

# Assignment 3

## Convolutional Neural Network

Name : Eric Ly

NetID : ehl7

Github Link : [https://github.com/keshavsbhandari/CS4347/tree/master/assign3\\_convolution\\_nn](https://github.com/keshavsbhandari/CS4347/tree/master/assign3_convolution_nn)

### Architecture Details:

For this project, I used my main computer at my apartment to do the testing. As doing these project model runs on my laptop would have taken an entire month probably to even finish a few testing's. The computer has a i7 8700k, I really wanted to get the tensorboard to run using the GPU as that would have significantly reduce the time of running each test and training methods. But I never got it to work so I just stuck with the CPU tensorboard. I had attempted to use the Google Colabotory in order to achieve faster results but was out of my scope of abilities to setup.

### Experiment Results:

This report was created and observed when running the kerasmain101.py using Jupyter-Lab, Python 3 and CPU processing to get the results. All of the testing results can be viewed at the [link](#).

### Learning Rates:

- Messing with learning rates can have a profound impact. I've changed it with the default parameters all the same and set the learning rates from factors of 10.
- So (0.0001 to 0.00001). This gave an increase in accuracy percentage by a factor of nearly 10%.  
Kerasmain101 learning\_rate: 0.001 -> **47.97%**  
Kerasmain101 learning\_rate: 0.0001 -> **54.29%**

**Optimizers:** To yield a more informative results of the training I had used several different optimizers that the Keras protocol provides.

### RMSProp:

- The default RMSProp yielded me a **47.97%** accuracy when defaulted to the original parameters given by the project.

#### Changing Epoch 10 -> 2

Doing this had a significant impact in decreasing the overall accuracy average to being **39.15%**. So, a negative gain as this would require many more runs to train the model to improve the accuracy.

#### Changing Epoch 10 -> 5

Doing this had a significant impact in decreasing the overall accuracy average to being **45.16%**. So, a slight decrease in terms of gaining a better average accuracy when compared to the original Epoch being 10.

#### Changing Epoch 10 -> 100

Doing this would have had a greater impact but would take ages to run as the Epoch was so much higher compared to the original 10. I had only run it up to 20 epochs before aborting as it would have taken a

significant time to finish the model. However, with only running up to 20 epochs. The results were already getting better. This was only after 1 run to train the model. The average accuracy was already at **47.24%**. So running this Epoch at 100 and running it 3 times would have given us a significant improvement but also a significant usage of time.

### **Changing the Horizontal Flip to a Vertical Flip:**

- After editing the **ImageDataGenerator** and changing the Horizontal flip to a **Vertical flip**. The results yielded a **45.26%** so a decrease when compared to the original model.

**Note :** I had changed the Horizontal Flip to a **Vertical Flip** and **never changed it back**.

### **SGD:**

- After having a decent idea what the training was going to give. I changed the optimizer to SGD and this model yielded a measly **19.14%**. The worst one out of all the different testing's I did.

### **Adagrad:**

- After 3 runs and loading the model the accuracy for this optimizer only gave average accuracy of **31.48%**. Which compared to everything it is one of the lowest results. So definitely not suited for this particular training.

### **Adadelta:**

- Of the different optimizers I used during the testing for this project. This yielded the highest result of the different optimizers used. It had accuracy average of **51.39%**.

### **Train Size:**

- When increasing the training size by a factor of 2. There was a huge increase in the overall accuracy but increasing the training size to anything else provided little improvements.
- Average Accuracy of **38%**.

### **Final Results:**

- Running these models and test proves that there is a key factor in optimization needed to achieve better results. However, to achieve these results many tests have to be ran and compared in order to find a better approach. To achieving a higher accuracy.
- Doing this type of work on a regular dualCore laptop proved futile for the laptop as it would have taken ages to process. Even when running on the i7 8700k these testing would take upwards of an hour for each run alone. That's also disregarding changing the model to use a different parameter or optimizer.