Image Super Resolution : Comparison of state-of-the-art methods

Swapnoneel Kayal
Department of ME
IIT Bombay
Mumbai, India
200100154@iitb.ac.in

Nikhil Tiwari
Department of ME
IIT Bombay
Mumbai, India
200010050@iitb.ac.in

K S Varun
Department of ME
IIT Bombay
Mumbai, India
200100088@iitb.ac.in

Abstract*—This project report presents a comparative analysis of six state-of-the-art models for image super-resolution: SRCNN, FSRCNN, EDSR, DIP, SRGAN and ESRGAN. These models aim to enhance the resolution and details of low-resolution images using different approaches. We evaluate their performance on various image datasets, comparing the obtained results based on objective metrics and visual quality. SRGAN and ESRGAN utilize a generative adversarial network (GAN) framework and perceptual loss function to capture high-frequency details and improve visual quality whereas SRCNN and FSRCNN utilize the convolutional neural network (CNN). EDSR focuses on optimizing the network architecture using deep residual networks, while DIP formulates the super-resolution problem as an optimization task, leveraging the inherent structure of low-resolution images. The results show that ESRGAN outperforms the other methods in terms of both reconstruction accuracy and perceptual quality, but it also has the highest computational complexity. DIP performs well in terms of reconstruction accuracy, but it produces less visually appealing results. SRCNN and FSRCNN are efficient in terms of computational complexity, but they are outperformed by the other methods in terms of reconstruction accuracy and perceptual quality. SRGAN and EDSR achieve good balance between reconstruction accuracy and perceptual quality, but they are computationally more complex compared to SRCNN and FSRCNN. Overall, the comparison provides insights into the strengths and weaknesses of each method and can guide the selection of appropriate super-resolution techniques for different applications. This project provides a comprehensive comparison of these models, highlighting their strengths and weaknesses. Researchers and practitioners can use these findings to make informed decisions when selecting an appropriate model for image enhancement and reconstruction tasks.

***Index Terms*—Image Super Resolution, Convolutional Neural Networks, Generative Adversarial Networks, Residual Networks**

I. Introduction

Image Super Resolution (ISR) can be defined as increasing the size of small images while keeping the drop in quality to minimum, or restoring high resolution images from rich details obtained from low resolution images. This problem is quite complex since there exist multiple solutions for a given low resolution image. This has numerous applications like satellite and aerial image analysis, medical image processing, compressed image/video enhancement etc. There are many forms of image enhancement which includes noise-reduction, up-scaling image and color adjustments. Our main target is to reconstruct super resolution image or high resolution image using different models by up-scaling low resolution image such that texture detail in the reconstruction SR images is not lost.

A. Convolutional Neural Networks

Convolutional neural networks (CNN) date back decades and deep CNNs have recently shown an explosive popularity partially due to its success in image classification. They have also been successfully applied to other computer vision fields, such as object detection, face recognition, and pedestrian detection. Several factors are of central importance in this progress: (i) the efficient training implementation on modern powerful GPUs, (ii) the proposal of the Rectified Linear Unit (ReLU) which makes convergence much faster while still presents good quality, and (iii) the easy access to an abundance of data (like ImageNet) for training larger models.

CNNs have also been proven to be better than conventional methods in image restoration (super-resolution).

B. Residual Networks

Residual networks can also be used for image super resolution. The key importance of **ResNet** lies in its ability to address the problem of vanishing gradients in deep neural networks. As neural networks grow deeper, the gradients that flow backward during the training process tend to become extremely small, which hinders effective learning. This phenomenon is known as the vanishing gradient problem. ResNet introduced the concept of residual learning, where the network learns to model the residual mapping between input and output, rather than trying to learn the entire mapping from scratch. This is achieved through the use of skip connections or shortcuts, which allow the gradients to flow more easily through the network. By propagating the gradients directly to earlier layers, ResNet enables the training of very deep networks

(e.g., hundreds of layers) without suffering from degradation in performance.

Another popular method for super resolution is **Deep Image Prior**. In this a randomly-initialized neural network can be used as a handcrafted prior with excellent results in standard inverse problems such as denoising, super-resolution, and inpainting. Furthermore, the same prior can be used to invert deep neural representations to diagnose them, and to restore images based on flash-no flash input pairs.

C. Generative Adversarial Networks

GANs are a class of AI algorithms used in Unsupervised Machine Learning. Their architecture consists of two networks : (1) Generator and (2) Discriminator, pitting one against the other hence the term "adversarial". The main focus for GANs is to generate data from scratch. In order to understand GANs, we need to understand what these two networks do. In a nutshell, a generative model generates new data that fits the distribution of the training data and does not know anything about the classes of the data. The discriminator network on the other hand discriminates between two (or more) different classes of data.

Therefor, within a GAN, the generator tries to produce some data from probability distribution and the discriminator acts like a judge. The discriminator decides whether the input is coming from the true training dataset of fake (generated) data. Generator tries to optimize data so that it can match the true training data while the discriminator guides the generator to produce realistic data.

