Test Results

Sougata Moi
m23mac008@iitj.ac.in
moisougata15@gmail.com

Github link : https://github.com/sm1899/deeplens

Common Test I. Multi-Class Classification

Notebook link: https://www.kaggle.com/code/moisougata/multi-class-classification

Task : build a model for classifying the images into lenses using **PyTorch** or **Keras**.

Methodology

To tackle this task I've used the Fine-Tuned **EfficientNet architecture**[1], pre-trained on ImageNet. This architecture has demonstrated exceptional performance across various image classification tasks. To adapt the model for our specific task, I've fine-tuned it using The given dataset.

- The Fine-Tuned EfficientNet architecture consists of **convolutional layers followed by fully connected layers**.
- I've **modified the initial layers** to accommodate the grayscale input images and adjusted the fully connected layers to match the number of output classes

Reason to use Transfer learning : pre-training allows the model to learn generic features from diverse images, which can then be fine-tuned on smaller, task-specific datasets with comparatively fewer labeled examples. Leveraging transfer learning with EfficientNet can accelerate the training process and improve the model's performance on specialized tasks.

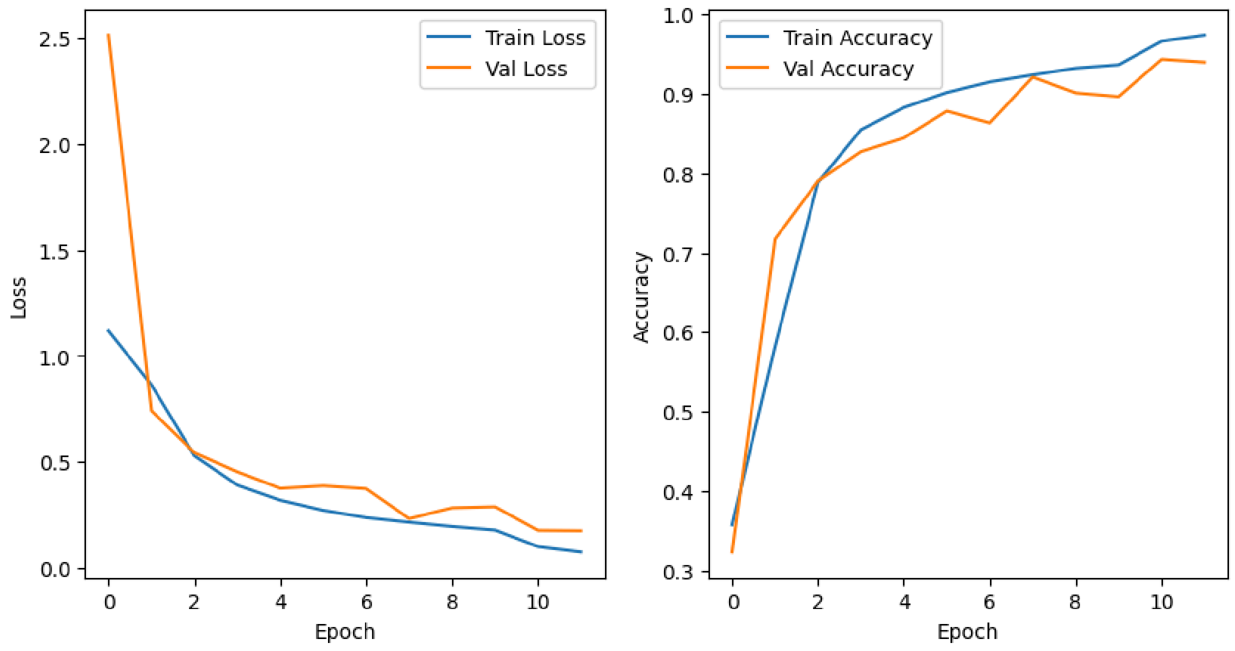
Train-Test Split : The given dataset has been divided into 90 : 10 for training and testing purpose,10% of the training dataset has been used for validation during training

