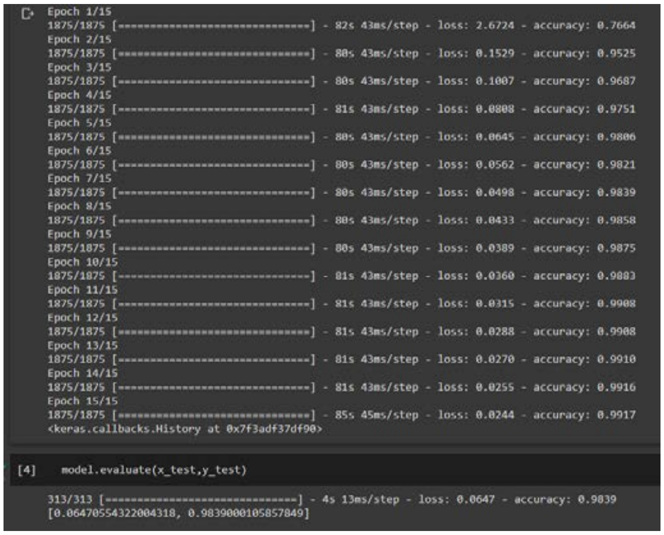
CNN MNIST Report

First model:

1. 2 convolution layers
2. 4 activation functions (relu, relu, relu, softmax)
3. 2 max pools
4. 1 fully connected layer
5. The output layer
6. Learning rate= default, Epochs = 15, patch size =32

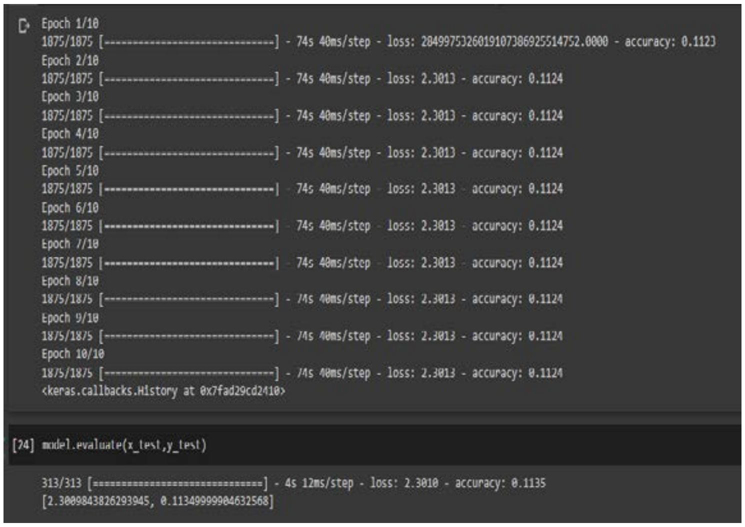
Epochs:

suitable = 15 the higher the epochs the better the accuracy becomes but it takes along time



```
Epoch 1/15
1875/1875 [==============================] - 82s 43ms/step - loss: 2.6724 - accuracy: 0.7664
Epoch 2/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.1529 - accuracy: 0.9525
Epoch 3/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.1007 - accuracy: 0.9687
Epoch 4/15
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0808 - accuracy: 0.9751
Epoch 5/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0645 - accuracy: 0.9806
Epoch 6/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0562 - accuracy: 0.9821
Epoch 7/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0498 - accuracy: 0.9839
Epoch 8/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0433 - accuracy: 0.9858
Epoch 9/15
1875/1875 [==============================] - 80s 43ms/step - loss: 0.0389 - accuracy: 0.9875
Epoch 10/15
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0360 - accuracy: 0.9883
Epoch 11/15
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0315 - accuracy: 0.9908
Epoch 12/15
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0288 - accuracy: 0.9908
Epoch 13/15
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0270 - accuracy: 0.9910
Epoch 14/15
1875/1875 [==============================] - 81s 43ms/step - loss: 0.0255 - accuracy: 0.9916
Epoch 15/15
1875/1875 [==============================] - 85s 45ms/step - loss: 0.0244 - accuracy: 0.9917
<keras.callbacks.History at 0x7f3adf37df90>

[4]   model.evaluate(x_test,y_test)

313/313 [==============================] - 4s 13ms/step - loss: 0.0647 - accuracy: 0.9839
[0.06470554322004318, 0.9839000105857849]
```

Epochs =15



```
Epoch 1/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2849975326019107386925514752.0000 - accuracy: 0.1123
Epoch 2/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 3/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 4/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 5/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 6/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 7/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 8/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 9/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
Epoch 10/10
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3013 - accuracy: 0.1124
<keras.callbacks.History at 0x7fad29cd2410>

[24] model.evaluate(x_test,y_test)

313/313 [==============================] - 4s 12ms/step - loss: 2.3010 - accuracy: 0.1135
[2.3009843826293945, 0.11349999904632568]
```

Epochs =10

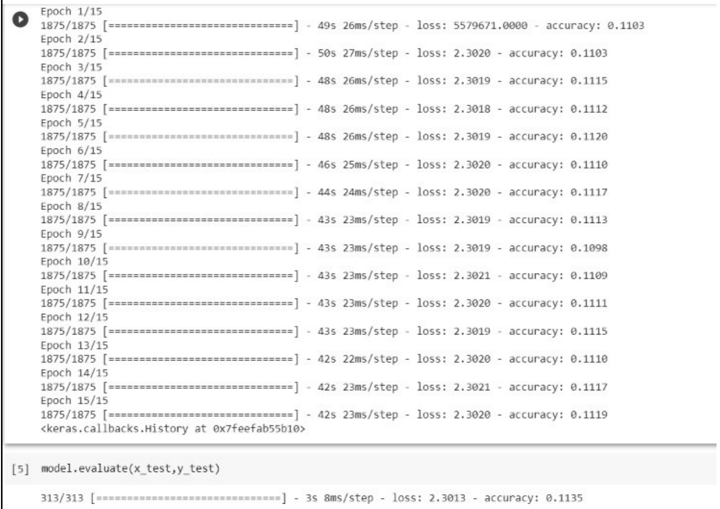
Second model:

1. 2 convolution layers
2. 4 activation functions (relu, relu, relu, softmax)
3. 2 max pools
4. 1 fully connected layer
5. The output layers
6. Learning rate= default, Epochs = 15, patch size =32

