DEEP LEARNING CS60010

ASSIGNMENT 3

FINAL REPORT

GROUP NO 2

NAME	ROLL NO
Writabrata Bhattacharya	22BM6JP58
Manvinder Chahar	22BM6JP25
Shashwat Naidu	20CS10055
Lagnesh T S	22BM6JP22

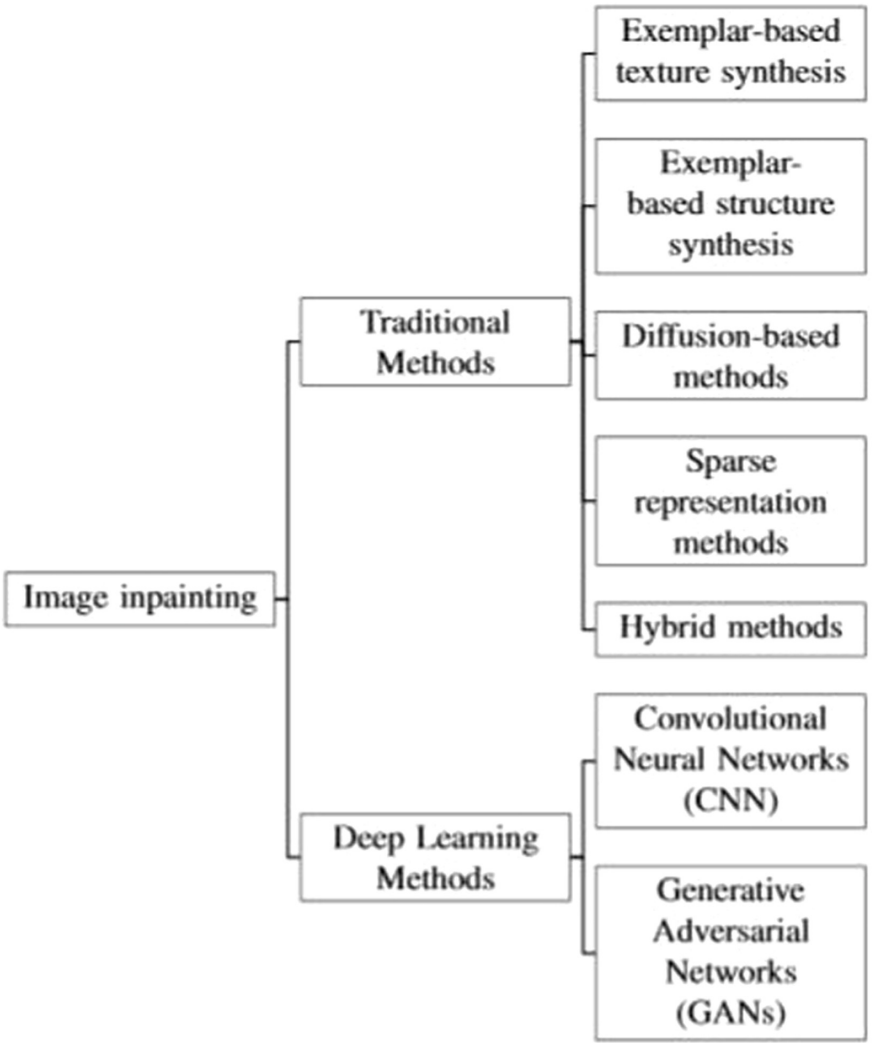
INTRODUCTION

The task was to build models with a generative ability to reconstruct images' lost/damaged parts. Input images were provided with missing sections (masks) in the form of squares of 75 x 75. Each image consists of two such sections and data regarding where this section is located was provided in a CSV file. The output was expected to be the whole image without the missing sections.

We found this task quite relatable to image inpainting. Image inpainting refers to the process of filling in missing data in a designated region of the visual input. Hence we referred to various image inpainting-related research papers and works done in this domain to help us complete the challenge.

DIFFERENT MODELS OF IMAGE INPAINTING

We can categorize the solution to image inpainting by the below flowchart:



The above clearly categorizes the ways to solve image inpainting into broad areas: Deep Learning Methods and Traditional methods. We didn't dwell much into the traditional methods of Image Inpainting as Deep Learning Models are proven to provide better accuracy and results. Though we did use a GAN Based Diffusion model which works on principles similar to a traditional diffusion method. So below we proceed further into the Deep Learning Methods we used:

CNN BASED:

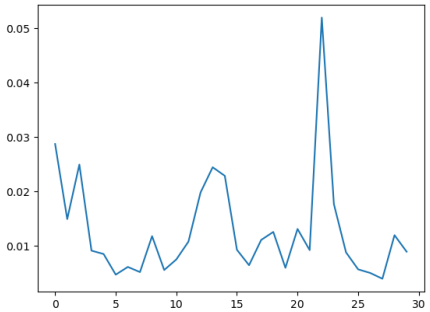
1) **Partial Convolution based model** (**Paper Link**)

Description: Partial convolution is a technique used in image inpainting, which is the process of filling in missing or damaged regions in an image. It is a type of convolution operation that takes into account the presence of masked or missing pixels in the input image during the convolution process.
In traditional convolution, the filter/kernel is applied uniformly to the entire image, including the missing regions, which can result in artifacts and blurring in the inpainted areas. Partial convolution, on the other hand, adaptively applies the filter based on the availability of information in the input image.

