GNR638 Mini Project 2 Report (for Problem 1)

Team: Kavya Gupta (22B1053), Atharva Bendale (22B0901), Soham Dahane (22B0941)

Important Details:-

- Implemented the **UNet** model architecture for image deblurring.
- The script **inference.py** loads the model and calculates output of a set of low resolution images and calculates the PSNR values (**run this in the same directory in which the mp2_test folder is present**)
- The file **deblur_image.py** is the *main training script*, you can do "python3 deblur_image.py --help" for help regarding the required inputs
- All the training logs are saved inside **training.log** file
- The file **model_summary.txt** contains information regarding the numbers of params in the model (Total parameters 8,438,403)
- The file **unet_model** contains the code for the UNet network, the file **unet_utils** contains code for any helper functions/classes used in the model
- An updated **eval.py** is also given in the submission. Make sure it is inside the "mp2_test" folder before running inference.py
- The final checkpoint is stored in **best_model.pth**.

PSNR Score achieved : 26.3968

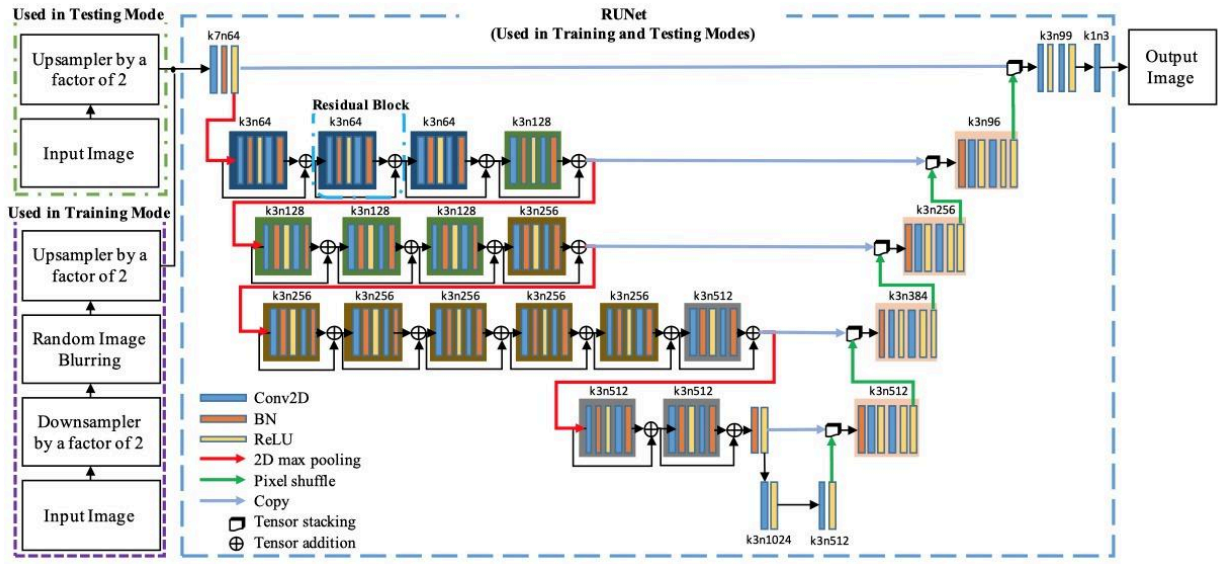
Model Architecture:-

- Model used: U-Net
- Exact layers detail in **model_summary.txt**

The architecture of U-Net is unique in that it consists of a contracting path and an expansive path. The contracting path contains encoder layers that capture contextual information and reduce the spatial resolution of the input, while the expansive path contains decoder layers that decode the encoded data and use the information from the contracting path via skip connections to generate a segmentation map.

