

# Assignment -01

## Convolutional Neural Network

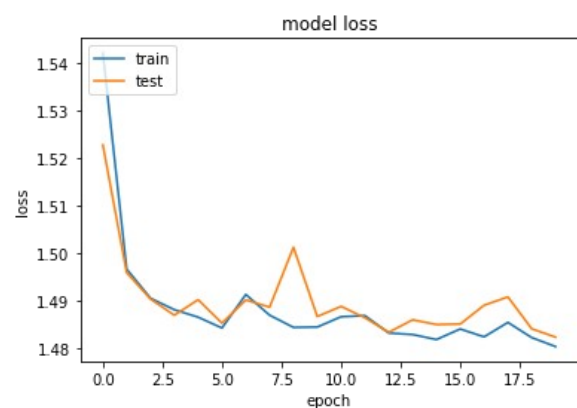
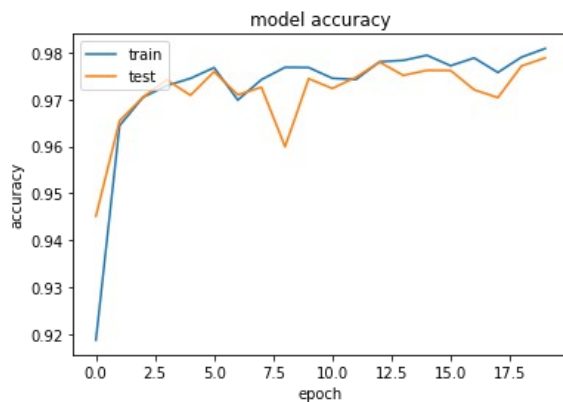
### Q3. (Part-1)

**Making a CNN model from scratch to classify the images of the line dataset into the respective 96 classes and MNIST dataset into 10 classes**

CNN Network
1. 7x7 Convolutional Layer with 32 filters and stride of 1.
2. ReLU Activation Layer, Batch Normalization Layer.
3. 2x2 Max Pooling layer with a stride of 2.
4. Fully connected layer with 1024 output units.
5. ReLU Activation Layer.
6. Adam optimizer.
7. Categorical crossentropy loss.

### MNIST Dataset result (60-40 train test split)

**Training and testing accuracy and loss curve:-**



**Accuracy : 97.89%**

**Baseline Error: 2.11%**

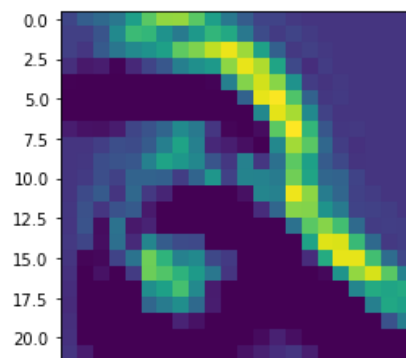
## Confusion matrix:-

```
tf.Tensor(
[[2665    1    0    0    6    2   15    1   15    7]
 [    0 3118    5    2    5    2    0    4    2    0]
 [    6    15 2765   28    6    0    0   13   22    5]
 [    1    4    6 2726    1   34    0   14   12   10]
 [    1    4    0    0 2739    0    4    4    6   14]
 [    2    1    2    5    0 2466   20    0   12    7]
 [    4    7    0    0   11    6 2730    0   10    0]
 [    1   13   20    3    6    1    0 2835    9   23]
 [    7   10    3    6    8   10    6    3 2613   14]
 [    5    6    0    3   43    7    1    7   13 2751]], shape=(10, 10),
dtype=int32)
```

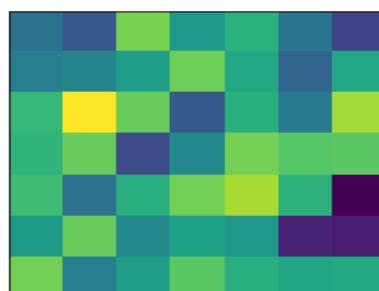
## Precision, recall, Fi-score:-

	precision	recall	f1-score	support
0	0.99	0.99	0.98	2712
1	0.98	0.99	0.99	3138
2	0.99	0.97	0.98	2860
3	0.98	0.97	0.98	2808
4	0.97	0.99	0.98	2772
5	0.98	0.98	0.98	2515
6	0.98	0.99	0.98	2768
7	0.98	0.97	0.98	2911
8	0.96	0.97	0.97	2680
9	0.97	0.97	0.97	2836

