Names :
Gasser Ahmed Mohamed **Youssef Salah Fathy**
20190150 **20190636**

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=150, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
200/200 [==============================] - 323s 2s/step - loss: 12.8914
- accuracy: 0.5152 - val_loss: 0.7072 - val_accuracy: 0.7783
Epoch 2/5
200/200 [==============================] - 295s 1s/step - loss: 0.4285
- accuracy: 0.8721 - val_loss: 0.2886 - val_accuracy: 0.9187
Epoch 3/5
200/200 [==============================] - 294s 1s/step - loss: 0.2385
- accuracy: 0.9316 - val_loss: 0.2461 - val_accuracy: 0.9343
Epoch 4/5
200/200 [==============================] - 295s 1s/step - loss: 0.1795
- accuracy: 0.9488 - val_loss: 0.1896 - val_accuracy: 0.9477
Epoch 5/5
200/200 [==============================] - 295s 1s/step - loss: 0.1476
- accuracy: 0.9586 - val_loss: 0.2212 - val_accuracy: 0.9372
938/938 [==============================] - 66s 71ms/step - loss: 0.2212
- accuracy: 0.9372
> 93.717
Test time: 66s
Training time: 300.4s
Epoch 1/5
200/200 [==============================] - 296s 1s/step - loss: 9.6880
- accuracy: 0.1784 - val_loss: 2.0529 - val_accuracy: 0.1941
Epoch 2/5
200/200 [==============================] - 296s 1s/step - loss: 1.9640
- accuracy: 0.2013 - val_loss: 1.9609 - val_accuracy: 0.2059
Epoch 3/5
200/200 [==============================] - 294s 1s/step - loss: 1.9034
- accuracy: 0.2103 - val_loss: 1.8941 - val_accuracy: 0.2144
Epoch 4/5
200/200 [==============================] - 296s 1s/step - loss: 1.8309
- accuracy: 0.2260 - val_loss: 1.8015 - val_accuracy: 0.2275
Epoch 5/5
200/200 [==============================] - 296s 1s/step - loss: 1.7802
- accuracy: 0.2436 - val_loss: 1.7761 - val_accuracy: 0.2988
938/938 [==============================] - 67s 71ms/step - loss: 1.7761
- accuracy: 0.2988
> 29.883
Test time: 67s Training time: 295.6s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=160, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  679478
Epoch 1/5
188/188 [==============================] - 410s 2s/step - loss: 10.9789
- accuracy: 0.1118 - val_loss: 2.3017 - val_accuracy: 0.1118
Epoch 2/5
188/188 [==============================] - 412s 2s/step - loss: 2.3012
- accuracy: 0.1129 - val_loss: 2.3015 - val_accuracy: 0.1118
Epoch 3/5
188/188 [==============================] - 413s 2s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 4/5
188/188 [==============================] - 412s 2s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 5/5
188/188 [==============================] - 413s 2s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
938/938 [==============================] - 89s 94ms/step - loss: 2.3016
- accuracy: 0.1118
> 11.180
Test time: 89s
Training time: 412s
Epoch 1/5
188/188 [==============================] - 413s 2s/step - loss: 6.6328
- accuracy: 0.1046 - val_loss: 2.3020 - val_accuracy: 0.1129
Epoch 2/5
188/188 [==============================] - 414s 2s/step - loss: 2.3019
- accuracy: 0.1118 - val_loss: 2.3015 - val_accuracy: 0.1129
Epoch 3/5
188/188 [==============================] - 415s 2s/step - loss: 2.3017
- accuracy: 0.1118 - val_loss: 2.3012 - val_accuracy: 0.1129
Epoch 4/5
188/188 [==============================] - 415s 2s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3011 - val_accuracy: 0.1129
Epoch 5/5
188/188 [==============================] - 415s 2s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3011 - val_accuracy: 0.1129
938/938 [==============================] - 89s 94ms/step - loss: 2.3011
- accuracy: 0.1129
> 11.293
Test time: 89s         Training time: 414.4
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (5*5) and another Pooling (Max_Pooling) = (5*5) and SGD with batch_size=130, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
231/231 [==============================] - 211s 910ms/step - loss:
5.1156 - accuracy: 0.1049 - val_loss: 2.3022 - val_accuracy: 0.1033
Epoch 2/5
231/231 [==============================] - 210s 912ms/step - loss:
2.3016 - accuracy: 0.1069 - val_loss: 2.3018 - val_accuracy: 0.1118
Epoch 3/5
231/231 [==============================] - 212s 920ms/step - loss:
2.3012 - accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 4/5
231/231 [==============================] - 213s 923ms/step - loss:
2.3010 - accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 5/5
231/231 [==============================] - 214s 926ms/step - loss:
2.3009 - accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
938/938 [==============================] - 48s 52ms/step - loss: 2.3016
- accuracy: 0.1118
> 11.180
Test time: 48s
Training time: 212s
Epoch 1/5
231/231 [==============================] - 210s 907ms/step - loss:
7.0156 - accuracy: 0.3725 - val_loss: 1.1621 - val_accuracy: 0.6149
Epoch 2/5
231/231 [==============================] - 210s 910ms/step - loss:
0.8922 - accuracy: 0.7355 - val_loss: 0.7388 - val_accuracy: 0.8025
Epoch 3/5
231/231 [==============================] - 210s 910ms/step - loss:
0.6494 - accuracy: 0.8115 - val_loss: 0.6054 - val_accuracy: 0.8219
Epoch 4/5
231/231 [==============================] - 210s 912ms/step - loss:
0.3437 - accuracy: 0.9087 - val_loss: 0.2245 - val_accuracy: 0.9479
Epoch 5/5
231/231 [==============================] - 211s 913ms/step - loss:
0.1830 - accuracy: 0.9547 - val_loss: 0.1838 - val_accuracy: 0.9564
938/938 [==============================] - 47s 50ms/step - loss: 0.1838
- accuracy: 0.9564
> 95.640
Test time: 47s
Training time: 210.2s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (5*5) and another Pooling (Max_Pooling) = (5*5) and SGD with batch_size=120, Optimizer= SGD, learning_rate=0.1 and momentum=0.4 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
250/250 [==============================] - 222s 883ms/step - loss:
224856721734005227520.0000 - accuracy: 0.1111 - val_loss: 2.3020 -
val_accuracy: 0.1118
Epoch 2/5
250/250 [==============================] - 214s 857ms/step - loss:
2.3012 - accuracy: 0.1129 - val_loss: 2.3017 - val_accuracy: 0.1118
Epoch 3/5
250/250 [==============================] - 214s 857ms/step - loss:
2.3012 - accuracy: 0.1122 - val_loss: 2.3019 - val_accuracy: 0.1118
Epoch 4/5
250/250 [==============================] - 214s 855ms/step - loss:
2.3011 - accuracy: 0.1129 - val_loss: 2.3020 - val_accuracy: 0.1118
Epoch 5/5
250/250 [==============================] - 214s 858ms/step - loss:
2.3011 - accuracy: 0.1129 - val_loss: 2.3018 - val_accuracy: 0.1118
938/938 [==============================] - 50s 53ms/step - loss: 2.3018
- accuracy: 0.1118
> 11.180
Test time: 50s
Training time: 215.6s
Epoch 1/5
250/250 [==============================] - 210s 840ms/step - loss:
46109188.0000 - accuracy: 0.1110 - val_loss: 2.3011 - val_accuracy:
0.1129
Epoch 2/5
250/250 [==============================] - 208s 834ms/step - loss:
2.3019 - accuracy: 0.1118 - val_loss: 2.3014 - val_accuracy: 0.1129
Epoch 3/5
250/250 [==============================] - 209s 835ms/step - loss:
2.3019 - accuracy: 0.1118 - val_loss: 2.3010 - val_accuracy: 0.1129
Epoch 4/5
250/250 [==============================] - 209s 835ms/step - loss:
2.3019 - accuracy: 0.1113 - val_loss: 2.3010 - val_accuracy: 0.1129
