**MNIST HANDWRITTEN DIGIT CLASSIFICATION**

**Internship Project at Neurons Solution**

**Course: Artificial Intelligence (Robotics)**

**Group#: 1**

**Group Members:**

**Sana Akbar**

**Shehla Ibrar**

**Ubaida Waheed**

**Muneeba Mehmood**

Table of Contents

[Introduction: 2](#_Toc113590347)

[MNIST Handwritten Digit Classification: 2](#_Toc113590348)

[Algorithm: 2](#_Toc113590349)

[Dry run of Algorithm: 2](#_Toc113590350)

[Tools and Technologies: 2](#_Toc113590351)

[Libraries: 3](#_Toc113590352)

[Tools: 3](#_Toc113590353)

[Language: 3](#_Toc113590354)

[Frontend: 3](#_Toc113590355)

[Working: 3](#_Toc113590356)

[Output Screenshot: 6](#_Toc113590357)

[Conclusion: 6](#_Toc113590358)

# Introduction:

Humans can visually sense the world around them by using their eyes and brains. Computer visions work on enabling computers to see and process images in the same way that human vision does. There are many algorithms that are developed in the field of computer vision to recognize images. Artificial Neural Networks are interconnected systems composed of many simple processing elements, called as neurons, operating in parallel whose function is controlled by the network structure, connection strengths and the processing performed at computing elements or nodes. The goal of this work is to create a model to identify and determine the handwritten digits from its data sets with better accuracy and aim to complete this by using the concepts of Convolutional Neural Network and MNIST dataset. The scope of the project is to improvise the accuracy in recognizing and make image processing techniques to robots or any automated machine. Utilizing these deep learning techniques, a high amount of accuracy can be obtained in recognizing handwritten digits. This technique can be used to make the model test with the robotics or in creating artificial brain.

# MNIST Handwritten Digit Classification:

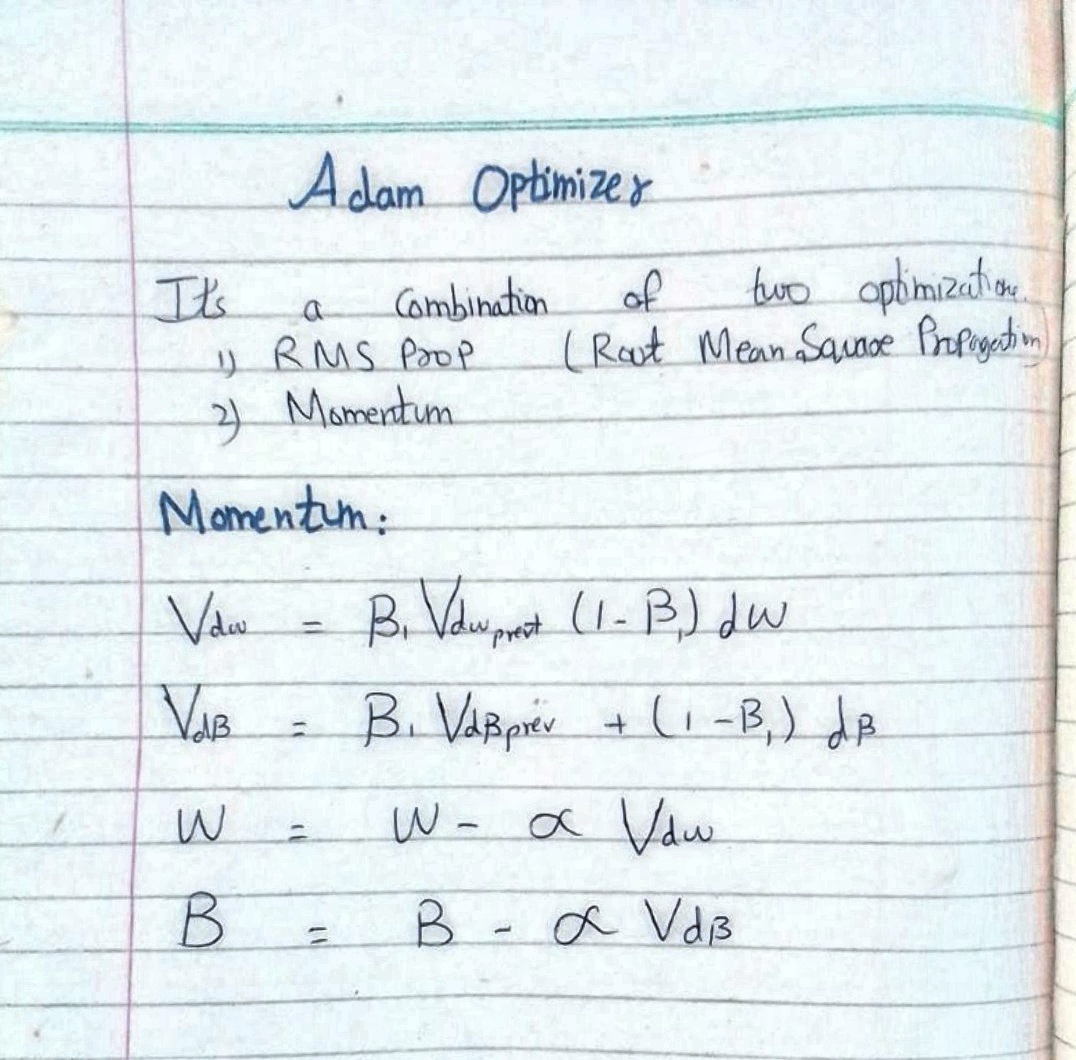
MNIST is a built-in dataset provided by Keras., a library of deep learning. The acronym stands for **“Modified National Institute of Standards and Technology**”. Training of the network is done by this dataset. MNIST dataset has a training set of 60,000 examples and a test set of 10,000 examples. All the grayscale images of handwritten single digits between 0 and 9 in the dataset are of 28 ×28 pixels.