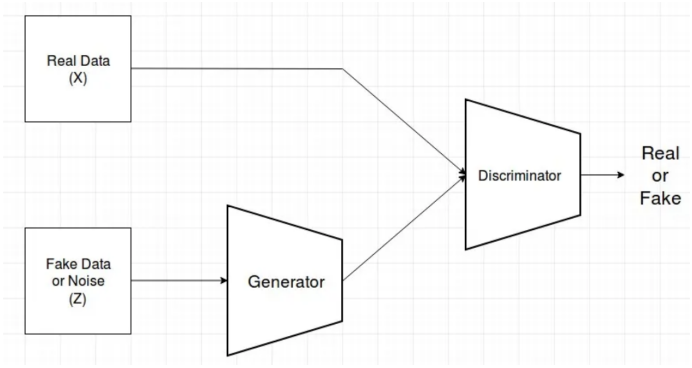

Fig. 1. Basic Architecture of GAN. Extracted from [1]

Discriminator and Generator are both learning at the same learning, and once Generator is trained it knowns enough about the distribution of the training samples so that it can now generate new samples which share very similar properties.

We will be using 2 different GAN architectures namely, **SRGAN** and **ESRGAN** for image super resolution.

II. Background

A. Super-Resolution Convolutional Neural Network (SRCNN)

SRCNN provides superior accuracy compared with state-of-the-art example-based methods. With moderate numbers of filters and layers, this method achieves fast speed for practical online usage even on a CPU. This is faster than a number of example-based methods, because it is fully feed-forward and does not need to solve any optimization problem on usage. Experiments show that the restoration quality of the network can be further improved when (i) larger and more diverse datasets are available, and/or (ii) a larger and deeper model is used. On the contrary, larger datasets/models can present challenges for existing example-based methods. SRCNN can cope with three channels of color images simultaneously to achieve improved super-resolution performance.

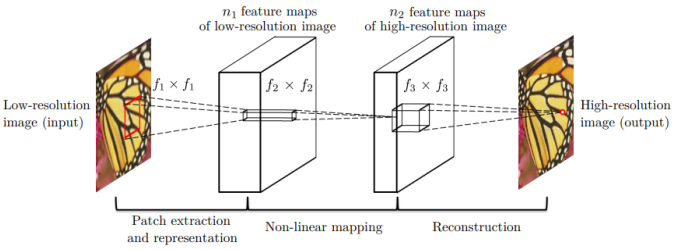

Fig. 2. SRCNN architecture

Consider a single low-resolution image, we first upscale it to the desired size using bicubic interpolation, which is the only pre-processing we perform3. Let us denote the interpolated image as Y. Our goal is to recover from Y an image F(Y) that is as similar as possible to the ground truth high-resolution image X. We wish to learn a mapping F, which conceptually consists of three operations:

- **Patch extraction and representation:** this operation extracts (overlapping) patches from the low-resolution image Y and represents each patch as a high-dimensional vector. These vectors comprise a set of feature maps, of which the number equals to the dimensionality of the vectors.
- **Non-linear mapping:** this operation non-linearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch. These vectors comprise another set of feature maps.
- **Reconstruction:** this operation aggregates the above high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be similar to the ground truth X.

B. Fast Super-Resolution Convolutional Neural Network (FSRCNN)

Though SRCNN is already faster than most previous learning-based methods, the processing speed on large images is still unsatisfactory. To approach real-time, we should accelerate SRCNN for at least 17 times while keeping the previous performance. When we delve into the network structure, we find two inherent limitations that restrict its running speed. First, as a pre-processing step, the original LR image needs to be upsampled to the desired size using bicubic interpolation to form the input. The second restriction lies on the costly non-linear mapping step.

According to the above observations, we investigate a more concise and efficient network structure for fast and accurate image SR. To solve the first problem, we adopt a deconvolution layer to replace the bicubic interpolation. To further ease the computational burden, we place the deconvolution layer at the end of the network. For the second problem, we add a shrinking and an expanding layer at the beginning and the end of the mapping layer separately to restrict mapping in a low-dimensional feature space. Furthermore, we decompose a single wide mapping layer into several layers with a fixed filter size 3 × 3.

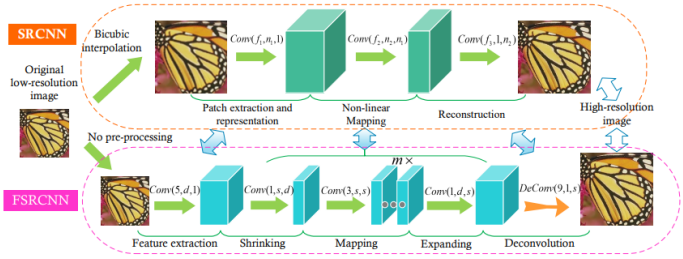

Fig. 3. Network structure comparison between SRCNN and FSRCNN

As shown in Figure 2, FSRCNN can be decomposed into five parts – feature extraction, shrinking, mapping, expanding and deconvolution. The first four parts are convolution layers, while the last one is a deconvolution layer.

Experiments show that the proposed model, named as Fast Super-Resolution Convolutional Neural Networks (FSRCNN) , achieves a speed-up of more than 40× with even superior performance than the SRCNN.

