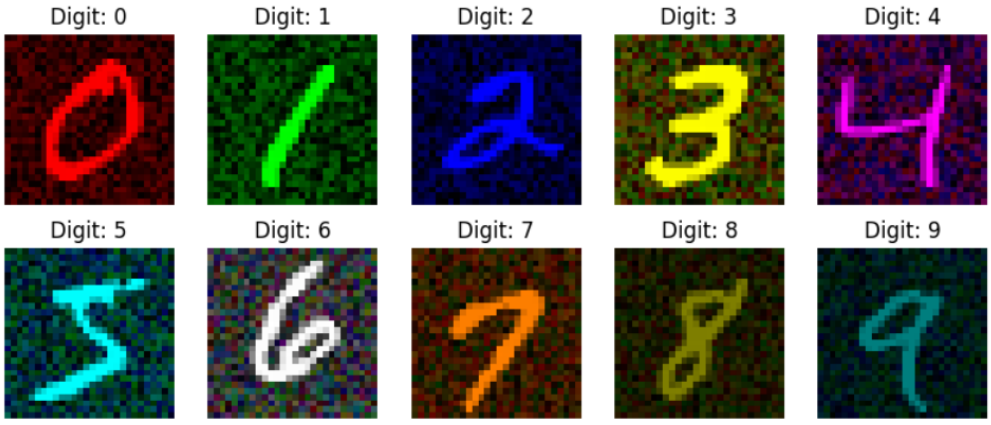
CNN-Tasks

Task-0:

First an MNIST Dataset is downloaded from online and then coloured using a class coloured MNIST where the foreground of the digits are coloured with 0-9 follows:

```
[1,0,0], [0,1,0], [0,0,1], [1,1,0], [1,0,1],
          [0,1,1], [1,1,1], [1,0.5,0], [0.5,0.5,0], [0,0.5,0.5]
```

The background is coloured with noise related to that.



Task-1:

A simple 3-layer CNN is implemented first it is convolved with 32 different filters and then pooled with max of 3*3 to 1 value.
In the second convolution 32 are used to get 64 and then 128 in the third.
Final feature map will have 128 feature maps of 2*2 size.

80% of the easy set used for training, 20% for validation the accuracies obtained
Hard test with random colours is checked

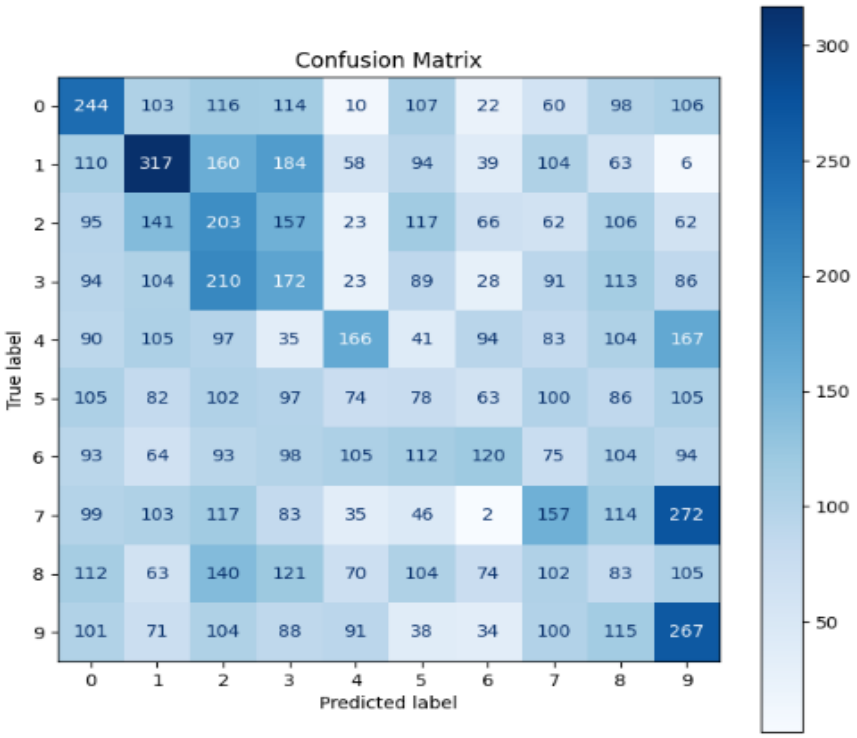
Accuracies obtained:

Train: 95-97%
Validation: 95-96%
Test: 15-20%

The expected results are achieved.
The model is more trained toward seeing the colour, its accuracy is less and the model is trained on colour would be proved in Task-3 using grad-cam heat map.