Epoch 5/5
250/250 [==============================] - 208s 834ms/step - loss:
2.3019 - accuracy: 0.1118 - val_loss: 2.3011 - val_accuracy: 0.1129
938/938 [==============================] - 50s 53ms/step - loss: 2.3011
- accuracy: 0.1129
> 11.293
Test time: 50s
Training time: 208.8s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (5*5) and another Pooling (Max_Pooling) = (5*5) and SGD with batch_size=180, Optimizer= SGD, learning_rate=0.2 and momentum=0.5 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
167/167 [==============================] - 207s 1s/step - loss:
5637571072.0000 - accuracy: 0.1123 - val_loss: 2.3018 - val_accuracy:
0.1118
Epoch 2/5
167/167 [==============================] - 207s 1s/step - loss: 2.3014
- accuracy: 0.1120 - val_loss: 2.3018 - val_accuracy: 0.1118
Epoch 3/5
167/167 [==============================] - 208s 1s/step - loss: 2.3014
- accuracy: 0.1124 - val_loss: 2.3020 - val_accuracy: 0.1118
Epoch 4/5
167/167 [==============================] - 210s 1s/step - loss: 2.3014
- accuracy: 0.1116 - val_loss: 2.3020 - val_accuracy: 0.1118
Epoch 5/5
167/167 [==============================] - 212s 1s/step - loss: 2.3015
- accuracy: 0.1129 - val_loss: 2.3018 - val_accuracy: 0.1118
938/938 [==============================] - 51s 54ms/step - loss: 2.3018
- accuracy: 0.1118
> 11.180
Test time: 51s
Training time: 208.8s
Epoch 1/5
167/167 [==============================] - 208s 1s/step - loss: nan -
accuracy: 0.0992 - val_loss: nan - val_accuracy: 0.0987
Epoch 2/5
167/167 [==============================] - 211s 1s/step - loss: nan -
accuracy: 0.0987 - val_loss: nan - val_accuracy: 0.0987
Epoch 3/5
167/167 [==============================] - 212s 1s/step - loss: nan -
accuracy: 0.0987 - val_loss: nan - val_accuracy: 0.0987
Epoch 4/5
167/167 [==============================] - 213s 1s/step - loss: nan -
accuracy: 0.0987 - val_loss: nan - val_accuracy: 0.0987
Epoch 5/5
167/167 [==============================] - 211s 1s/step - loss: nan -
accuracy: 0.0987 - val_loss: nan - val_accuracy: 0.0987
938/938 [==============================] - 50s 53ms/step - loss: nan -
accuracy: 0.0987
> 9.870
Test time: 50s
Training time: 211s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=110, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  679478
Epoch 1/5
273/273 [==============================] - 421s 2s/step - loss: 4.5805
- accuracy: 0.1071 - val_loss: 2.3018 - val_accuracy: 0.1118
Epoch 2/5
273/273 [==============================] - 419s 2s/step - loss: 2.3012
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 3/5
273/273 [==============================] - 418s 2s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 4/5
273/273 [==============================] - 419s 2s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3017 - val_accuracy: 0.1118
Epoch 5/5
273/273 [==============================] - 420s 2s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3017 - val_accuracy: 0.1118
938/938 [==============================] - 89s 95ms/step - loss: 2.3017
- accuracy: 0.1118
> 11.180
Test time: 89s
Training time: 419.4s
Epoch 1/5
273/273 [==============================] - 422s 2s/step - loss: 5.8615
- accuracy: 0.1103 - val_loss: 2.3017 - val_accuracy: 0.1129
Epoch 2/5
273/273 [==============================] - 423s 2s/step - loss: 2.3017
- accuracy: 0.1118 - val_loss: 2.3012 - val_accuracy: 0.1129
Epoch 3/5
273/273 [==============================] - 423s 2s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3011 - val_accuracy: 0.1129
Epoch 4/5
273/273 [==============================] - 423s 2s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3010 - val_accuracy: 0.1129
Epoch 5/5
273/273 [==============================] - 423s 2s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3010 - val_accuracy: 0.1129
938/938 [==============================] - 90s 95ms/step - loss: 2.3010
- accuracy: 0.1129
> 11.293
Test time: 90s
Training time: 422.8s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=100, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
300/300 [==============================] - 303s 1s/step - loss: 7.3657
- accuracy: 0.1128 - val_loss: 2.3018 - val_accuracy: 0.1118
Epoch 2/5
300/300 [==============================] - 303s 1s/step - loss: 2.3012
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 3/5
300/300 [==============================] - 303s 1s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 4/5
300/300 [==============================] - 304s 1s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 5/5
300/300 [==============================] - 301s 1s/step - loss: 2.3009
- accuracy: 0.1129 - val_loss: 2.3017 - val_accuracy: 0.1118
938/938 [==============================] - 67s 71ms/step - loss: 2.3017
- accuracy: 0.1118
> 11.180
Test time: 67s
Training time: 302.8
Epoch 1/5
300/300 [==============================] - 296s 984ms/step - loss:
7.1069 - accuracy: 0.6844 - val_loss: 0.3762 - val_accuracy: 0.8902
Epoch 2/5
300/300 [==============================] - 295s 985ms/step - loss:
0.2771 - accuracy: 0.9240 - val_loss: 0.2224 - val_accuracy: 0.9418
Epoch 3/5
300/300 [==============================] - 295s 985ms/step - loss:
0.1769 - accuracy: 0.9519 - val_loss: 0.2039 - val_accuracy: 0.9460
Epoch 4/5
300/300 [==============================] - 295s 985ms/step - loss:
0.1333 - accuracy: 0.9629 - val_loss: 0.1412 - val_accuracy: 0.9630
Epoch 5/5
300/300 [==============================] - 295s 984ms/step - loss:
0.1141 - accuracy: 0.9683 - val_loss: 0.1473 - val_accuracy: 0.9608
938/938 [==============================] - 66s 71ms/step - loss: 0.1473
- accuracy: 0.9608
> 96.083
Test time: 66s
Training time: 295.2s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=200, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
150/150 [==============================] - 273s 2s/step - loss: 12.1197
- accuracy: 0.5410 - val_loss: 0.7202 - val_accuracy: 0.7617
Epoch 2/5
150/150 [==============================] - 276s 2s/step - loss: 0.5724
- accuracy: 0.8086 - val_loss: 0.4237 - val_accuracy: 0.8753
Epoch 3/5
150/150 [==============================] - 269s 2s/step - loss: 0.3070
- accuracy: 0.9180 - val_loss: 0.3022 - val_accuracy: 0.9204
Epoch 4/5
150/150 [==============================] - 301s 2s/step - loss: 0.1991
- accuracy: 0.9477 - val_loss: 0.2091 - val_accuracy: 0.9433
Epoch 5/5
150/150 [==============================] - 285s 2s/step - loss: 0.1580
- accuracy: 0.9576 - val_loss: 0.1779 - val_accuracy: 0.9528
938/938 [==============================] - 63s 67ms/step - loss: 0.1779
- accuracy: 0.9528
> 95.277
Test time: 63s
Training time: 280.8s
Epoch 1/5
150/150 [==============================] - 281s 2s/step - loss: 9.4759
- accuracy: 0.6538 - val_loss: 0.3297 - val_accuracy: 0.9100
Epoch 2/5
150/150 [==============================] - 278s 2s/step - loss: 0.2584
- accuracy: 0.9271 - val_loss: 0.2118 - val_accuracy: 0.9434
Epoch 3/5
150/150 [==============================] - 284s 2s/step - loss: 0.1574
- accuracy: 0.9565 - val_loss: 0.1600 - val_accuracy: 0.9546
Epoch 4/5
150/150 [==============================] - 275s 2s/step - loss: 0.1187