## Visualizing Convnet feature:-



## Visualizing Convnet Filters:-

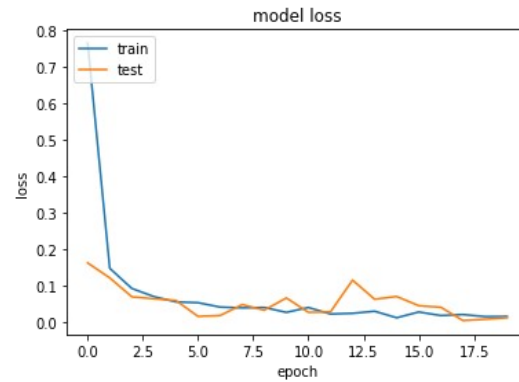
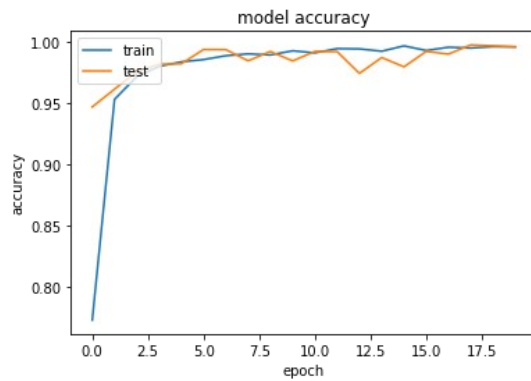


## Line dataset (60-40 train-test split)

**Accuracy : 99.63%**

**Baseline Error: 0.37%**

### Training and testing accuracy and loss curve:-



### Confusion matrix:-

```
[[383    0    0 ...    0    0    0]
 [  0  414    0 ...    0    0    0]
 [  0    0  404 ...    0    0    0]
 ...
 [  0    0    0 ...  386    0    0]
 [  0    0    0 ...    0  390    0]
 [  0    0    0 ...    0    0  408]]
```

### Q3. Part-2

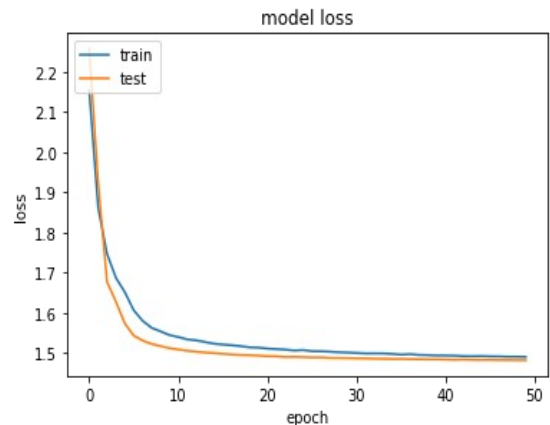
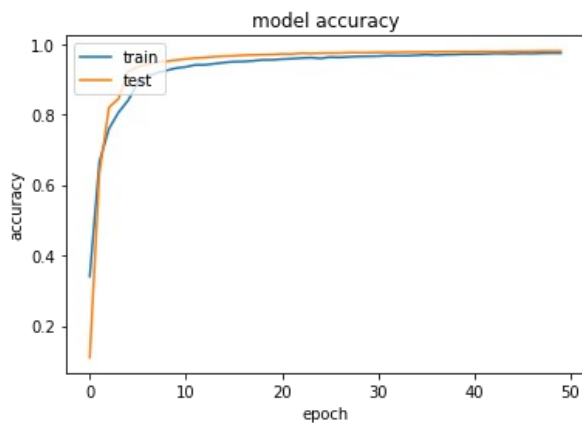
**Create own network architecture :-** Here we are coming with our own network architecture.

**MNIST Dataset:-** MNIST dataset have total 10 classes. So we create a network in such a way that these 10 classes are correctly classified with maximum classification accuracy.

**Variation 1:-**

Hyperparameters	
Epochs	50
Optimizer	SGD
Learning rate	0.001
Dropout	0.5
Batch size	200
Training & testing (60-40)	42000 & 28000
Conv layer + FC layer	3 conv + 2 Fc

