COMP8720 First Project Report

Navjot Kaur (110124316)

Jaspreet Kaur (110122756)

Niharika Khurrana (110124290)

17 October 2023

**Abstract**

In the realm of machine learning, Convolutional Neural Networks (CNNs) play a pivotal role in image recognition. This report describes the deployment and testing of a Convolutional Neural Network (CNN) that uses the Keras framework to categorise handwritten digits (0–9) from the MNIST dataset. The approach is described in depth in the paper, beginning with the import and visualisation of data and moving on to the data preparation, CNN architecture development, model fitting, and assessment steps. In order to provide light on how the network operates, the research also contains visualisations of random records and intermediary layer outputs.

**1.Introduction**1.1 Background

Handwritten digit recognition, pivotal in computer vision, uses the MNIST dataset with 60,000 training and 10,000 test images. Despite its simplicity, the dataset's complex patterns challenge algorithms. Convolutional Neural Networks (CNNs) have revolutionized image processing due to their automatic feature learning, especially for tasks like handwritten digit recognition.

1.2 Objective

The primary objective of this study is to develop a robust CNN model also compare it with its competitor models to know its capability of accurately classifying handwritten digits within the MNIST dataset. The report provides a detailed account of the implementation steps and explores the experimental results.

**2. Data Exploration and Visualization**

2.1 Dataset Overview

The MNIST dataset serves as the foundation of our study, comprising a total of 70,000 images. Among these, 60,000 images are utilized for training the model, while 10,000 images are reserved for rigorous testing, ensuring the model's robustness. These images consist of handwritten digits, forming the basis of our analysis.

2.2 Data Visualization

In this section, random samples from the dataset are meticulously examined and visualized. By doing so, we aim to gain insights into the diverse nature of the handwritten digits. Visualizing these samples provides a tangible understanding of the dataset's intricacies, laying the groundwork for our subsequent modeling and analysis efforts.

A yellow and blue label

Description automatically generated with medium confidence  
 Fig no. 1: Randomly selected training images with labels

A graph of a number of red and pink bars

Description automatically generated with medium confidence  
 Fig no. 2 : Digit Count in Mnist Dataset

A blue and green binary code

Description automatically generated  
 Fig no 3: Random record with the pixel value

**3. Data Preparation**

3.1 Reshape Data:

The dataset's 28x28 pixel images are reshaped into a 4D tensor (number of samples, height, width, channels) to align with the CNN model's requirements. Each image becomes a 28x28x1 array, denoting grayscale.

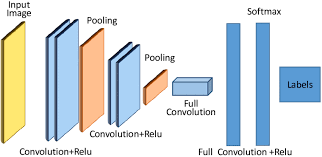
3.2 Rescaling (Normalization):

Pixel values, initially ranging from 0 to 255, are normalized to a 0-1 range. This normalization simplifies neural network training, aiding convergence by ensuring consistent input scales.

3.3 Convert Input Data Type: Int to Float:

Pixel values, initially integers, are converted to floating-point numbers. This precision ensures accurate calculations for subsequent neural network operations.

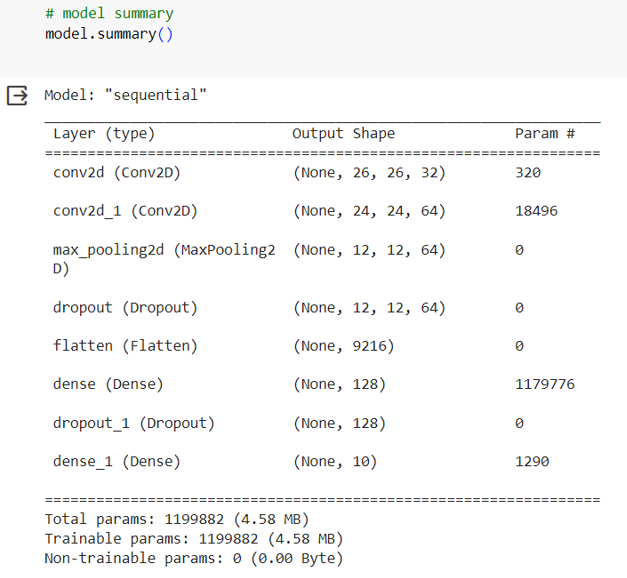
3.4 Sample Selection for Training:

A random subset of 20,000 images is chosen from the training dataset. This streamlined selection facilitates efficient experimentation, reducing computational load while preserving dataset diversity for robust model training.  
  