Hyperparemetres

batch_size = 32
num_epochs = 12
lr = .001

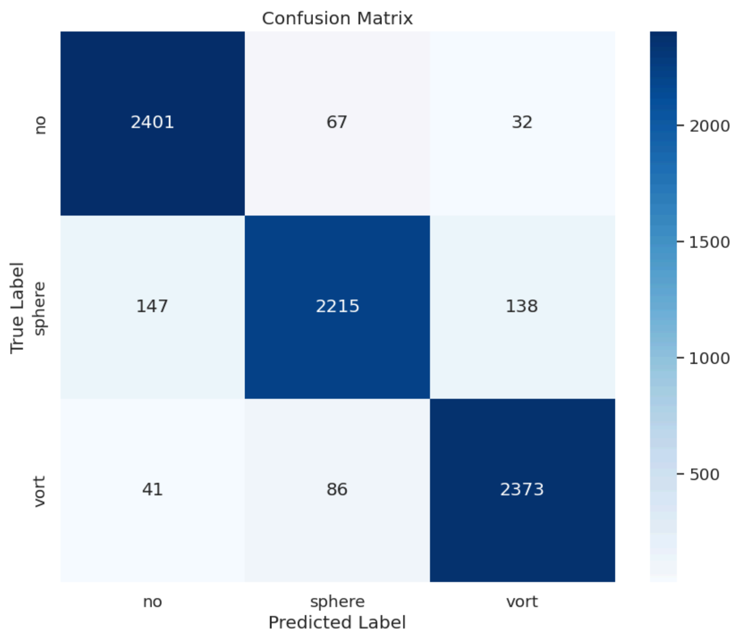
Result

After 12 epochs the model achieves 93.93% accuracy on the validation dataset

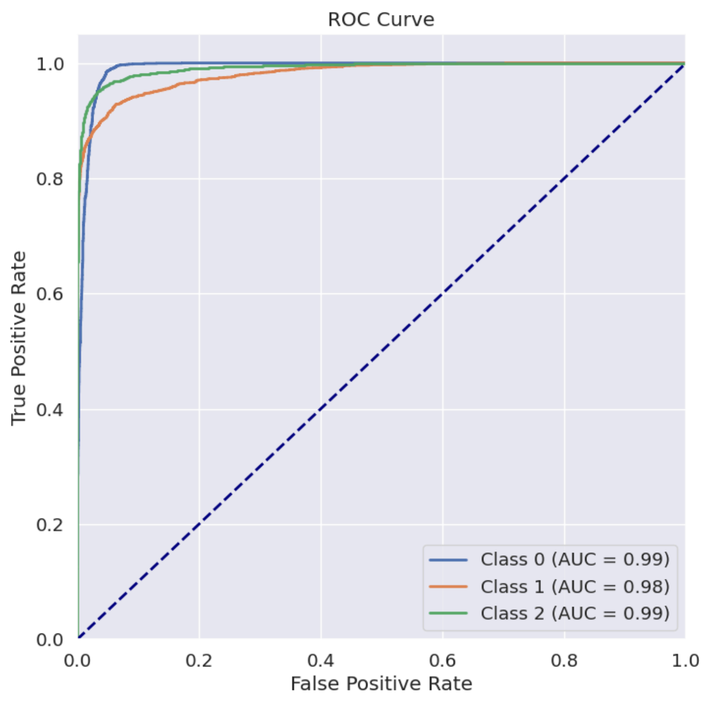


Test dataset: Micro-average AUC Score: **0.9895** , Accuracy : **93.18%**

Confusion matrix



Roc Curve



Specific Test III. Image Super-resolution

Task III.A

Notebook link: https://www.kaggle.com/code/moisougata/image-super-resolution-task-iii-a

Methodology

I've used two variants of the Super-Resolution Generative Adversarial Network (SRGAN)[2] architecture.

- The first variant utilizes L2 loss, optimizing pixel-wise similarity between the generated and ground truth images. This model basically behaves like a Enhanced Deep Residual Networks (EDSR) with MSE loss[3].
- The second variant incorporates both L2 and adversarial loss, encouraging the model to generate visually realistic images. Additionally, i've integrated EfficientNet loss in the second model , leveraging features learned by the EfficientNet architecture during classification task to guide the super-resolution process.

