

# Reflection Essay

(This is a reflection on the things you learnt, it is not the presentation mentioned above. To be more clear, the presentation is the 'answer paper', the video of 10 minutes is a proxy for an interview/viva (you can add it to your portfolio –sometimes it is hard to evaluate a PPT), and the ReflectionEssay is – well, a reflection for what went wrong, how you recovered, half-baked ideas, what-if scenarios, basically anything else that you want to share in a textual form. Think of how we present the answers in our grading guidelines in Tasks 04, 05 etc.). [Delete this before Submission]

## 1 Learning from the project

1. Learnt creating dataloader, model and training the model in PyTorch framework.
2. Learnt the architecture of U-Net. How it can be applied to multiple computer vision applications such as Depth Estimation.
3. Usage of transfer learning in Depth estimation using a U-net type architecture
4. Various Loss functions: DSSIM :  $(1-SSIM)/2$ , structural dissimilarity metric, pixel wise L1 Loss applied to gradients of image, weighted combination of DSSIM and L1 loss.
5. Learnt how to implement Generative Adversarial Networks such as pix2pix in PyTorch and that sometimes simpler networks without adversarial training may perform better than the complex GAN networks.
6. Deep learning based approaches for vision problems does not require a lot of domain knowledge however, can give poor results when used on a different data other than it is trained (Poor generalization properties).
7. We thought that if the depth map was to be considered as probability distribution in 2D, we can use the KL divergence loss to minimize the error. We tried training our proposed networks using the KL divergence loss, but the reconstruction results were much worse. Thus, we inferred that treating the depth map as a probability distribution is not the right approach.

## 2 Things that went wrong

1. We wanted to use Google Cloud credits for training our models because of the large data set requirements and tremendous complexity of the pre trained model, however we were unfamiliar with the usage methods and despite increasing our usage quota, we were unable obtain a google colab-oratory notebook with the desired GPU specification in any of the server zones. We were limited by the compute power, so the team members contrubuted equally to purchase a Google Colaboratory pro account. The training of our models was done on this paid colab account.
2. The intricate changes we wanted to make in the model, and the implementation of a better deep learning pipeline for efficient training of the network was not possible in the interest of time. We spent too much time in understanding the architecture, working of PyTorch, and experimentation with different loss functions and known networks that we couldn't get to that stage.

## 3 Half-Baked Ideas

1. **Pix2Pix** : We wanted to implement a GAN based depth map estimation approach where a discriminator would help the generator network learn the correct depth map without requiring an explicit loss function.