**4. Experimentation and Results**4.1 Model Overview   
The CNN architecture comprises multiple convolutional and pooling layers, followed by fully connected layers and a softmax output layer.  
  
   
 Fig no. 4 – CNN architecture  
  
4.2 Model Visualization  
The model is visualized and the output will provide a summary of the convolutional neural network (CNN) architecture that has been constructed.

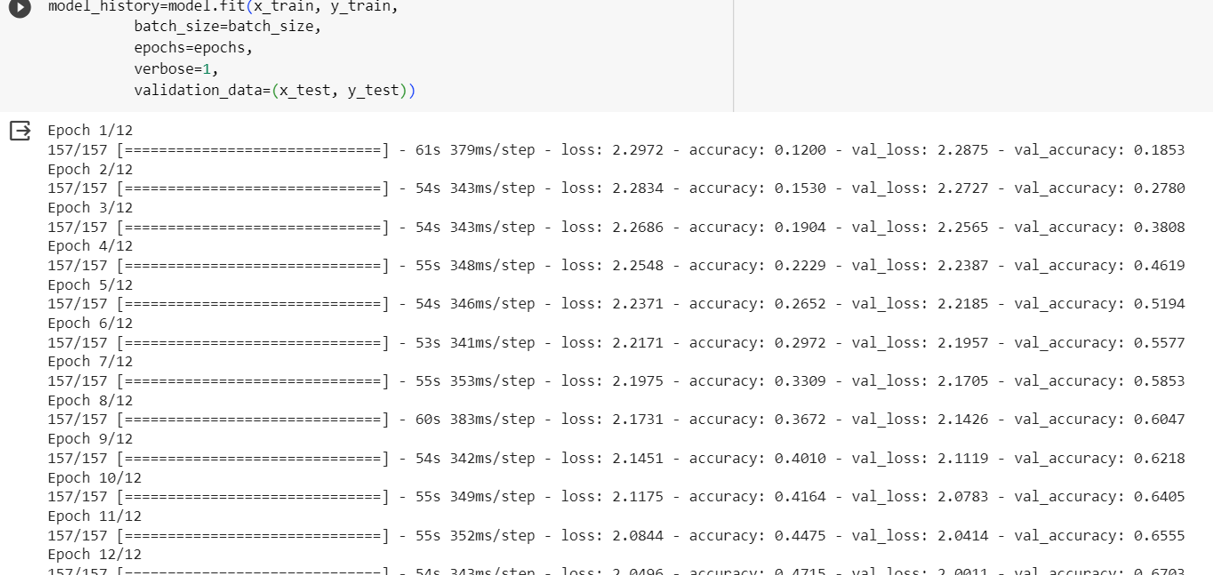
Here's what each part of the output represents:

Layer Type: Indicates the type of layer in the neural network (e.g., Conv2D, MaxPooling2D, Dense, Dropout).

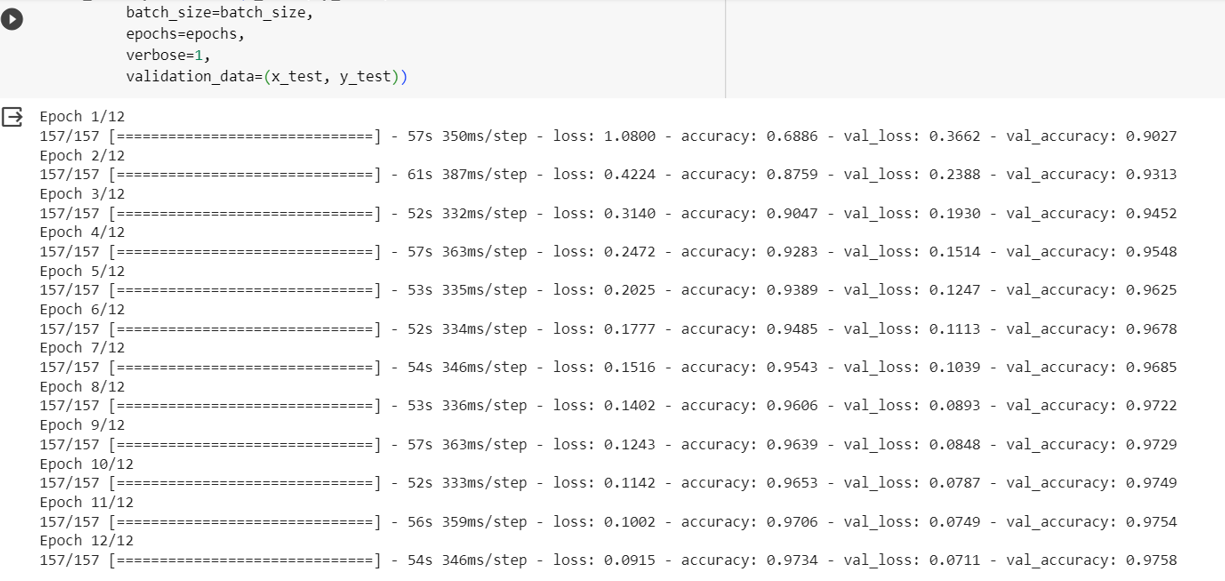
Output Shape: Describes the shape of the output tensor after the layer has been applied. It shows the dimensions of the data at that specific layer.