**Training and testing accuracy and loss curve:-**



**Accuracy : 98.01%**

**Baseline Error: 1.99%**

### Confusion matrix:-

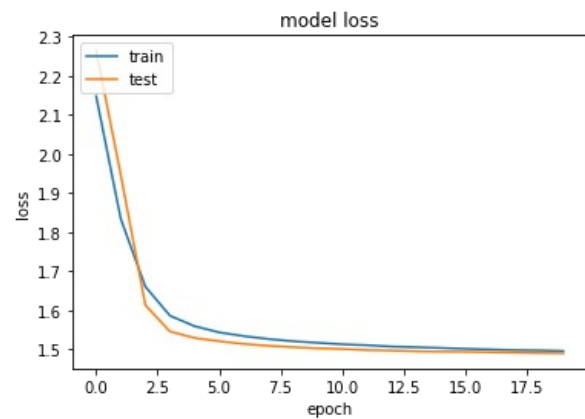
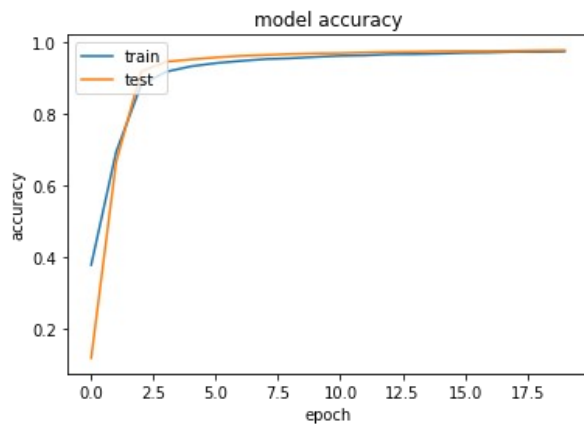
```
tf.Tensor(
[[2709  3  4  0  1  8  5  4  7  4]
 [  0 3037 11  4  0  2  1 12  3  0]
 [  5  5 2776 12  6  2  1 45 11  4]
 [  1  2 16 2786  0 13  0 18  4  7]
 [  6  2  2  0 2683  0 19  0  7 25]
 [  6  1  4  6  4 2472 11  4 10  2]
 [  9  4  0  0  6 10 2663  0  2  2]
 [  4  6 23  6  5  1  0 2887  2 13]
 [  1  1  7  1  8 12 13  7 2736 14]
 [  8  3  0  2 13 19  1 15  9 2694]], shape=(10, 10),
dtype=int32)
```

Classes	Precision	Recall	F1-score	Total no. Of images per class
0	0.99	0.99	0.99	2745
1	0.99	0.99	0.99	3070
2	0.98	0.97	0.97	2867
3	0.99	0.98	0.98	2847
4	0.98	0.98	0.98	2744
5	0.97	0.98	0.98	2520
6	0.98	0.99	0.98	2696
7	0.96	0.98	0.97	2947
8	0.98	0.98	0.98	2800
9	0.97	0.97	0.97	2764

## Variation-2 :-

Hyperparameters	
Epochs	20
Optimizer	SGD
Learning rate	0.001
Dropout	0.25
Batch size	200
Training & testing (60-40)	42000 & 28000
Conv layer + FC layer	3 conv + 2 Fc

## Training and testing accuracy and loss curve:-



**Accuracy : 97.64%**

**Baseline Error: 2.36%**

## Confusion matrix:-

```
tf.Tensor(
[[2750  2  6  0  7  3  8  0  9  5]
 [  0 3143 15  2  1  1  1 10  1  1]
 [  2  6 2736 15  5  1  3 33 11  2]
 [  0  6  23 2808  0  9  0 25  7  7]
 [  4  3  0  0 2671  0 12  1  7 22]
 [  4  6  1 12  3 2494  1  4  4  3]
 [ 13  3  3  0 11  7 2701  0  3  1]
 [  4  6 27 13  9  0  0 2847  1  8]
 [  2  5  7  3 13 15  8  1 2670 22]
 [ 16  2  0  3 14 19  1 10  8 2608]], shape=(10, 10),
dtype=int32)
```

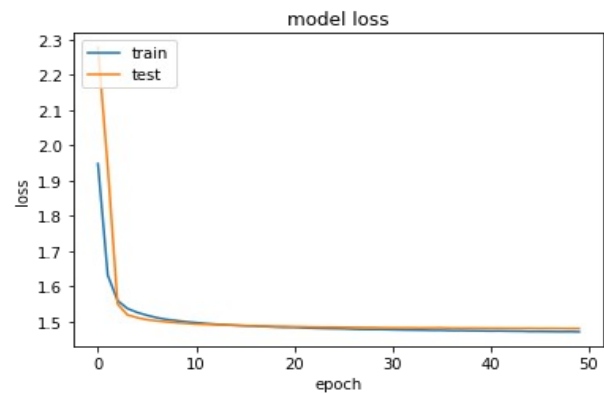
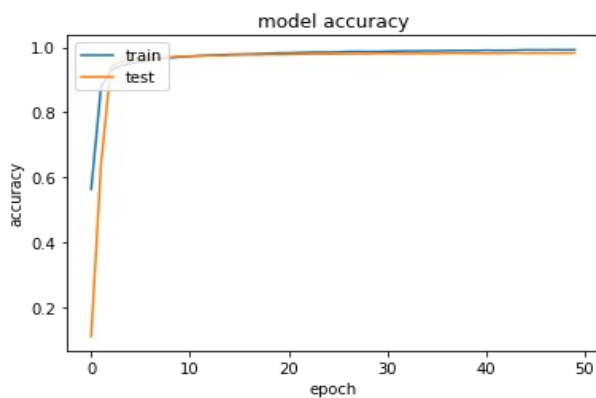
**Precision, Recall and F1-score of each class:-**