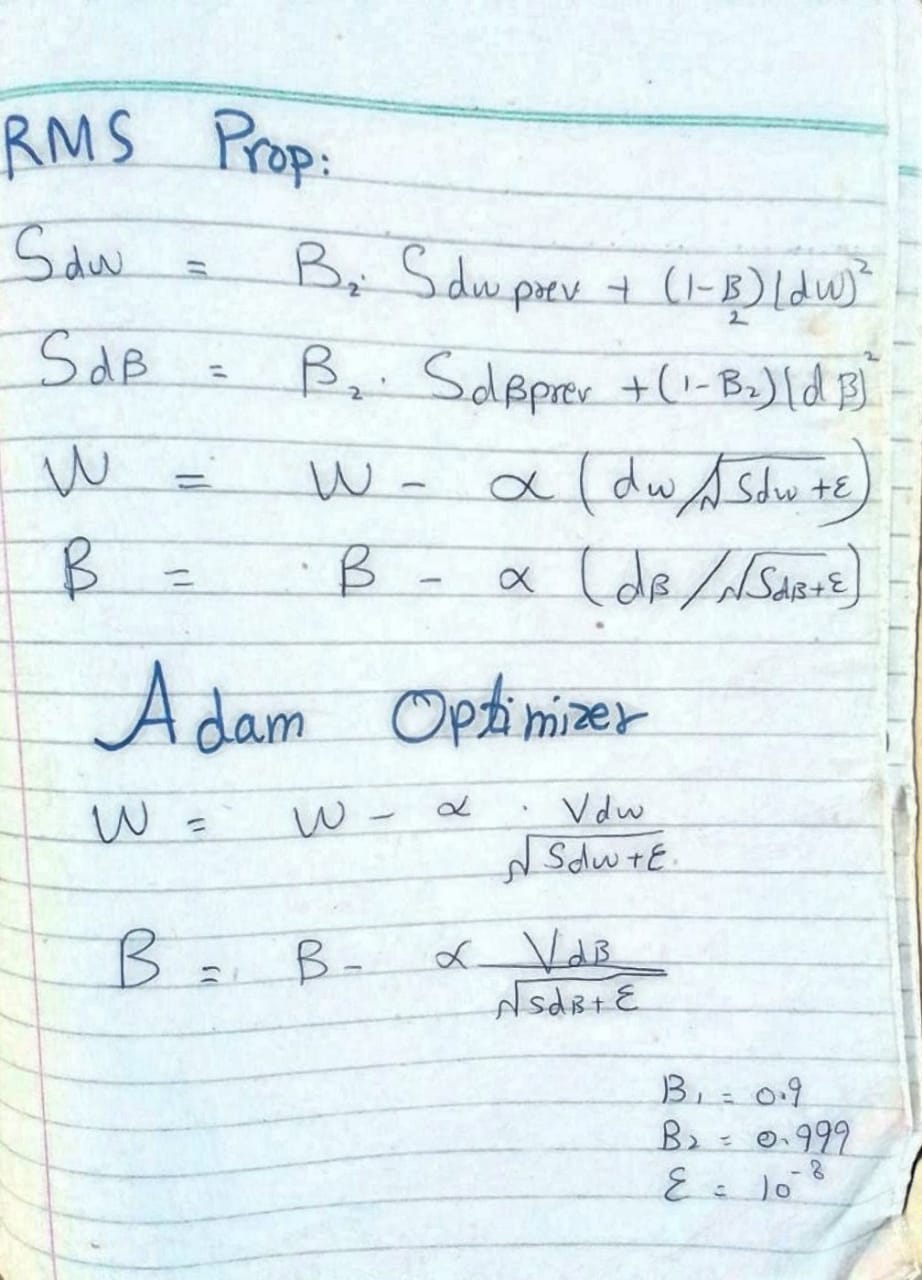
# Algorithm:

**Adam Optimizer** is the combination of two algorithms, RMSprop (Root Mean Square Propagation) and Momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of Momentum by using moving average of the gradient. We used Adam optimizer algorithm because of its following features.

* Straightforward to implement.
* Computationally efficient.
* Little memory requirements.
* Faster computation time.
* Hyper-parameters have intuitive interpretation and typically require little tuning.

# Dry run of Algorithm:





In Momentum, the weight updation is given by this equation where we use this Vdw and in RMS prop , the wight updation is given by Sdw. When take dw square , we represent it as Sdw and when we don’t take square , we represent it with vdw.

For Adam optimizer, we combine both momentum as well as the RMS prop into one single equation so the weight updation in adam will be given by combining both Vdw of the momentum and Sdw from the RMS prop into one and it will be given by Vdw divided by Sdw plus epsilon and similarly for B , it will be given by VdB divided by SdB plus epsilon .

# Tools and Technologies:

We have used the following tools and technologies:

## Libraries:

TensorFlow, NumPy, Keras.

## Tools:

Anaconda, PyCharm, Jupiter Notebook, Visual Studio Code, Google Browsers.

## Language:

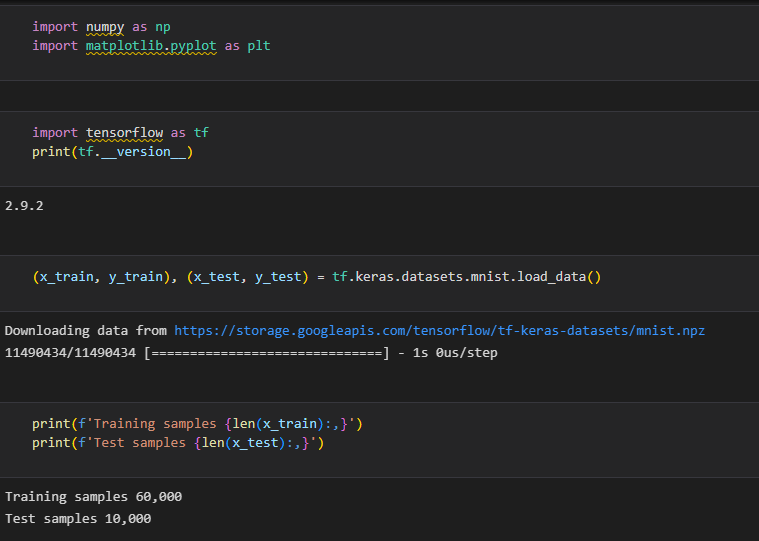
Python.

## Frontend:

For frontend, we have used Streamlit. Streamlit is an open-source app framework in Python language. It helps us create web apps for data science and machine learning in a short time.