C. Enhanced Deep ResNet for Super Resolution

We compare the building blocks of each network model from original ResNet, SRResNet, and our EDSR. We remove the batch normalization layers from our network. Since batch normalization layers normalize the features, they get rid of range flexibility from networks by normalizing the features, it is better to remove them. We experimentally show that this simple modification increases the performance substantially.

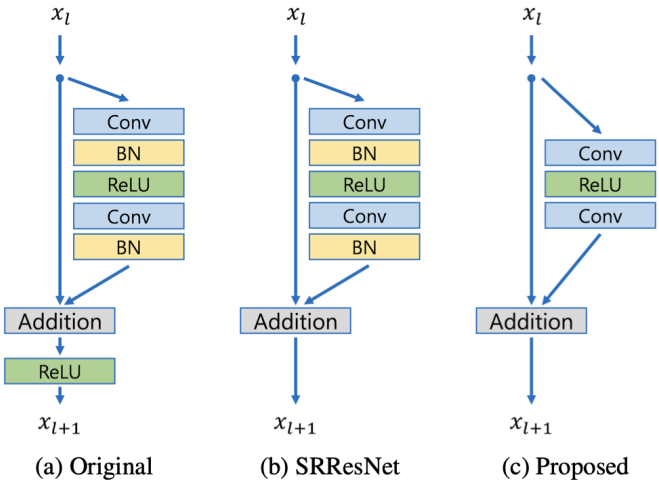

Fig. 4. Comparison between EDSR, ResNEt, SRResnet

Our proposed residual blocks are used to construct the baseline (single-scale) model. The structure is similar to SRResNet, but the model does not include ReLU activation layers outside the residual blocks. Additionally, residual scaling layers are not present in the baseline model as only 64 feature maps are used for each convolution layer. In the final single-scale model (EDSR), the baseline model is expanded by setting B = 32, F = 256, with a scaling factor of 0.1. The model architecture is displayed.

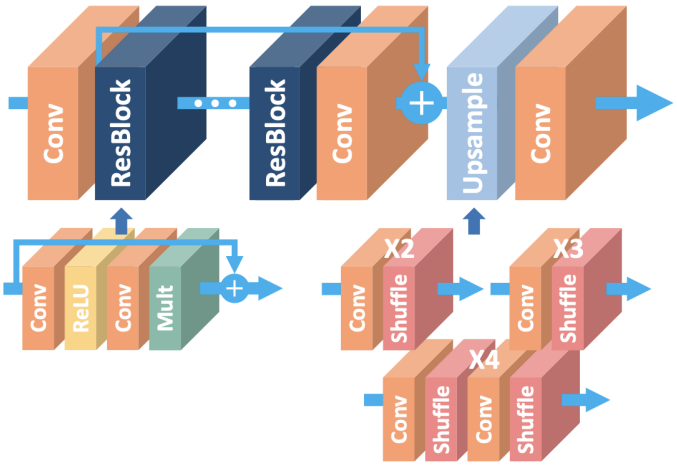

Fig. 5. The architecture of the proposed EDSR

D. Super Resolution GAN

Super-resolution GAN applies a deep network in combination with an adversary network to produce higher resolution images. During the training phase, a high-resolution image (HR) is downsampled to a low-resolution image (LR). A GAN generator upsamples LR images to super-resolution images (SR). We use a discriminator to distinguish the HR images and backpropagate the GAN loss to train the discriminator and the generator.

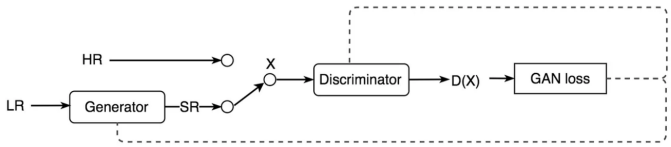

Fig. 6. SRGAN. Extracted from [3]

Below is the network design for the generator and the discriminator. It mostly composes of convolution layers, batch normalization and parameterized ReLU (PRelU). The generator also implements skip connections similar to ResNet. The convolution layer with "k3n64s1" stands for 3x3 kernel filters outputting 64 channels with stride 1.

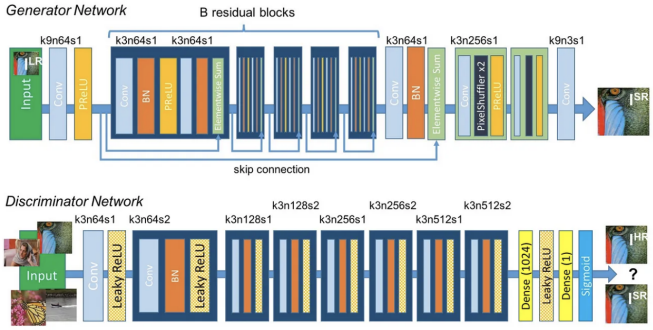

Fig. 7. SRGAN architecture. Extracted from [3]

SRGAN uses a perceptual loss measuring the MSE of features extracted by a VGG-19 network. For a specific layer within VGG-19, we want their features to be matched (Minimum MSE for features).

E. Enhanced Super Resolution GAN

As the name suggests, this is an improved version of previous SRGAN implementation. The overall high-level architecture design of the network is retained but few new concepts are added and changed which ultimately lead to increase in efficiency of the network. The three main improvements over the SRGAN are as follows :

1) **Network Architecture** : The network structure of Generator is improved by introducing the Residual-in-Residual Dense Block (RRDB), which increase the capacity of the network and makes the training easier too. Also, all the Batch Normalization (BN) layers were removed.