- accuracy: 0.9656 - val_loss: 0.1195 - val_accuracy: 0.9663
Epoch 5/5
150/150 [==============================] - 267s 2s/step - loss: 0.0955
- accuracy: 0.9716 - val_loss: 0.1436 - val_accuracy: 0.9613
938/938 [==============================] - 70s 74ms/step - loss: 0.1436
- accuracy: 0.9613
> 96.130
Test time: 70s
Training time: 277s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=240, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
125/125 [==============================] - 263s 2s/step - loss: 15.6976
- accuracy: 0.0989 - val_loss: 2.3024 - val_accuracy: 0.1023
Epoch 2/5
125/125 [==============================] - 260s 2s/step - loss: 2.2537
- accuracy: 0.0930 - val_loss: 2.2866 - val_accuracy: 0.1009
Epoch 3/5
125/125 [==============================] - 258s 2s/step - loss: 2.2986
- accuracy: 0.1042 - val_loss: 2.3020 - val_accuracy: 0.1118
Epoch 4/5
125/125 [==============================] - 259s 2s/step - loss: 2.3014
- accuracy: 0.1129 - val_loss: 2.3017 - val_accuracy: 0.1118
Epoch 5/5
125/125 [==============================] - 260s 2s/step - loss: 2.3011
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
938/938 [==============================] - 59s 62ms/step - loss: 2.3016
- accuracy: 0.1118
> 11.180
Test time: 59s
Training time: 260s
Epoch 1/5
125/125 [==============================] - 256s 2s/step - loss: 15.6656
- accuracy: 0.6499 - val_loss: 0.3548 - val_accuracy: 0.8997
Epoch 2/5
125/125 [==============================] - 255s 2s/step - loss: 0.2799
- accuracy: 0.9196 - val_loss: 0.2308 - val_accuracy: 0.9363
Epoch 3/5
125/125 [==============================] - 255s 2s/step - loss: 0.1906
- accuracy: 0.9454 - val_loss: 0.1657 - val_accuracy: 0.9532
Epoch 4/5
125/125 [==============================] - 255s 2s/step - loss: 0.1417
- accuracy: 0.9597 - val_loss: 0.1688 - val_accuracy: 0.9539
Epoch 5/5
125/125 [==============================] - 255s 2s/step - loss: 0.1128
- accuracy: 0.9676 - val_loss: 0.1631 - val_accuracy: 0.9548
938/938 [==============================] - 58s 62ms/step - loss: 0.1631
- accuracy: 0.9548
> 95.483
Test time: 66s
Training time: 255.2s
```

padding="same", activation function="relu" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=230, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  679478
Epoch 1/5
131/131 [==============================] - 360s 3s/step - loss: 14.0676
- accuracy: 0.1110 - val_loss: 2.3021 - val_accuracy: 0.1118
Epoch 2/5
131/131 [==============================] - 360s 3s/step - loss: 2.3018
- accuracy: 0.1129 - val_loss: 2.3017 - val_accuracy: 0.1118
Epoch 3/5
131/131 [==============================] - 359s 3s/step - loss: 2.3014
- accuracy: 0.1129 - val_loss: 2.3016 - val_accuracy: 0.1118
Epoch 4/5
131/131 [==============================] - 358s 3s/step - loss: 2.3011
- accuracy: 0.1129 - val_loss: 2.3015 - val_accuracy: 0.1118
Epoch 5/5
131/131 [==============================] - 360s 3s/step - loss: 2.3010
- accuracy: 0.1129 - val_loss: 2.3015 - val_accuracy: 0.1118
938/938 [==============================] - 79s 84ms/step - loss: 2.3015
- accuracy: 0.1118
> 11.180
Test time: 79s
Training time: 359.4
Epoch 1/5
131/131 [==============================] - 356s 3s/step - loss: 12.2485
- accuracy: 0.1092 - val_loss: 2.3020 - val_accuracy: 0.1129
Epoch 2/5
131/131 [==============================] - 356s 3s/step - loss: 2.3020
- accuracy: 0.1118 - val_loss: 2.3016 - val_accuracy: 0.1129
Epoch 3/5
131/131 [==============================] - 354s 3s/step - loss: 2.3017
- accuracy: 0.1118 - val_loss: 2.3013 - val_accuracy: 0.1129
Epoch 4/5
131/131 [==============================] - 354s 3s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3012 - val_accuracy: 0.1129
Epoch 5/5
131/131 [==============================] - 353s 3s/step - loss: 2.3016
- accuracy: 0.1118 - val_loss: 2.3011 - val_accuracy: 0.1129
938/938 [==============================] - 78s 84ms/step - loss: 2.3011
- accuracy: 0.1129
> 11.293
Test time: 78s
Training time: 354.6s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (6*6) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=100, Optimizer= SGD, learning_rate=0.1 and momentum=0.3 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
300/300 [==============================] - 296s 984ms/step - loss: 2.2783 - accuracy: 0.1774 - val_loss: 2.1900 - val_accuracy: 0.4839
Epoch 2/5
300/300 [==============================] - 291s 971ms/step - loss: 1.6608 - accuracy: 0.5461 - val_loss: 1.0970 - val_accuracy: 0.6894
Epoch 3/5
300/300 [==============================] - 292s 975ms/step - loss: 0.8231 - accuracy: 0.7822 - val_loss: 0.6464 - val_accuracy: 0.8355
Epoch 4/5
300/300 [==============================] - 291s 969ms/step - loss: 0.4870 - accuracy: 0.8868 - val_loss: 0.3838 - val_accuracy: 0.9125
Epoch 5/5
300/300 [==============================] - 293s 976ms/step - loss: 0.3121 - accuracy: 0.9258 - val_loss: 0.3300 - val_accuracy: 0.9094
938/938 [==============================] - 65s 70ms/step - loss: 0.3300 - accuracy: 0.9094
> 90.943
Test time:  65s
Training time: 292.6s
Epoch 1/5
300/300 [==============================] - 293s 977ms/step - loss: 2.1979 - accuracy: 0.2813 - val_loss: 1.8296 - val_accuracy: 0.5522
Epoch 2/5
300/300 [==============================] - 291s 971ms/step - loss: 1.2009 - accuracy: 0.6884 - val_loss: 0.7636 - val_accuracy: 0.8082
Epoch 3/5
300/300 [==============================] - 292s 973ms/step - loss: 0.5832 - accuracy: 0.8598 - val_loss: 0.4205 - val_accuracy: 0.9066
Epoch 4/5
300/300 [==============================] - 292s 973ms/step - loss: 0.3456 - accuracy: 0.9196 - val_loss: 0.2658 - val_accuracy: 0.9387
Epoch 5/5
300/300 [==============================] - 290s 968ms/step - loss: 0.2459 - accuracy: 0.9391 - val_loss: 0.2118 - val_accuracy: 0.9480
938/938 [==============================] - 65s 69ms/step - loss: 0.2118 - accuracy: 0.9480
> 94.800
Test time:  65s
Training time: 291.6s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (6*6) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=100, Optimizer= SGD, learning_rate=0.5 and momentum=0.7 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
300/300 [==============================] - 295s 982ms/step - loss:
1.2944 - accuracy: 0.5410 - val_loss: 0.3631 - val_accuracy: 0.8903
Epoch 2/5
300/300 [==============================] - 292s 974ms/step - loss:
0.6279 - accuracy: 0.8102 - val_loss: 0.3624 - val_accuracy: 0.8852
Epoch 3/5
300/300 [==============================] - 293s 977ms/step - loss:
0.6420 - accuracy: 0.8137 - val_loss: 0.7655 - val_accuracy: 0.7417
Epoch 4/5
300/300 [==============================] - 292s 974ms/step - loss:
0.3353 - accuracy: 0.8990 - val_loss: 0.2154 - val_accuracy: 0.9365
Epoch 5/5
300/300 [==============================] - 292s 973ms/step - loss:
0.2758 - accuracy: 0.9154 - val_loss: 0.2366 - val_accuracy: 0.9281
938/938 [==============================] - 66s 70ms/step - loss: 0.2366
- accuracy: 0.9281
> 92.810
Test time: 66s
Training time: 292.8s
Epoch 1/5
300/300 [==============================] - 292s 972ms/step - loss:
1.1758 - accuracy: 0.5888 - val_loss: 0.4829 - val_accuracy: 0.8355
Epoch 2/5