# Working:

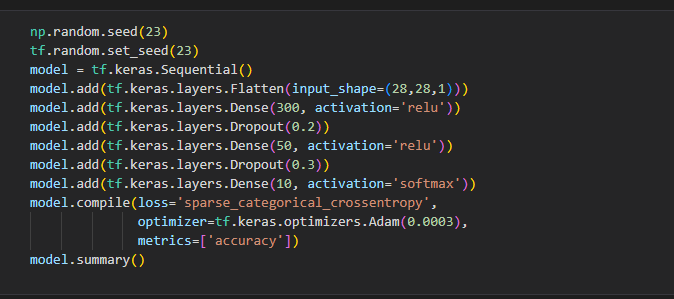
* **Importing the modules and loading the MNIST dataset from Keras:**



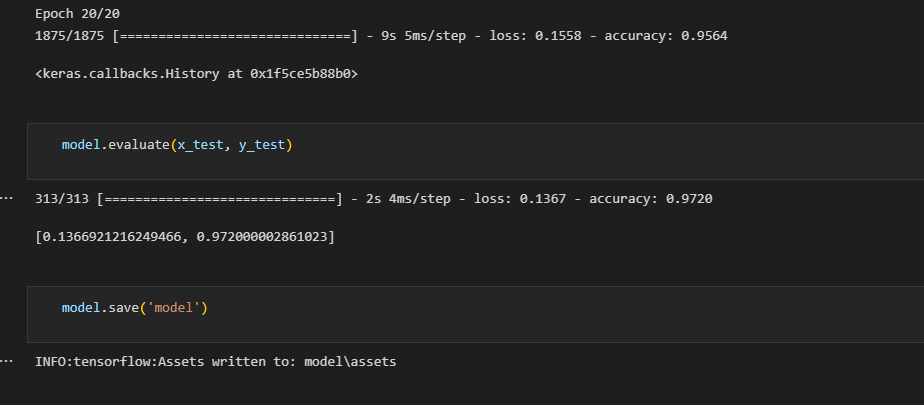
* **Seeing the Handwritten digit, Image.**



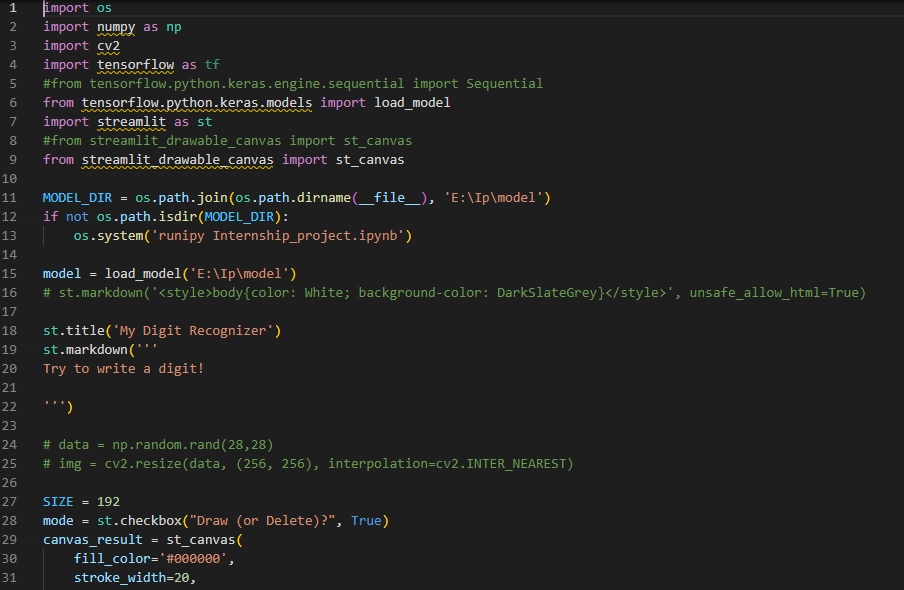
* **Flatten the Training Data:**



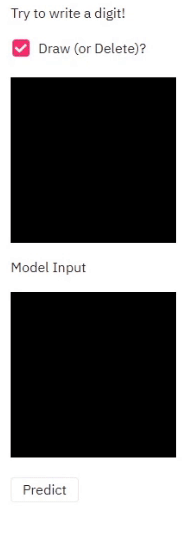
* **Evaluating the accuracy test:**



* **GUI frontend:**



# Output Screenshot:



# Conclusion:

Here we have demonstrated a model which can recognize handwritten digit. Later it can be extended for character recognition and real-time person’s handwriting. Handwritten digit recognition is the first step to the vast field of Artificial Intelligence and Computer Vision. As seen from the results of the experiment, CNN proves to be far better than other classifiers. The results can be made more accurate with more convolution layers and a greater number of hidden neurons. It can completely abolish the need for typing. Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning.