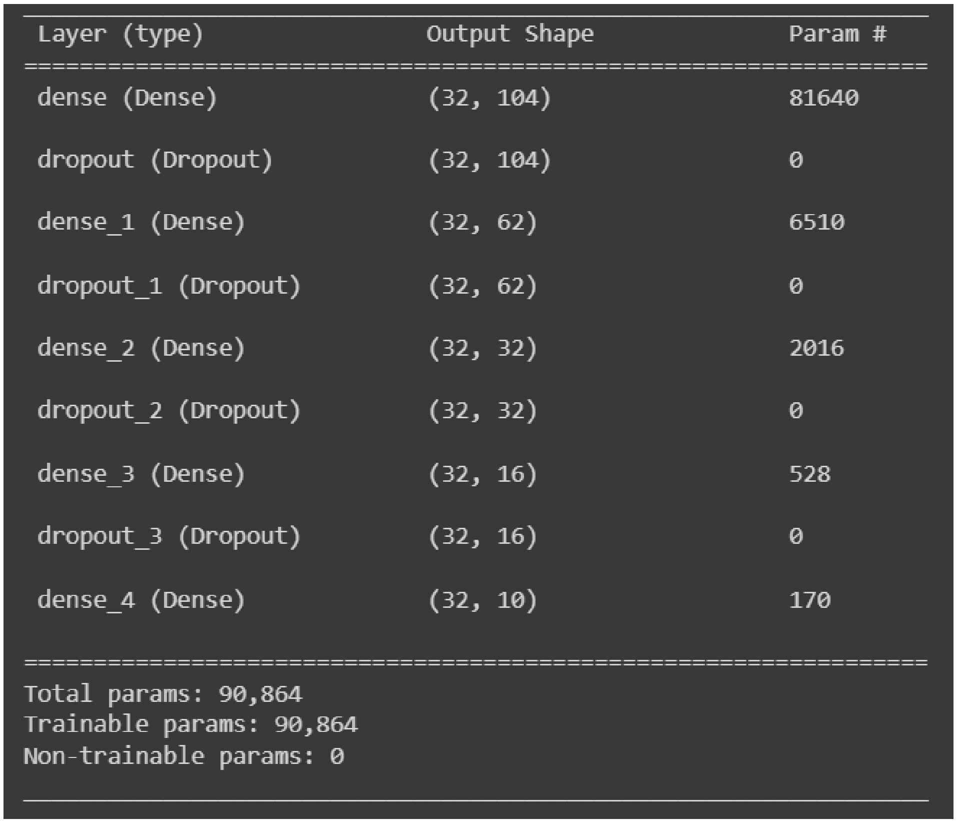
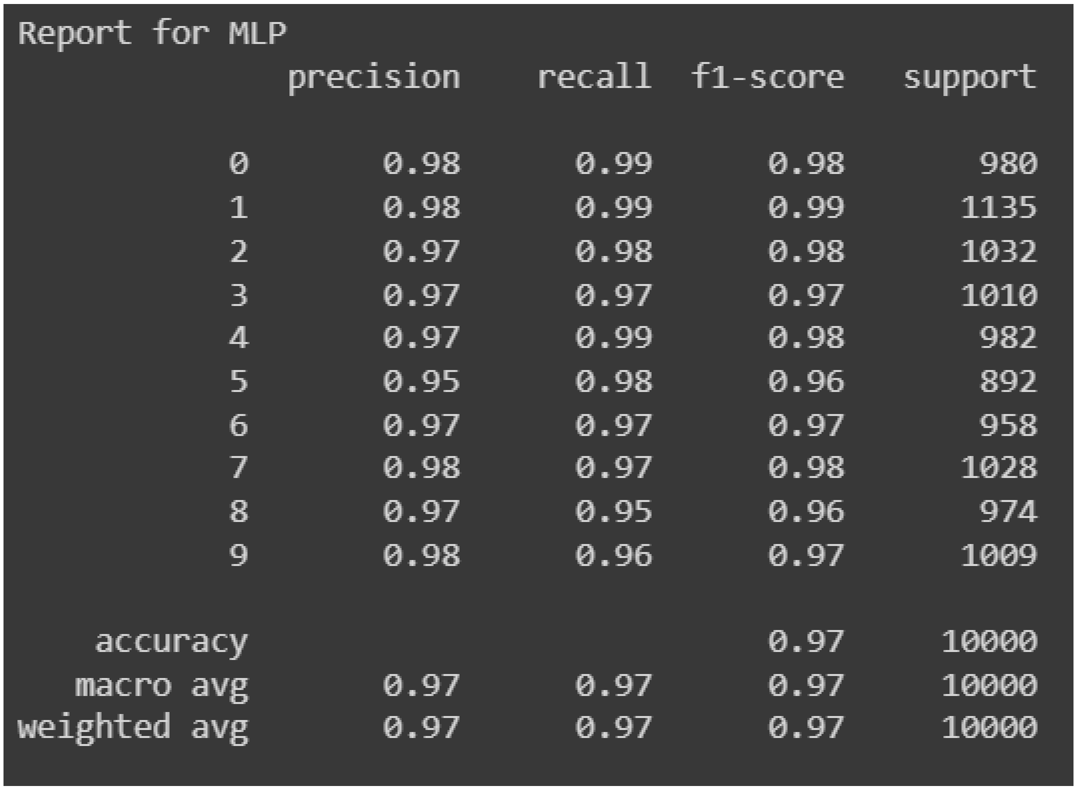
Analysis of MLP & CNN

