Helwan University

كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

Faculty of Computers and Artificial Intelligence

Computer Science Department

2022/2023

Selected Topics in Computer Science 2

Project Documentation

Team No. 19

ID	Name	Grade
201900636	محمد اشرف محمد عبد الغني	
201900617	محمد احمد ابراهيم سليمان	
201900660	محمد خالد بن الوليد	
201900661	محمد خالد حسين	
201900632	محمد اشرف ابو سريع	
201900582	كريم كامل احمد	
201900627	محمد احمد محمد كامل محمد	
201900588	كمال عبدالحليم محمد كمال الجنزوري	

Delivered to:

Dr. Wessam El-Behaidy

Eng. Salma Doma

Project Documentation

In this documentation we will briefly discuss the main concept that was introduced in our selected paper and then we will choose another dataset and we will apply the same algorithm that was applied in the selected Paper.

The selected paper

Name: Image Classification using Convolutional Neural Networks.

Authors:

1-Muthukrishnan Ramprasath (Sr. Assistant professor, Department of Computer Science & Engineering, Madanapalle Institute of Technology & Science, Andhra Pradesh, INDIA)

2-Shanmugasundaram Hariharan (Professor, Department of Computer Science and Engineering Saveetha Engineering College, Chennai, India)

3-M. Vijay Anand (Professor, Department of Computer Science and Engineering, Saveetha Engineering College, Chennai, India)

Publisher: International Journal of Pure and Applied Mathematics.

Publication Date: 2018

The Used dataset in Paper: The MNIST dataset of handwritten digits.

Image Dimension: 28*28

No. of classes: 10 classes

Introduction:-

Image classification is the process of a computer analyzing an image and determining which ‘class’ it belongs to. (Or a chance that the image belongs to a ‘class.’). A class is essentially a label, for instance, ‘car’, ‘animal’, ‘building’, and so on. Therefore, if you input an image of a sheep. Image classification is the process of the computer analyzing the image and telling you it’s a sheep. (Or the probability that it’s a sheep.)

Image classification is a very easy task for humans, but for computers, it’s a bit of a hard challenge. Experts used raw pixel data for early image classification. Computers would then breakdown images into individual pixels. The problem is that two pictures of the same object can look very different. For instance, they could have different backgrounds, angles, poses, etc. This made it quite the challenge for computers to correctly ‘see’ and categorize images.

You can do image classification in two ways: Supervised Classification, and Unsupervised Classification. In this paper, the author discusses the use of the Unsupervised learning algorithm in the underwater fish recognition framework for classifying images.

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep learning is the key technology behind self-driving cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost. It is the key to voice control in consumer devices like phones, tablets, TVs, and hands-free speakers.

Deep learning is getting a lot of attention lately and for a good reason. It’s achieving results that were not possible before. Deep learning allows computers to perform classification tasks directly from images, text, or sound.

Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. You can train deep learning models using a large set of labeled data and neural network architectures that contain many layers.

Convolutional neural networks (CNNs) are a specialized type of artificial neural networks. CNNs use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. CNNs were specifically designed to process pixel data and are used in image recognition and processing.

How do the authors apply CNN to their dataset?

The system uses gray-scale images as input images having 28x28 sizes. For the first layer in CCN, they applied 32 filters on input images, each image size is 3x3 producing 32 feature maps of size 26x26. The second layer is applying 64 filters, each of size 3x3 producing 64 feature maps of size 24x24. Max pooling layer acts as the third layer, which is used to down-sample the Input Images to 12x12 by using a sub-sampling window of size 2x2. Layer 4 is a fully connected layer having 128 neurons and uses the SoftMax activation function for the classification of images and producing the output image.

The layers of a Convolutional Neural Network have neurons arranged in three dimensions: width, height, and depth. The word depth refers to the third dimension of an activation volume, not to the depth of a full Neural Network - which can refer to the total number of layers in a network -.