<b>Classes</b>	<b>Precision</b>	<b>Recall</b>	<b>F1-score</b>	<b>Total no. Of images per class</b>
0	0.98	0.99	0.98	2745
1	0.99	0.99	0.99	3070
2	0.97	0.97	0.97	2867
3	0.98	0.97	0.98	2847
4	0.98	0.98	0.98	2744
5	0.98	0.98	0.98	2520
6	0.99	0.99	0.99	2696
7	0.97	0.98	0.97	2947
8	0.98	0.97	0.98	2800
9	0.97	0.97	0.97	2764

### Variation-3 :-

Hyperparameters	
Epochs	50
Optimizer	SGD
Learning rate	0.001
Dropout	0.25
Batch size	200
Training & testing (60-40)	42000 & 28000
Conv layer + FC layer	3 conv with varying no. of filters + 2 Fc

### Training and testing accuracy and loss curve:-



**Accuracy : 98.28%**

**Baseline Error: 1.72%**

### Confusion matrix:-

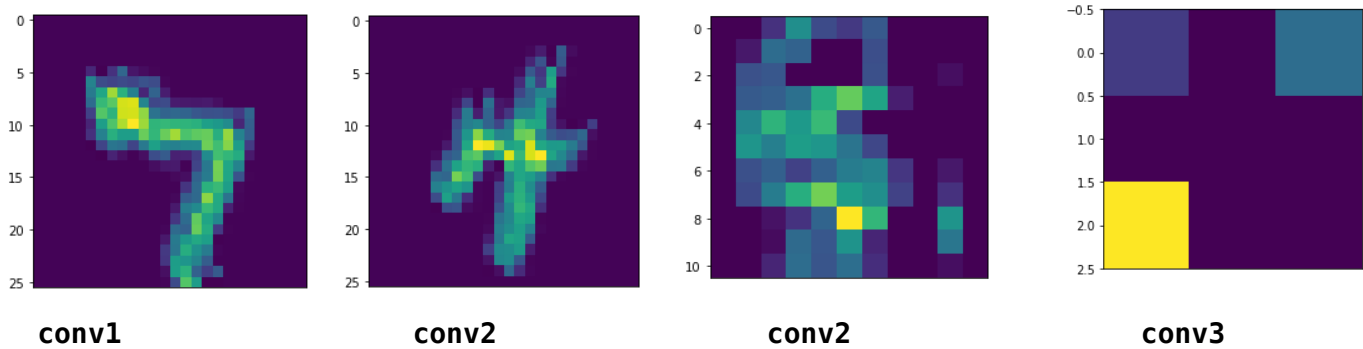
```
tf.Tensor(
[[2725  1  0  4  2  1  1  2  4]
 [ 1 3076 10  1  1  0  0  6  0]
 [ 1  2 2830  5  2  0  2 16  3]
 [ 0  0 12 2790  0  9  0  6  2]
 [ 2  1  0  0 2779  0  3  1  1]
 [ 2  0  1  3  5 2544  3  2  5]
 [ 4  4  0  0  3  7 2679  0  2]
 [ 0  4 11  2  1  0  0 2863  0]
 [ 1  3  3  1  2  4  2  1 2744]
 [ 6  1  0  1  8  2  0  6  4]
 dtype=int32), shape=(10, 10),
```