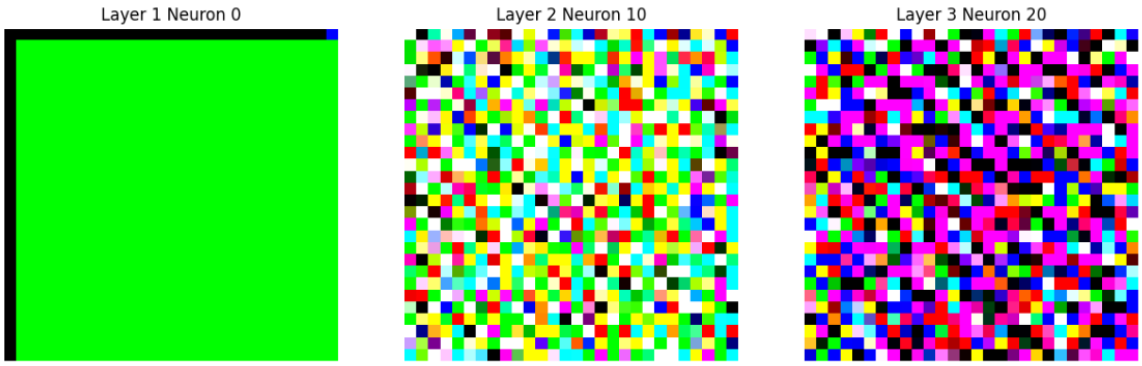
When the red ‘ 1 ’ is given input the model predicted ‘ 0 ‘ because of the colour.

The confusion matrix is below



Task-2:

For analysing how each neuron is observing at each stage we use hook function which can save the images or values that are generated and deleted in between at each layer we use a hook function to jump and the images at 3 layers look like below:



From the image it can be observed that the first layer is mostly observing the colour part and the second is observing different colours. It is going to more complex features like colour transition at block and other features and the third looks into more complex features it responds to shapes, fragments and other patterns.

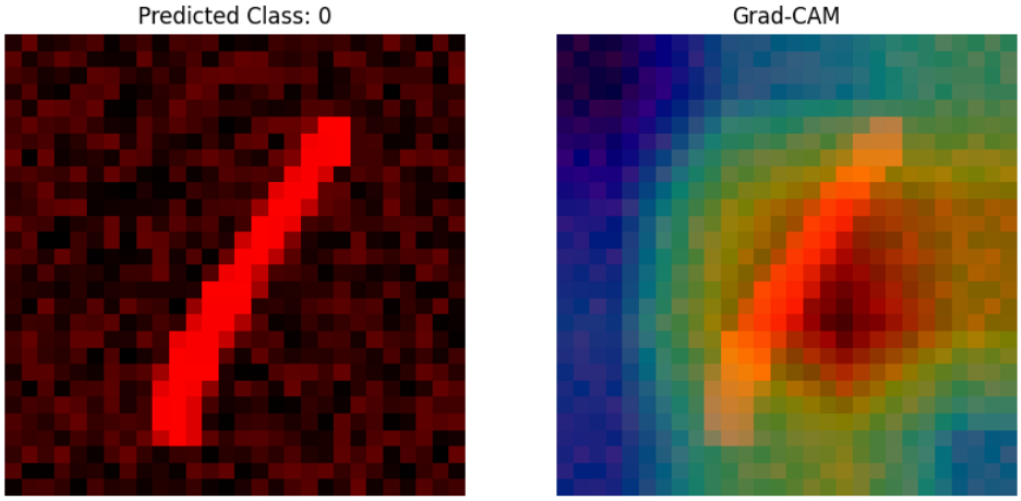
Task -3:

Here the grad cam is implemented from which a heatmap is generated from giving input and in the heatmap it has dense colour on the things the model focussed to give that result which can be obtained by using the below function.

$$Lc(gradcam) = ReLU(\sum_{k} \alpha^{c}_{k} A^{k})$$

α^{c}_{k} is the gradient of channel k with respect to the c which is the highest logit.