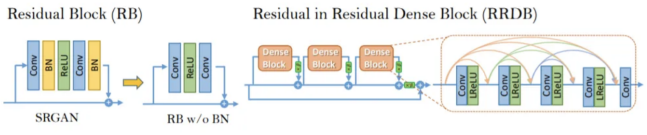

Fig. 8. RRDB. Extracted from [2]

2) **Adversial Loss** : The second enhancement made is the improving the discriminator using the concept of Relativistic average GAN (RaGAN) which makes the discriminator to judge "whether one image is more realistic than the other" rather than "whether one image is real or fake".

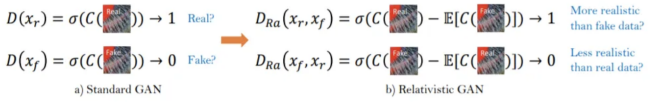

Fig. 9. Difference between standard discriminator and relativistic discriminator. Extracted from [2]

3) **Perceptual Loss** : The perceptual loss is introduced in super-resolution to optimize super-resolution model in feature space instead of pixel space. The perceptual loss is improved in the ESRGAN by using the features before activation which could lead to brightness consistency and texture recovery. The perceptual loss is implemented by using VGG features before activation instead of after activation as in SRGAN.

F. Deep Image Prior

Deep convolutional networks have become a popular tool for image generation and restoration. Generally, their excellent performance is imputed to their ability to learn realistic image priors from a large number of example images. The structure of a generator network is sufficient to capture a great deal of low-level image statistics prior to any learning. In order to do so, we show that a randomly-initialized neural network can be used as a handcrafted prior with excellent results in standard inverse problems such as denoising, super-resolution, and inpainting. Furthermore, the same prior can be used to invert deep neural representations to diagnose them, and to restore images based on flash-no flash input pairs.

Image restoration problems the goal is to recover original image x having a corrupted image x0. Such problems are often formulated as an optimization task:

$$\min_x E(x, x_0) + R(x)$$

where E(x, x0) is a data term and R(x) is an image prior. The data term is usually easy to design for super-resolution. Trend is to capture the prior R(x) with a ConvNet by training it using large number of examples. We first notice, that for a surjective g(theta-¿x) the following procedure in theory is equivalent to

$$\min_\theta E(g(\theta), x_0) + R(g(\theta))$$

In practice g dramatically changes how the image space is searched by an optimization method. Furthermore, by selecting a "good" (possibly injective) mapping g , we could get rid of the prior term. We define g(theta) as f(z) , where f is a deep ConvNet with parameters theta and z is a fixed input, leading to the formulation

$$\min_\theta E(f(z); x_0)$$

Here, the network f is initialized randomly and input z is filled with noise and fixed.

Instead of searching for the answer in the image space we now search for it in the space of neural network's parameters. We emphasize that we never use a pretrained network or an image database. Only corrupted image x0 is used in the restoration process.

III. Evaluation Metric : PSNR

Peak signal-to-noise ratio definition (PSNR) is most commonly used as a quality estimation for the loss of quality through different codecs and image compression where the signal is the original image and the noise is error created by compressing the image.

PSNR is very common for evaluating image enhancement techniques, such as Super resolution where the signal is the original/ground truth image and the noise is the error not recovered by the model.

Although PSNR is a logarithm based metric, it is based on the MSE.

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

Fig. 10. Peak signal-to-noise ratio. Extracted from [8].

IV. Training Details

A. SRCNN

1) Dataset Used: The model has been trained on a 91-image dataset used in the original SRCNN paper.

2) Hyperparameters: Learning rate = 0.001, No. of epochs = 25

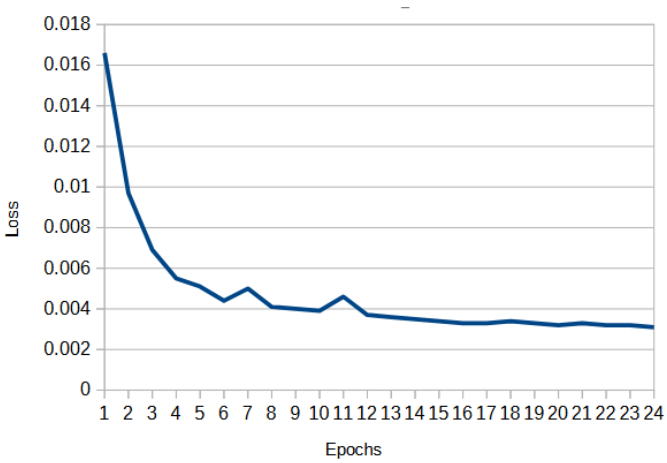

Fig. 11. Loss vs Epoch for SRCNN

B. FSRCNN

1) Dataset Used: The model has been trained on a 91-image dataset used in the original SRCNN paper.