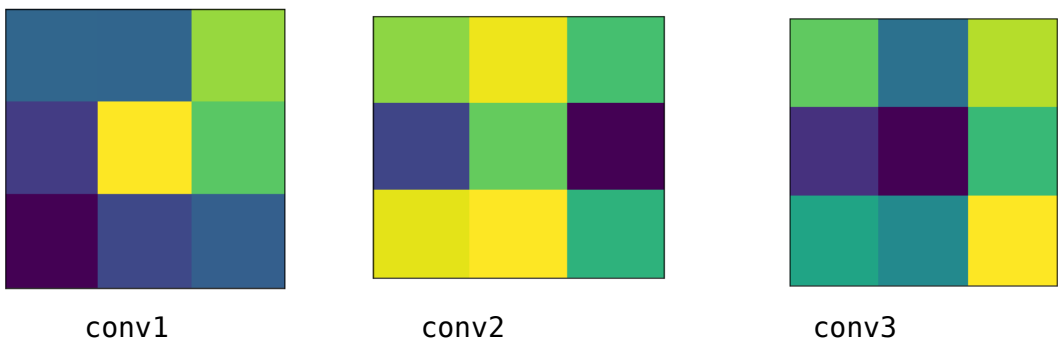
**Precision, Recall and F1-score of each class:-**

classes	precision	recall	f1-score	No. Of images
0	0.99	0.99	0.99	2740
1	0.99	0.99	0.99	3095
2	0.99	0.99	0.99	2862
3	1.00	0.99	0.99	2825
4	0.99	0.99	0.99	2796
5	0.99	0.99	0.99	2565
6	1.00	0.99	0.99	2699
7	0.99	0.99	0.99	2882
8	0.99	0.99	0.99	2772
9	0.99	0.99	0.99	2764

**Visualizing Intermediate Layer Activations:-**



**Visualizing Convnet Filters:-**

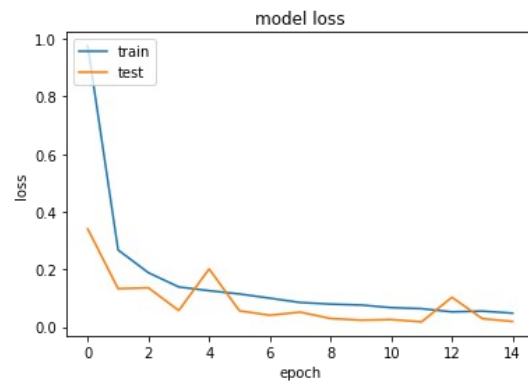
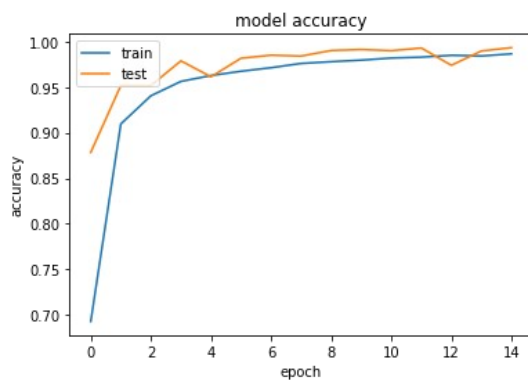


## LINE Dataset

### Variation1:-

Hyperparameters	
Epochs	15
Optimizer	Adam
Learning rate	0.001
Dropout	0.25
Batch size	200
Training & testing (60-40)	42000 & 28000
Conv layer + FC layer	2 conv + 2 Fc

### Training and testing accuracy and loss curve:-



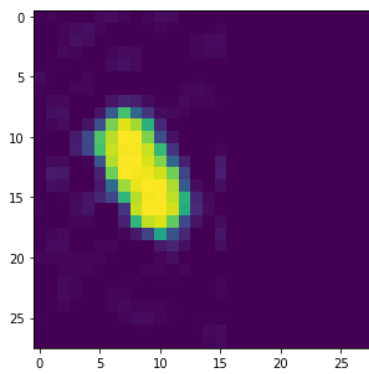
**Accuracy : 99.34%**

**Baseline Error: 0.66%**

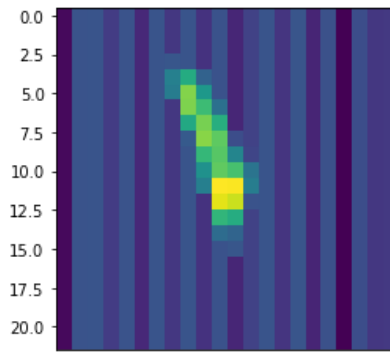
### Confusion metrix:-

```
[[376  0  0 ...  0  0  0]
 [  0 407  0 ...  0  0  0]
 [  0  0 404 ...  0  0  0]
 ...
 [  0  0  0 ... 386  0  0]
 [  0  0  0 ...  0 390  0]
 [  0  0  0 ...  0  0 408]]
```