For calculating gradients hook function is used again and the gradients and the feature maps are saved and the gradients are calculated with respect to the highest logit (predicted number). These gradients indicate how sensitive the class score is to each feature map activation.
Grad-CAM then forms the heatmap by taking a weighted sum of the feature maps using these weights and ReLU is applied to keep only positive contributions.



In the image it is focusing on the red colour that is why it predicted 0.

Task-4:

Now correcting the model such that it gives good accuracy on hard tests two methods are implemented details are given below.

Method-1

Colour penalty:
While the model is training for a given image not only that colour but same image with grey colour is also given for training then while calculating loss function it calculates what are the common features in both the images which helps the model to learn about the shape.
In this way the model can be corrected.

Train Accuracy: 97.08958333333334
Validation Accuracy: 96.95833333333333
Hard Test Accuracy: 83.64

Method-2

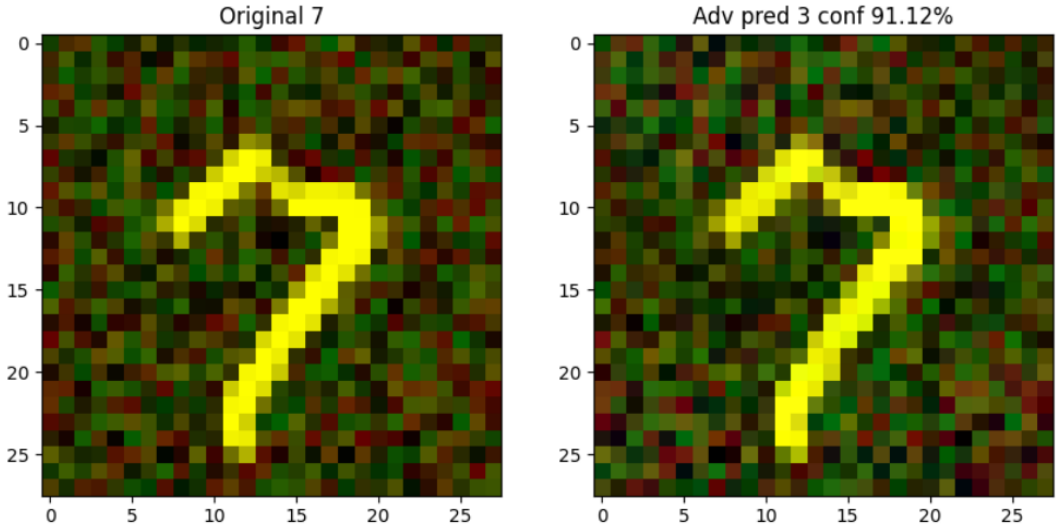
Number of epochs are increased to 5 and the it is observed in task-2 that the first layer sees for colour so a Dropout is kept for 0.5 to increase accuracy on the shape not on the colour.

Training Accuracy: 99.01%
Validation Accuracy: 98.99%
Hard Test Accuracy: 86.60%

Task-5:

In this an attack code should be implemented such that the model will answer 7 as 3 with 90% confidence like that a function needs to be written.
Now gradients are calculated with respect to 7 and using these gradients, pixels are changed such that the change is <=0.05 and then

the confidence for 3 is observed and when it goes to 0.9 if the sum of confidence for all numbers scaled to 1 it works.
The image shows both as 0.05 is so small that the human eye cannot estimate the change.



Task-6:
To prove that model depends only on colour first the neuron activation numbers are obtained while training and then the neurons which determine the number most is taken and kept to 0 to check will the model change the result and it changes this proves that this neuron is triggering the colour so it is proved that result is being based on some specific neurons.