Param # (Trainable Parameters): Represents the number of trainable parameters in the layer. These parameters are the weights that the neural network learns during training to make predictions.  
    
 Fig no. 5 – CNN architecture parameter

**5. Model Fitting and Evaluation**



Model1 training 12 epocs

   
Model 2 training 12 epocs

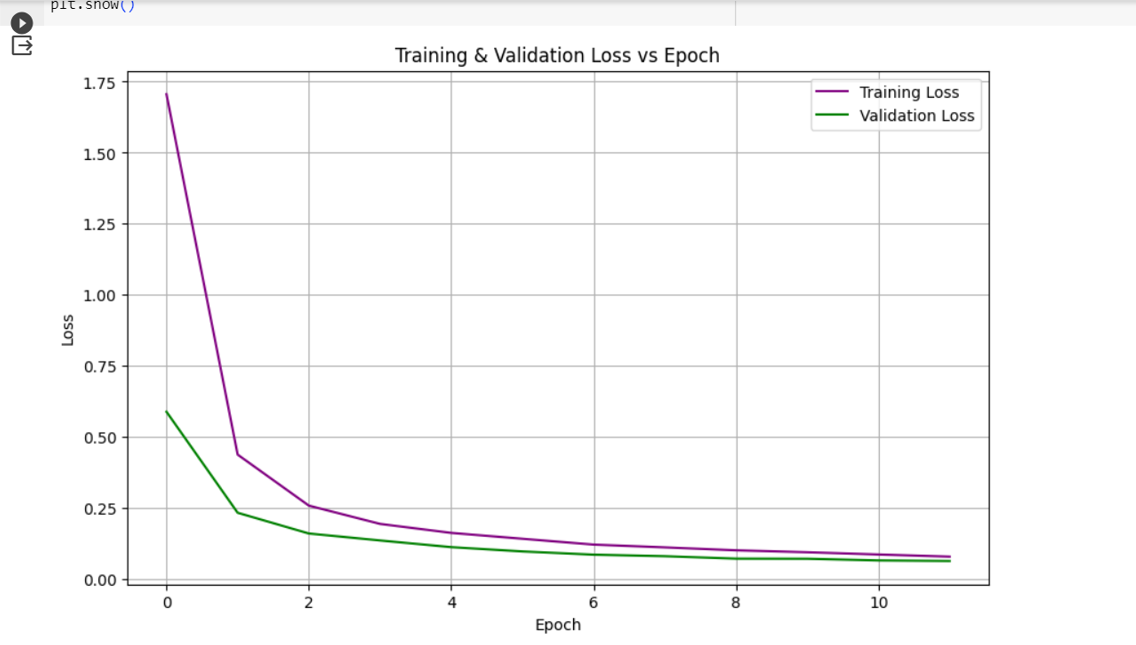
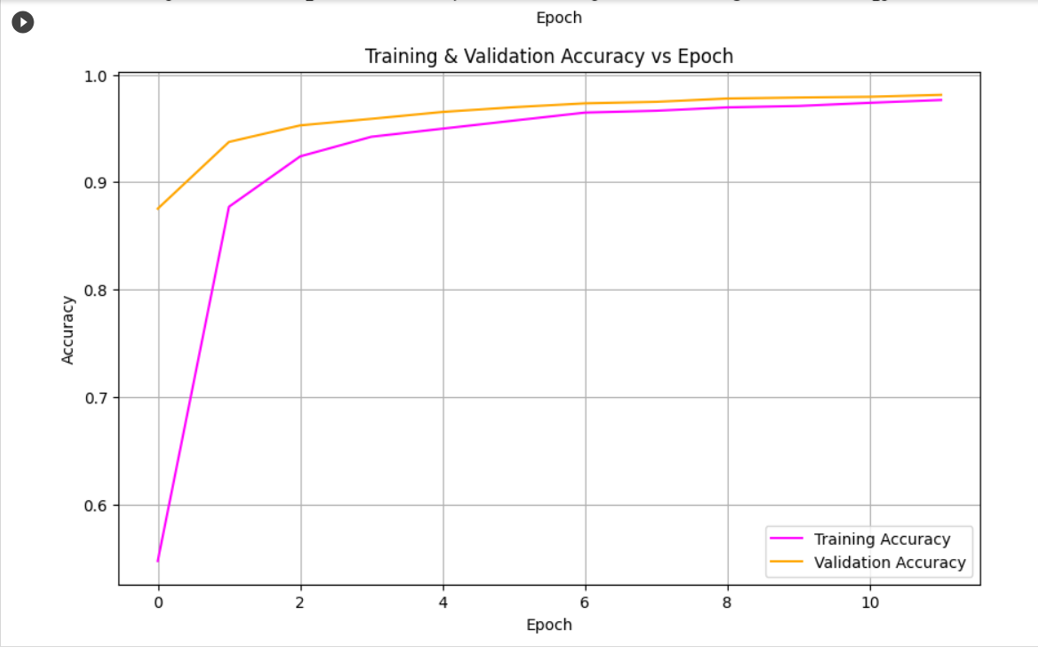
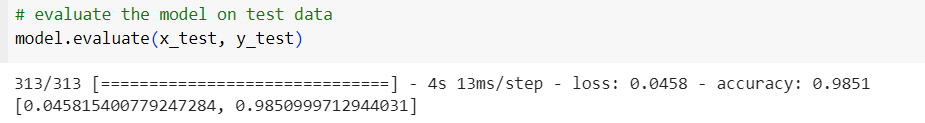
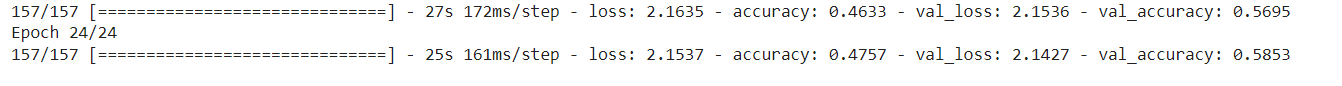
**Evaluation**  
  