CNN Algorithm: -

1. Batch size = 128, no of classes 10, number of epochs = 5

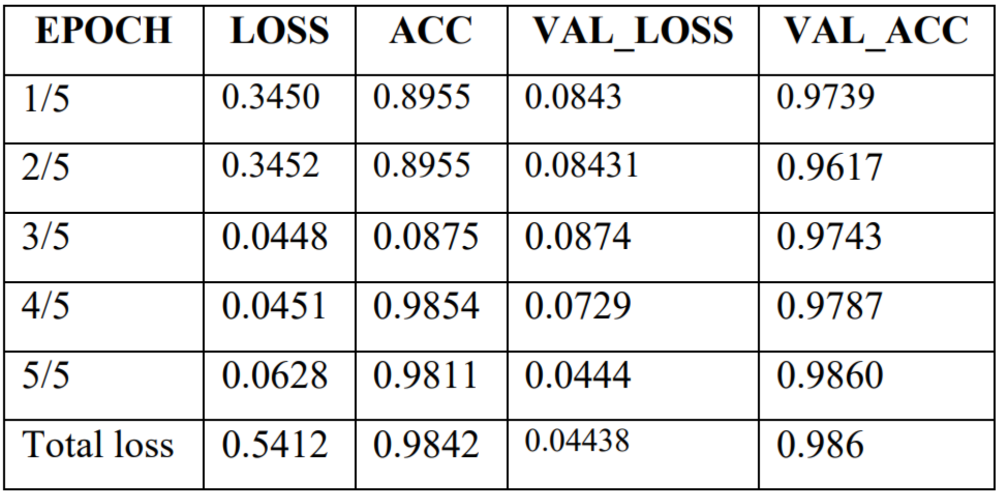
2. Dimension of input image 28 ×28

3. Loading the input images from MNIST data set.

4. Variable exploration: X=test data set (10000,28,28,1), Train data set (60000,28,28,1)

5. Creating and compiling the models.

6. Training the network.



EPOCH	LOSS	ACC	VAL_LOSS	VAL_ACC
1/5	0.3450	0.8955	0.0843	0.9739
2/5	0.3452	0.8955	0.08431	0.9617
3/5	0.0448	0.0875	0.0874	0.9743
4/5	0.0451	0.9854	0.0729	0.9787
5/5	0.0628	0.9811	0.0444	0.9860
Total loss	0.5412	0.9842	0.04438	0.986

Table 1. The loss and accuracy of all epochs

Project Details

Data set :

1. description : The Arabic Handwritten Digits Databases

2. Link : https://datacenter.aucegypt.edu/shazeem/

ADBase & MADBase

1. Layers:

   1 hidden layer Conv2D 32 Unit with (3*3) Filter and relu Activition function

   1 hidden layer Conv2D 64 Unit with (3*3) Filter and relu Activition function

   1 hidden layer MaxPooling2D Pool size is (2*2)