Learning rate:

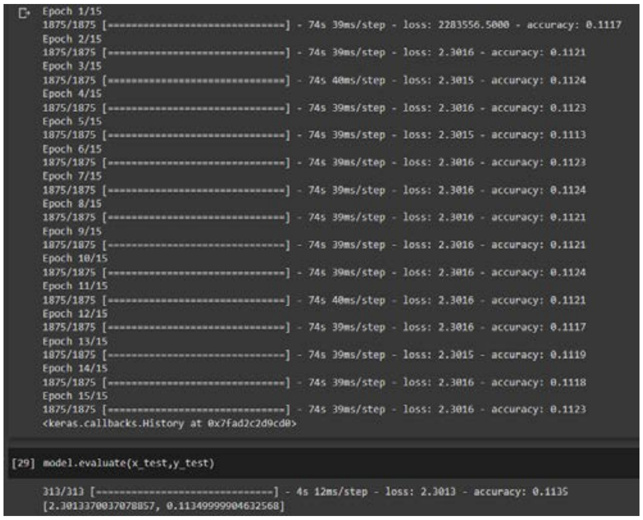
suitable = default

trying different learning rate did not get a higher accuracy but the default learning rate gave the best at all others



```
Epoch 1/15
1875/1875 [==============================] - 49s 26ms/step - loss: 5579671.0000 - accuracy: 0.1103
Epoch 2/15
1875/1875 [==============================] - 50s 27ms/step - loss: 2.3020 - accuracy: 0.1103
Epoch 3/15
1875/1875 [==============================] - 48s 26ms/step - loss: 2.3019 - accuracy: 0.1115
Epoch 4/15
1875/1875 [==============================] - 48s 26ms/step - loss: 2.3018 - accuracy: 0.1112
Epoch 5/15
1875/1875 [==============================] - 48s 26ms/step - loss: 2.3019 - accuracy: 0.1120
Epoch 6/15
1875/1875 [==============================] - 46s 25ms/step - loss: 2.3020 - accuracy: 0.1110
Epoch 7/15
1875/1875 [==============================] - 44s 24ms/step - loss: 2.3020 - accuracy: 0.1117
Epoch 8/15
1875/1875 [==============================] - 43s 23ms/step - loss: 2.3019 - accuracy: 0.1113
Epoch 9/15
1875/1875 [==============================] - 43s 23ms/step - loss: 2.3019 - accuracy: 0.1098
Epoch 10/15
1875/1875 [==============================] - 43s 23ms/step - loss: 2.3021 - accuracy: 0.1109
Epoch 11/15
1875/1875 [==============================] - 43s 23ms/step - loss: 2.3020 - accuracy: 0.1111
Epoch 12/15
1875/1875 [==============================] - 43s 23ms/step - loss: 2.3019 - accuracy: 0.1115
Epoch 13/15
1875/1875 [==============================] - 42s 22ms/step - loss: 2.3020 - accuracy: 0.1110
Epoch 14/15
1875/1875 [==============================] - 42s 23ms/step - loss: 2.3021 - accuracy: 0.1117
Epoch 15/15
1875/1875 [==============================] - 42s 23ms/step - loss: 2.3020 - accuracy: 0.1119
<keras.callbacks.History at 0x7feefab55b10>

[5] model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 2.3013 - accuracy: 0.1135
```

Learning rate =0.1



```
Epoch 1/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2283556.5000 - accuracy: 0.1117
Epoch 2/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1121
Epoch 3/15
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3015 - accuracy: 0.1124
Epoch 4/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1123
Epoch 5/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3015 - accuracy: 0.1113
Epoch 6/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1123
Epoch 7/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1124
Epoch 8/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1121
Epoch 9/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1121
Epoch 10/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1124
Epoch 11/15
1875/1875 [==============================] - 74s 40ms/step - loss: 2.3016 - accuracy: 0.1121
Epoch 12/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1117
Epoch 13/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3015 - accuracy: 0.1119
Epoch 14/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1118
Epoch 15/15
1875/1875 [==============================] - 74s 39ms/step - loss: 2.3016 - accuracy: 0.1123
<keras.callbacks.History at 0x7fad2c2d9cd0>

[29] model.evaluate(x_test,y_test)

313/313 [==============================] - 4s 12ms/step - loss: 2.3013 - accuracy: 0.1135
[2.3013370037078857, 0.11349999904632568]
```

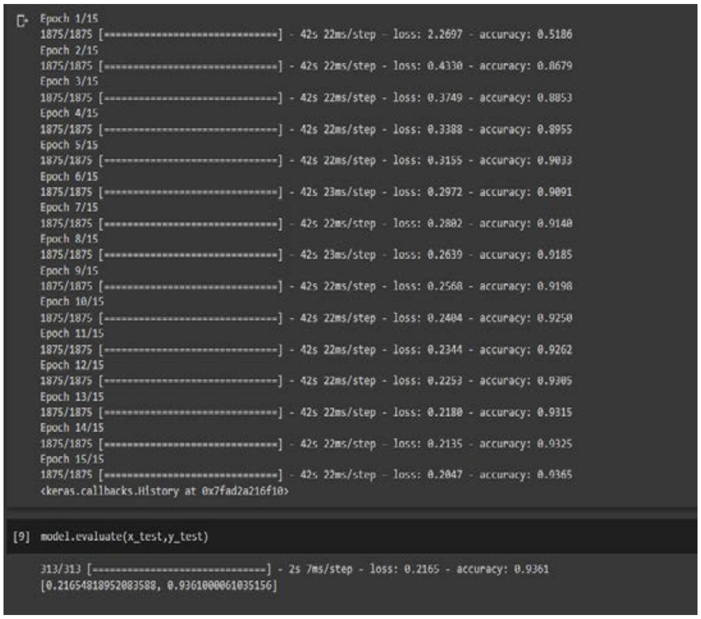
Learning rate =0.05

Third model:

1. 1 convolution layers
2. 3 activation functions (relu, relu, softmax)
3. 2 max pools
4. 1 fully connected layer
5. The output layer
6. Learning rate= default, Epochs = 15, patch size =32

