**A Project Report on**

**DIGIT RECOGNITION USING MNIST**

**(A Machine Learning model to recognize an input digit trained on MNIST dataset)**

**Submitted by :**

**Name:-** Seikh Aftave.

**INTRODUCTION:**

The "Digit Recognition using MNIST" predicts the handwritten digits from 0 – 9. It utilizes the MNIST dataset, comprising 70,000 grayscale images with a resolution of 28 by 28 pixels, the project showcases the efficacy of deep learning in image categorization assignments.

**PURPOSE OF THE PROJECT:**

The purpose of the project is to identify and predict the handwritten input digits from 0 -9. It can be used in many real-life scenarios such as automated-data entry, Optical Character Recognition (OCR), etc.

**APPROACH:**

­ In this project Convolutional Neural Network (CNN), which is a supervised Neural Network model is used to develop the machine learning model. As CNN is able to automatically learn spatial hierarchies of characteristics from input images, they are especially well-suited for this kind of work. Typically, the network consists of multiple layers: pooling layers to lower dimensionality, convolutional layers to extract features by the application of filters, and fully connected layers for classification. In order to reduce classification mistakes, the network is fed photos during the training phase. The network then uses gradient descent and backpropagation to modify its weights.

It also demonstrates how well they can recognize and infer intricate patterns from visual input. And to assess the model’s performance accuracy, loss and other measures are used. CNNs' efficacy in digit recognition, with applications in fields like automated data entry and optical character recognition (OCR) is highlighted in this project.

**USED TECHNOLOGY:**

**1)Python:**

A very easy to learn and human friendly programming language, having many advantages over other programming languages such as having a large amount of libraries. It is useful for a variety of tasks such as creating games, analysing data etc.

**2)Numpy:**

NumPy is a library for python, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays.

**3)Pygame:**

Pygame is a set of python modules used for writing games. It is used to create computer graphics, interface etc.

**4)Sys:**

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment.

**5)Tokenize:**

The tokenize module provides a lexical scanner for python source code, implemented in python.

**6)Keras:**

Keras is an open-source library that provides a Python interface for artificial neural networks. It is integrated into the Tensorflow library.

**7)TensorFlow:**

TensorFlow is a free and open-source software library for machine learning and artificial intelligence.

**8)Matplotlib:**

Matplotlib is a plotting library for python. It is used to plot data points in different plotting diagrams.

**9)Pandas:**

Pandas is a software library for python for data manipulation and data analysis. It offers data structures and operations for manipulating numericals and data sets.

**10)OpenCV:**

OpenCV is a library of programming functions for real\_time computer vision. It is used for Object Detection and video/image processing.

**SOFTWARES REQUIRED:**

1)VS Code

2)Jupyter Notebook

3)Anaconda

**STEPS:**

1)Import the necessary libraries in “reco.ipynb”.

2)Load the dataset in x\_train,y\_train & x\_test,y\_test and check if they are successfully loaded.

3)The data type of x\_train & x\_test is changed to float32 and a new dimension is added at the end.

4)The data in y\_train & y\_test is converted into a binary matrix. Where 1 is placed in place of integers and 0 in all other positions.

5) An empty sequential model is created and required layers are added.

6) Configuring the data using different parameters such as (optimiser=’adam’).

7)The model is trained through 5 epochs reaching an accuracy of 0.9889 and loss of 0.0345.

8)After training the data it is saved.

9) To test the model an interface is created.

10)Import the necessary libraries in “prjct.py” also load the model.

11)Different parameters, arguments and pygame is initialised.

12)A graphical user interface is created using pygame and user input is given there.

13)The input number is taken as an image and resize it to 28x28 to match the images in the dataset.

14)The resized and standardised image is then processed through the model to predict the number.

15)The prediction is then shown on the top-left side of the GUI inside a rectangle.