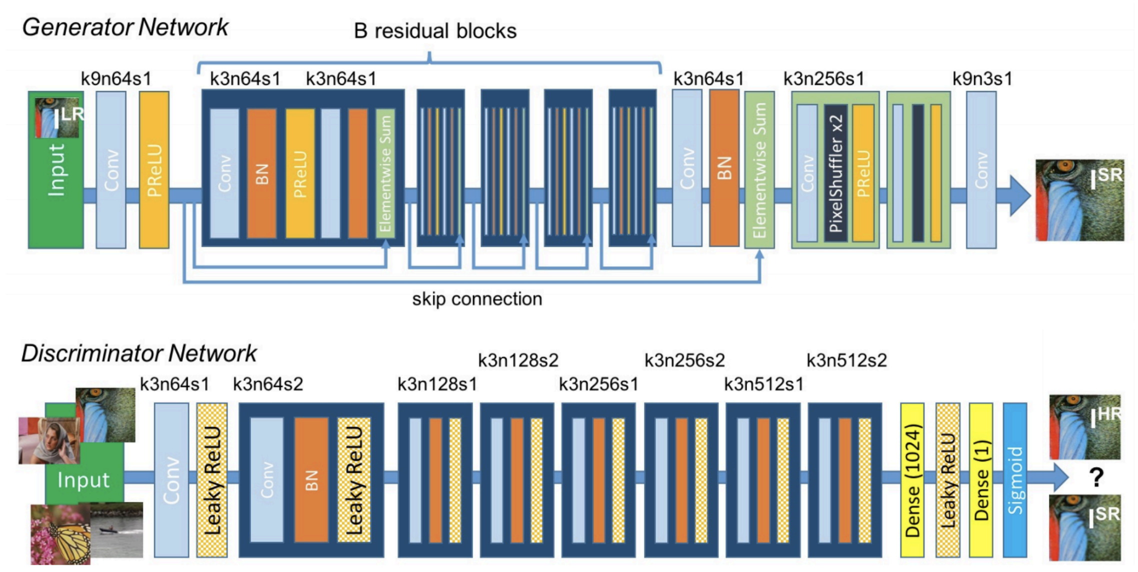


Figure 4: Architecture of Generator and Discriminator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer.

Hyperparemetres

For The first model

LEARNING_RATE = .0005
NUM_EPOCHS = 8
BATCH_SIZE = 63

For The second model

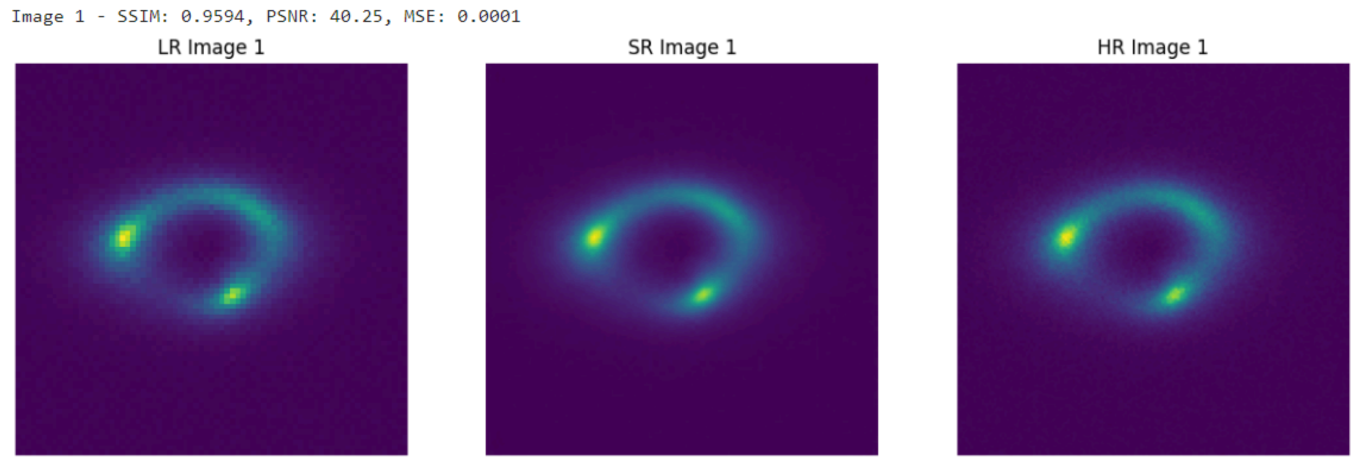
NUM_EPOCHS = 25
LEARNING_RATE = .000001
BATCH_SIZE = 32