CNN and parameters of the CNN:

removing one Convolution layer gave acceptable accuracy also it took a small time to train



```
Epoch 1/15
1875/1875 [==============================] - 42s 22ms/step - loss: 2.2697 - accuracy: 0.5186
Epoch 2/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.4330 - accuracy: 0.8679
Epoch 3/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.3749 - accuracy: 0.8853
Epoch 4/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.3388 - accuracy: 0.8955
Epoch 5/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.3155 - accuracy: 0.9033
Epoch 6/15
1875/1875 [==============================] - 42s 23ms/step - loss: 0.2972 - accuracy: 0.9091
Epoch 7/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2802 - accuracy: 0.9140
Epoch 8/15
1875/1875 [==============================] - 42s 23ms/step - loss: 0.2639 - accuracy: 0.9185
Epoch 9/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2568 - accuracy: 0.9198
Epoch 10/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2404 - accuracy: 0.9250
Epoch 11/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2344 - accuracy: 0.9262
Epoch 12/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2253 - accuracy: 0.9305
Epoch 13/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2180 - accuracy: 0.9315
Epoch 14/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2135 - accuracy: 0.9325
Epoch 15/15
1875/1875 [==============================] - 42s 22ms/step - loss: 0.2047 - accuracy: 0.9365
<keras.callbacks.History at 0x7fad2a216f10>

[9] model.evaluate(x_test,y_test)

313/313 [==============================] - 2s 7ms/step - loss: 0.2165 - accuracy: 0.9361
[0.21654818952083588, 0.9361000061035156]
```

Parameters before removing CNL=202698

Parameters after removing CNL=804170

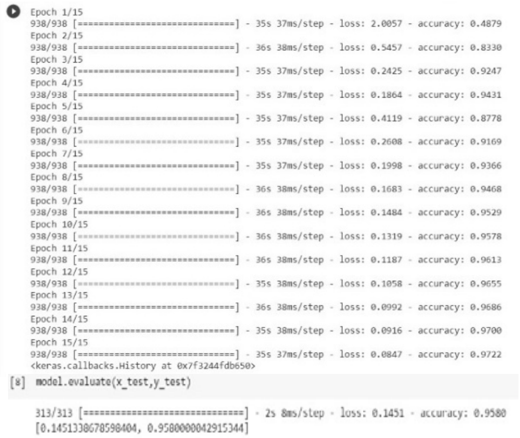
Fourth model:

1. 1 convolution layers
2. 3 activation functions (relu, relu, softmax)
3. 2 max pools
4. 1 fully connected layer
5. The output layer
6. Learning rate= default, Epochs = 15, patch size =64

Patch size:

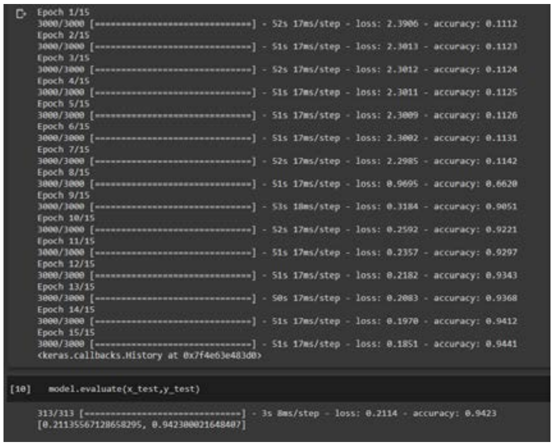
suitable = 64

the higher the patch size the petter the accuracy come also it took small time to train



```
Epoch 1/15
938/938 [==============================] - 35s 37ms/step - loss: 2.0057 - accuracy: 0.4879
Epoch 2/15
938/938 [==============================] - 36s 38ms/step - loss: 0.5457 - accuracy: 0.8330
Epoch 3/15
938/938 [==============================] - 35s 37ms/step - loss: 0.2425 - accuracy: 0.9247
Epoch 4/15
938/938 [==============================] - 35s 37ms/step - loss: 0.1864 - accuracy: 0.9431
Epoch 5/15
938/938 [==============================] - 35s 37ms/step - loss: 0.4119 - accuracy: 0.8778
Epoch 6/15
938/938 [==============================] - 35s 37ms/step - loss: 0.2608 - accuracy: 0.9169
Epoch 7/15
938/938 [==============================] - 35s 37ms/step - loss: 0.1998 - accuracy: 0.9366
Epoch 8/15
938/938 [==============================] - 36s 38ms/step - loss: 0.1683 - accuracy: 0.9468
Epoch 9/15
938/938 [==============================] - 36s 38ms/step - loss: 0.1484 - accuracy: 0.9529
Epoch 10/15
938/938 [==============================] - 36s 38ms/step - loss: 0.1319 - accuracy: 0.9578
Epoch 11/15
938/938 [==============================] - 36s 38ms/step - loss: 0.1187 - accuracy: 0.9613
Epoch 12/15
938/938 [==============================] - 35s 38ms/step - loss: 0.1058 - accuracy: 0.9655
Epoch 13/15
938/938 [==============================] - 36s 38ms/step - loss: 0.0992 - accuracy: 0.9686
Epoch 14/15
938/938 [==============================] - 35s 38ms/step - loss: 0.0916 - accuracy: 0.9700
Epoch 15/15
938/938 [==============================] - 35s 37ms/step - loss: 0.0847 - accuracy: 0.9722
<keras.callbacks.History at 0x7f3244fdb650>
[8] model.evaluate(x_test,y_test)

313/313 [==============================] - 2s 8ms/step - loss: 0.1451 - accuracy: 0.9580
[0.1451338678598404, 0.9580000042915344]
```

