

MNIST Handwritten Digit Classification



This project demonstrates a **basic Machine Learning / Deep Learning model** built using **TensorFlow and Keras** to classify handwritten digits (0-9) from the popular **MNIST dataset**. It is a beginner-friendly project suitable for learning neural networks and for showcasing on a GitHub profile.

💡 Project Overview

The MNIST dataset contains **70,000 grayscale images** of handwritten digits: - **60,000** images for training - **10,000** images for testing

Each image is of size **28x28 pixels**. The goal of this project is to correctly identify the digit present in the image using a **fully connected neural network (ANN)**.

🌟 Features

- Uses TensorFlow & Keras
 - Data preprocessing and normalization
 - Artificial Neural Network (ANN)
 - Model training and evaluation
 - Accuracy calculation
 - Confusion Matrix visualization
 - Beginner-friendly and well-structured
-

👉 Technologies Used

- Python 3
 - TensorFlow / Keras
 - NumPy
 - Matplotlib
 - Seaborn
 - Jupyter Notebook
-

👂 Project Structure

```
MNIST-Digit-Classification/
|
├── Untitled.ipynb      # Jupyter Notebook (Model Implementation)
└── README.md           # Project Documentation
```

Installation & Setup

1. Clone the repository:

```
git clone https://github.com/your-username/MNIST-Digit-  
Classification.git
```

2. Navigate to the project directory:

```
cd MNIST-Digit-Classification
```

3. Install required dependencies:

```
pip install tensorflow numpy matplotlib seaborn
```

4. Open the notebook:

```
jupyter notebook Untitled.ipynb
```

Model Architecture

- Input Layer: 784 neurons (28×28 flattened image)
- Dense Layer: 10 neurons
- Activation Function: **Softmax**
- Loss Function: **Sparse Categorical Crossentropy**
- Optimizer: **Adam**
- Epochs: 5

Model Evaluation

- Accuracy is evaluated on test data
- Confusion Matrix is plotted to visualize predictions
- Model predicts handwritten digits from 0 to 9

Sample Output

- Accuracy Score
- Predicted vs Actual Labels
- Confusion Matrix Heatmap

Learning Outcomes

- Understanding image classification
 - Working with TensorFlow & Keras
 - Data preprocessing for neural networks
 - Model evaluation techniques
-

Author

Harsh Mishra

 Beginner ML / AI Enthusiast

 Exploring Deep Learning & Data Science

Future Improvements

- Use CNN instead of ANN
 - Increase epochs for better accuracy
 - Add model saving/loading
 - Deploy using Flask or Streamlit
-

License

This project is open-source and free to use for learning and educational purposes.

 If you like this project, don't forget to give it a star on GitHub! 