300/300 [==============================] - 292s 975ms/step - loss:
0.3390 - accuracy: 0.8933 - val_loss: 0.2101 - val_accuracy: 0.9368
Epoch 3/5
300/300 [==============================] - 292s 975ms/step - loss:
0.2251 - accuracy: 0.9290 - val_loss: 0.2570 - val_accuracy: 0.9183
Epoch 4/5
300/300 [==============================] - 292s 976ms/step - loss:
0.2068 - accuracy: 0.9345 - val_loss: 0.2529 - val_accuracy: 0.9130
Epoch 5/5
300/300 [==============================] - 292s 974ms/step - loss:
0.1849 - accuracy: 0.9425 - val_loss: 0.1514 - val_accuracy: 0.9537
938/938 [==============================] - 67s 71ms/step - loss: 0.1514
- accuracy: 0.9537
> 95.373
Test time: 67s
Training time: 292s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (6*6) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=50, Optimizer= SGD, learning_rate=0.6 and momentum=0.8 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
600/600 [==============================] - 304s 506ms/step - loss:
2.2702 - accuracy: 0.1338 - val_loss: 2.3138 - val_accuracy: 0.1118
Epoch 2/5
600/600 [==============================] - 301s 502ms/step - loss:
2.3133 - accuracy: 0.1062 - val_loss: 2.3159 - val_accuracy: 0.1118
Epoch 3/5
600/600 [==============================] - 304s 506ms/step - loss:
2.3124 - accuracy: 0.1093 - val_loss: 2.3276 - val_accuracy: 0.1118
Epoch 4/5
600/600 [==============================] - 303s 505ms/step - loss:
2.3115 - accuracy: 0.1061 - val_loss: 2.3211 - val_accuracy: 0.1033
Epoch 5/5
600/600 [==============================] - 302s 503ms/step - loss:
2.3163 - accuracy: 0.0998 - val_loss: 2.3185 - val_accuracy: 0.0977
938/938 [==============================] - 66s 70ms/step - loss: 2.3185
- accuracy: 0.0977
> 9.770
Test time: 66s
Training time: 302.8s
Epoch 1/5
600/600 [==============================] - 301s 501ms/step - loss:
2.2224 - accuracy: 0.1488 - val_loss: 2.0669 - val_accuracy: 0.2508
Epoch 2/5
600/600 [==============================] - 302s 503ms/step - loss:
2.3560 - accuracy: 0.1116 - val_loss: 2.4343 - val_accuracy: 0.1022
Epoch 3/5
600/600 [==============================] - 304s 507ms/step - loss:
2.2948 - accuracy: 0.1416 - val_loss: 2.1377 - val_accuracy: 0.1975
Epoch 4/5
600/600 [==============================] - 306s 509ms/step - loss:
2.3879 - accuracy: 0.1045 - val_loss: 2.3760 - val_accuracy: 0.1129
Epoch 5/5
600/600 [==============================] - 305s 508ms/step - loss:
2.3922 - accuracy: 0.1066 - val_loss: 2.4175 - val_accuracy: 0.1056
938/938 [==============================] - 67s 71ms/step - loss: 2.4175
- accuracy: 0.1056
> 10.557
Test time: 67s
Training time: 303.6s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (6*6) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=50, Optimizer= SGD, learning_rate=0.1 and momentum=0.2 and 20 Neurons and 10 Neurons at o/p layer :-

```
number of parameters =  671798
Epoch 1/5
600/600 [==============================] - 306s 510ms/step - loss:
2.0370 - accuracy: 0.2931 - val_loss: 1.3285 - val_accuracy: 0.6164
Epoch 2/5
600/600 [==============================] - 305s 508ms/step - loss:
0.8334 - accuracy: 0.7672 - val_loss: 0.5245 - val_accuracy: 0.8692
Epoch 3/5
600/600 [==============================] - 304s 508ms/step - loss:
0.3834 - accuracy: 0.9072 - val_loss: 0.2831 - val_accuracy: 0.9296
Epoch 4/5
600/600 [==============================] - 303s 506ms/step - loss:
0.2465 - accuracy: 0.9374 - val_loss: 0.2106 - val_accuracy: 0.9470
Epoch 5/5
600/600 [==============================] - 307s 511ms/step - loss:
0.1896 - accuracy: 0.9478 - val_loss: 0.1947 - val_accuracy: 0.9452
938/938 [==============================] - 66s 70ms/step - loss: 0.1947
- accuracy: 0.9452
> 94.523
Test time: 66s
Training time: 305s
Epoch 1/5
600/600 [==============================] - 299s 497ms/step - loss:
2.0138 - accuracy: 0.2912 - val_loss: 1.3618 - val_accuracy: 0.5731
Epoch 2/5
600/600 [==============================] - 299s 499ms/step - loss:
0.9115 - accuracy: 0.7357 - val_loss: 0.6058 - val_accuracy: 0.8418
Epoch 3/5
600/600 [==============================] - 298s 497ms/step - loss:
0.4359 - accuracy: 0.8893 - val_loss: 0.2978 - val_accuracy: 0.9262
Epoch 4/5
600/600 [==============================] - 297s 495ms/step - loss:
0.2593 - accuracy: 0.9321 - val_loss: 0.1909 - val_accuracy: 0.9528
Epoch 5/5
600/600 [==============================] - 297s 496ms/step - loss:
0.1898 - accuracy: 0.9488 - val_loss: 0.1559 - val_accuracy: 0.9590
938/938 [==============================] - 67s 71ms/step - loss: 0.1559
- accuracy: 0.9590
> 95.897
Test time: 67s
Training time:  298s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (6*6) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=100, Optimizer= SGD, learning_rate=0.2 and momentum=0.4 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
300/300 [==============================] - 299s 996ms/step - loss:
1.7408 - accuracy: 0.4257 - val_loss: 1.1132 - val_accuracy: 0.6035
Epoch 2/5
300/300 [==============================] - 308s 1s/step - loss: 0.5735
- accuracy: 0.8484 - val_loss: 0.3113 - val_accuracy: 0.9263
Epoch 3/5
300/300 [==============================] - 311s 1s/step - loss: 0.2677
- accuracy: 0.9294 - val_loss: 0.2218 - val_accuracy: 0.9397
Epoch 4/5
300/300 [==============================] - 312s 1s/step - loss: 0.1910
- accuracy: 0.9468 - val_loss: 0.1772 - val_accuracy: 0.9504
Epoch 5/5
300/300 [==============================] - 314s 1s/step - loss: 0.1636
- accuracy: 0.9522 - val_loss: 0.1435 - val_accuracy: 0.9579
938/938 [==============================] - 68s 72ms/step - loss: 0.1435
- accuracy: 0.9579
> 95.787
Test time: 68s
Training time: 308.8s
Epoch 1/5
300/300 [==============================] - 294s 980ms/step - loss:
1.7588 - accuracy: 0.3953 - val_loss: 0.9430 - val_accuracy: 0.7402
Epoch 2/5
300/300 [==============================] - 293s 976ms/step - loss:
0.6174 - accuracy: 0.8306 - val_loss: 0.3763 - val_accuracy: 0.9051
Epoch 3/5
300/300 [==============================] - 294s 979ms/step - loss:
0.3138 - accuracy: 0.9158 - val_loss: 0.2367 - val_accuracy: 0.9358
Epoch 4/5
300/300 [==============================] - 295s 982ms/step - loss:
0.2133 - accuracy: 0.9408 - val_loss: 0.2126 - val_accuracy: 0.9385
Epoch 5/5
300/300 [==============================] - 293s 977ms/step - loss:
0.1583 - accuracy: 0.9561 - val_loss: 0.1404 - val_accuracy: 0.9603
938/938 [==============================] - 65s 70ms/step - loss: 0.1404
- accuracy: 0.9603
> 96.030
Test time: 65s
Training time:  293.8s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (6*6) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=200, Optimizer= SGD, learning_rate=0.1 and momentum=0.4 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
150/150 [==============================] - 295s 2s/step - loss: 2.2874
- accuracy: 0.1552 - val_loss: 2.2504 - val_accuracy: 0.2459
Epoch 2/5
150/150 [==============================] - 294s 2s/step - loss: 2.0380
- accuracy: 0.3600 - val_loss: 1.7274 - val_accuracy: 0.4865
Epoch 3/5
150/150 [==============================] - 294s 2s/step - loss: 1.4579
- accuracy: 0.5723 - val_loss: 1.2232 - val_accuracy: 0.6649
Epoch 4/5
150/150 [==============================] - 294s 2s/step - loss: 1.0393
- accuracy: 0.7186 - val_loss: 0.8806 - val_accuracy: 0.7679
Epoch 5/5
150/150 [==============================] - 294s 2s/step - loss: 0.7403
- accuracy: 0.8199 - val_loss: 0.6165 - val_accuracy: 0.8561
938/938 [==============================] - 67s 71ms/step - loss: 0.6165
- accuracy: 0.8561
> 85.610
Test time: 67s
Training time:  294.2s
Epoch 1/5
150/150 [==============================] - 296s 2s/step - loss: 2.2872
- accuracy: 0.1591 - val_loss: 2.2493 - val_accuracy: 0.1129
Epoch 2/5
150/150 [==============================] - 294s 2s/step - loss: 2.0762
- accuracy: 0.4132 - val_loss: 1.7496 - val_accuracy: 0.5531
Epoch 3/5
150/150 [==============================] - 295s 2s/step - loss: 1.3947
- accuracy: 0.6156 - val_loss: 1.1176 - val_accuracy: 0.6773
Epoch 4/5
150/150 [==============================] - 295s 2s/step - loss: 0.9340
- accuracy: 0.7381 - val_loss: 0.7779 - val_accuracy: 0.7951
Epoch 5/5
150/150 [==============================] - 294s 2s/step - loss: 0.6717
- accuracy: 0.8330 - val_loss: 0.5710 - val_accuracy: 0.8719
938/938 [==============================] - 67s 71ms/step - loss: 0.5710
- accuracy: 0.8719
> 87.190
Test time: 67s
Training time: 294.8s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=100, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
300/300 [==============================] - 301s 1s/step - loss: 1.8367
- accuracy: 0.5300 - val_loss: 1.0886 - val_accuracy: 0.9058
Epoch 2/5
300/300 [==============================] - 300s 1s/step - loss: 0.6409
- accuracy: 0.9384 - val_loss: 0.3858 - val_accuracy: 0.9482
Epoch 3/5
300/300 [==============================] - 301s 1s/step - loss: 0.2603
- accuracy: 0.9643 - val_loss: 0.1997 - val_accuracy: 0.9676
Epoch 4/5
300/300 [==============================] - 300s 1s/step - loss: 0.1551
- accuracy: 0.9739 - val_loss: 0.1505 - val_accuracy: 0.9685
Epoch 5/5
300/300 [==============================] - 300s 1s/step - loss: 0.1136
- accuracy: 0.9776 - val_loss: 0.1186 - val_accuracy: 0.9732
938/938 [==============================] - 67s 72ms/step - loss: 0.1186
- accuracy: 0.9732
> 97.317
Test time: 67s
Training time:  300.4s
Epoch 1/5
300/300 [==============================] - 300s 999ms/step - loss:
1.9449 - accuracy: 0.3927 - val_loss: 1.2559 - val_accuracy: 0.7643
Epoch 2/5
300/300 [==============================] - 300s 1000ms/step - loss:
0.7911 - accuracy: 0.8574 - val_loss: 0.4706 - val_accuracy: 0.9449
Epoch 3/5
300/300 [==============================] - 300s 1s/step - loss: 0.3262
- accuracy: 0.9553 - val_loss: 0.2214 - val_accuracy: 0.9635
Epoch 4/5
300/300 [==============================] - 300s 1s/step - loss: 0.1771
- accuracy: 0.9691 - val_loss: 0.1722 - val_accuracy: 0.9646
Epoch 5/5
300/300 [==============================] - 300s 1s/step - loss: 0.1262
- accuracy: 0.9748 - val_loss: 0.1278 - val_accuracy: 0.9726
938/938 [==============================] - 69s 73ms/step - loss: 0.1278
- accuracy: 0.9726
> 97.263
Test time: 69s
Training time: 300s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=200, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
150/150 [==============================] - 298s 2s/step - loss: 1.9945
- accuracy: 0.3706 - val_loss: 1.4641 - val_accuracy: 0.7982
Epoch 2/5
150/150 [==============================] - 297s 2s/step - loss: 1.0392
- accuracy: 0.8993 - val_loss: 0.7060 - val_accuracy: 0.9360
Epoch 3/5
150/150 [==============================] - 296s 2s/step - loss: 0.5116
- accuracy: 0.9530 - val_loss: 0.3793 - val_accuracy: 0.9572
Epoch 4/5
150/150 [==============================] - 296s 2s/step - loss: 0.2915
- accuracy: 0.9685 - val_loss: 0.2373 - val_accuracy: 0.9700
Epoch 5/5
150/150 [==============================] - 296s 2s/step - loss: 0.1901
- accuracy: 0.9753 - val_loss: 0.1667 - val_accuracy: 0.9748
938/938 [==============================] - 67s 71ms/step - loss: 0.1667
- accuracy: 0.9748
> 97.477
Test time: 67s
Training time:  296.6s
Epoch 1/5
150/150 [==============================] - 297s 2s/step - loss: 2.0356
- accuracy: 0.4133 - val_loss: 1.5403 - val_accuracy: 0.7399
Epoch 2/5
150/150 [==============================] - 297s 2s/step - loss: 1.1654
- accuracy: 0.8440 - val_loss: 0.8461 - val_accuracy: 0.9225
Epoch 3/5
150/150 [==============================] - 297s 2s/step - loss: 0.6331
- accuracy: 0.9397 - val_loss: 0.4492 - val_accuracy: 0.9573
Epoch 4/5
150/150 [==============================] - 296s 2s/step - loss: 0.3600
- accuracy: 0.9592 - val_loss: 0.2793 - val_accuracy: 0.9641
Epoch 5/5
150/150 [==============================] - 296s 2s/step - loss: 0.2377
- accuracy: 0.9669 - val_loss: 0.2021 - val_accuracy: 0.9700
938/938 [==============================] - 67s 71ms/step - loss: 0.2021
- accuracy: 0.9700
> 97.003
Test time: 67s
Training time: 296.6s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=100, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  679478
Epoch 1/5
300/300 [==============================] - 367s 1s/step - loss: 1.1234
- accuracy: 0.7321 - val_loss: 0.3937 - val_accuracy: 0.9491
Epoch 2/5
300/300 [==============================] - 366s 1s/step - loss: 0.2625
- accuracy: 0.9617 - val_loss: 0.1852 - val_accuracy: 0.9687
Epoch 3/5
300/300 [==============================] - 365s 1s/step - loss: 0.1431
- accuracy: 0.9749 - val_loss: 0.1186 - val_accuracy: 0.9766
Epoch 4/5
300/300 [==============================] - 364s 1s/step - loss: 0.1007
- accuracy: 0.9796 - val_loss: 0.1104 - val_accuracy: 0.9734
Epoch 5/5
300/300 [==============================] - 365s 1s/step - loss: 0.0766
- accuracy: 0.9837 - val_loss: 0.0933 - val_accuracy: 0.9772
938/938 [==============================] - 79s 85ms/step - loss: 0.0933
- accuracy: 0.9772
> 97.720
Test time: 79s
Training time: 365.4s
Epoch 1/5
300/300 [==============================] - 364s 1s/step - loss: 1.5312
- accuracy: 0.5722 - val_loss: 0.5802 - val_accuracy: 0.9338
Epoch 2/5
300/300 [==============================] - 366s 1s/step - loss: 0.3314
- accuracy: 0.9543 - val_loss: 0.1977 - val_accuracy: 0.9680
Epoch 3/5
300/300 [==============================] - 365s 1s/step - loss: 0.1647
- accuracy: 0.9702 - val_loss: 0.1342 - val_accuracy: 0.9739
Epoch 4/5
300/300 [==============================] - 366s 1s/step - loss: 0.1179
- accuracy: 0.9755 - val_loss: 0.1080 - val_accuracy: 0.9757
Epoch 5/5
300/300 [==============================] - 366s 1s/step - loss: 0.0913
- accuracy: 0.9806 - val_loss: 0.0868 - val_accuracy: 0.9793
938/938 [==============================] - 79s 84ms/step - loss: 0.0868
- accuracy: 0.9793
> 97.933
Test time: 79s
Training time: 365.4s
```

padding="same", activation function="sigmoid" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=150, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  679478
Epoch 1/5
200/200 [==============================] - 363s 2s/step - loss: 1.5189
- accuracy: 0.6022 - val_loss: 0.7405 - val_accuracy: 0.8745
Epoch 2/5
200/200 [==============================] - 362s 2s/step - loss: 0.4777
- accuracy: 0.9380 - val_loss: 0.3227 - val_accuracy: 0.9549
Epoch 3/5