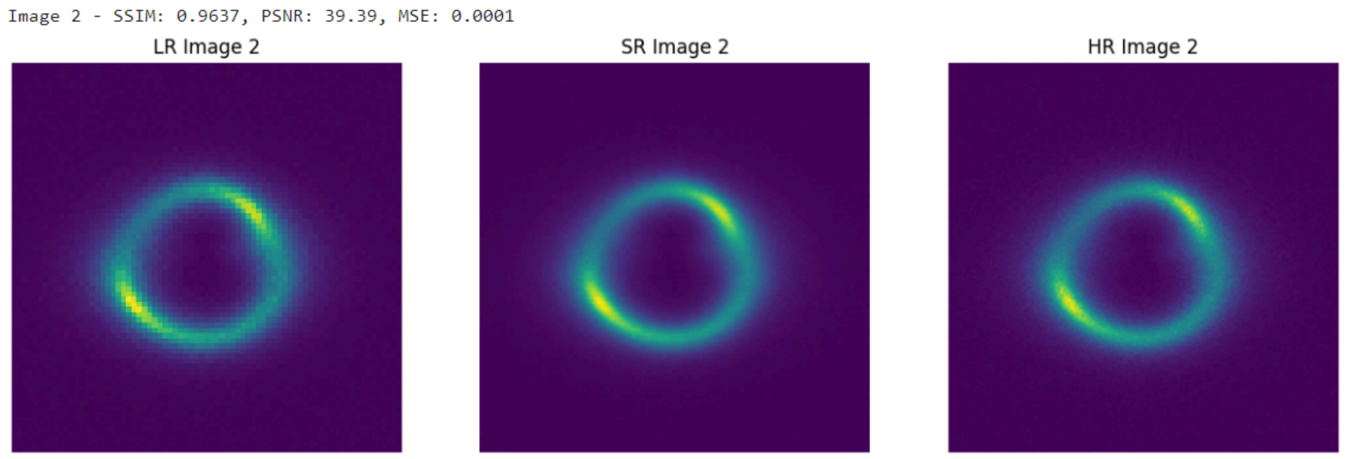
Result

The **First model** achieve following metrics on the Validation Dataset

- Mean SSIM = 0.9725
- Mean PSNR = 41.3440
- Mean MSE = 0.0001

Some SR image created by the first model from the val dataset plotted below with their corresponding metrics