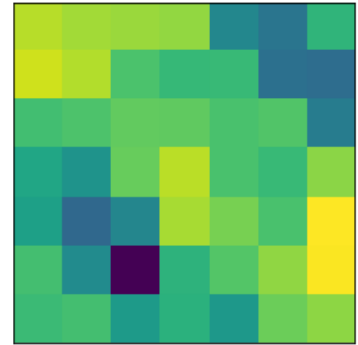
## Visualizing Intermediate Layer Activations:-



Input image



Conv Feature

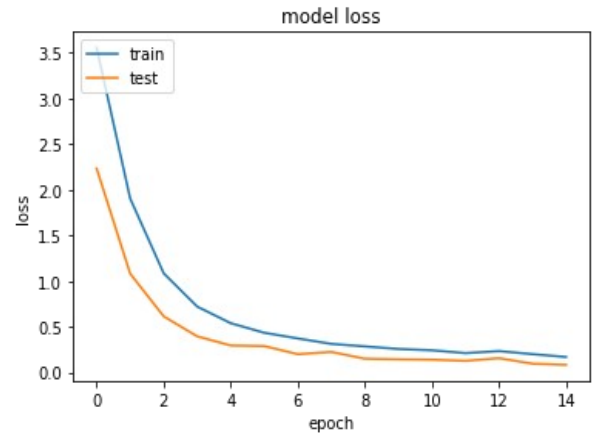
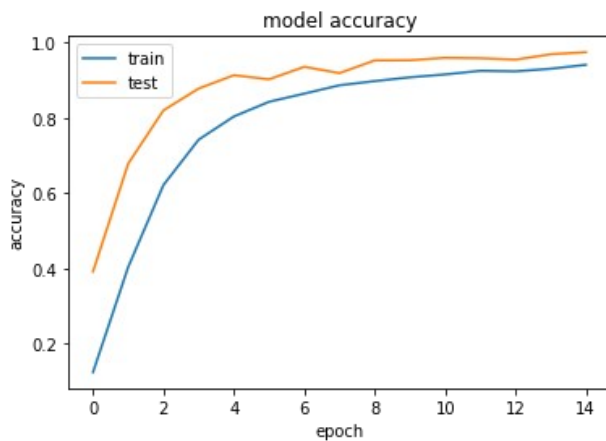


Filters

## Variation2:-

Hyperparameters	
Epochs	15
Optimizer	Adam
Learning rate	0.001
Dropout	0.5
Batch size	200
Training & testing (60-40)	57600 & 38400
Conv layer + FC layer	2 conv + 2 Fc

## Training and testing accuracy and loss curve:-

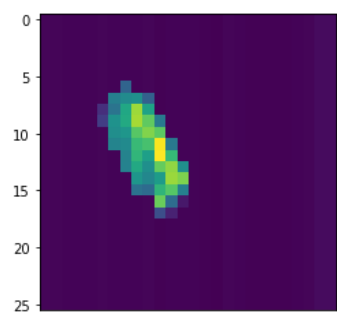


**Accuracy : 97.37%**  
**Baseline Error: 2.63%**

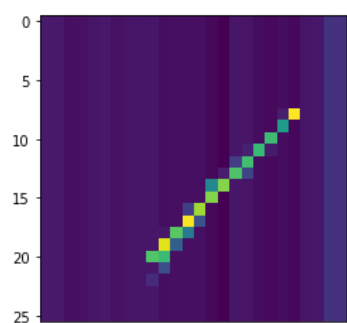
**Confusion matrix:-**

```
[[383    0    0 ...    0    0    0]
 [   0 394    0 ...    0    0    0]
 [   0    0 404 ...    0    0    0]
 ...
 [   0    0    0 ... 386    0    0]
 [   0    0    0 ...    0 390    0]
 [   0    0    0 ...    0    0 408]]
```

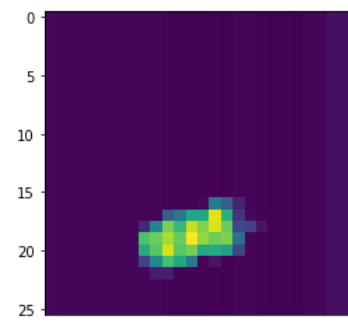
**Visualizing Intermediate Layer Activations:-**



conv1



conv1



conv1

**Visualizing Convnet Filters:-**



Conv1(filter14)



Conv1(filter14)