this photo represents the out if the patch size =64



```
Epoch 1/15
3000/3000 [==============================] - 52s 17ms/step - loss: 2.3906 - accuracy: 0.1112
Epoch 2/15
3000/3000 [==============================] - 51s 17ms/step - loss: 2.3013 - accuracy: 0.1123
Epoch 3/15
3000/3000 [==============================] - 52s 17ms/step - loss: 2.3012 - accuracy: 0.1124
Epoch 4/15
3000/3000 [==============================] - 51s 17ms/step - loss: 2.3011 - accuracy: 0.1125
Epoch 5/15
3000/3000 [==============================] - 51s 17ms/step - loss: 2.3009 - accuracy: 0.1126
Epoch 6/15
3000/3000 [==============================] - 51s 17ms/step - loss: 2.3002 - accuracy: 0.1131
Epoch 7/15
3000/3000 [==============================] - 52s 17ms/step - loss: 2.2985 - accuracy: 0.1142
Epoch 8/15
3000/3000 [==============================] - 51s 17ms/step - loss: 0.9695 - accuracy: 0.6620
Epoch 9/15
3000/3000 [==============================] - 53s 18ms/step - loss: 0.3184 - accuracy: 0.9051
Epoch 10/15
3000/3000 [==============================] - 52s 17ms/step - loss: 0.2592 - accuracy: 0.9221
Epoch 11/15
3000/3000 [==============================] - 51s 17ms/step - loss: 0.2357 - accuracy: 0.9297
Epoch 12/15
3000/3000 [==============================] - 51s 17ms/step - loss: 0.2182 - accuracy: 0.9343
Epoch 13/15
3000/3000 [==============================] - 50s 17ms/step - loss: 0.2083 - accuracy: 0.9368
Epoch 14/15
3000/3000 [==============================] - 51s 17ms/step - loss: 0.1970 - accuracy: 0.9412
Epoch 15/15
3000/3000 [==============================] - 51s 17ms/step - loss: 0.1851 - accuracy: 0.9441
<keras.callbacks.History at 0x7f4e63e483d0>

[10]   model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.2114 - accuracy: 0.9423
[0.21135567128658295, 0.942300021648407]
```

this photo represents the out if the patch size =20

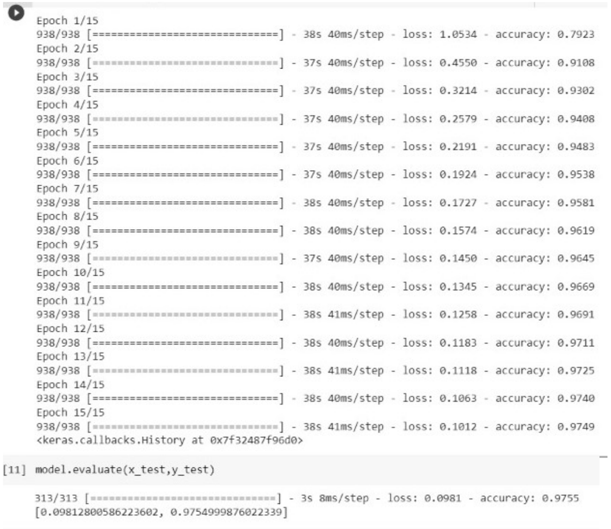
Fifth model:

1. 1 convolution layers
2. 3 activation functions (relu, relu, softmax)
3. 2 max pools
4. 1 fully connected layer
5. The output layer
6. Learning rate= default, Epochs = 15, patch size =64

Activations:

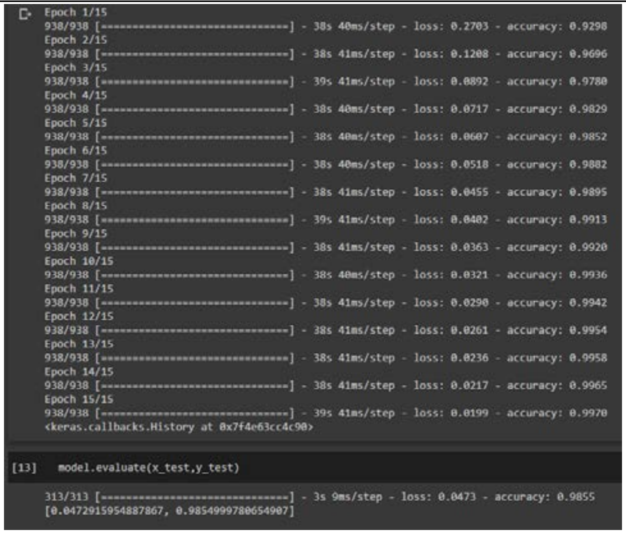
suitable = tanh

the tanh function gave the best accuracy of the others



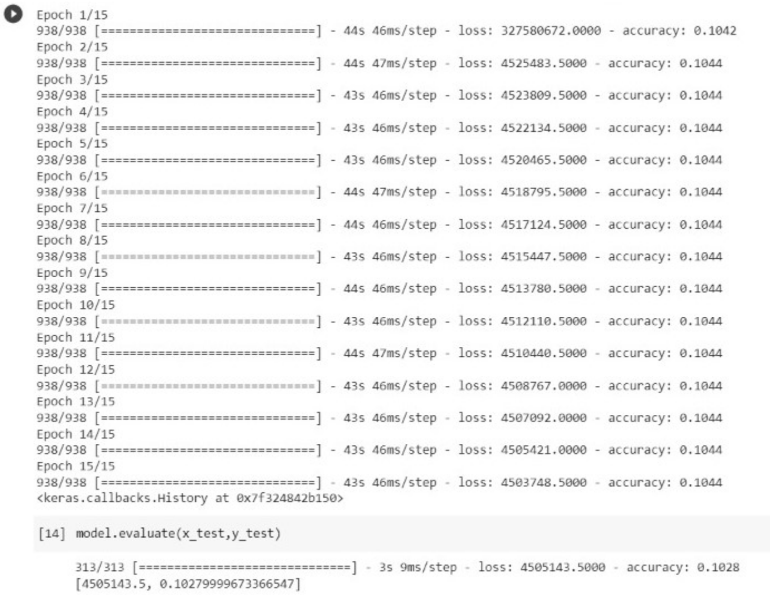
```
Epoch 1/15
938/938 [==============================] - 38s 40ms/step - loss: 1.0534 - accuracy: 0.7923
Epoch 2/15
938/938 [==============================] - 37s 40ms/step - loss: 0.4550 - accuracy: 0.9108
Epoch 3/15
938/938 [==============================] - 37s 40ms/step - loss: 0.3214 - accuracy: 0.9302
Epoch 4/15
938/938 [==============================] - 37s 40ms/step - loss: 0.2579 - accuracy: 0.9408
Epoch 5/15
938/938 [==============================] - 37s 40ms/step - loss: 0.2191 - accuracy: 0.9483
Epoch 6/15
938/938 [==============================] - 37s 40ms/step - loss: 0.1924 - accuracy: 0.9538
Epoch 7/15
938/938 [==============================] - 38s 40ms/step - loss: 0.1727 - accuracy: 0.9581
Epoch 8/15
938/938 [==============================] - 38s 40ms/step - loss: 0.1574 - accuracy: 0.9619
Epoch 9/15
938/938 [==============================] - 37s 40ms/step - loss: 0.1450 - accuracy: 0.9645
Epoch 10/15
938/938 [==============================] - 38s 40ms/step - loss: 0.1345 - accuracy: 0.9669
Epoch 11/15
938/938 [==============================] - 38s 41ms/step - loss: 0.1258 - accuracy: 0.9691
Epoch 12/15
938/938 [==============================] - 38s 40ms/step - loss: 0.1183 - accuracy: 0.9711
Epoch 13/15
938/938 [==============================] - 38s 41ms/step - loss: 0.1118 - accuracy: 0.9725
Epoch 14/15
938/938 [==============================] - 38s 40ms/step - loss: 0.1063 - accuracy: 0.9740
Epoch 15/15
938/938 [==============================] - 38s 41ms/step - loss: 0.1012 - accuracy: 0.9749
<keras.callbacks.History at 0x7f32487f96d0>

[11] model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.0981 - accuracy: 0.9755
[0.09812800586223602, 0.9754999876022339]
```