Fig no. 6-Training and Validation Metrics: Loss

  
Fig no. 7-Training and Validation Metrics: accuracy

**6. Results:**  
Fig no. 8- result from notebook for 24 epocs on Model 2

  
 Fig no. 9- result from notebook for 24 epocs on Model 1

**7. Experiments with training:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Training model sequential** | **Accuracy** | **Layers** | **Epochs** | **Optimizer** | **Learning rate** |
| Model 1 | 67 | 2 | 12 | Adadelta | No Lr |
| Model 2 | 98 | 3 | 12 | Adam | 1e-4 |
| Model 1 | 58 | 2 | 24 | Adadelta | No Lr |
| Model 2 | 99 | 3 | 24 | Adam | 1e-4 |

We trained the model based on different parameters and evaluated the score on train and test and found the above result. We can further train and test the model on other numerous parameters and increase the efficiency of the classifier.

**8.Compairing our results with the existing models**

There are many Deep learning and Machine learning models developed by various researchers for classification. Some of them are discussed below in the image. Source of this literature review is a publication found on research gate with the title *Study\_and\_Develop\_a\_Convolutional\_Neural\_Network\_for\_MNIST\_Handwritten\_Digit\_Classification* by Brijesh Kumar



**9. Participants and their Responsibilities:**

|  |  |
| --- | --- |
| **Participant** | **Responsibility** |
| Jaspreet Kaur | Visualization/Report: Responsible for creating visual representations of the data |
| Navjot Kaur | Data Processing/Report: Handles cleaning and structuring the dataset for analysis. |
| Niharika Khurrana | Model Training/Report : Builds and fine-tunes machine learning models for the project |

**10. References**   
1. <https://www.researchgate.net/figure/A-generic-CNN-Architecture_fig1_344294512>   
2. <https://medium.com/analytics-vidhya/a-brief-study-of-convolutional-neural-network-cnn-using-mnist-digit-recognizer-e054cf8863bf>