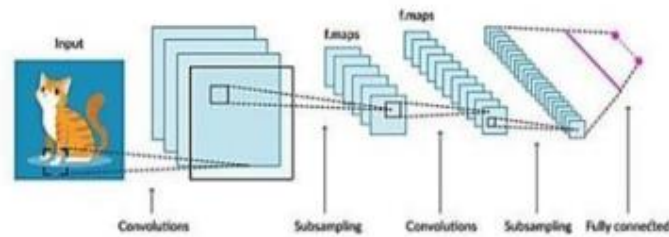# 1-Architecture used in the Paper:

4.1: convolutional neural networks model

In the paper, the algorithm used for the image multiclassification consisted of several layers. Firstly, the input layer to allow the input of the images. The size of the images inputted was 256x256 pixels. This results in the input layer consisting of 65,536 (total number of pixels in the image). The inputted data is then fed into the CNN layer. Since this is a multiclassification algorithm, The activation function used was SoftMax while the loss function was categorical crossentropy. This layer consists of multiple nodes that allows for feature extraction to allow the image to be processed. The output of The CNN layer is then passed to the ANN layer which will classify the data accordingly and classify the images respectively.

2-Dataset details:

Dataset name: MNIST digits classification dataset

Dataset source: https://keras.io/api/datasets/mnist/

Number of Classes: 10

Dataset labels: (0-1-2-3-4-5-6-7-8-9)

Dataset description: The MNIST dataset has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

3-Implementation details:

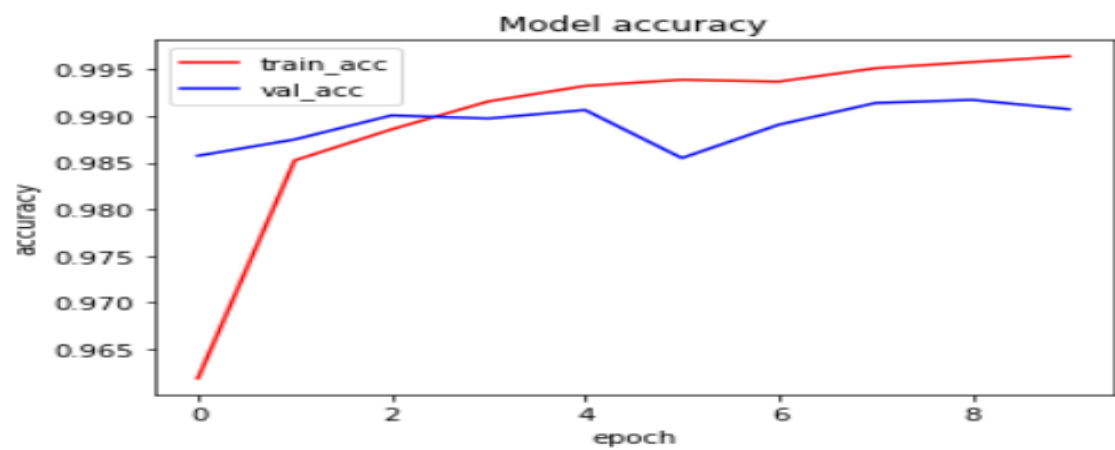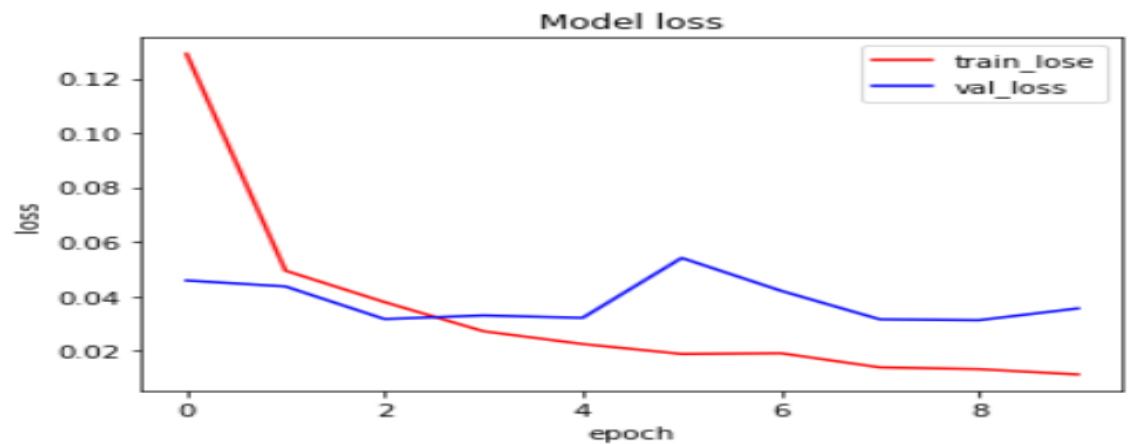Training set ratio: 68% (48000)

Validation set ratio: 18% (12000)

Testing set ratio: 14% (10000)

we use 2 convolution layers and 2 dense layers

4-Results and visualizations

❌ 313/313 - 2s - loss: 0.0305 - accuracy: 0.9913 - 2s/epoch - 7ms/step

Test accuracy: 0.9912999868392944

Model loss

Model accuracy


```
[[ 975    1    0    0    0    1    0    0    3    0]
 [   2 1122    2    2    1    2    2    0    2    0]
 [   0    0 1030    0    0    0    0    1    1    0]
 [   0    0    3 1001    0    2    0    1    3    0]
 [   0    0    0    0  979    0    2    0    0    1]
 [   1    0    0    4    0  884    3    0    0    0]
 [   1    1    0    0    2    1  951    0    2    0]
 [   0    2    9    0    1    0    0 1013    1    2]
 [   0    0    2    0    0    0    0    0  972    0]
 [   1    0    0    1   12    3    0    3    3  986]]
0.9913
```