CNNs vs CapsNet

Group members:

1) Aditya Das (ad3574)
2) Siddhant Gada (scg2151)
3) Chinmay Joshi (caj2163)

Motivation for this project

While CNNs have been used massively for Computer Vision tasks like Object Recognition, Facial Recognition, Image Classification etc, it has several shortcomings. Some of the drawbacks are:

- The classification or prediction made by the CNN is done by taking a picture and checking the presence of certain components in the picture. If it finds these components, then the image gets classified accordingly. For example, assume that a CNN is trained on Faces of humans, for facial recognition tasks. Now, if we show this CNN a distorted image of a person, where the position of the eyes has been switched with the position of the mouth etc, then the CNN would still classify such a distorted image as a human face. This happens because the CNN only checks for the presence of features like eyes, nose, lips etc. But the CNN fails to take the relative positions and locations of such components to each other.
- CNN does not take the position and orientation of the object in an image into consideration when it tries to make a prediction. The main reason why they don't work really well is because they can only make predictions on an image if there is only a slight variance between the train images and the test images. So, if the CNN is trained on images of people sitting on a chair, then they will only be able to identify people in the test images where people are sitting in the same positions, and the CNN will give an incorrect prediction if there is a slight change in the positioning of the people in the train images and test images.
- There is a problem in the way the CNN routes the data from the lower levels (the layers which are close to the input layer) to the higher levels (the layers which are close to the output layer). So, instead of sending the information from all the neurons, it is better to route the part of image to particular neurons which deal with those features. For example, given an image of a face, if a neuron at low levels has learned to identify mouth, then it is logical that this information should be sent to higher level neurons that are good at recognizing images containing faces, and not images containing cars. So, by routing all of the information from the lower level to the higher levels, we are increasing the

computation cost as well as sending data to the neurons which are not made to learn a particular feature.

So, to overcome these drawbacks, researchers came up with a new Deep Learning architecture called Capsule Networks. CapsNet have been touted to solve all of these problems, and through this project we want to confirm if CapsNet can actually solve those problems, and if not, then how does it compare with CNNs for the same test case.

Research Questions

Through this project we want to answer the following questions:

1) How much more time does CapsNet take to train, as compared to CNNs?
2) Can CapsNet perform better than CNNs when tested on rotated images?
3) Can CapsNet perform better than CNNs when tested on images which have been modified by translating objects in it?
4) Can CapsNet perform better than CNNs when tested on images whose pixel values have been manipulated?
5) Can CapsNet do better image classification than CNNs on large datasets like CIFAR10?
6) Does CapsNet have the ability to replace CNNs in today's world?

Data used

For this project we have used image datasets which are publicly available. The datasets include:

1) MNIST Dataset: This dataset consists of images of handwritten digits from 0-9. The images in this dataset are grayscale, and have the dimension of 28x28. The link to this dataset is: http://yann.lecun.com/exdb/mnist/index.html
2) CIFAR10 Dataset: This dataset consists of images having 10 different classes. The classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. These images are color images, each having dimensions of 32x32.
   The link to this dataset is: https://www.cs.toronto.edu/~kriz/cifar.html

Group Members Tasks

1) Aditya Das: Since I have a lot of experience in the domain of Image Processing and Computer Vision, I am working on writing code which can be used to generate the test images to test the CNNs and CapsNet on. This task will be done using image manipulation libraries in Python like PIL, matplotlib etc. I will also be helping Chinmay Joshi with developing CNNs for this project. I have published a research paper in a International Conference, and the project for the research paper was accomplished using a CNN. So, I believe I can have a major contribution for developing this project.

2) Chinmay Joshi: I have previously taken a Deep Learning class at Columbia University under Prof Iddo Drori, so I have a lot of experience in developing Neural Network models, and especially CNNs. So, my major contribution for this project will be to create and implement necessary CNNs for testing purposes. We plan to use some popular CNNs architectures like VGG16, Inception, ResNet50. Training these CNNs on the aforementioned datasets will be a huge task as we will need to preprocess all the images present in the dataset before we supply it to the CNNs. I will also be configuring the Virtual Instance on Google Cloud Platform, as we will be needing a GPU to train our CapsNet and CNNs.
3) Siddhant Gada: I will be focussing on making the CapsNet and making it run. I have great knowledge about CapsNet as I had researched about them in great depths to come up with my midterm paper for this course. I will be making and modifying the CapsNet to train the two datasets of MNIST and CIFAR10. The entire preprocessing the and the inputs for the CapsNet will change as there are differences in the images between both the datasets. Training both these datasets on the CapsNet will be a notorious task as it can easily take more than 20 hours to train. I will also be analyzing how the CapsNet performs when being presented with the test images which will be created by Aditya Das.

