

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## VIII Semester Project

### MONTHLY PROGRESS REPORT - I

Batch No.	35	
Title of the project:	IMAGE REGENERATION USING GENERATIVE MODELS	
Team members:	Abhijith C.	1MV14CS004
	Raghava G. Dhanya	1MV14CS077
	Shashank S.	1MV14CS131
Name of the Guide:	Sushila Shidnal	
Duration:	From sometime to sometime	

#### Details Of Work Carried Out:

Under Basic GAN, we implemented the DCGAN architecture from Alec Radford *et al*<sup>1</sup>. We were able to reproduce the results on the MNIST dataset and on facial images with a high degree of visual accuracy. The training with CelebA dataset which has 200k examples each with 28x28 pixels took around 30 hours to complete 1600 epochs on a modest home computer system.



Figure 1: GAN output after 40000 epochs

Seeing as the training would take an enormous amount of time on standard computing systems we searched for other options. After considering AWS and Google Colaboratory, we decided on Colaboratory for its ease of use. It allows us to implement code in the form of Jupyter notebooks stored on the Google drive. Colaboratory has Nvidia Tesla GPU back-end support and 13 GB of RAM which helps improve the training speeds. For comparison it took 70 minutes to train the same DCGAN on MNIST dataset which has 60000 examples each with 28x28 pixels for 60000 epochs, It is more than 8x speedup even if the number of epochs are ignored. We can see the results in figure 1.

The next step in the process was to implement a vanilla CapsNet. This would act as the base code which we will augment to finally shape it into a GAN. We implemented CapsNet architecture similar to that of in Geoffrey Hinton *et al.*<sup>2</sup>. We were able to reproduce the results but with longer training times.

<sup>1</sup>Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015.

<sup>2</sup>Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 3856–3866. Curran Associates, Inc., 2017.

## Time-line:



**[Completed] Feb Week 3: Basic GAN**  
Implement a vanilla GAN with MNIST data.

**[Completed] Feb Week 4: Basic CapsNet**  
Implement a CapsNet Classifier on MNIST data.

**Mar Week 3: Discriminator using CapsNet**  
Implement a binary CapsNet Classifier and train it as discriminator.

**Mar Week 4: GAN with CapsNet Discriminator**  
Plug the CapsNet discriminator to GAN.

**Apr Week 1: Generator using CapsNet**  
Try to implement CapsNet based generator.

**Apr Week 2: Fully CapsNet based GAN**  
Plug the CapsNet based generator into GAN to create a fully CapsNet based GAN.

**Apr Week 3: Training and testing**  
Train and test the model on face dataset.

**Apr Week 4: Compare results and continue testing**  
Compare the resulting model with current state of the art models.

**May Week 1: Optimize the model**  
Tune the hyper-parameters to improve the model.

**May Week 2: Train final model and start work on GUI**  
Train the final model for face completion and start working on GUI front-end.

---

**Head of the Department**

**Project Guide**

**Project Coordinator**