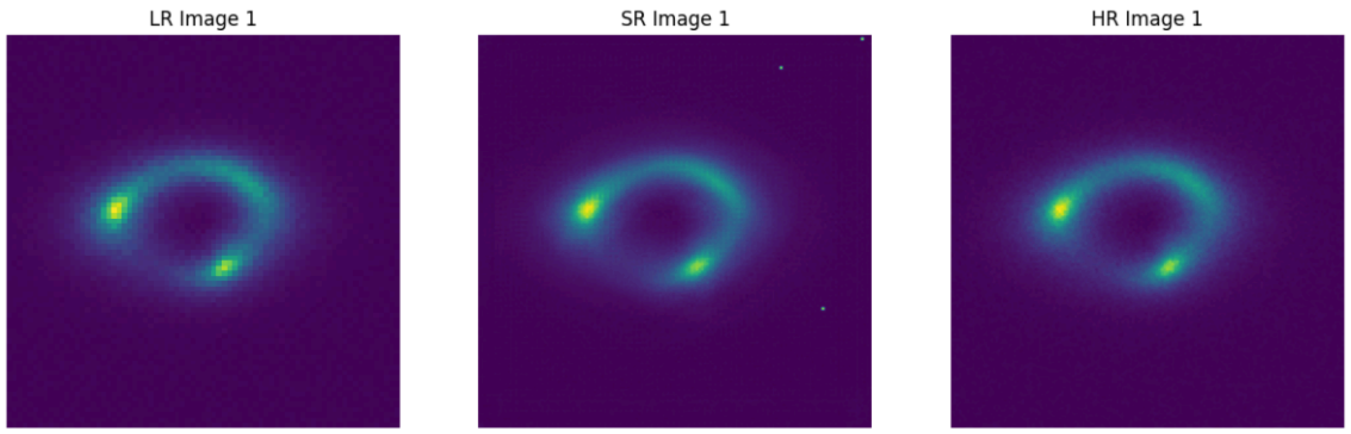


The **Second model** achieve following metrics on the Validation Dataset

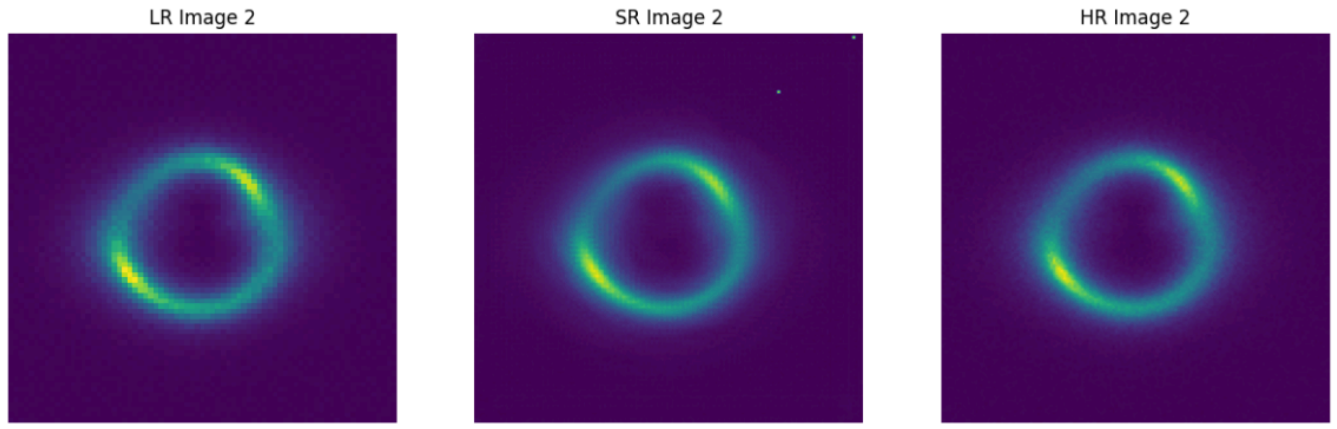
- Mean SSIM = 0.9415
- Mean PSNR = 37.5205
- Mean MSE = 0.0002

Although The second model lacks in metrics , it produces more visually closer SR to HR in my opinion. The reason for performing poorly on metrics is due to the **artifacts** in the Generated SR image.AS GAN based SR models are notoriously Famous for producing unpleasant and undesirable artifacts,this result is expected , we can further explore different Models which Detect and Delete the Artifacts of GAN-based Real-World Super-Resolution[4.

```
Image 1 - SSIM: 0.9354, PSNR: 37.56, MSE: 0.0002
```



```
Image 2 - SSIM: 0.9395, PSNR: 36.37, MSE: 0.0002
```



Task III.B

Notebook link: https://www.kaggle.com/moisougata/image-super-resolution-test-iii-b

Methodology

I have used the previous two models and fine-tuned them using the small dataset of HR/LR image pairs collected from HSC and HST telescopes.