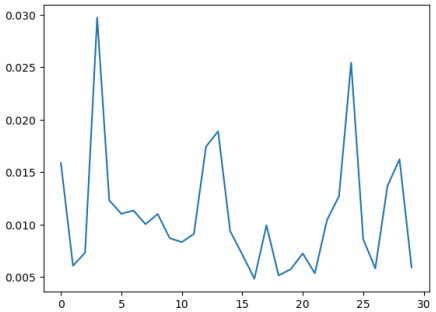
Notebook Link: Code Link for PconvNet

Results: This model gave us our best submission with an RMSE Score of 0.23212 on the private dataset. But we didn't save the checkpoint for this model hence we selected our second-best submission for the leaderboard u can clearly see in the submissions that this was our best submission score-wise.

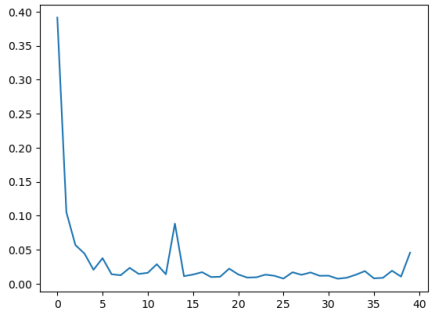
Loss Plots for the above model for various hyperparameters:



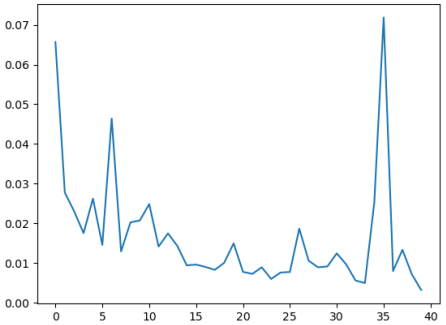
Epoch = 30, lr = 0.0005



Epoch=30, lr = 0.001



Epoch = 40, lr = 0.00005



Epoch = 40, lr = 0.0001

GAN BASED:

1) **Pretrained Stable Diffusion:** (Hugging Face API)

   Description: The pre-trained stable-diffusion-2-inpainting model from Hugging Face is used to predict the masked portions of the image. The Stable Diffusion model is basically a conditional GAN model. It consists of a text encoder, variational autoencoder and U-Net architecture.

   Below is the notebook for running the pre-trained model, we couldn't work towards finetuning this model for further use to our dataset.

   Notebook Link:

   Results: We attained an RMSE Score of 0.2748 on using this model (this was one of our first submissions)

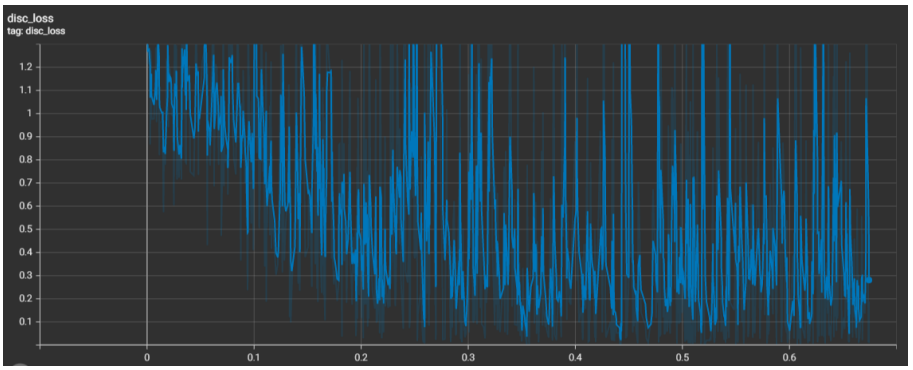
2) **Pix2Pix Model from Scratch:** (Paper Link)

   Description: The Pix2Pix architecture consists of two main components: a generator network and a discriminator network. The generator network takes an input image and generates an output image. The discriminator network takes a pair of images (an input image and a corresponding output image) and classifies whether the output image is a plausible result of the generator network or not. The generator network has U-Net architecture with skip connections between encoders and decoders, while Discriminator has PatchGAN architecture. During training, the generator and discriminator networks are trained in an adversarial manner. The generator is trained to generate realistic output images that can fool the discriminator, while the discriminator is trained to distinguish between real and fake image pairs. The overall objective of the Pix2Pix architecture is the minimization of pixel-wise L1 and adversarial loss.