2) Hyperparameters: Learning rate = 0.001, No. of epochs = 20

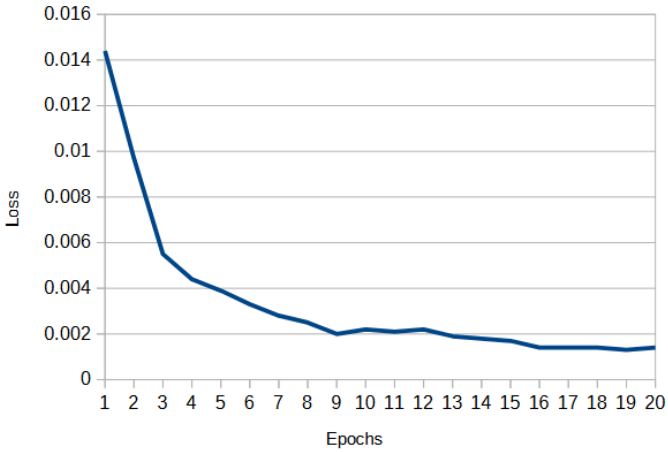

Fig. 12. Loss vs Epoch for FSRCNN

C. EDSR

1) Dataset Used: The model has been trained on DIV2K dataset.

2) Hyperparameters: Learning rate = 0.001 ϵ=1e-8 No. of epochs = 20

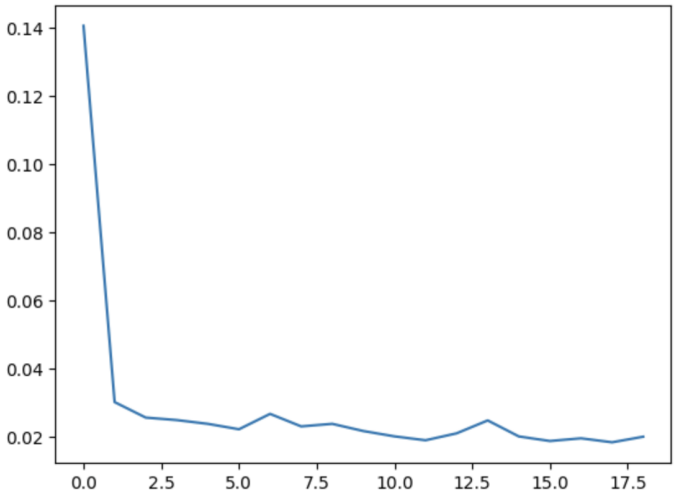

Fig. 13. Loss vs Epoch for EDSR

D. Deep Image Prior

1) Hyperparameters: Learning rate = 0.01

Here optimization happens on the same image for 2000 epochs

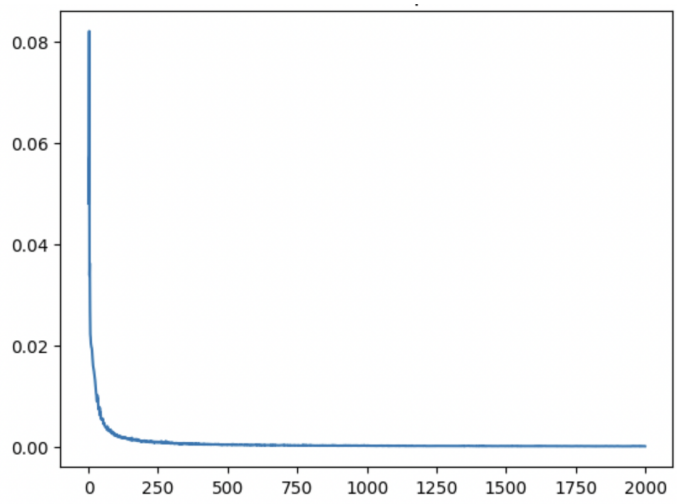

Fig. 14. Loss vs Epoch for DIP

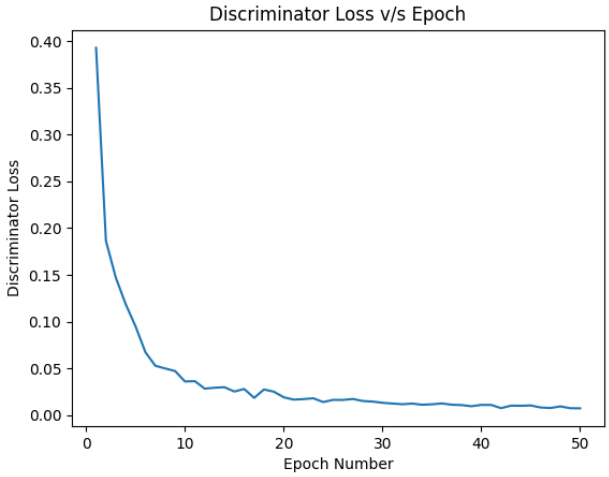

Fig. 16. Discriminator Loss vs Epoch for SRGAN

E. SRGAN

1) Dataset Used: The original paper for SRGAN had trained their networks on the renowned ImageNet image recognition dataset. Although, we understand that it is beneficial to train such complex models on large amounts of data, the dataset proved to be too heavy and hence we decided to train this model on the tf_flowers dataset which consists of 3670 images. The dataset does seem too small however it turned out to be just enough for a toy dataset to access the performance of the model.