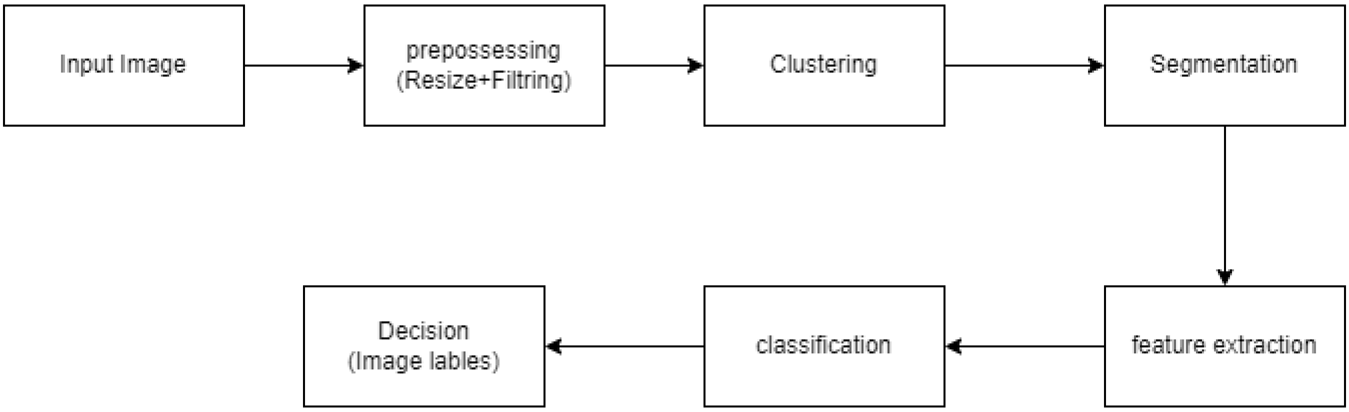
1 hidden layer Flatten

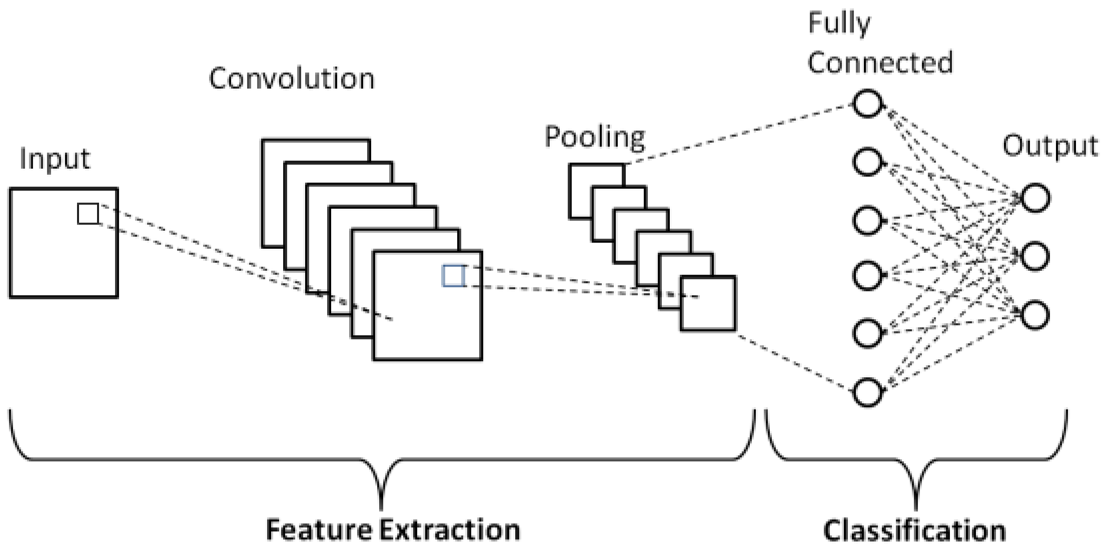
1 hidden Dense layer with 128 units and relu function

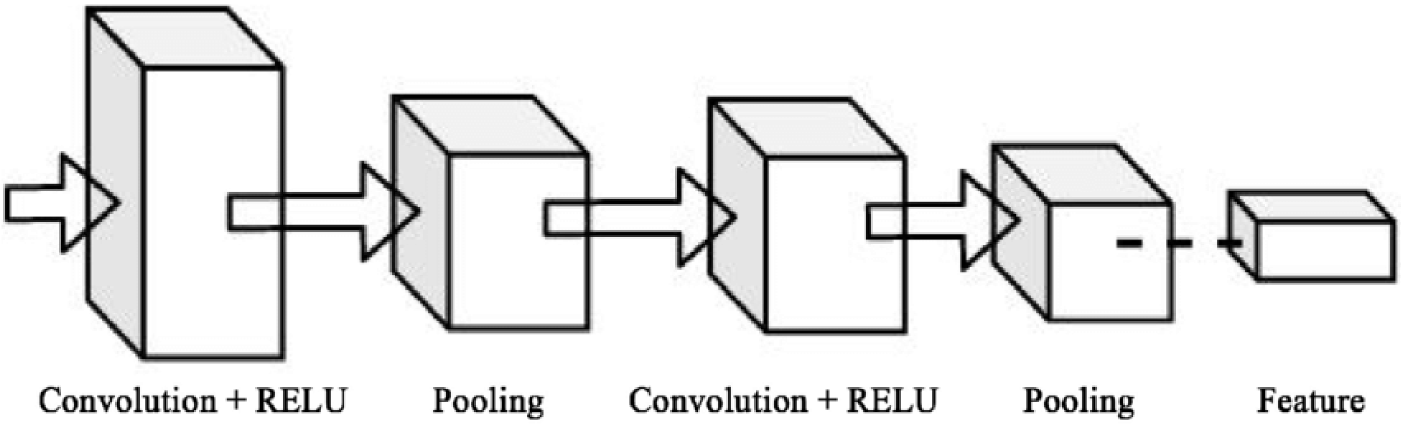
Output layer with 10 units and Softmax function

2. Loss Function: categorical_crossentropy
3. Optimizer: Adam
4. Learning Rate: 0.001
5. Metrics: Accuracy
6. Epochs: 20
7. Batch Size: 64
8. Total params: 1,199,882
9. Trainable params: 1,199,882
10. Non-trainable params: 0