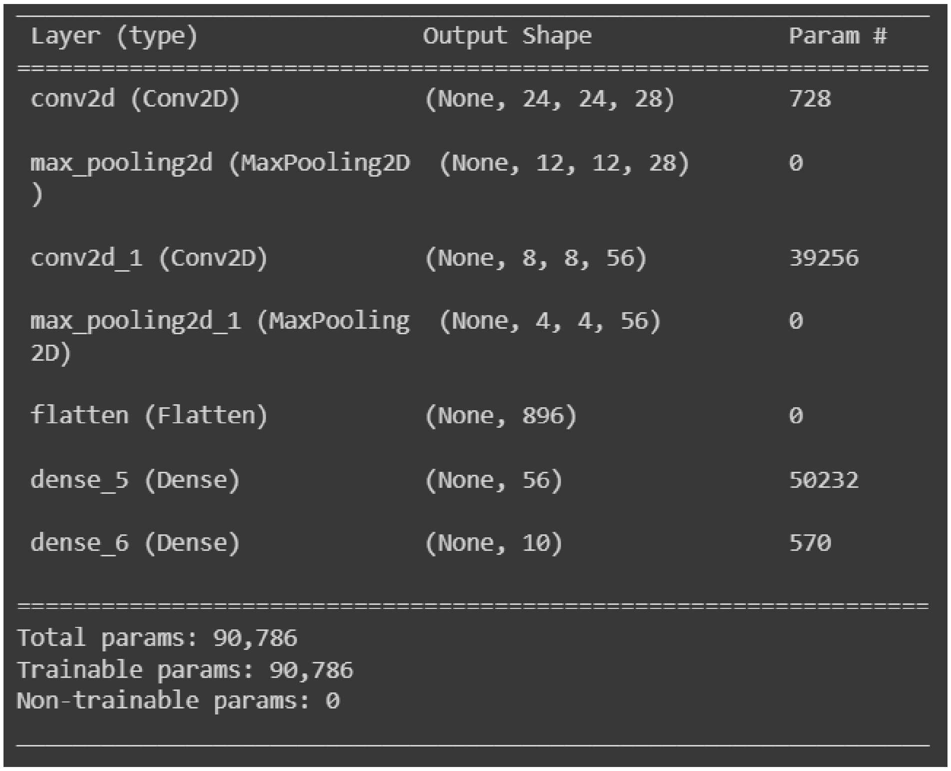


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (32, 104)                 81640