2) Training - Validation Split: The first 600 images were used as a validation dataset whereas the rest of the images comprised of the training dataset.

3) Hyperparameters: The values for most of the hyperparameters are inspired by the original research paper. However, for computational reasons, we decided to train the network for 50 epochs. Both the generator as well as the discriminator was optimized using SGD algorithm with a learning rate of 0.001.

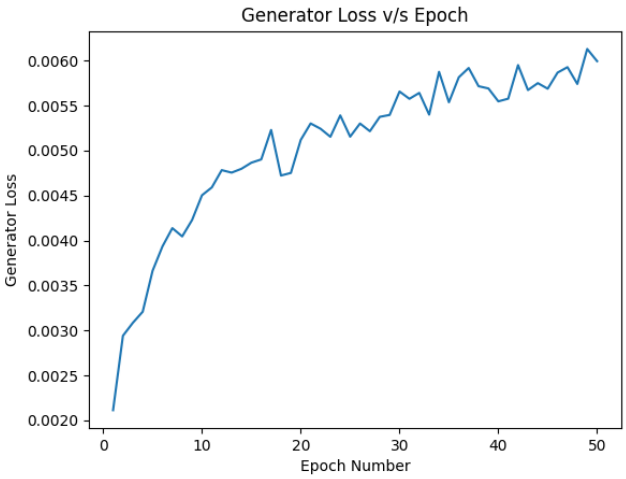

Fig. 15. Generator Loss vs Epoch for SRGAN

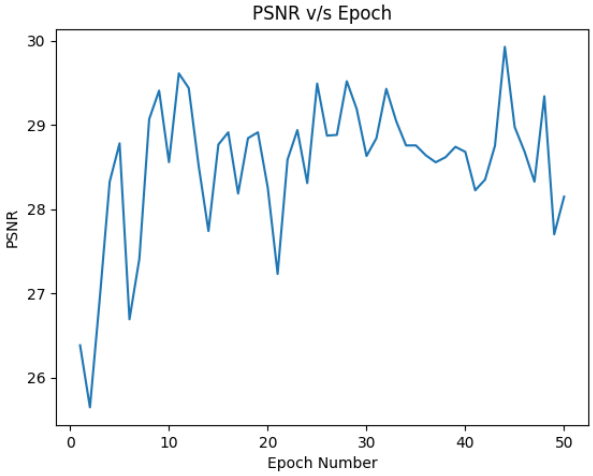

Fig. 17. PSNR vs Epoch for SRGAN

F. ESRGAN

1) Training - Validation Split: The training - validation split was 90 - 10 which means 90 % of the data was used as a training set whereas the rest 10 % was used as the validation set.

2) Hyperparameters:

- SCALING FACTOR = 4
- FEATURE MAPS = 64
- RESIDUAL BLOCKS = 16
- LEAKY ALPHA = 0.2
- DISC BLOCKS = 4
- RESIDUAL SCALAR = 0.2
- PRETRAIN LR = 1e-4
- FINETUNE LR = 3e-5
- PRETRAIN EPOCHS = 1500
- FINETUNE EPOCHS = 1000
- TRAIN BATCH SIZE = 10