this photo represents the out if the Activation = sigmoid



```
Epoch 1/15
938/938 [==============================] - 38s 40ms/step - loss: 0.2703 - accuracy: 0.9298
Epoch 2/15
938/938 [==============================] - 38s 41ms/step - loss: 0.1208 - accuracy: 0.9696
Epoch 3/15
938/938 [==============================] - 39s 41ms/step - loss: 0.0892 - accuracy: 0.9780
Epoch 4/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0717 - accuracy: 0.9829
Epoch 5/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0607 - accuracy: 0.9852
Epoch 6/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0518 - accuracy: 0.9882
Epoch 7/15
938/938 [==============================] - 38s 41ms/step - loss: 0.0455 - accuracy: 0.9895
Epoch 8/15
938/938 [==============================] - 39s 41ms/step - loss: 0.0402 - accuracy: 0.9913
Epoch 9/15
938/938 [==============================] - 38s 41ms/step - loss: 0.0363 - accuracy: 0.9920
Epoch 10/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0321 - accuracy: 0.9936
Epoch 11/15
938/938 [==============================] - 38s 41ms/step - loss: 0.0290 - accuracy: 0.9942
Epoch 12/15
938/938 [==============================] - 38s 41ms/step - loss: 0.0261 - accuracy: 0.9954
Epoch 13/15
938/938 [==============================] - 38s 41ms/step - loss: 0.0236 - accuracy: 0.9958
Epoch 14/15
938/938 [==============================] - 38s 41ms/step - loss: 0.0217 - accuracy: 0.9965
Epoch 15/15
938/938 [==============================] - 39s 41ms/step - loss: 0.0199 - accuracy: 0.9970
<keras.callbacks.History at 0x7f4e63cc4c90>

[13]  model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 9ms/step - loss: 0.0473 - accuracy: 0.9855
[0.0472915954887867, 0.9854999780654907]
```

this photo represents the out if the Activation = tanh



```
Epoch 1/15
938/938 [==============================] - 44s 46ms/step - loss: 327580672.0000 - accuracy: 0.1042
Epoch 2/15
938/938 [==============================] - 44s 47ms/step - loss: 4525483.5000 - accuracy: 0.1044
Epoch 3/15
938/938 [==============================] - 43s 46ms/step - loss: 4523809.5000 - accuracy: 0.1044
Epoch 4/15
938/938 [==============================] - 43s 46ms/step - loss: 4522134.5000 - accuracy: 0.1044
Epoch 5/15
938/938 [==============================] - 43s 46ms/step - loss: 4520465.5000 - accuracy: 0.1044
Epoch 6/15
938/938 [==============================] - 44s 47ms/step - loss: 4518795.5000 - accuracy: 0.1044
Epoch 7/15
938/938 [==============================] - 44s 46ms/step - loss: 4517124.5000 - accuracy: 0.1044
Epoch 8/15
938/938 [==============================] - 43s 46ms/step - loss: 4515447.5000 - accuracy: 0.1044
Epoch 9/15
938/938 [==============================] - 44s 46ms/step - loss: 4513780.5000 - accuracy: 0.1044
Epoch 10/15
938/938 [==============================] - 43s 46ms/step - loss: 4512110.5000 - accuracy: 0.1044
Epoch 11/15
938/938 [==============================] - 44s 47ms/step - loss: 4510440.5000 - accuracy: 0.1044
Epoch 12/15
938/938 [==============================] - 43s 46ms/step - loss: 4508767.0000 - accuracy: 0.1044
Epoch 13/15
938/938 [==============================] - 43s 46ms/step - loss: 4507092.0000 - accuracy: 0.1044
Epoch 14/15
938/938 [==============================] - 43s 46ms/step - loss: 4505421.0000 - accuracy: 0.1044
Epoch 15/15
938/938 [==============================] - 43s 46ms/step - loss: 4503748.5000 - accuracy: 0.1044
<keras.callbacks.History at 0x7f324842b150>

[14] model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 9ms/step - loss: 4505143.5000 - accuracy: 0.1028
[4505143.5, 0.10279999673366547]
```

this photo represents the out if the Activation = selu

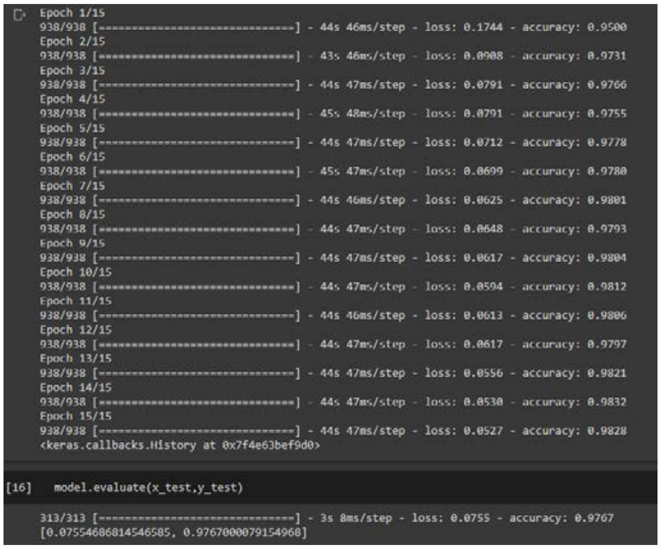
sixth model:

1. 1 convolution layers
2. 4 activation functions (tanh, tanh, softmax)
3. 2 max pools
4. 1 fully connected layer
5. The output layer
6. Learning rate= default, Epochs = 15, patch size =64

Optimizer:

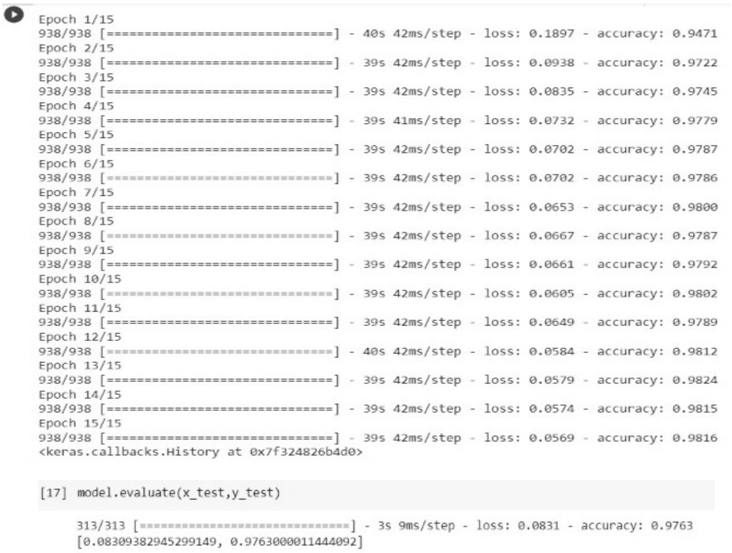
suitable = SGD

the difference all most can be noticed but still the SGD gave the best of them



```
Epoch 1/15
938/938 [==============================] - 44s 46ms/step - loss: 0.1744 - accuracy: 0.9500
Epoch 2/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0908 - accuracy: 0.9731
Epoch 3/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0791 - accuracy: 0.9766
Epoch 4/15
938/938 [==============================] - 45s 48ms/step - loss: 0.0791 - accuracy: 0.9755
Epoch 5/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0712 - accuracy: 0.9778
Epoch 6/15
938/938 [==============================] - 45s 47ms/step - loss: 0.0699 - accuracy: 0.9780
Epoch 7/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0625 - accuracy: 0.9801
Epoch 8/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0648 - accuracy: 0.9793
Epoch 9/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0617 - accuracy: 0.9804
Epoch 10/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0594 - accuracy: 0.9812
Epoch 11/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0613 - accuracy: 0.9806
Epoch 12/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0617 - accuracy: 0.9797
Epoch 13/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0556 - accuracy: 0.9821
Epoch 14/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0530 - accuracy: 0.9832
Epoch 15/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0527 - accuracy: 0.9828
<keras.callbacks.History at 0x7f4e63bef9d0>

[16]  model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.0755 - accuracy: 0.9767
[0.07554686814546585, 0.9767000079154968]
```

this photo represents the out if the Optimizer = nadam



```
Epoch 1/15
938/938 [==============================] - 40s 42ms/step - loss: 0.1897 - accuracy: 0.9471
Epoch 2/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0938 - accuracy: 0.9722
Epoch 3/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0835 - accuracy: 0.9745
Epoch 4/15
938/938 [==============================] - 39s 41ms/step - loss: 0.0732 - accuracy: 0.9779
Epoch 5/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0702 - accuracy: 0.9787
Epoch 6/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0702 - accuracy: 0.9786
Epoch 7/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0653 - accuracy: 0.9800
Epoch 8/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0667 - accuracy: 0.9787
Epoch 9/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0661 - accuracy: 0.9792
Epoch 10/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0605 - accuracy: 0.9802
Epoch 11/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0649 - accuracy: 0.9789
Epoch 12/15
938/938 [==============================] - 40s 42ms/step - loss: 0.0584 - accuracy: 0.9812
Epoch 13/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0579 - accuracy: 0.9824
Epoch 14/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0574 - accuracy: 0.9815
Epoch 15/15
938/938 [==============================] - 39s 42ms/step - loss: 0.0569 - accuracy: 0.9816
<keras.callbacks.History at 0x7f324826b4d0>

[17] model.evaluate(x_test,y_test)

    313/313 [==============================] - 3s 9ms/step - loss: 0.0831 - accuracy: 0.9763
    [0.08309382945299149, 0.9763000011444092]
```

this photo represents the out if the Optimizer = adam

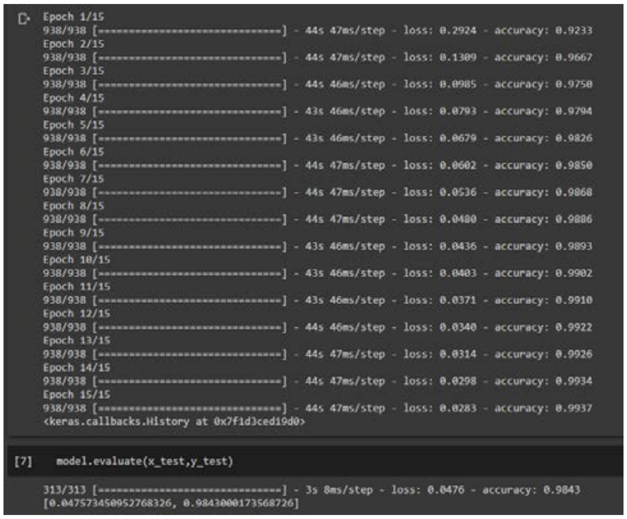
sixth model:

1. 1 convolution layers
2. 3 activation functions (tanh, tanh, softmax)
3. 2 max pools
4. Droop out layer
5. 1 fully connected layer
6. The output layer
7. Learning rate= default, Epochs = 15, patch size =64

Droop out:

Suitable= case 3

The droop out layer had a small effect on the accuracy and time that is why maybe we do not need to add a droop out layer