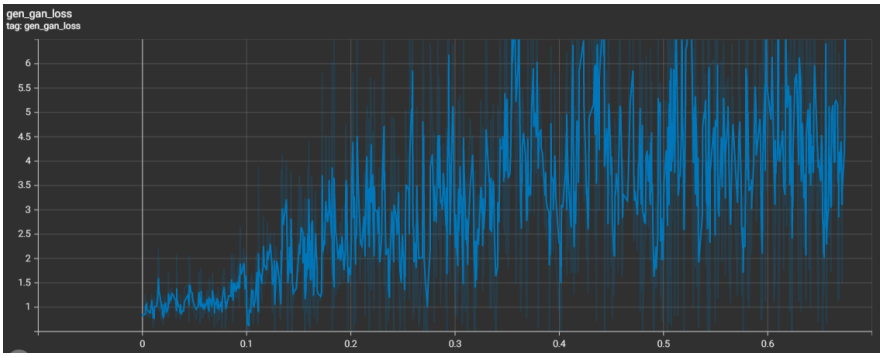
   Inference Notebook Link: Code Link Model Inference, Link to model

   Results: We got an RMSE score of 0.23352 on Private Leaderboard with this model

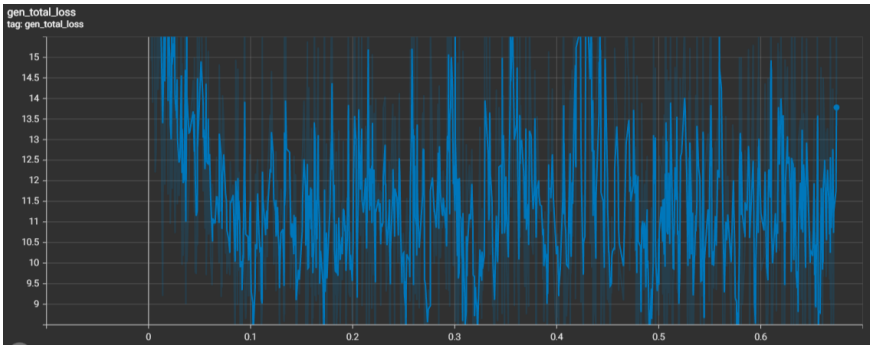
Loss plots:



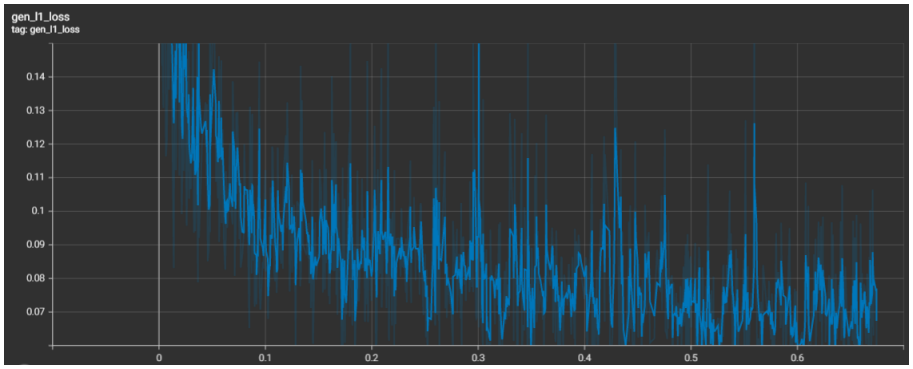
Discriminator Loss



Generator Gan Loss



Generator Total Loss



Generator L1 Loss

DATA PREPROCESSING

- **Data Augmentation:** We tried increasing the training dataset by generating random masks for the ground truth examples and making more masked images. But this gave worse results, maybe too many data points caused the overfitting of the model. We tried this data augmentation on both the CNN-based model as well as GAN model and both the results degraded on using this.

  Link to dataset(This contains 14K training images, all the ground truth images were run once to create a random mask on it): Extra Training Dataset Link
  Code to make it: Making Extra Masks Code

- **Applied Randomness**: Before training the pix2pix model, the images are first resized, and randomly cropped. Then they are horizontally flipped with a probability of 0.5. Finally, the pixel values of the images are normalized to a range between -1 and 1.

- **Mask Image Creation:** For using the pre-trained StableDiffusion model for prediction, an additional input apart from the masked image needs to be created. That input should be an image of the same size as the original image filled/highlighted with white pixels at the positions of the mask and with black pixels at the remaining positions. For this purpose, a function create_mask was created (which was used for the Pconvnet model too).