 dropout (Dropout)           (32, 104)                 0

 dense_1 (Dense)             (32, 62)                  6510

 dropout_1 (Dropout)         (32, 62)                  0

 dense_2 (Dense)             (32, 32)                  2016

 dropout_2 (Dropout)         (32, 32)                  0

 dense_3 (Dense)             (32, 16)                  528

 dropout_3 (Dropout)         (32, 16)                  0

 dense_4 (Dense)             (32, 10)                  170

=================================================================
Total params: 90,864
Trainable params: 90,864
Non-trainable params: 0
_________________________________________________________________
```

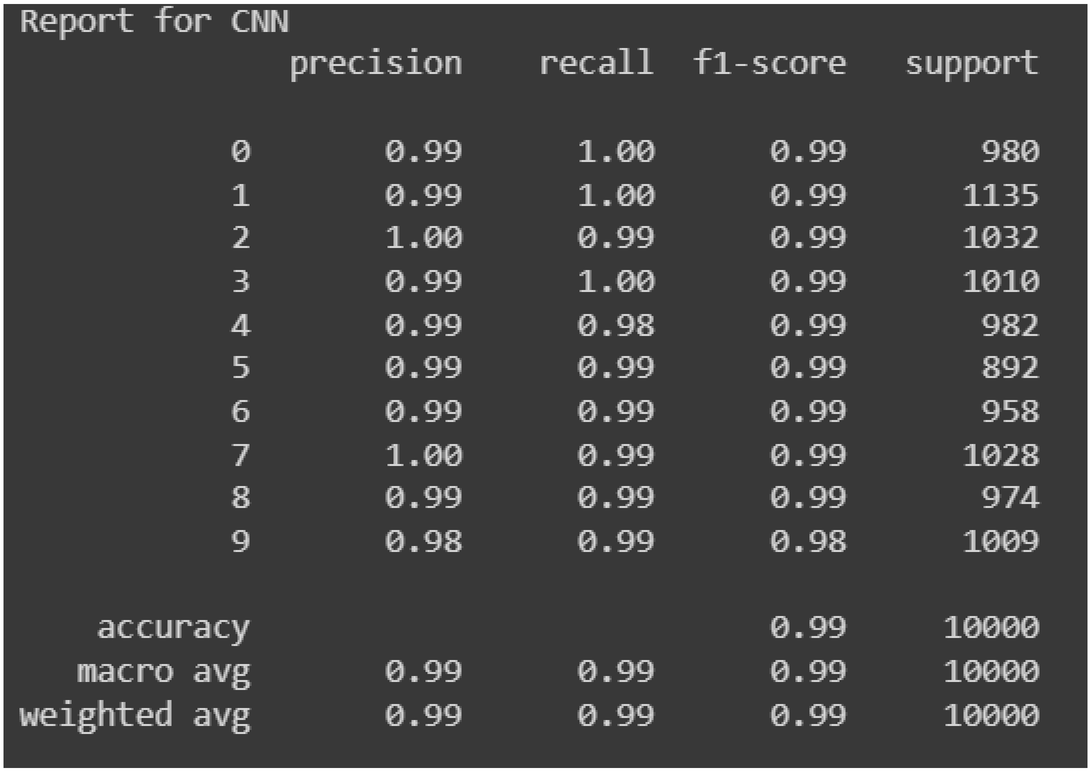


```
Report for MLP
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       980
           1       0.98      0.99      0.99      1135
           2       0.97      0.98      0.98      1032
           3       0.97      0.97      0.97      1010
           4       0.97      0.99      0.98       982
           5       0.95      0.98      0.96       892
           6       0.97      0.97      0.97       958
           7       0.98      0.97      0.98      1028
           8       0.97      0.95      0.96       974
           9       0.98      0.96      0.97      1009