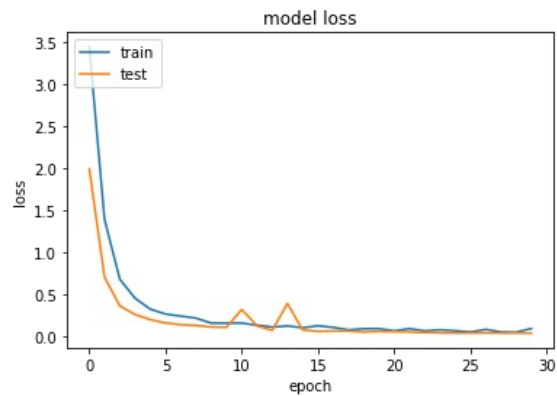
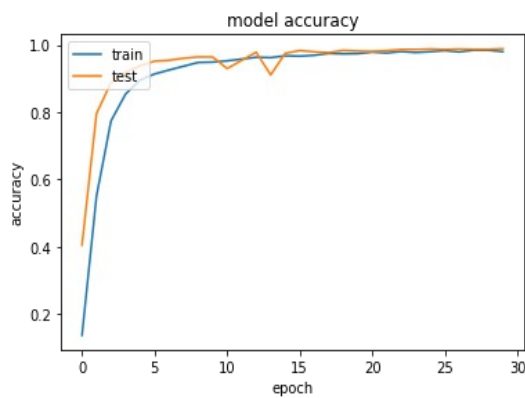


Conv1(filter14)

### Variation3:-

Hyperparameters	
Epochs	15
Optimizer	SGD
Learning rate	0.001
Dropout	0.25
Batch size	200
Training & testing (60-40)	57600 & 38400
Conv layer + FC layer	3 conv + 2 Fc

### Training and testing accuracy and loss curve:-



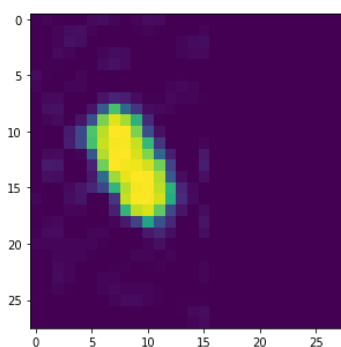
**Accuracy : 98.83%**

**Baseline Error: 1.17%**

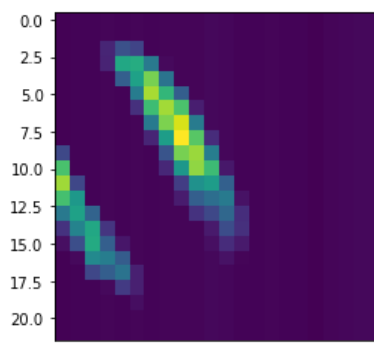
### Confusion Matrix:-

```
[[383    0    0 ...    0    0    0]
 [   0 399    0 ...    0    0    0]
 [   0    0 404 ...    0    0    0]
 ...
 [   0    0    0 ... 386    0    0]
 [   0    0    0 ...    0 390    0]
 [   0    0    0 ...    0    0 408]]
```

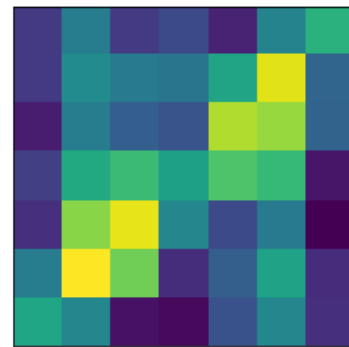
### Visualizing Intermediate Layer Activations:-



**Input image**



**Conv Feature**

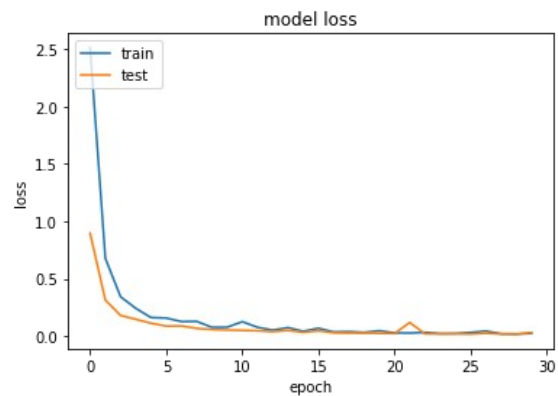
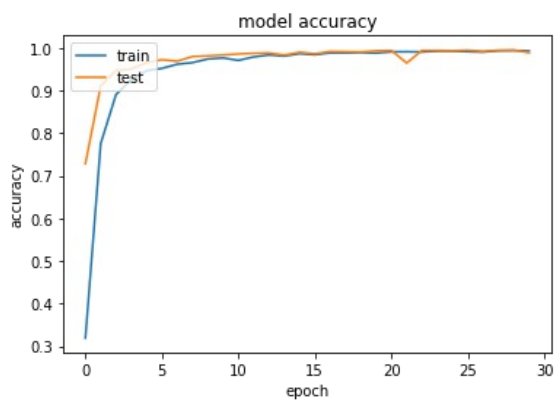