200/200 [==============================] - 362s 2s/step - loss: 0.2301
- accuracy: 0.9675 - val_loss: 0.1780 - val_accuracy: 0.9724
Epoch 4/5
200/200 [==============================] - 365s 2s/step - loss: 0.1445
- accuracy: 0.9767 - val_loss: 0.1297 - val_accuracy: 0.9769
Epoch 5/5
200/200 [==============================] - 363s 2s/step - loss: 0.1039
- accuracy: 0.9821 - val_loss: 0.1020 - val_accuracy: 0.9791
938/938 [==============================] - 79s 84ms/step - loss: 0.1020
- accuracy: 0.9791
> 97.907
Test time: 79s
Training time: 363s
Epoch 1/5
200/200 [==============================] - 359s 2s/step - loss: 1.3451
- accuracy: 0.6863 - val_loss: 0.6369 - val_accuracy: 0.9264
Epoch 2/5
200/200 [==============================] - 358s 2s/step - loss: 0.4224
- accuracy: 0.9474 - val_loss: 0.2753 - val_accuracy: 0.9615
Epoch 3/5
200/200 [==============================] - 358s 2s/step - loss: 0.2134
- accuracy: 0.9692 - val_loss: 0.1698 - val_accuracy: 0.9731
Epoch 4/5
200/200 [==============================] - 357s 2s/step - loss: 0.1413
- accuracy: 0.9765 - val_loss: 0.1325 - val_accuracy: 0.9741
Epoch 5/5
200/200 [==============================] - 359s 2s/step - loss: 0.1016
- accuracy: 0.9817 - val_loss: 0.0957 - val_accuracy: 0.9810
938/938 [==============================] - 79s 85ms/step - loss: 0.0957
- accuracy: 0.9810
> 98.097
Test time: 79s
Training time: 358.2s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=100, Optimizer= SGD, learning_rate=0.1 and momentum=0.3 and 25 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  67665
Epoch 1/5
300/300 [==============================] - 110s 365ms/step - loss:
0.8797 - accuracy: 0.7124 - val_loss: 0.3449 - val_accuracy: 0.8969
Epoch 2/5
300/300 [==============================] - 110s 367ms/step - loss:
0.3244 - accuracy: 0.9037 - val_loss: 0.2609 - val_accuracy: 0.9222
Epoch 3/5
300/300 [==============================] - 110s 367ms/step - loss:
0.2421 - accuracy: 0.9266 - val_loss: 0.2165 - val_accuracy: 0.9364
Epoch 4/5
300/300 [==============================] - 111s 370ms/step - loss:
0.2013 - accuracy: 0.9390 - val_loss: 0.2017 - val_accuracy: 0.9385
Epoch 5/5
300/300 [==============================] - 110s 368ms/step - loss:
0.1976 - accuracy: 0.9391 - val_loss: 0.2237 - val_accuracy: 0.9309
938/938 [==============================] - 31s 33ms/step - loss: 0.2237
- accuracy: 0.9309
> 93.093
Test time: 31s
Training time: 110.2s
Epoch 1/5
300/300 [==============================] - 110s 365ms/step - loss:
0.6892 - accuracy: 0.7850 - val_loss: 0.2979 - val_accuracy: 0.9115
Epoch 2/5
300/300 [==============================] - 109s 364ms/step - loss:
0.2772 - accuracy: 0.9146 - val_loss: 0.2223 - val_accuracy: 0.9315
Epoch 3/5
300/300 [==============================] - 109s 365ms/step - loss:
0.2187 - accuracy: 0.9345 - val_loss: 0.1854 - val_accuracy: 0.9449
Epoch 4/5
300/300 [==============================] - 109s 362ms/step - loss:
0.2003 - accuracy: 0.9377 - val_loss: 0.2012 - val_accuracy: 0.9388
Epoch 5/5
300/300 [==============================] - 110s 366ms/step - loss:
0.1862 - accuracy: 0.9431 - val_loss: 0.1739 - val_accuracy: 0.9471
938/938 [==============================] - 31s 33ms/step - loss: 0.1739
- accuracy: 0.9471
> 94.713
Test time: 31s
Training time: 109.4s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=120, Optimizer= SGD, learning_rate=0.2 and momentum=0.4 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  134060
Epoch 1/5
250/250 [==============================] - 230s 917ms/step - loss:
1.4722 - accuracy: 0.4730 - val_loss: 0.6461 - val_accuracy: 0.7828
Epoch 2/5
250/250 [==============================] - 228s 914ms/step - loss:
0.3575 - accuracy: 0.8906 - val_loss: 0.3841 - val_accuracy: 0.8779
Epoch 3/5
250/250 [==============================] - 228s 914ms/step - loss:
0.2399 - accuracy: 0.9268 - val_loss: 0.2104 - val_accuracy: 0.9347
Epoch 4/5
250/250 [==============================] - 226s 905ms/step - loss:
0.1867 - accuracy: 0.9437 - val_loss: 0.1990 - val_accuracy: 0.9376
Epoch 5/5
250/250 [==============================] - 227s 907ms/step - loss:
0.1801 - accuracy: 0.9447 - val_loss: 0.2056 - val_accuracy: 0.9370
938/938 [==============================] - 60s 64ms/step - loss: 0.2056
- accuracy: 0.9370
> 93.703
Test time: 60s
Training time: 227.8s
Epoch 1/5
250/250 [==============================] - 228s 910ms/step - loss:
2.0055 - accuracy: 0.2593 - val_loss: 1.0489 - val_accuracy: 0.6319
Epoch 2/5
250/250 [==============================] - 227s 907ms/step - loss:
1.0590 - accuracy: 0.6536 - val_loss: 1.2916 - val_accuracy: 0.5443
Epoch 3/5
250/250 [==============================] - 226s 905ms/step - loss:
0.4906 - accuracy: 0.8529 - val_loss: 0.3742 - val_accuracy: 0.8940
Epoch 4/5
250/250 [==============================] - 227s 907ms/step - loss:
0.3739 - accuracy: 0.8881 - val_loss: 0.4798 - val_accuracy: 0.8511
Epoch 5/5
250/250 [==============================] - 227s 907ms/step - loss:
0.3358 - accuracy: 0.8991 - val_loss: 0.2610 - val_accuracy: 0.9242
938/938 [==============================] - 60s 64ms/step - loss: 0.2610
- accuracy: 0.9242
> 92.417
Test time: 60s
Training time: 227s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (5*5) and another Pooling (Max_Pooling) = (5*5) and SGD with batch_size=130, Optimizer= SGD, learning_rate=0.3 and momentum=0.5 and 22 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  64540
Epoch 1/5
231/231 [==============================] - 168s 294ms/step - loss:
1.6305 - accuracy: 0.4223 - val_loss: 1.0120 - val_accuracy: 0.6773
Epoch 2/5
231/231 [==============================] - 67s 292ms/step - loss:
0.8697 - accuracy: 0.7171 - val_loss: 0.6535 - val_accuracy: 0.7942
Epoch 3/5
231/231 [==============================] - 68s 293ms/step - loss:
0.6343 - accuracy: 0.8014 - val_loss: 0.6602 - val_accuracy: 0.7882
Epoch 4/5
231/231 [==============================] - 68s 293ms/step - loss:
0.5289 - accuracy: 0.8358 - val_loss: 0.4689 - val_accuracy: 0.8526
Epoch 5/5
231/231 [==============================] - 67s 292ms/step - loss:
0.4657 - accuracy: 0.8562 - val_loss: 0.4872 - val_accuracy: 0.8493
938/938 [==============================] - 20s 22ms/step - loss: 0.4872
- accuracy: 0.8493
> 84.930
Test time: 20s
Training time: 87.5s
Epoch 1/5
231/231 [==============================] - 69s 298ms/step - loss:
2.2355 - accuracy: 0.1558 - val_loss: 2.3221 - val_accuracy: 0.0879
Epoch 2/5
231/231 [==============================] - 68s 293ms/step - loss:
2.3055 - accuracy: 0.1132 - val_loss: 2.3768 - val_accuracy: 0.1129
Epoch 3/5
231/231 [==============================] - 68s 294ms/step - loss:
1.9997 - accuracy: 0.2325 - val_loss: 1.6329 - val_accuracy: 0.3876
Epoch 4/5
231/231 [==============================] - 68s 293ms/step - loss:
1.1757 - accuracy: 0.5868 - val_loss: 0.9422 - val_accuracy: 0.6741
Epoch 5/5
231/231 [==============================] - 67s 292ms/step - loss:
0.8474 - accuracy: 0.7214 - val_loss: 0.8463 - val_accuracy: 0.7224
938/938 [==============================] - 20s 22ms/step - loss: 0.8463
- accuracy: 0.7224
> 72.243
Test time: 20s
Training time: 68s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=140, Optimizer= SGD, learning_rate=0.4 and momentum=0.6 and 27 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  126106
Epoch 1/5
215/215 [==============================] - 177s 820ms/step - loss:
2.3362 - accuracy: 0.1013 - val_loss: 2.3261 - val_accuracy: 0.1118
Epoch 2/5
215/215 [==============================] - 174s 810ms/step - loss:
2.3145 - accuracy: 0.1055 - val_loss: 2.3189 - val_accuracy: 0.1033
Epoch 3/5
215/215 [==============================] - 174s 809ms/step - loss:
2.3087 - accuracy: 0.1074 - val_loss: 2.3193 - val_accuracy: 0.0977
Epoch 4/5
215/215 [==============================] - 174s 808ms/step - loss:
2.3082 - accuracy: 0.1037 - val_loss: 2.3111 - val_accuracy: 0.1033
Epoch 5/5
215/215 [==============================] - 173s 806ms/step - loss:
2.3086 - accuracy: 0.1018 - val_loss: 2.3129 - val_accuracy: 0.1033
938/938 [==============================] - 49s 53ms/step - loss: 2.3129
- accuracy: 0.1033
> 10.327
Test time: 49s
Training time: 174.4s
Epoch 1/5
215/215 [==============================] - 172s 801ms/step - loss:
2.3426 - accuracy: 0.1014 - val_loss: 2.3594 - val_accuracy: 0.0879
Epoch 2/5
215/215 [==============================] - 172s 803ms/step - loss:
2.3139 - accuracy: 0.0993 - val_loss: 2.3067 - val_accuracy: 0.0990
Epoch 3/5
215/215 [==============================] - 172s 802ms/step - loss:
2.3105 - accuracy: 0.1035 - val_loss: 2.3144 - val_accuracy: 0.0979
Epoch 4/5
215/215 [==============================] - 173s 803ms/step - loss:
2.3081 - accuracy: 0.1043 - val_loss: 2.3107 - val_accuracy: 0.1129
Epoch 5/5
215/215 [==============================] - 172s 801ms/step - loss:
2.3081 - accuracy: 0.1024 - val_loss: 2.3060 - val_accuracy: 0.1056
938/938 [==============================] - 49s 53ms/step - loss: 2.3060
- accuracy: 0.1056
> 10.557
Test time: 49s
Training time: 172.2s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=150, Optimizer= SGD, learning_rate=0.1 and momentum=0.5 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
200/200 [==============================] - 311s 2s/step - loss: 1.6975
- accuracy: 0.4092 - val_loss: 0.5776 - val_accuracy: 0.8473
Epoch 2/5
200/200 [==============================] - 309s 2s/step - loss: 0.3988
- accuracy: 0.8832 - val_loss: 0.3487 - val_accuracy: 0.8952
Epoch 3/5
200/200 [==============================] - 308s 2s/step - loss: 0.2469
- accuracy: 0.9265 - val_loss: 0.2022 - val_accuracy: 0.9394
Epoch 4/5
200/200 [==============================] - 304s 2s/step - loss: 0.1717
- accuracy: 0.9497 - val_loss: 0.1604 - val_accuracy: 0.9543
Epoch 5/5
200/200 [==============================] - 304s 2s/step - loss: 0.1555
- accuracy: 0.9537 - val_loss: 0.1586 - val_accuracy: 0.9530
938/938 [==============================] - 68s 73ms/step - loss: 0.1586
- accuracy: 0.9530
> 95.303
Test time: 68s
Training time: 307.2s
Epoch 1/5
200/200 [==============================] - 300s 1s/step - loss: 2.2980
- accuracy: 0.1308 - val_loss: 2.1267 - val_accuracy: 0.2086
Epoch 2/5
200/200 [==============================] - 304s 2s/step - loss: 2.0124
- accuracy: 0.2458 - val_loss: 1.4329 - val_accuracy: 0.5282
Epoch 3/5
200/200 [==============================] - 307s 2s/step - loss: 0.7979
- accuracy: 0.7548 - val_loss: 0.4210 - val_accuracy: 0.8739
Epoch 4/5
200/200 [==============================] - 306s 2s/step - loss: 0.3284
- accuracy: 0.9054 - val_loss: 0.2459 - val_accuracy: 0.9285
Epoch 5/5
200/200 [==============================] - 302s 2s/step - loss: 0.2402
- accuracy: 0.9301 - val_loss: 0.2261 - val_accuracy: 0.9351
938/938 [==============================] - 69s 73ms/step - loss: 0.2261
- accuracy: 0.9351
> 93.510
Test time: 69s
Training time: 303.8s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=130, Optimizer= SGD, learning_rate=0.3 and momentum=0.3 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
231/231 [==============================] - 309s 1s/step - loss: 2.3255
- accuracy: 0.1062 - val_loss: 2.3202 - val_accuracy: 0.0977
Epoch 2/5
231/231 [==============================] - 308s 1s/step - loss: 2.3179
- accuracy: 0.1056 - val_loss: 2.3653 - val_accuracy: 0.0996
Epoch 3/5
231/231 [==============================] - 308s 1s/step - loss: 2.3166
- accuracy: 0.1085 - val_loss: 2.3208 - val_accuracy: 0.1118
Epoch 4/5
231/231 [==============================] - 308s 1s/step - loss: 2.3186
- accuracy: 0.1032 - val_loss: 2.3072 - val_accuracy: 0.1118
Epoch 5/5
231/231 [==============================] - 307s 1s/step - loss: 2.3138
- accuracy: 0.1063 - val_loss: 2.3310 - val_accuracy: 0.0993
938/938 [==============================] - 68s 72ms/step - loss: 2.3309
- accuracy: 0.0993
> 9.933
Test time: 68s
Training time: 308s
Epoch 1/5
231/231 [==============================] - 307s 1s/step - loss: 2.3277
- accuracy: 0.1016 - val_loss: 2.3170 - val_accuracy: 0.1129
Epoch 2/5
231/231 [==============================] - 307s 1s/step - loss: 2.3155
- accuracy: 0.1141 - val_loss: 2.4903 - val_accuracy: 0.1022
Epoch 3/5
231/231 [==============================] - 306s 1s/step - loss: 1.7163
- accuracy: 0.3693 - val_loss: 1.3232 - val_accuracy: 0.5264
Epoch 4/5
231/231 [==============================] - 309s 1s/step - loss: 0.7824
- accuracy: 0.7363 - val_loss: 0.5750 - val_accuracy: 0.8154
Epoch 5/5
231/231 [==============================] - 310s 1s/step - loss: 0.5019
- accuracy: 0.8399 - val_loss: 0.6294 - val_accuracy: 0.7951
938/938 [==============================] - 70s 74ms/step - loss: 0.6294
- accuracy: 0.7951
> 79.513
Test time: 70s
Training time: 307.8s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=200, Optimizer= SGD, learning_rate=0.2 and momentum=0.3 and 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
150/150 [==============================] - 305s 2s/step - loss: 2.3074
- accuracy: 0.1260 - val_loss: 2.2914 - val_accuracy: 0.0928
Epoch 2/5
150/150 [==============================] - 308s 2s/step - loss: 2.1174
- accuracy: 0.2185 - val_loss: 1.2670 - val_accuracy: 0.6128
Epoch 3/5
150/150 [==============================] - 312s 2s/step - loss: 1.0586
- accuracy: 0.6547 - val_loss: 0.8688 - val_accuracy: 0.7021
Epoch 4/5
150/150 [==============================] - 303s 2s/step - loss: 1.1827
- accuracy: 0.5990 - val_loss: 2.1506 - val_accuracy: 0.1832
Epoch 5/5
150/150 [==============================] - 303s 2s/step - loss: 0.9039
- accuracy: 0.6927 - val_loss: 0.3727 - val_accuracy: 0.9002
938/938 [==============================] - 68s 73ms/step - loss: 0.3727
- accuracy: 0.9002
> 90.023
Test time: 68s
Training time: 306.2s
Epoch 1/5
150/150 [==============================] - 299s 2s/step - loss: 2.0191
- accuracy: 0.2620 - val_loss: 0.9286 - val_accuracy: 0.7144
Epoch 2/5
150/150 [==============================] - 310s 2s/step - loss: 0.5257
- accuracy: 0.8450 - val_loss: 0.2392 - val_accuracy: 0.9335
Epoch 3/5
150/150 [==============================] - 304s 2s/step - loss: 0.2455
- accuracy: 0.9271 - val_loss: 0.2295 - val_accuracy: 0.9298
Epoch 4/5
150/150 [==============================] - 307s 2s/step - loss: 0.1954
- accuracy: 0.9429 - val_loss: 0.1844 - val_accuracy: 0.9447
Epoch 5/5
150/150 [==============================] - 305s 2s/step - loss: 0.1581
- accuracy: 0.9527 - val_loss: 0.1694 - val_accuracy: 0.9479
938/938 [==============================] - 72s 76ms/step - loss: 0.1694
- accuracy: 0.9479
> 94.790
Test time: 72s
Training time: 305s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (3*3) and another Pooling (Max_Pooling) = (3*3) and SGD with batch_size=150, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  679478
Epoch 1/5
200/200 [==============================] - 359s 2s/step - loss: 0.9881
- accuracy: 0.7521 - val_loss: 0.3762 - val_accuracy: 0.9406
Epoch 2/5
200/200 [==============================] - 359s 2s/step - loss: 0.2364
- accuracy: 0.9597 - val_loss: 0.2127 - val_accuracy: 0.9539
Epoch 3/5
200/200 [==============================] - 361s 2s/step - loss: 0.1408
- accuracy: 0.9703 - val_loss: 0.1190 - val_accuracy: 0.9733
Epoch 4/5
200/200 [==============================] - 361s 2s/step - loss: 0.1001
- accuracy: 0.9774 - val_loss: 0.0990 - val_accuracy: 0.9758
Epoch 5/5
200/200 [==============================] - 360s 2s/step - loss: 0.0876
- accuracy: 0.9785 - val_loss: 0.1078 - val_accuracy: 0.9719
938/938 [==============================] - 79s 84ms/step - loss: 0.1078
- accuracy: 0.9719
> 97.187
Test time: 79s
Training time: 360s
Epoch 1/5
200/200 [==============================] - 364s 2s/step - loss: 0.6192
- accuracy: 0.8539 - val_loss: 0.1963 - val_accuracy: 0.9611
Epoch 2/5
200/200 [==============================] - 365s 2s/step - loss: 0.1481
- accuracy: 0.9687 - val_loss: 0.1226 - val_accuracy: 0.9736
Epoch 3/5
200/200 [==============================] - 365s 2s/step - loss: 0.0962
- accuracy: 0.9780 - val_loss: 0.0980 - val_accuracy: 0.9759
Epoch 4/5
200/200 [==============================] - 365s 2s/step - loss: 0.0743
- accuracy: 0.9818 - val_loss: 0.0870 - val_accuracy: 0.9767
Epoch 5/5
200/200 [==============================] - 363s 2s/step - loss: 0.0647
- accuracy: 0.9839 - val_loss: 0.0868 - val_accuracy: 0.9754
938/938 [==============================] - 79s 84ms/step - loss: 0.0868
- accuracy: 0.9754
> 97.537
Test time: 79s
Training time: 364.4s
```

padding="same", activation function="tanh" , activation function at o/p layer="softmax" ,kernel_initializer='he_uniform', input_shape=(28, 28, 1), Convolution Conv2D with kernel size (4*4) and also 5 epochs and 2 folds and use Pooling (Max_Pooling) = (4*4) and another Pooling (Max_Pooling) = (4*4) and SGD with batch_size=220, Optimizer= ADAM, 20 Neurons and 10 Neurons at o/p layer :-

```
Number of parameters =  671798
Epoch 1/5
137/137 [==============================] - 258s 2s/step - loss: 0.8279 - accuracy: 0.7979 - val_loss: 0.3189 - val_accuracy: 0.9380
Epoch 2/5
137/137 [==============================] - 257s 2s/step - loss: 0.2165 - accuracy: 0.9560 - val_loss: 0.1739 - val_accuracy: 0.9619
Epoch 3/5
137/137 [==============================] - 257s 2s/step - loss: 0.1364 - accuracy: 0.9689 - val_loss: 0.1373 - val_accuracy: 0.9654
Epoch 4/5
137/137 [==============================] - 258s 2s/step - loss: 0.0984 - accuracy: 0.9779 - val_loss: 0.1478 - val_accuracy: 0.9601
Epoch 5/5
137/137 [==============================] - 259s 2s/step - loss: 0.0813 - accuracy: 0.9801 - val_loss: 0.1081 - val_accuracy: 0.9701
938/938 [==============================] - 58s 62ms/step - loss: 0.1081 - accuracy: 0.9701
> 97.010
Test time: 58s
Training time: 257.8s
Epoch 1/5
137/137 [==============================] - 259s 2s/step - loss: 1.0502 - accuracy: 0.7595 - val_loss: 0.3898 - val_accuracy: 0.9398
Epoch 2/5
137/137 [==============================] - 259s 2s/step - loss: 0.2616 - accuracy: 0.9557 - val_loss: 0.1917 - val_accuracy: 0.9617
Epoch 3/5
137/137 [==============================] - 261s 2s/step - loss: 0.1589 - accuracy: 0.9664 - val_loss: 0.1364 - val_accuracy: 0.9712
Epoch 4/5
137/137 [==============================] - 260s 2s/step - loss: 0.1180 - accuracy: 0.9737 - val_loss: 0.1492 - val_accuracy: 0.9594
Epoch 5/5
137/137 [==============================] - 261s 2s/step - loss: 0.1008 - accuracy: 0.9749 - val_loss: 0.1231 - val_accuracy: 0.9674
938/938 [==============================] - 59s 62ms/step - loss: 0.1231 - accuracy: 0.9674
> 96.737
Test time: 59s
Training time: 260s
```

Source code:

```
from matplotlib import pyplot as plt
from sklearn.model_selection import KFold
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.optimizers import Adam

# load train and test dataset
def load_dataset():
  # load dataset
  (X_train, Y_train), (X_test, Y_test) = mnist.load_data()
  # reshape dataset to have a single channel
  X_train = X_train.reshape((X_train.shape[0], 28, 28, 1))
  X_test = X_test.reshape((X_test.shape[0], 28, 28, 1))
  # one hot encode target values
  Y_train = to_categorical(Y_train)
  testY = to_categorical(Y_test)
  return X_train, Y_train, X_test, Y_test


# define cnn model
def define_model():
  model = Sequential()
  model.add(Conv2D(48,  (12, 12) ,padding="same", activation="relu",
kernel_initializer='he_uniform', input_shape=(28, 28, 1)))
kernel_initializer to initialise values in the Conv2D class before
actually training the model.
  model.add(MaxPooling2D((3, 3)))
  model.add(Conv2D(48,  (12, 12) ,padding="same", activation='relu',
kernel_initializer='he_uniform'))
  model.add(Conv2D(48,  (12, 12) ,padding="same", activation='relu',
kernel_initializer='he_uniform'))
  model.add(MaxPooling2D((3, 3)))
  model.add(Flatten())
#Flatten to return a copy of the array in one dimensional rather than
two dimensional a multidimensional array.
```

```
  model.add(Dense(20, activation='relu',
kernel_initializer='he_uniform'))
  model.add(Dense(10, activation='softmax'))
#Dense--->  implements the operation: output = activation(dot(input,
kernel) + bias)
# activation function : sigmoid , relu , relu
  # compile model
  opt = Adam() # optimizer : adam , sgd
  model.compile(optimizer=opt, loss='categorical_crossentropy',
metrics=['accuracy'])
  return model

# evaluate a model using k-fold cross-validation
def evaluate_model(dataX, dataY, n_folds=2):
  scores, histories = list(), list()
  # prepare cross validation
  kfold = KFold(n_folds, shuffle=True, random_state=1)
  # enumerate splits
  for train_ix, test_ix in kfold.split(dataX):
    # define model
    model = define_model()
    # select rows for train and test
    X_train, Y_train, X_test, Y_test = dataX[train_ix],
dataY[train_ix], dataX[test_ix], dataY[test_ix]
    # fit model
    history = model.fit(X_train, Y_train, epochs=5, batch_size=230,
validation_data=(X_test, Y_test))
    # evaluate model
    _, acc = model.evaluate(X_test, Y_test)
    print('> %.3f' % (acc * 100.0))

    # stores scores
    scores.append(acc)
    histories.append(history)
  return scores, histories
model = define_model()

print("Number of parameters = ",model.count_params())

def getAccuracy(scores):

  print('Accuracy:  n=%d' % (len(scores)))
```

```
def test():
  # load dataset
  X_train, Y_train, X_test, Y_test = load_dataset()
  # evaluate model
  scores, histories = evaluate_model(X_train, Y_train)


# entry point, run the test harness
test()
```