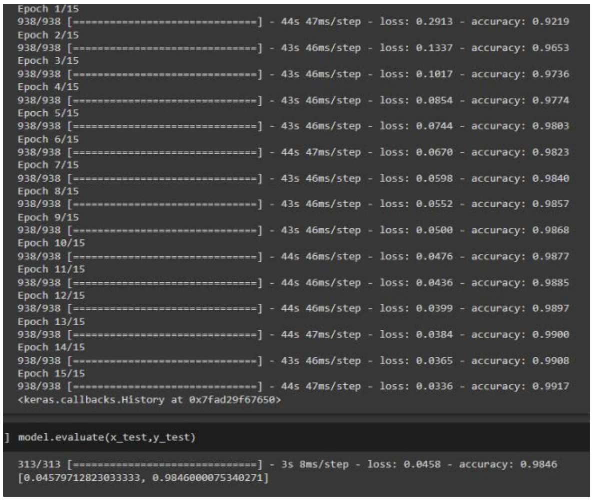


```
Epoch 1/15
938/938 [==============================] - 44s 47ms/step - loss: 0.2924 - accuracy: 0.9233
Epoch 2/15
938/938 [==============================] - 44s 47ms/step - loss: 0.1309 - accuracy: 0.9667
Epoch 3/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0985 - accuracy: 0.9758
Epoch 4/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0793 - accuracy: 0.9794
Epoch 5/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0679 - accuracy: 0.9826
Epoch 6/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0602 - accuracy: 0.9850
Epoch 7/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0536 - accuracy: 0.9868
Epoch 8/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0480 - accuracy: 0.9886
Epoch 9/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0436 - accuracy: 0.9893
Epoch 10/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0403 - accuracy: 0.9902
Epoch 11/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0371 - accuracy: 0.9910
Epoch 12/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0340 - accuracy: 0.9922
Epoch 13/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0314 - accuracy: 0.9926
Epoch 14/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0298 - accuracy: 0.9934
Epoch 15/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0283 - accuracy: 0.9937
<keras.callbacks.History at 0x7f1d3ced19d0>

[7]   model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.0476 - accuracy: 0.9843
[0.047573450952768326, 0.9843000173568726]
```

this photo represents the out if the

droop out layer before the first FCL

percent 0.15 case 1

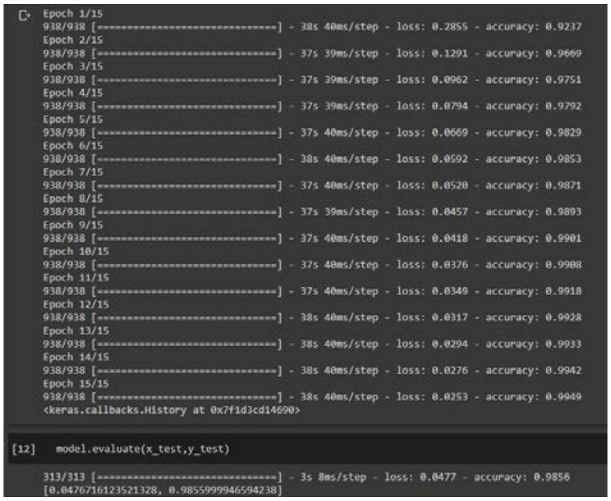


```
Epoch 1/15
938/938 [==============================] - 44s 47ms/step - loss: 0.2913 - accuracy: 0.9219
Epoch 2/15
938/938 [==============================] - 43s 46ms/step - loss: 0.1337 - accuracy: 0.9653
Epoch 3/15
938/938 [==============================] - 43s 46ms/step - loss: 0.1017 - accuracy: 0.9736
Epoch 4/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0854 - accuracy: 0.9774
Epoch 5/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0744 - accuracy: 0.9803
Epoch 6/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0670 - accuracy: 0.9823
Epoch 7/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0598 - accuracy: 0.9840
Epoch 8/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0552 - accuracy: 0.9857
Epoch 9/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0500 - accuracy: 0.9868
Epoch 10/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0476 - accuracy: 0.9877
Epoch 11/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0436 - accuracy: 0.9885
Epoch 12/15
938/938 [==============================] - 44s 46ms/step - loss: 0.0399 - accuracy: 0.9897
Epoch 13/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0384 - accuracy: 0.9900
Epoch 14/15
938/938 [==============================] - 43s 46ms/step - loss: 0.0365 - accuracy: 0.9908
Epoch 15/15
938/938 [==============================] - 44s 47ms/step - loss: 0.0336 - accuracy: 0.9917
<keras.callbacks.History at 0x7fad29f67650>

] model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.0458 - accuracy: 0.9846
[0.04579712823033333, 0.9846000075340271]
```

this photo represents the out if the

droop out layer before the first FCL

percent 0.30 case 2

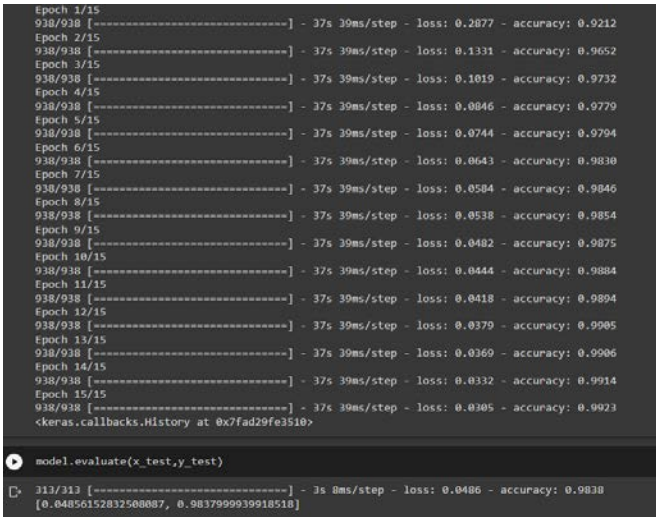


```
Epoch 1/15
938/938 [==============================] - 38s 40ms/step - loss: 0.2855 - accuracy: 0.9237
Epoch 2/15
938/938 [==============================] - 37s 39ms/step - loss: 0.1291 - accuracy: 0.9669
Epoch 3/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0962 - accuracy: 0.9751
Epoch 4/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0794 - accuracy: 0.9792
Epoch 5/15
938/938 [==============================] - 37s 40ms/step - loss: 0.0669 - accuracy: 0.9829
Epoch 6/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0592 - accuracy: 0.9853
Epoch 7/15
938/938 [==============================] - 37s 40ms/step - loss: 0.0520 - accuracy: 0.9871
Epoch 8/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0457 - accuracy: 0.9893
Epoch 9/15
938/938 [==============================] - 37s 40ms/step - loss: 0.0418 - accuracy: 0.9901
Epoch 10/15
938/938 [==============================] - 37s 40ms/step - loss: 0.0376 - accuracy: 0.9908
Epoch 11/15
938/938 [==============================] - 37s 40ms/step - loss: 0.0349 - accuracy: 0.9918
Epoch 12/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0317 - accuracy: 0.9928
Epoch 13/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0294 - accuracy: 0.9933
Epoch 14/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0276 - accuracy: 0.9942
Epoch 15/15
938/938 [==============================] - 38s 40ms/step - loss: 0.0253 - accuracy: 0.9949
<keras.callbacks.History at 0x7f1d3cd14690>

[12]   model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.0477 - accuracy: 0.9856
[0.0476716123521328, 0.9855999946594238]
```