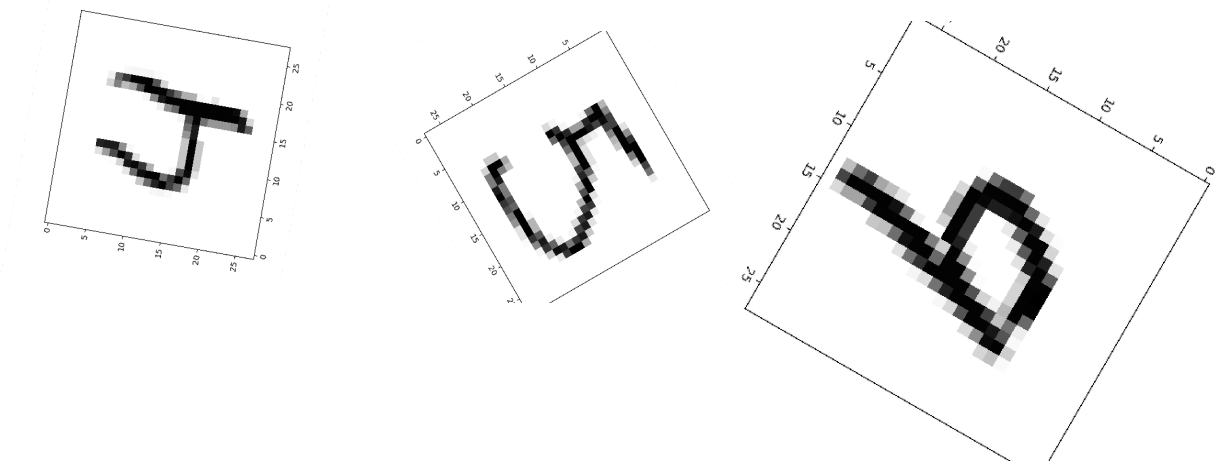
What have we achieved till now?

1) Developed a few testing images based on Mnist.
2) Training of the CapsNet on Mnist is still in progress
   (CapsNet based on https://www.kaggle.com/kmader/capsulenet-on-mnist)
3) Code for various CNN models has been completed
   (Using pre trained VGG, Inception etc. (https://keras.io/applications/))
4) Created and configured instances on GCP.

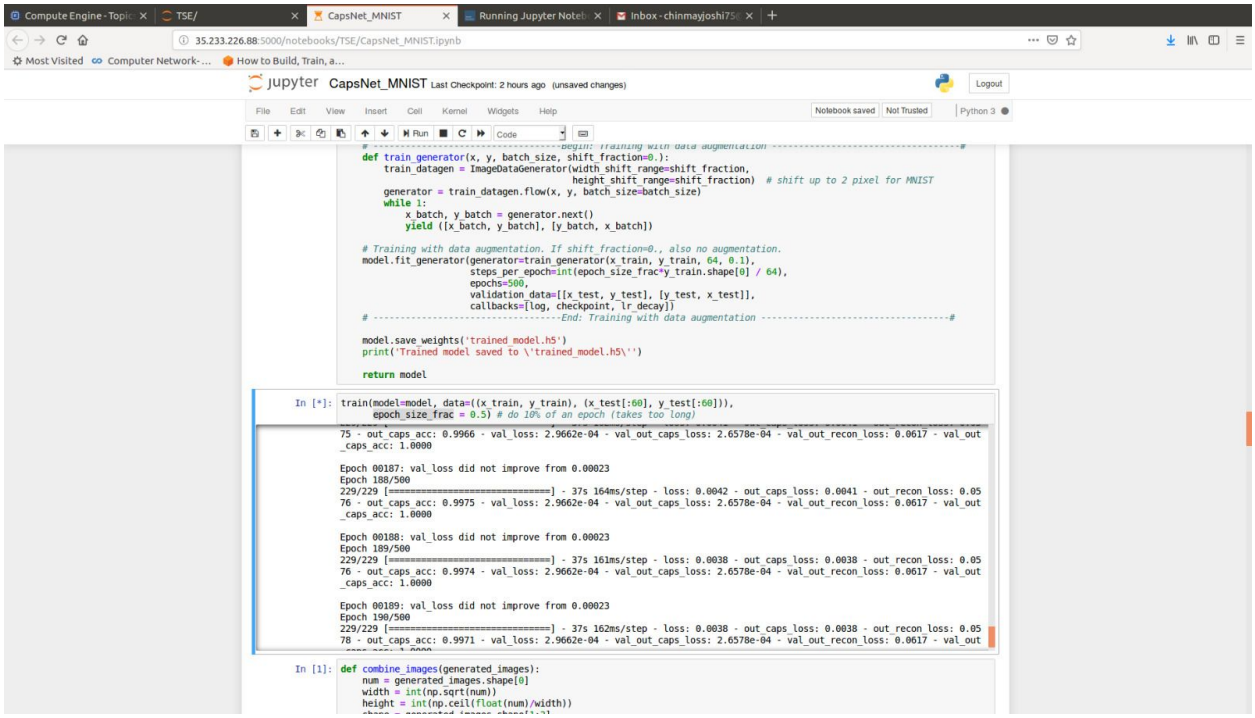
Future Milestones?

1) Create more testing data based on Mnist and Cifar-10
2) Train the CapsNet on Cifar-10
3) Train the various CNN models which we have developed
4) Test the CNN and CapsNet models on our test images.
5) Analyze the performance of CapsNet and CNNs
6) Develop Inferences
7) If possible : Implement it for other common datasets

Screenshots



Screenshot of some test images which we generated



Screenshot of CapsNet running on Google Cloud Platform

Deliverables

Code for all the implementation, image processing etc via github

Graph and analysis using a PPT

Demo using python scripts