Hyperparemetres

For The first model

LEARNING_RATE = .0001
NUM_EPOCHS = 10

BATCH_SIZE = 64

For The second model

NUM_EPOCHS = 13
LEARNING_RATE = .000001
BATCH_SIZE = 64

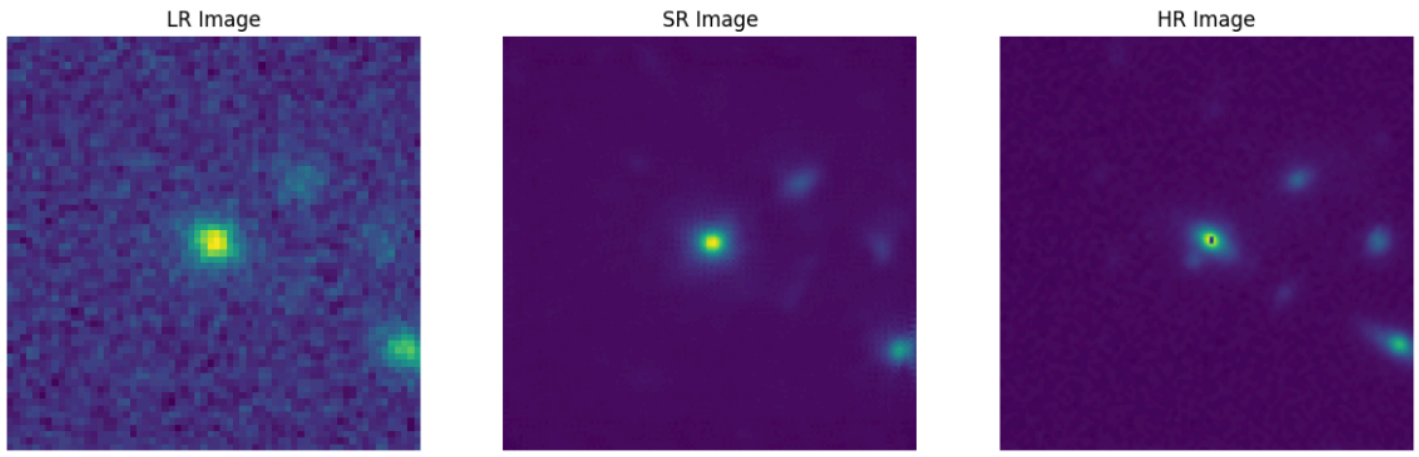
Result

The **First model** achieve following metrics on the Validation Dataset

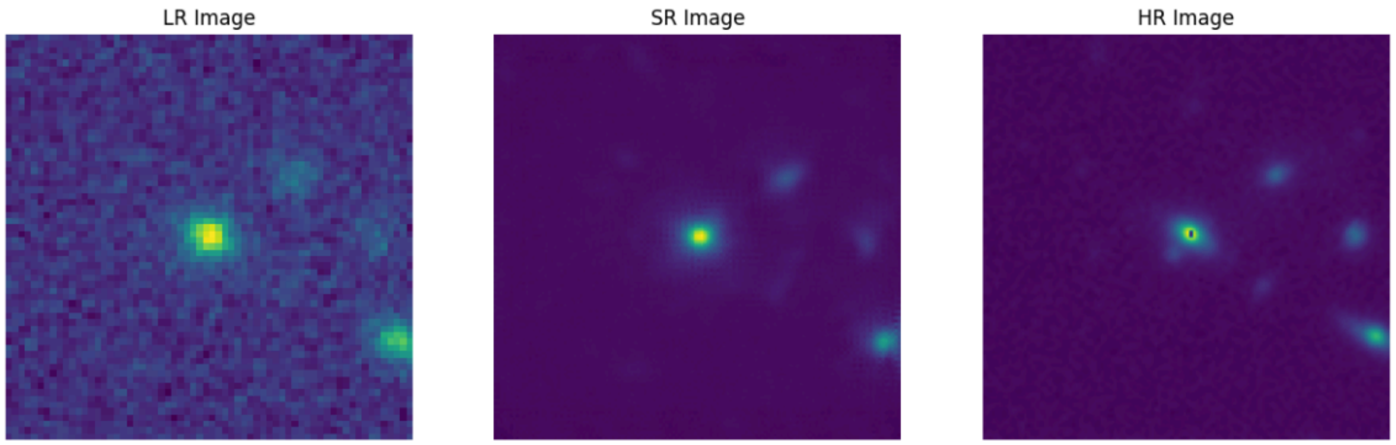
- Mean SSIM = 0.8431
- Mean PSNR = 34.5012
- Mean MSE = 0.0008

Some SR image created by the first model from the val dataset plotted below with their corresponding metrics

```
Image 1 - SSIM: 0.8822, PSNR: 33.18,MSE: 0.0005
```



