an even more complex and deeper structure than SRGAN's architecture,boosting performance by incorporating multiple deeper connections.

Removal of Batch Normalization from each blocks has proven to increase performance and reduce computational complexity
Besides the improved structure of the generator, we also enhance the discriminator based on the Relativistic GAN,different from the standard discriminator D in SRGAN, which estimates the probability that one input image x is real and natural, a relativistic discriminator tries to predict the probability that a real image xr is relatively more realistic than a fake one xf.
Improvement in perceptual loss along with architecture makes it a powerful solution for the SR problem.

In the original architecture about 23 RRDB layers were present for the generator network along with 2 convolution layers for upsampling.

The discriminator network utilizes VGG like architecture and changes its adversarial loss to incorporate realism by behaving as relativistic discriminator

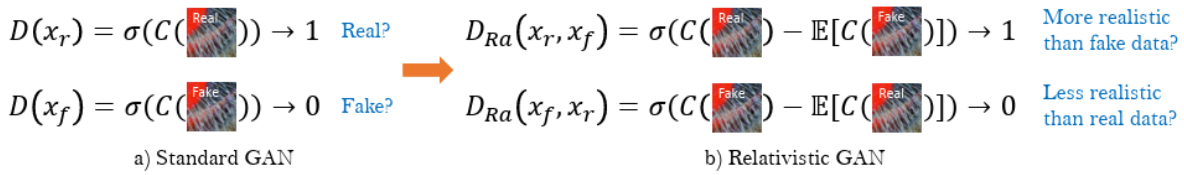


Fig. 5: Difference between standard discriminator and relativistic discriminator.

The original model is optimized on perceptual loss which involves 2 components that factor in the perceptions of the human visual system
1) Content Loss
2)Adversarial Loss
Content loss is similar to VGG loss and is calculated before activation on the feature as it has been observed that there is lots of loss of information after activation function is applied.

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

Here r is the upscaling factor,W,H are the dimensions of the feature map and HR and LR resemble feature maps of low resolution image and high resolution image respectively
Adversarial loss is the binary cross entropy loss given by the discriminator making the generator know about how far it was from generating an actual image

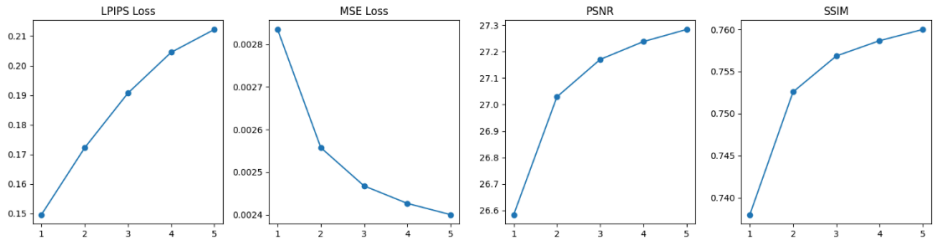
$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

Here G_{θ_G} refers to the generated output of the LR image.This loss is summed over all the examples in the current mini batch

In addition to the improved architecture, to facilitate training a very deep network residual scaling i.e., scaling down the residuals by multiplying a constant between 0 and 1 before adding them to the main path to prevent instability is also incorporated.

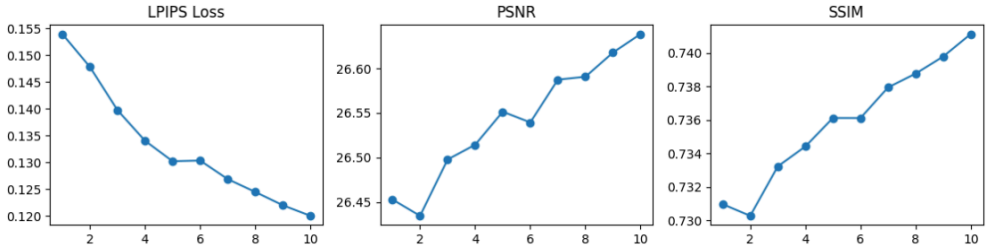
Training Performance

When optimized with MSE loss:



Progressive decrease in MSE Loss and substantial increment in the values of PSNR and SSIM is observed in this case. While the value of LPIPS skyrockets with increasing number of epochs.

When optimized with Perceptual Loss:



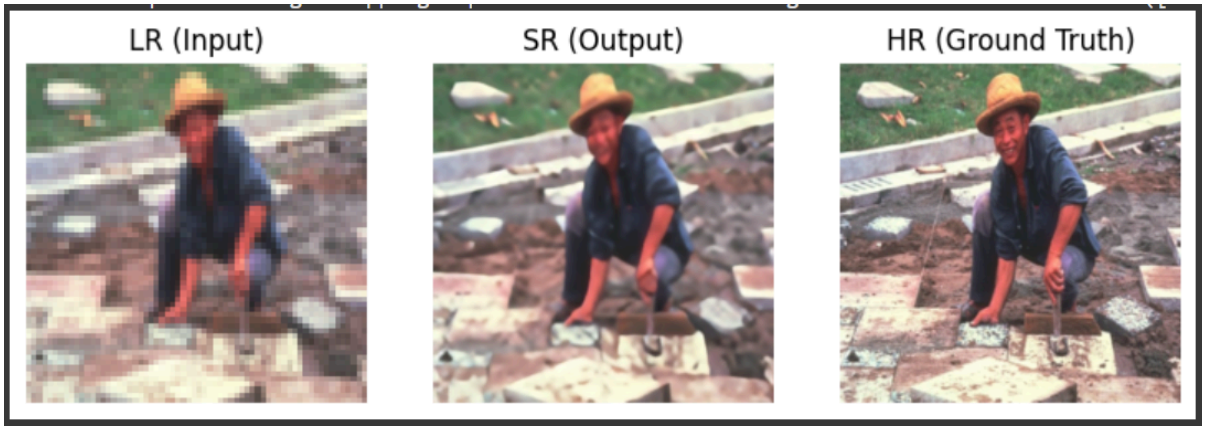
PSNR and SSIM gradually increase their way up to 27.39 and 0.7073 respectively while LPIPS decreases through number of epochs.

Evaluation and Comparison

After training on our training dataset and evaluating it on the test dataset by setting the evaluation metric as PSNR , SSIM and LPIPS I obtained

```
Test PSNR: 28.2055
Test SSIM: 0.7394
Test LPIPS:0.2205
```

The results when visualized:



The results obtained were a slight different with less pixel accuracy but high perceptual quality when the model was optimized by using perceptual loss.

```
Test PSNR: 27.3911
Test SSIM: 0.7073
Test LPIPS: 0.1257
```

This model proposes better perceptual quality and trades it off with the PSNR and SSIM which is quite frequently seen in many models that PSNR and SSIM may not be completely sufficient to predict whether the generated image has reached the mark of the HR image.

This model was pitted against Bicubic interpolation technique,SRCNN,OriginalESRGAN model and provided the following results:

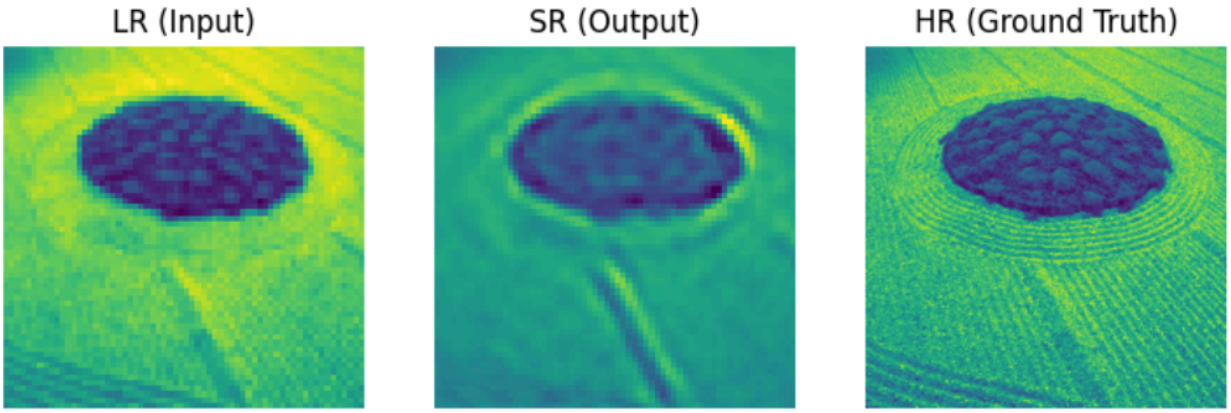
OriginalESRGAN:

```
Test PSNR: 27.3863
Test SSIM: 0.7033
```

```
Test LPIPS: 0.1972
```

SRCNN

```
TestPSNR: 27.5452
Test SSIM: 0.7262
```



Bicubic:

```
TestPSNR : 25.7640
Test SSIM : 0.6608
```

Our model has performed very well as compared to the standard models used in this field and can produce even better results.
Our model has a better pixel wise accuracy and structural similarity in the generated images if we compare it to other models. But,It lags behind the original ESRGAN model in terms of perceptual quality i.e outputs look more natural or realistic to a human observer.that means some refinements are definitely needed.These refinements are listed in the future prospects

Future prospects
In this part we will discuss how the current model's working can be enhanced further to provide even better results on the given dataset and other futuristic approaches (not tested till date)

By tuning the hyperparameters such as batch size, learning rate, and number of freezable layers can tend to provide good results.

We can mimic the training like the one employed in the Original ESRGAN by optimizing the perceptual loss that might help in increasing details in the generated images through data augmentation.

As we know , transformers have proved to provide remarkable results on super resolution related tasks because of their ability to capture long term dependencies so we can employ a transformer network such as Swin and employ it on this dataset and fine tune it .
In the transformer network, if employed,as an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN . In this hybrid model, the patch embedding projection is applied to patches extracted from a CNN feature map.

Rather than incorporating positional encoding to gather information about position of the patches we can incorporate ROPE(Rotary positional embedding) into our transformer

network that would help us to provide an even larger context window to enrich our embeddings.This would be a very complex task as it will change the complete working of a vision transformer, this approach needs to researched upon and implemented but may provide to be fruitful.

Appendix

ESRGAN: Enhanced Super Resolution Generative adversarial network
GAN:Generative Adversarial network
HR: High Resolution images
IDE:Integrated development network
LPIPS:Learned Perceptual Image Patch Similarity
LR: Low resolution images
PSNR: Peak Signal- to- noise ratio
SRCNN: Super resolution convolutional neural network
SSIM:Structural Similarity Index Measure.
SWIN:Shifted window
MSE:Mean Squared Error
VGG:Visual Geometry group

References

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network:*Christian Ledig, Lucas Theis, Ferenc Husz´ar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi Twitter*

Generative Adversarial Nets: *Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio*

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks: *Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, Xiaoou Tang*