- INFER BATCH SIZE = 10

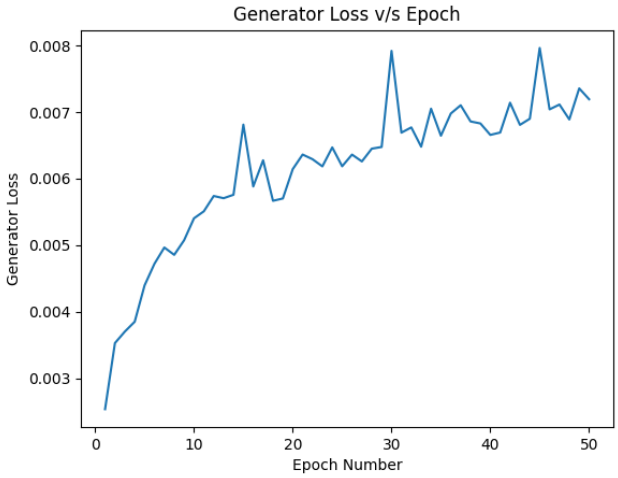

Fig. 18. Generator Loss vs Epoch for ESRGAN

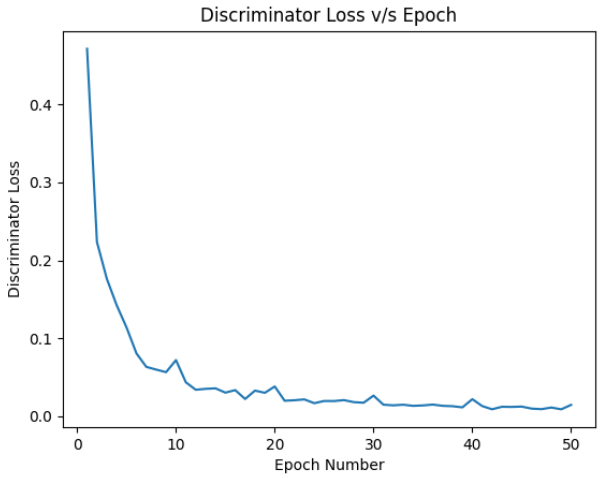

Fig. 19. Discriminator Loss vs Epoch for ESRGAN

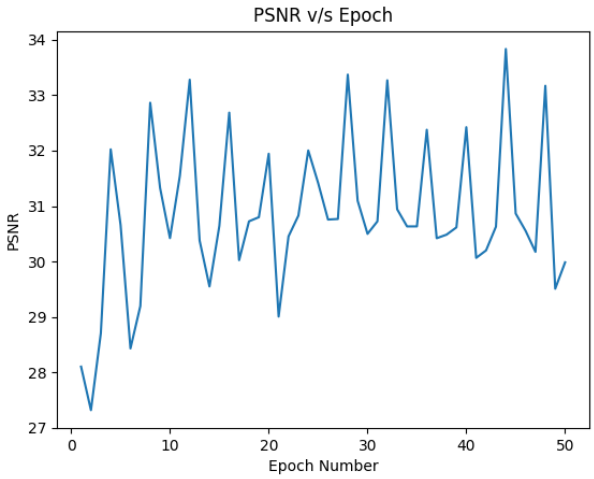

Fig. 20. PSNR vs Epoch for ESRGAN

V. Results

A. SRCNN

PSNR of a low-resolution image and its SRCNN reconstruction with respect to the original high-quality image

Fig. 21. Degraded Image PSNR: 27.24 — SRCNN Reconstruction PSNR: 29.66

The below plot shows the comparison of PSNR for image outputs from SRCNN with that of bicubic and sparse-coding images.

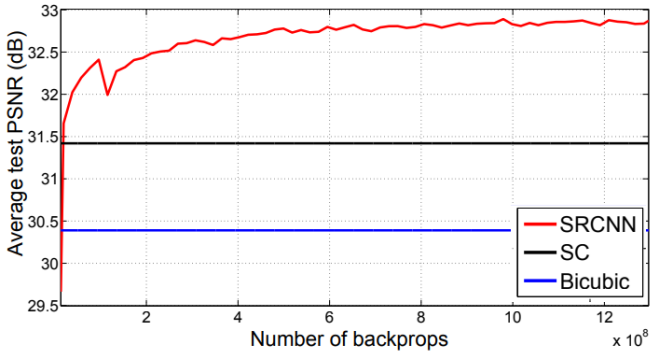

Fig. 22. PSNR Comparison (SRCNN, Bicubic, Sparse-coding)

B. FSRCNN

FSRCNN show superior restoration quality compared to that of SRCNN. Comparison of PSNR for FSRCNN image and bicubic image:

Fig. 23. Bicubic Image PSNR (Left): 24.04 — FSRCNN Reconstruction PSNR (Right): 28.68

The below plot shows the PSNR vs iterations for FRSCNN.

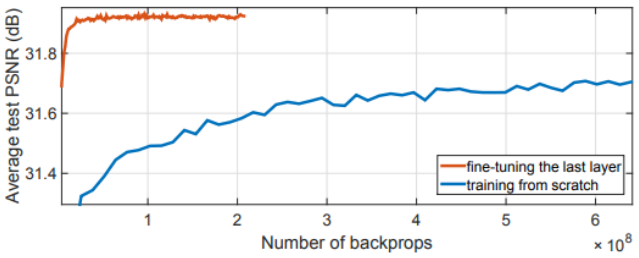

Fig. 24. PSNR Comparison vs iterations (FSRCNN)

C. EDSR

The following is comparison of PSNR for EDSR super resoluted image and bicubic image

Fig. 25. PSNR Comparison — PSNR after using EDSR is 33.17

The below plot shows increased PSNR for image outputs from EDSR as compared to PSNRs for bicubic images. The test has been performed on first 6 images.

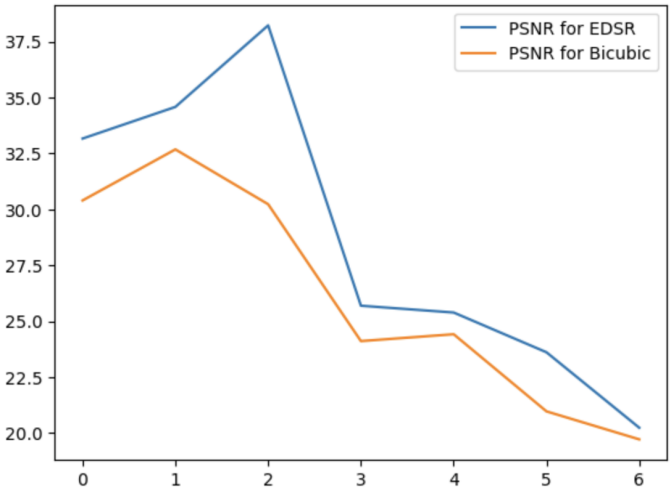

Fig. 26. PSNR Comparison (Horizontal axis denotes test image)

D. DIP (Deep Image Prior)

The following image shows how PSNR is being improvised in the image.

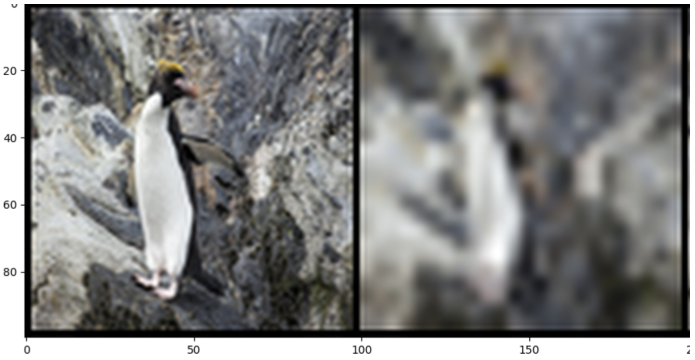

Fig. 27. PSNR Comparison (Horizontal axis denotes test image)

The below plot shows improvement of PSNR with increasing number of epochs.

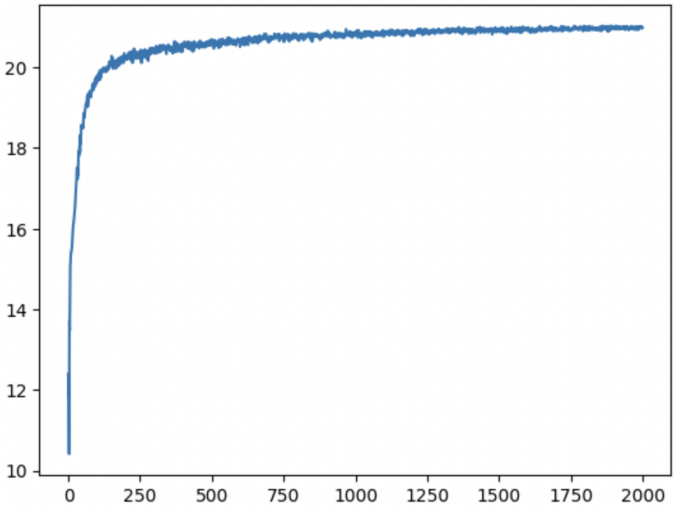

Fig. 28. PSNR Comparison (Horizontal axis denotes test image)

E. SRGAN & ESRGAN

The following comparison shows how the value of PSNR is being improved upon using SRGAN (left image) and ESRGAN (right image).

- PSNR after using SRGAN is 30.4
- PSRN after using ESRGAN is 33.7

Fig. 29. Another comparison between (a) ESRGAN (b) SRGAN (c) Bicubic (d) Low Resolution

VI. Conclusion and Limitations

Various image super-resolution methods have made significant progress in recent years, leveraging advancements in deep learning and computer vision. Different methods with different architectures try to tackle different challenges such as the trade-off between reconstruction accuracy as well as computational complexity, and the overall ability to generalize to different types of images. Overall, image super-resolution has the potential to be useful in many applications, such as medical imaging, satellite imagery etc. What we have presented in this report is the implementation and the comparison between different models like SRCNN (PSNR = 29.66), FSRCNN (PSNR = 28.68), EDSR (PSNR = 33.17) , DIP (PSNR = 21.27), SRGAN (PSNR = 30.4) and ESRGAN (PSNR = 33.7) since we feel that with the ongoing research and development, image super-resolution techniques are expected to continue improving, and we can be sure to expect to see even more sophisticated algorithms based on these.

Some of the limitations of our project are as follows :

- We were unable to standardize the training procedure. This was partially due to the fact that different models were trained by different members of the team. The other reason was that some complex models like SRGAN and ESRGAN required more training data or simply needed to be trained for a large number of epochs. However, we failed to realise this on time and used different training datasets, training-validation splits and also trained our models for different number of training epochs.
- We also did not have much time to experiment with the models with which we were working with. However, with the time given to us, we tried our best to play around with the crucial hyperparameters, in an attempt to understand it's significance and how it can possibly affect the final output and it's PSNR score.

VII. Future Work

- **Incorporating attention mechanisms** : Attention mechanisms have shown great potential in computer vision tasks, including image super-resolution. Future work could explore how to incorporate attention mechanisms into existing super-resolution models to improve their performance.
- **Adapting to diverse datasets** : Existing super-resolution techniques are often trained and tested on specific datasets. Future work could focus on developing models that can adapt to diverse datasets, such as those with different resolutions, image qualities, and content types.

Work Contribution

- **K S Varun** : Worked on Super-Resolution Convolutional Neural Network (SRCNN) and Fast Super-Resolution Convolutional Neural Network (FSRCNN)
- **Nikhil Tiwari** : Worked on Enhanced Deep ResNet (EDSR) and Deep Image Prior (DIP)
- **Swapnoneel Kayal** : Worked on Super Resolution Generative Adversial Network (SRGAN) and Enhanced Super Resolution Generative Adversial Network (ESRGAN)

References

1) Image Super-Resolution Using Deep Convolutional Networks
2) Super Resolution CNN for Image Restoration
3) Accelerating the Super-Resolution Convolutional Neural Network
4) FSRCNN - Keras
5) Single Image Super Resolution Using GANs — Keras
6) ESRGAN : Enhanced Super Resolution GAN
7) GAN — Super Resolution GAN (SRGAN)
8) Deep learning image enhancement insights on loss function engineering
9) Deep Image Prior Wikipedia
10) EDSR