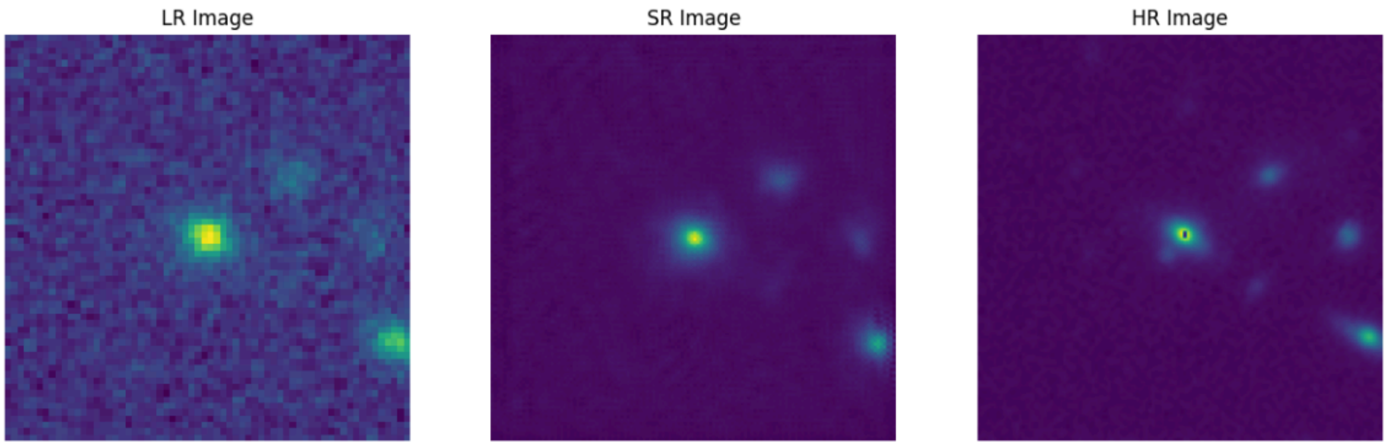
```
Image 2 - SSIM: 0.8822, PSNR: 33.18,MSE: 0.0005
```

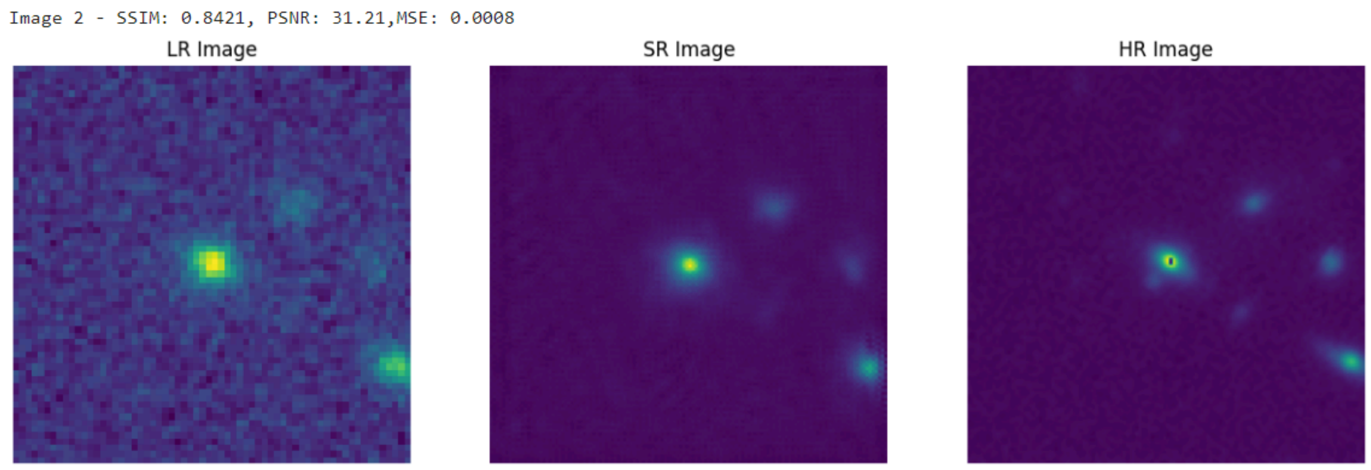


The **Second model** achieve following metrics on the Validation Dataset

- Mean SSIM = 0.7834
- Mean PSNR = 34.0317
- Mean MSE = 0.0009

```
Image 1 - SSIM: 0.8421, PSNR: 31.21,MSE: 0.0008
```





In this case we observe the l2 loss performs better than the gan, which makes sense since 300 samples is simply too small to even properly implement Transfer learning with GAN.
We could explore **Few Shot SR** models for further improvement .

Citations

1. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
2. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
3. ESRGAN Implementation
4. Detect and Delete the Artifacts of GAN-based Real-World Super-Resolution Models