    accuracy                           0.97     10000
   macro avg       0.97      0.97      0.97     10000
weighted avg       0.97      0.97      0.97     10000
```

CNN



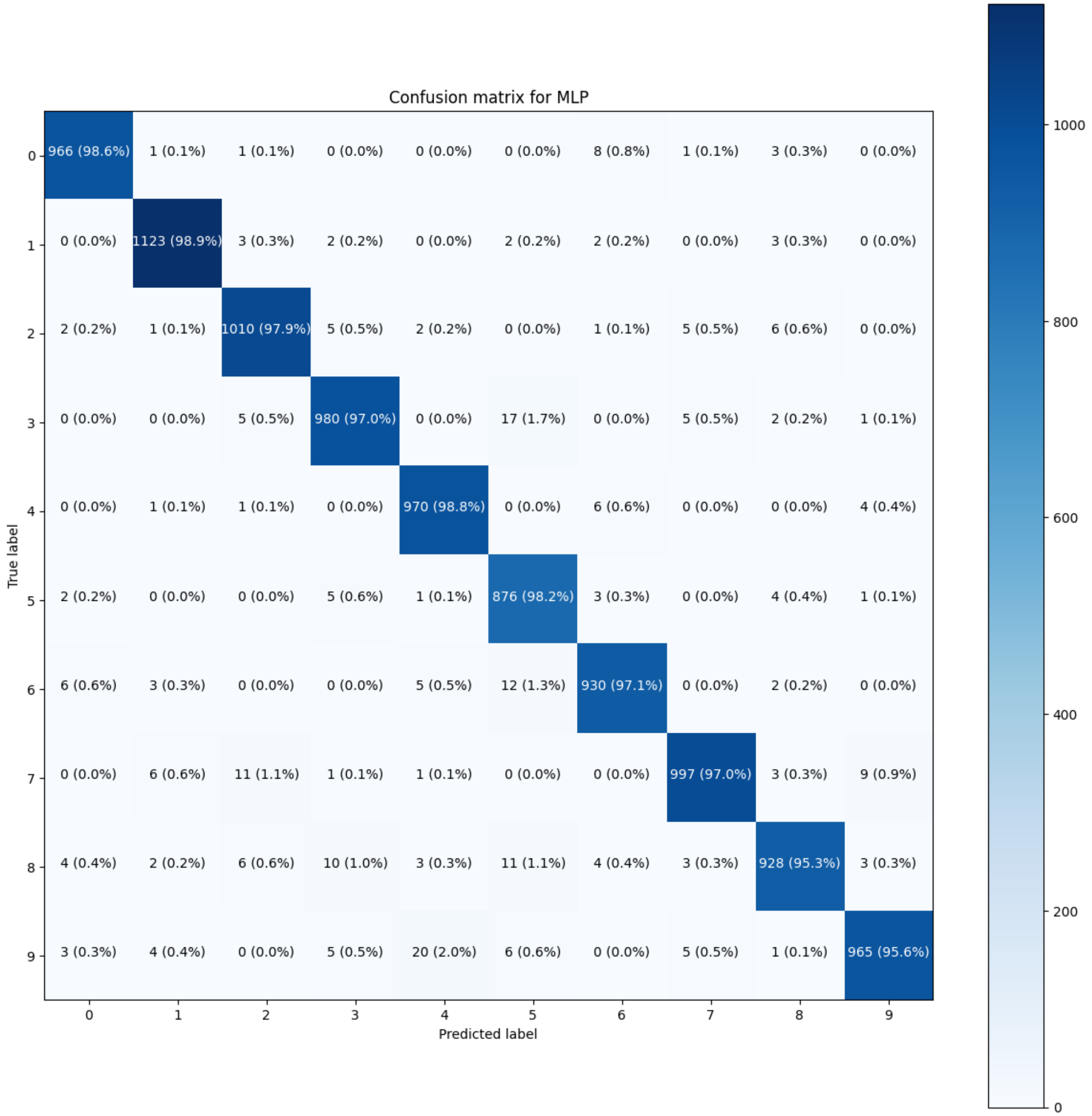
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 24, 24, 28)        728       
                                                                 
 max_pooling2d (MaxPooling2D  (None, 12, 12, 28)       0         
 )                                                               
                                                                 
 conv2d_1 (Conv2D)           (None, 8, 8, 56)          39256     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 4, 4, 56)         0         
 2D)                                                             
                                                                 
 flatten (Flatten)           (None, 896)               0         
                                                                 
 dense_5 (Dense)             (None, 56)                50232     
                                                                 
 dense_6 (Dense)             (None, 10)                570       
                                                                 
=================================================================
Total params: 90,786
Trainable params: 90,786
Non-trainable params: 0
_________________________________________________________________
```



```
Report for CNN
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       980
           1       0.99      1.00      0.99      1135
           2       1.00      0.99      0.99      1032
           3       0.99      1.00      0.99      1010
           4       0.99      0.98      0.99       982
           5       0.99      0.99      0.99       892
           6       0.99      0.99      0.99       958
           7       1.00      0.99      0.99      1028
           8       0.99      0.99      0.99       974
           9       0.98      0.99      0.98      1009

    accuracy                           0.99     10000
   macro avg       0.99      0.99      0.99     10000
weighted avg       0.99      0.99      0.99     10000
```

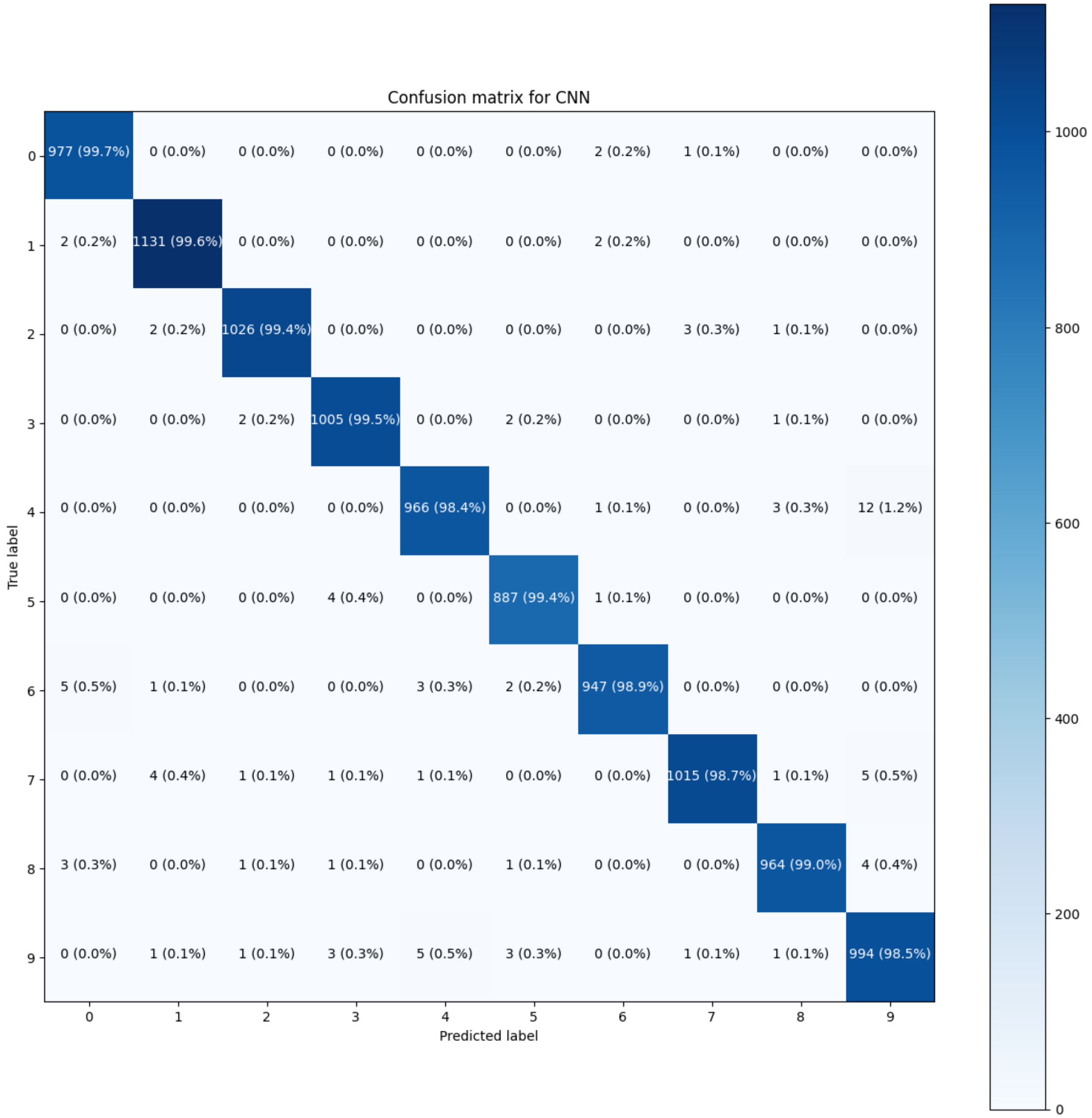
Results:
We can observe that the number of trainable parameters in MLP and CNN is almost equal. In CNN, most of the parameters are contributed from the dense layers.

From the prediction results obtained, we see that the CNN performs slightly better by 0.02 than MLP. We have also calculated various metrics to show prediction results.



Confusion matrix for MLP

	0	1	2	3	4	5	6	7	8	9
0	966 (98.6%)	1 (0.1%)	1 (0.1%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	8 (0.8%)	1 (0.1%)	3 (0.3%)	0 (0.0%)
1	0 (0.0%)	1123 (98.9%)	3 (0.3%)	2 (0.2%)	0 (0.0%)	2 (0.2%)	2 (0.2%)	0 (0.0%)	3 (0.3%)	0 (0.0%)
2	2 (0.2%)	1 (0.1%)	1010 (97.9%)	5 (0.5%)	2 (0.2%)	0 (0.0%)	1 (0.1%)	5 (0.5%)	6 (0.6%)	0 (0.0%)
3	0 (0.0%)	0 (0.0%)	5 (0.5%)	980 (97.0%)	0 (0.0%)	17 (1.7%)	0 (0.0%)	5 (0.5%)	2 (0.2%)	1 (0.1%)
4	0 (0.0%)	1 (0.1%)	1 (0.1%)	0 (0.0%)	970 (98.8%)	0 (0.0%)	6 (0.6%)	0 (0.0%)	0 (0.0%)	4 (0.4%)
5	2 (0.2%)	0 (0.0%)	0 (0.0%)	5 (0.6%)	1 (0.1%)	876 (98.2%)	3 (0.3%)	0 (0.0%)	4 (0.4%)	1 (0.1%)
6	6 (0.6%)	3 (0.3%)	0 (0.0%)	0 (0.0%)	5 (0.5%)	12 (1.3%)	930 (97.1%)	0 (0.0%)	2 (0.2%)	0 (0.0%)
7	0 (0.0%)	6 (0.6%)	11 (1.1%)	1 (0.1%)	1 (0.1%)	0 (0.0%)	0 (0.0%)	997 (97.0%)	3 (0.3%)	9 (0.9%)
8	4 (0.4%)	2 (0.2%)	6 (0.6%)	10 (1.0%)	3 (0.3%)	11 (1.1%)	4 (0.4%)	3 (0.3%)	928 (95.3%)	3 (0.3%)
9	3 (0.3%)	4 (0.4%)	0 (0.0%)	5 (0.5%)	20 (2.0%)	6 (0.6%)	0 (0.0%)	5 (0.5%)	1 (0.1%)	965 (95.6%)



	0	1	2	3	4	5	6	7	8	9
0	977 (99.7%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	2 (0.2%)	1 (0.1%)	0 (0.0%)	0 (0.0%)
1	2 (0.2%)	1131 (99.6%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	2 (0.2%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
2	0 (0.0%)	2 (0.2%)	1026 (99.4%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	3 (0.3%)	1 (0.1%)	0 (0.0%)
3	0 (0.0%)	0 (0.0%)	2 (0.2%)	1005 (99.5%)	0 (0.0%)	2 (0.2%)	0 (0.0%)	0 (0.0%)	1 (0.1%)	0 (0.0%)
4	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	966 (98.4%)	0 (0.0%)	1 (0.1%)	0 (0.0%)	3 (0.3%)	12 (1.2%)
5	0 (0.0%)	0 (0.0%)	0 (0.0%)	4 (0.4%)	0 (0.0%)	887 (99.4%)	1 (0.1%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
6	5 (0.5%)	1 (0.1%)	0 (0.0%)	0 (0.0%)	3 (0.3%)	2 (0.2%)	947 (98.9%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
7	0 (0.0%)	4 (0.4%)	1 (0.1%)	1 (0.1%)	1 (0.1%)	0 (0.0%)	0 (0.0%)	1015 (98.7%)	1 (0.1%)	5 (0.5%)
8	3 (0.3%)	0 (0.0%)	1 (0.1%)	1 (0.1%)	0 (0.0%)	1 (0.1%)	0 (0.0%)	0 (0.0%)	964 (99.0%)	4 (0.4%)
9	0 (0.0%)	1 (0.1%)	1 (0.1%)	3 (0.3%)	5 (0.5%)	3 (0.3%)	0 (0.0%)	1 (0.1%)	1 (0.1%)	994 (98.5%)

We can clearly see the false and true predictions from the above confusion matrix. i.e. the confusion of the model to predict the ground truth.

The MLP model is a fully connected neural network, which means that all input features are connected to every neuron in the hidden layers. It works well for problems where the input features are independent of each other, but it can struggle with problems where the input features have spatial dependencies or correlations.

On the other hand, the CNN model is specifically designed for image processing tasks, where the input features are arranged in a grid-like structure. It applies convolutional and pooling layers to extract meaningful features from the input images, which can be used to classify the images into different categories.

Compared to the MLP model, the CNN model generally performs better on image classification tasks, because it is able to capture the spatial dependencies and correlations in the input images. The CNN model also has fewer parameters than the MLP model, which makes it easier to train and less prone to overfitting.