**Project Requirement and Specification**

**On**

**MNIST Digit Classification Machine Learning**

**(CSE 3rd Semester Mini project)**

2020-2021

****

**Submitted to: Submitted by:**

Mr. Deepak Uniyal. Ayush Gaurav

(CC-CSE-3rd -Sem) Roll. No.: 2014377.

Guided By: B tech. CSE-3rd -Sem

Mr. Ashwini Singh. Session: 2020-2021

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING.**

**GRAPHIC ERA UNVERSITY, DEHRADUN.**

**CONTENTS**

* **About the Project.**
* **System Required.**
* **Module of Project.**
* **Output and Accuracy.**
* **Conclusions.**
* **References.**
  1. **About Project**

## Project Overview

In this project, we will discover the MNIST handwritten digit recognition problems and we will develop a machine learning Model. The all dataset is used by the help of Online data Scraping and by the help of UCI ML repository or Kaggle that will help of achieving Best Results.

## Project Highlights

This project is designed to get you familiar to working with dataset in python and applying basic machine learning technique using Scikit-Learn. By the using of Scikit-learn and dataset from the online platform i .e. UCI ML repository and Kaggle.

By the End of this project you will learn Things like:

* How to Import data from online repository and the features of a dataset.
* How to design a model using Machine Learning with higher Accuracy.
* How to determine the best-guess model for predictions from unseen data.
* How to use Scikit-Learn and Use of Python in Different Platform.

## Description

The MNIST problem is a dataset developed by Yann Le Cun, Corinna Cortes and Christopher Burges for evaluating machine learning models on the handwritten digit classification problem.

The dataset was constructed from a number of scanned document dataset available from the National Institute of Standards and Technology (NIST). This is where the name for the dataset comes from, as the Modified NIST or MNIST dataset. Images of digits were taken from a variety of scanned documents, normalized in size and cantered. This makes it an excellent dataset for evaluating models, allowing the developer to focus on the machine learning with very little data cleaning or preparation required. how to build a hand-written digit classifier using the MNIST dataset. For someone new to deep learning, this exercise is arguably the “Hello World” equivalent.

It consists of 70,000 labelled 28x28 pixel grayscale images of hand-written digits. The dataset is split into 60,000 training images and 10,000 test images. There are 10 classes (one for each of the 10 digits). The task at hand is to train a model using the 60,000 training images and subsequently test its classification accuracy on the 10,000 test images.

For, making this project in Simpler way and time-efficient, we have to use PYTHON as a Language and PYCHARM as the IDLE module.

## Software & Libraries

This project uses the following software and Python libraries:

* Python (Latest Version 3.0)
* Tensor flow.
* Kaggle or UCI ML repository.
* Jupiter-Notebook.

## System Requirements

RAM: 2 GB (Minimum).

STORAGE: Based upon Project.

OS: Window 7 or above.

## Accuracy

This Model gets an accuracy of 1.00 on validation data.

## Starting the Project

Follow the Instructions in Notebook.

### ----Install

This project requires **Python** and the following Python libraries installed:

* Jupyter Notebook.
* TensorFlow

### ----Code

Template code is provided in the notebook file. You will also be required to use the dataset file to complete your work. While some code has already been implemented to get you started, you will need to implement additional functionality when requested to successfully complete the project.

### ----Run

In a terminal or command window jupyter notebook file,

This will open the Jupyter Notebook software and project file in your browser.

## Modules of Project

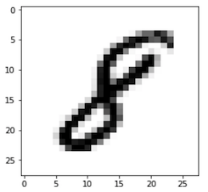
Training and Testing.

Analysis Model Performance.

Evaluating Model Performance.

## Conclusion and Discussion

In this section, the results of recognition system for offline handwritten Gurmukhi characters are presented. The K-Nearest-Neighbour has used in this study. The results are based on two feature extraction techniques, namely, diagonal and transitions features. As sated earlier, we have also experimented some partitioning strategies. The training and testing the data are done in Rapid Miner with 70% of the data in training and 30% for testing. The operator is first trained using the Training dataset and later the images in the testing dataset are send to the apply model operator to test the accuracy of the classifier. Figure 8 shows the main process layout of the Rapid Miner. All the dataset (training and testing) is combined and split into 70:30 ratio get independent results before sending it to be training and testing. Various classification models were used to check the accuracy of the model like Neural Net, K-NN, Random Forest, Decision Tree and Bagging with gradient boost.



A visualization of the sample image at index 7777.

## References

* Video Lect. On YOUTUBE.
* Introduction to ML(WEB)
* Using Libraries.

**THANK-YOU**