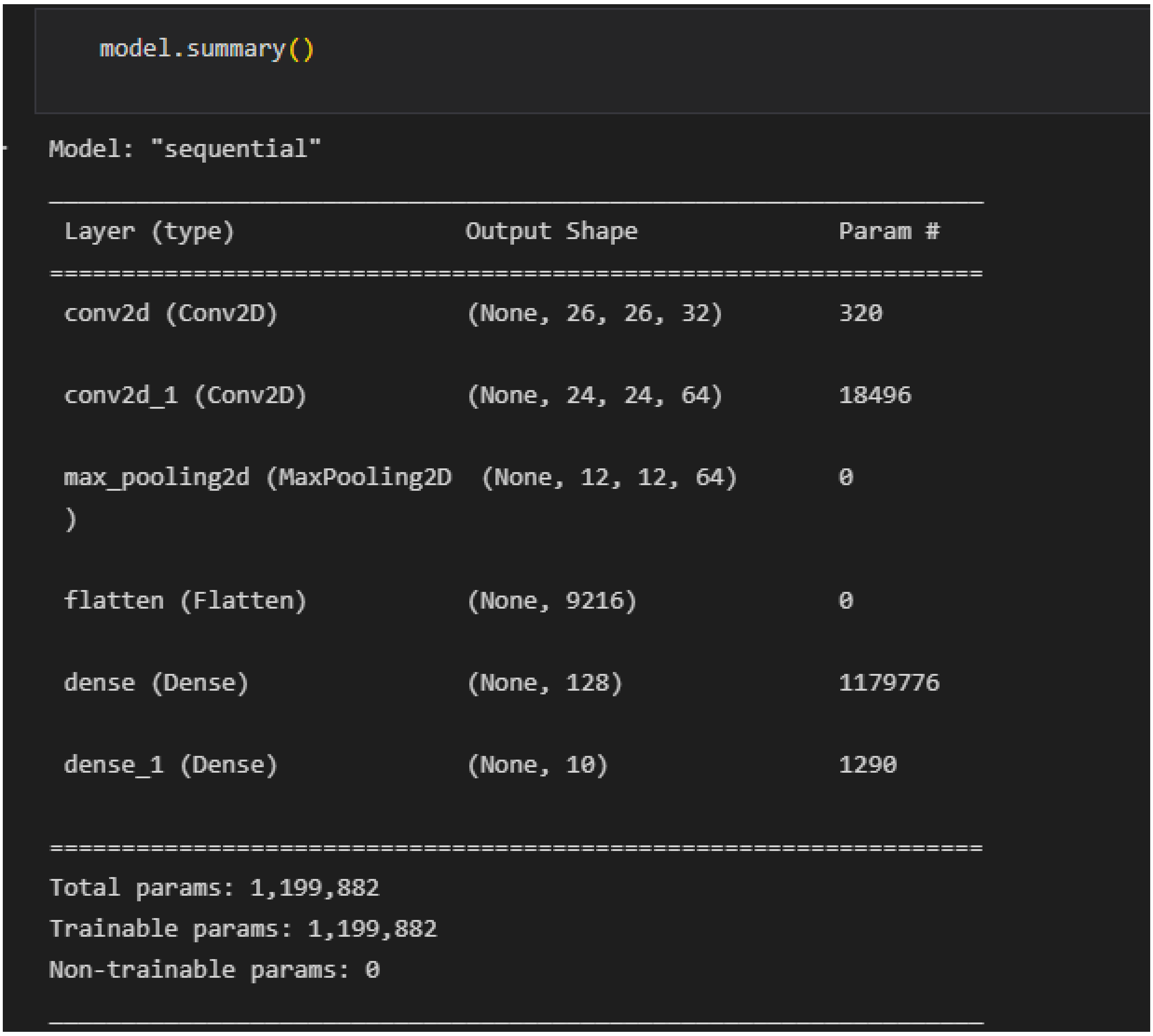
<u>Diagrams of our model:</u>







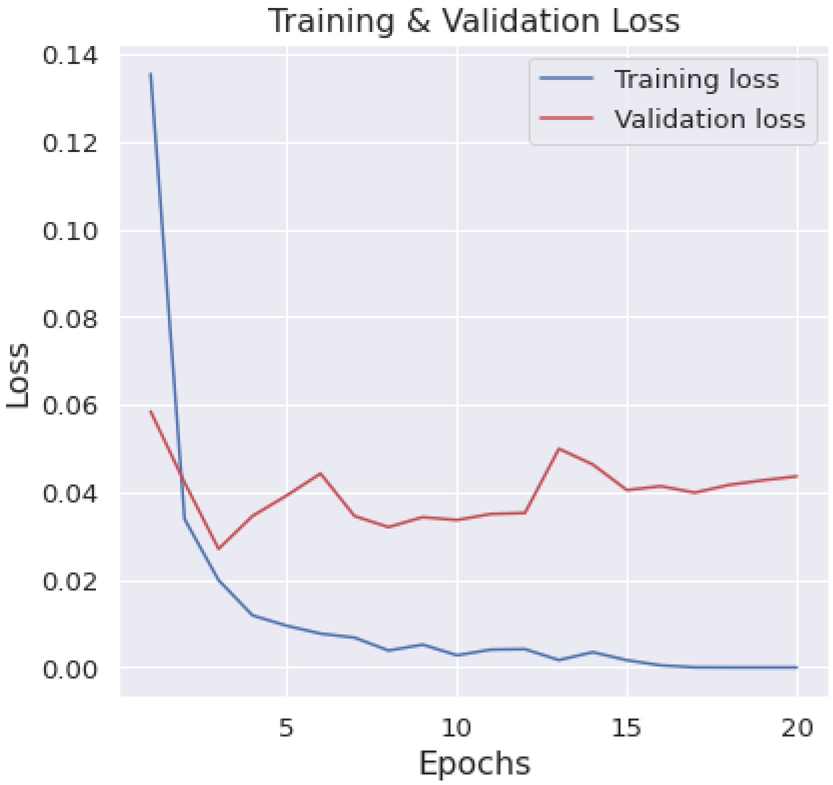
Model Summary :