**Filter**

## Variation4:-

Hyperparameters	
Epochs	15
Optimizer	Adam
Learning rate	0.001
Dropout	0.25
Batch size	200
Training & testing (60-40)	57600 & 38400
Conv layer + FC layer	3 conv + 2 Fc

## Training and testing accuracy and loss curve:-



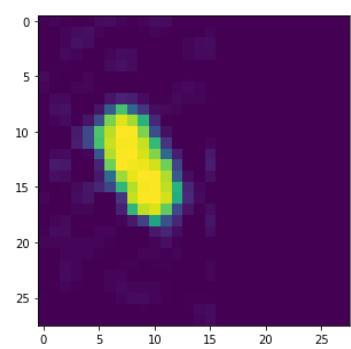
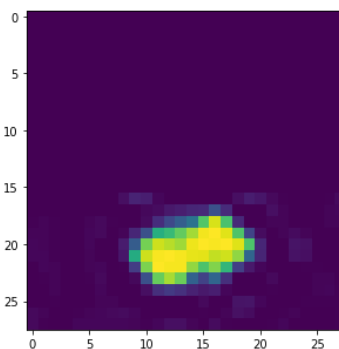
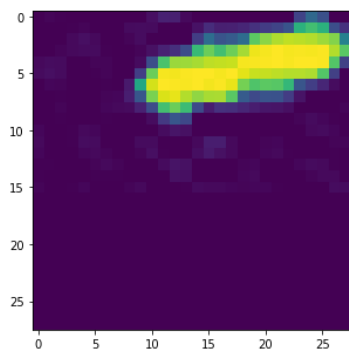
**Accuracy : 98.89%**

**Baseline Error : 1.11%**

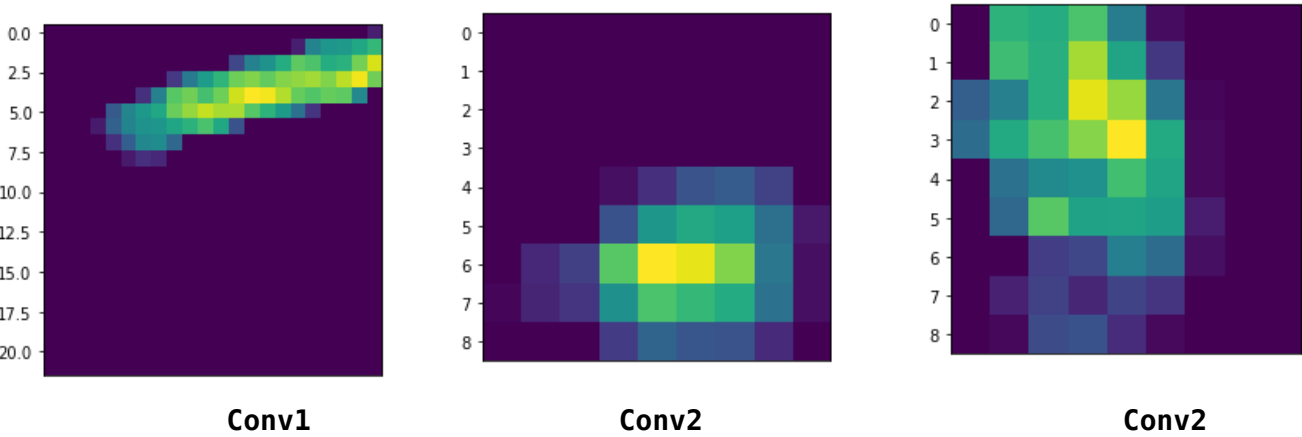
**Confusion matrix:-**

```
[[382   0   0 ...   0   0   0]
 [  0 411   0 ...   0   0   0]
 [  0   0 404 ...   0   0   0]
 ...
 [  0   0   0 ... 386   0   0]
 [  0   0   0 ...   0 390   0]
 [  0   0   0 ...   0   0 408]]
```

**Input images:-**



**Visualizing Intermediate Layer Activations:-**



**Visualizing Convnet Filters:-**