this photo represents the out if the

droop out layer before the output layer

percent 0.15 case 3



```
Epoch 1/15
938/938 [==============================] - 37s 39ms/step - loss: 0.2877 - accuracy: 0.9212
Epoch 2/15
938/938 [==============================] - 37s 39ms/step - loss: 0.1331 - accuracy: 0.9652
Epoch 3/15
938/938 [==============================] - 37s 39ms/step - loss: 0.1019 - accuracy: 0.9732
Epoch 4/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0846 - accuracy: 0.9779
Epoch 5/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0744 - accuracy: 0.9794
Epoch 6/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0643 - accuracy: 0.9830
Epoch 7/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0584 - accuracy: 0.9846
Epoch 8/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0538 - accuracy: 0.9854
Epoch 9/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0482 - accuracy: 0.9875
Epoch 10/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0444 - accuracy: 0.9884
Epoch 11/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0418 - accuracy: 0.9894
Epoch 12/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0379 - accuracy: 0.9905
Epoch 13/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0369 - accuracy: 0.9906
Epoch 14/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0332 - accuracy: 0.9914
Epoch 15/15
938/938 [==============================] - 37s 39ms/step - loss: 0.0305 - accuracy: 0.9923
<keras.callbacks.History at 0x7fad29fe3510>

model.evaluate(x_test,y_test)

313/313 [==============================] - 3s 8ms/step - loss: 0.0486 - accuracy: 0.9838
[0.04856152832508087, 0.9837999939918518]
```

this photo represents the out if the

droop out layer before the output layer

percent 0.30 case 4

Codes:

- The initial model

```
model.add(Conv2D(64, (3, 3),input_shape=(28,28,1)))
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2, 2)))
# ------------------------------------------------------------------------------------------------------
model.add(Conv2D(64, (3, 3)))
model.add(Activation("sigmoid"))
model.add(MaxPool2D(pool_size=(2, 2)))
# ------------------------------------------------------------------------------------------------------
model.add(Flatten())
model.add(Dense(64))
model.add(Activation("relu"))
# ------------------------------------------------------------------------------------------------------
model.add(Dense(10))
model.add(Activation("softmax"))
# ------------------------------------------------------------------------------------------------------
model.compile(loss="sparse_categorical_crossentropy",optimizer = "SGD",metrics=["accuracy"],)
model.fit(x_train,y_train,32,15, shuffle=True)
```

- Model after removing CNL layer

```
model.add(Conv2D(64, (3, 3),input_shape=(28,28,1)))
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2, 2)))
# ------------------------------------------------------------------------------------------------------
# ------------------------------------------------------------------------------------------------------
model.add(Flatten())
model.add(Dense(64))
model.add(Activation("relu"))
# ------------------------------------------------------------------------------------------------------
model.add(Dense(10))
model.add(Activation("softmax"))
# ------------------------------------------------------------------------------------------------------
model.compile(loss="sparse_categorical_crossentropy",optimizer = "SGD",metrics=["accuracy"],)
model.fit(x_train,y_train,32,15, shuffle=True)
```

- Model first droop out

```
model.add(Conv2D(64, (3, 3),input_shape=(28,28,1)))
model.add(Activation("tanh"))
model.add(MaxPool2D(pool_size=(2, 2)))
# ------------------------------------------------------------------------------------------------------
# ------------------------------------------------------------------------------------------------------
model.add(Flatten())
model.add(Dense(64))
model.add(Activation("tanh"))
model.add(Dropout(0.15))
```

```
# ----------------------------------------------------------------------------------------------------
model.add(Dense(10))
model.add(Activation("softmax"))
# ----------------------------------------------------------------------------------------------------
model.compile(loss="sparse_categorical_crossentropy",optimizer = "SGD",metrics=["accuracy"],)
model.fit(x_train,y_train,64,15, shuffle=True)
```

- Model second droop

```
model.add(Conv2D(64, (3, 3),input_shape=(28,28,1)))
model.add(Activation("tanh"))
model.add(MaxPool2D(pool_size=(2, 2)))
# ----------------------------------------------------------------------------------------------------
# ----------------------------------------------------------------------------------------------------
model.add(Flatten())
model.add(Dropout(0.15))
model.add(Dense(64))
model.add(Activation("tanh"))
# ----------------------------------------------------------------------------------------------------
model.add(Dense(10))
model.add(Activation("softmax"))
# ----------------------------------------------------------------------------------------------------
model.compile(loss="sparse_categorical_crossentropy",optimizer = "SGD",metrics=["accuracy"],)
model.fit(x_train,y_train,64,15, shuffle=True)
```

- the final preferred model

```
model.add(Conv2D(64, (3, 3),input_shape=(28,28,1)))
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2, 2)))
# ----------------------------------------------------------------------------------------------------
# ----------------------------------------------------------------------------------------------------
model.add(Flatten())
model.add(Dense(64))
model.add(Activation("relu"))
# ----------------------------------------------------------------------------------------------------
model.add(Dense(10))
model.add(Activation("softmax"))
# ----------------------------------------------------------------------------------------------------
model.compile(loss="sparse_categorical_crossentropy",optimizer = "SGD",metrics=["accuracy"],)
model.fit(x_train,y_train,32,15, shuffle=True)
```

Why?

This model was chosen depending on the results of the hyper prameters