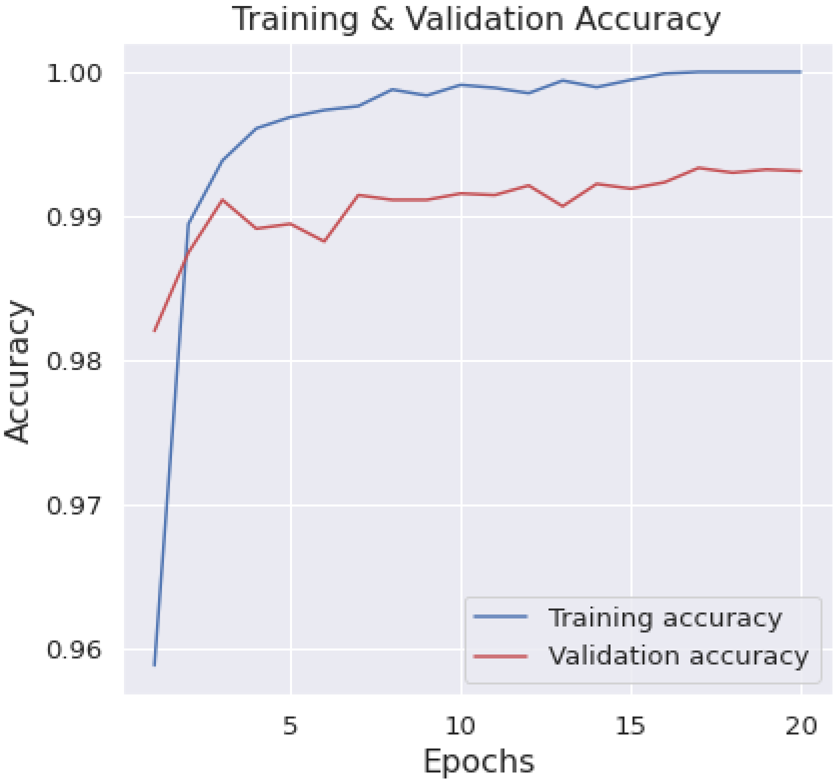


```
model.summary()
```

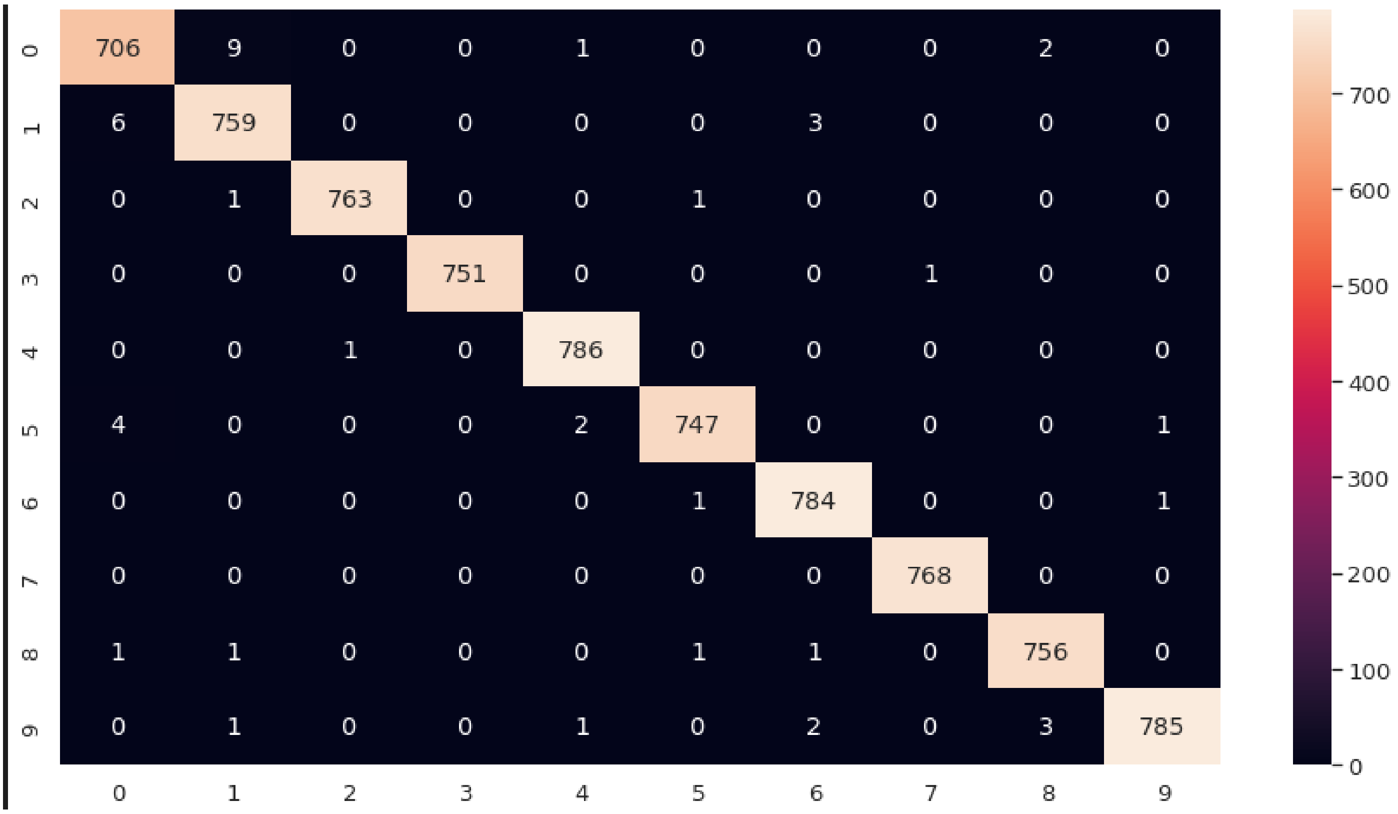
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        320

 conv2d_1 (Conv2D)           (None, 24, 24, 64)        18496

 max_pooling2d (MaxPooling2D  (None, 12, 12, 64)       0
 )

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 128)               1179776

 dense_1 (Dense)             (None, 10)                1290

=================================================================
Total params: 1,199,882
Trainable params: 1,199,882
Non-trainable params: 0
_________________________________________________________________
```

Loss Curve :


Training & Validation Loss
Training loss
Validation loss
Loss
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
5
10
15
20
Epochs


Training & Validation Accuracy
Training accuracy
Validation accuracy
Accuracy
1.00
0.99
0.98
0.97
0.96
5
10
15
20
Epochs

Confusion Matrix :



	0	1	2	3	4	5	6	7	8	9
0	706	9	0	0	1	0	0	0	2	0
1	6	759	0	0	0	0	3	0	0	0
2	0	1	763	0	0	1	0	0	0	0
3	0	0	0	751	0	0	0	1	0	0
4	0	0	1	0	786	0	0	0	0	0
5	4	0	0	0	2	747	0	0	0	1
6	0	0	0	0	0	1	784	0	0	1
7	0	0	0	0	0	0	0	768	0	0
8	1	1	0	0	0	1	1	0	756	0
9	0	1	0	0	1	0	2	0	3	785