

GNR638
MINI Project : 2
Abhishek Anand: 200260002
Kapil Bhagat: 200260022

Problem statement: Image deblurring using CNN

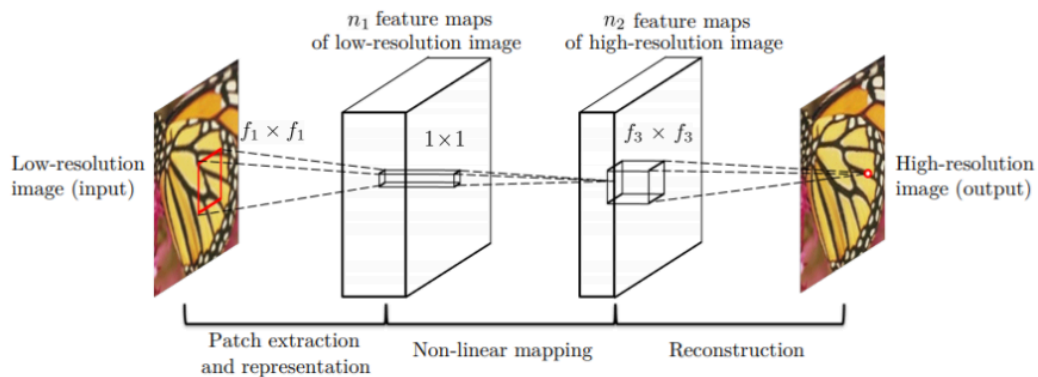
In this we have applied 3 types of Gaussian blur on the **24K** image datasets:

- a. Kernel size = 3×3 , sigma = 0.3
- b. Kernel size = 7×7 , sigma = 1
- c. Kernel size = 11×11 , sigma = 1.6

After this we have **72K** different blurred images.

We have created a Dataloader where I have taken three instances of blurred images as X_{train} and for those three instances, We have taken one instance of sharp image as y_{train} .

After this we have used three layers of **SRCNN (Super Resolution Convolutional Neural Network) architecture**:



Architecture of SRCNN:

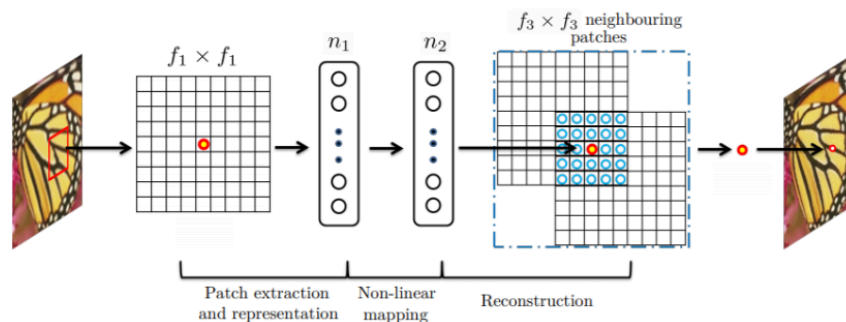


Image showing patch extraction and image reconstruction using SRCNN

We have experimented with different number of training parameters of model, As there is device and Kaggle constraint and also time constraints for each epoch, I have experimented on maximum of 4.8 M trainable parameters (as it takes about 10 hours for only epoch1 with 14M trainable parameters.)

We have used batch size = 2, and learning_rate = 0.0001 over all the models

SRCNN with 4.8M parameters:

First layer: **(3,2048)** with kernel_size = 9, padding = 2

Second layer: **(2048, 2048)** with kernel_size = 1, padding = 2

Third layer: **(2048, 3)** with kernel_size = 5, padding = 2

Activation function: **ReLU** in every layers

Total number of Trainable parameters = 4849667

Total number of epochs = 1

Total train loss after first epoch = 0.00044

Total validation loss after first epoch = 0.00030

Initially this model was supposed to run for 3 epochs, but we have to stop it after 1st epoch, because it was taking too much time for one epoch(6hrs for each epoch), but I have resultant image after this 1st epoch:

blur_0



sharp_0



Deblurred_0



SRCNN with 1.3M parameters:

First layer: **(3,1024)** with kernel_size = 9, padding = 2

Second layer: **(1024, 1024)** with kernel_size = 1, padding = 2

Third layer: **(1024, 3)** with kernel_size = 5, padding = 2

Activation function: **ReLU** in every layers

Total number of Trainable parameters = 1376259

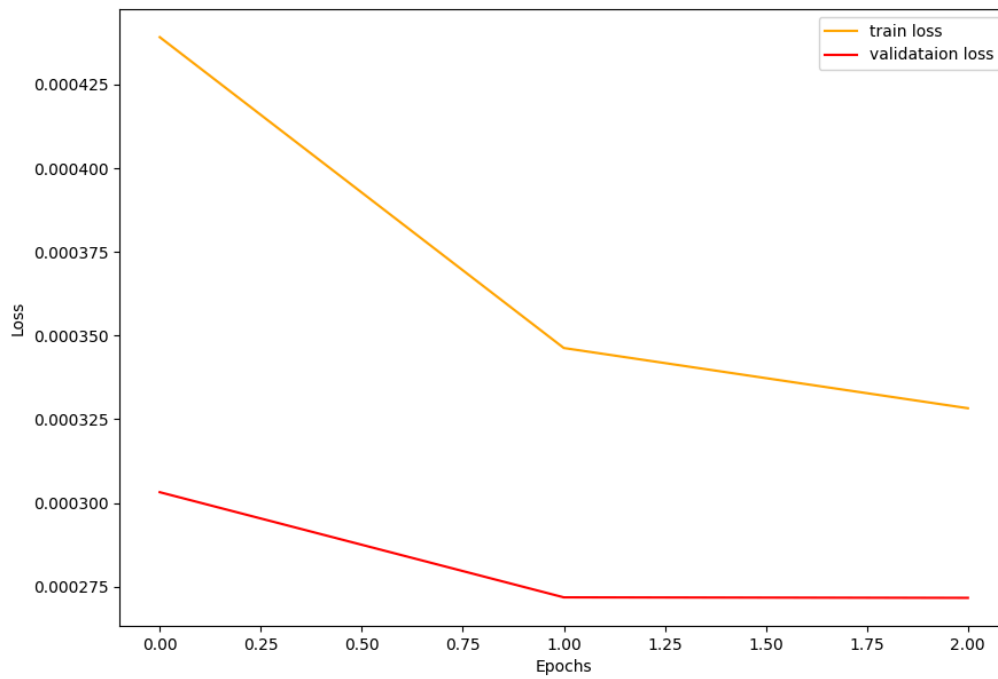
Total number of epochs = 3

Total train loss after 3rd epoch epoch = 0.00033

Total validation loss after 3rd epoch = 0.00027

Took **383.122 minutes to train**

- **Training and Validation loss curve:**



- **Outputs(deblurred images) for this model for every epochs:**

Blur_0:



sharp_0:



Val_deblurred_0:



Val_deblurred_1:



Val_deblurred_2:



- **Evaluation:**

We have evaluated the 1.3M model and calculate PSNR score on test deblurred data and test sharp images:

Total number of images: 150

Average PSNR between corresponding images: 25.914224295994213 dB

Generally scores between **20dB-60dB are considered acceptable**, but also greater than 30dB is considered a very good score. We got ~26dB, this is because of model architecture and less number of epochs.

We have trained the **SRCNN with ~30k parameters** for each gaussian blur data separately for 10 epochs each. We got very satisfying results:

1. **For Kernel size = 3x3 , sigma = 0.3 Gaussian blur with 10 epochs:**

Blur_0



Sharp_0



Val_deblurred_0:



Val_deblurred_9:



2. For Kernel size = 7×7 , sigma = 1 Gaussian blur with 10 epochs:

Blur_0:



Sharp_0:



Val_deblurred_9:



3. For Kernel size = 11x11 , sigma = 1.6 Gaussian blur with 10 epochs:

Blur_0:



Sharp_0:



Val_deblurred_9:



Qualitative Analysis:

We noticed some random patches or dots of different colors (red, green, blue) on the deblurred images, this is the reason why we got less **PSNR score**. We should have trained the SRCNN with **4.8M parameters for 5 epochs**, for better deblurred images.

Test_images_deblurred_output contains random patches:





- Some of the images are well deblurred:



Quantitative Analysis:

As we have showed two models with different number of trainable parameters:
We have also experimented with increasing the convolutional layers up to:

- ~14M trainable parameters (with training time 25 hrs/epoch),
- ~12M trainable parameters (with training time 20 hrs/epoch),
- ~8M trainable parameters (with training time 14 hrs/epoch),

but we have device **GPU limit constraints**. Kaggle's 30hrs limit was over also time constraints, so we can't train for these much trainable parameters.

Best model we can train smoothly on our device is SRCNN with 1.3M trainable parameters.

Code, checkpoints and test_deblurred_images folder link:

https://drive.google.com/drive/folders/1-9Iy-vWKWsrtRv0d-yV4-9D_bAl0sOtc?usp=sharing