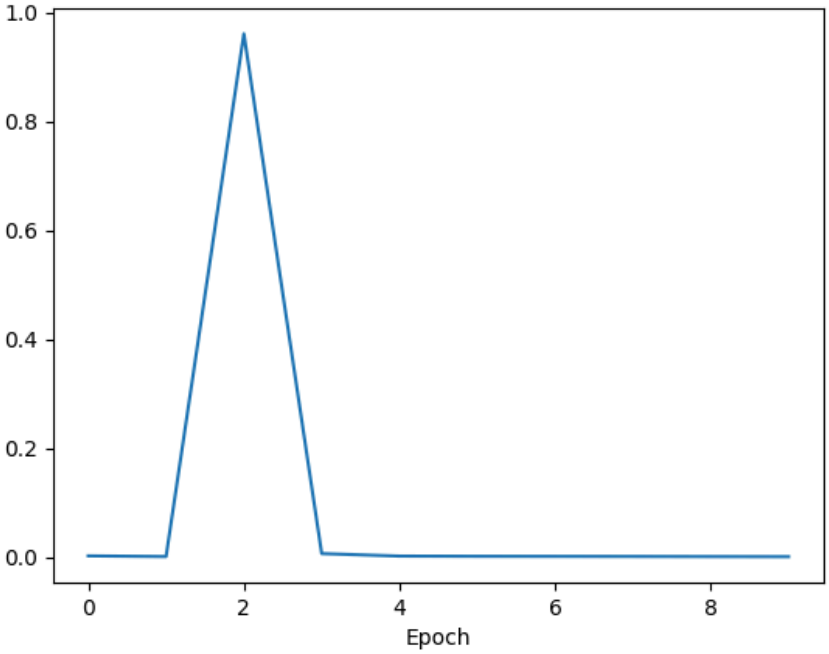
The contracting path in U-Net is responsible for identifying the relevant features in the input image. The encoder layers perform convolutional operations that reduce the spatial resolution of the feature maps while increasing their depth, thereby capturing increasingly abstract representations of the input. This contracting path is similar to the feedforward layers in other convolutional neural networks. On the other hand, the expansive path works on decoding the encoded data and locating the features while

maintaining the spatial resolution of the input. The decoder layers in the expansive path upsample the feature maps, while also performing convolutional operations. The skip connections from the contracting path help to preserve the spatial information lost in the contracting path, which helps the decoder layers to locate the features more accurately.

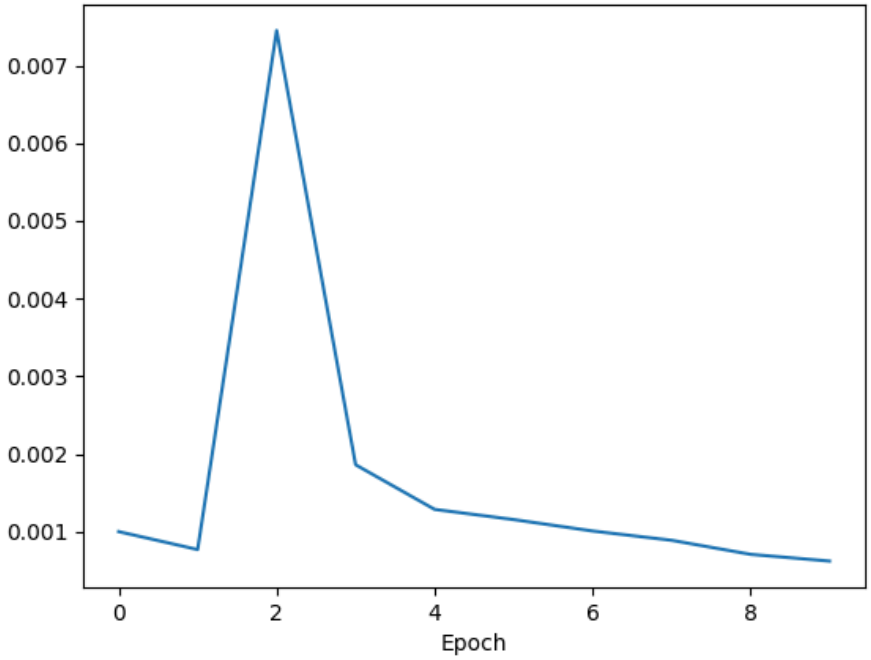


Above is a picture depicting Architecture in U-Net.

Training Curves:-



This is the train loss graph.



This is the test loss graph.

Training Details:-

Epoch Number	Train Loss	Test Loss
1	0.011	0.001
2	0.008	0.0007
3	0.0095	0.007
4	0.0025	0.0018
5	0.0015	0.0012
6	0.0014	0.0011
7	0.0012	0.0010
8	0.0010	0.0008
9	0.0009	0.0007
10	0.0007	0.0006

There is a spike in training and test loss from Epoch 2 to 3 but it reduces for further epochs. Exact log of training in training.log.
PSNR value kept increasing with every epoch.

Some Qualitative Results:-

Above is a blurred image and below is the sharpened image generated by our code.

Another example !

Another example!
