# Handwritten Character Recognition Project

## Overview

This project focuses on building a Handwritten Character Recognition system using deep learning and TensorFlow. The goal is to recognize handwritten characters with high accuracy and efficiency, leveraging a trained model that processes input data, performs predictions, and outputs the recognized characters.

## Features

* • Model Training: Trained using a deep learning model built with TensorFlow/Keras.
* • Data Preprocessing: Includes cleaning, normalization, and transformation of input data for better training performance.
* • Prediction: Capable of predicting handwritten characters based on input images.
* • User Interface: A basic interface for uploading handwritten images for recognition (if applicable).

## Installation

### Prerequisites

Make sure you have the following installed on your system:  
- Python 3.10 or higher  
- Virtual environment tools like `venv`  
- Git

### Setup Steps

1. Clone the repository:  
 ```bash  
 git clone https://github.com/AnkitaPimpalkar08/HandwrittenCharacterRecognition.git  
 cd HandwrittenCharacterRecognition  
 ```

2. Set up a virtual environment:  
 ```bash  
 python3 -m venv myenv  
 source myenv/bin/activate # On Windows use `myenv\Scripts\activate`  
 ```

3. Install dependencies:  
 ```bash  
 pip install -r requirements.txt  
 ```

## Usage

### Training the Model

1. Place your dataset in the appropriate data directory.  
2. Run the training script:  
 ```bash  
 python train\_model.py  
 ```

### Recognizing Characters

1. To recognize characters using a trained model, run:  
 ```bash  
 python recognize.py --input <image\_path>  
 ```

### User Interface (if applicable)

If you have a user interface for uploading handwritten images, follow the instructions provided in the UI module directory or documentation.

## Project Structure

- `train\_model.py`: Script for training the model.  
- `recognize.py`: Script for recognizing handwritten characters.  
- `data/`: Directory for storing datasets.  
- `models/`: Directory for storing trained models.  
- `myenv/`: Virtual environment directory (not included in version control).

## .gitignore

This project uses a `.gitignore` file to prevent tracking unnecessary files and directories. Key entries include:  
```  
# Virtual Environment  
myenv/  
\*.pyc  
\_\_pycache\_\_/  
  
# TensorFlow Large Files  
myenv/lib/python3.10/site-packages/tensorflow/libtensorflow\_cc.2.dylib  
myenv/lib/python3.10/site-packages/clang/native/libclang.dylib  
myenv/lib/python3.10/site-packages/tensorflow/compiler/mlir/stablehlo/stablehlo\_extension.so  
```

## Contributing

Contributions are welcome! If you'd like to improve or modify this project:  
1. Fork the repository.  
2. Create a feature branch (`git checkout -b feature-branch`).  
3. Commit your changes (`git commit -m 'Add a new feature'`).  
4. Push to the branch (`git push origin feature-branch`).  
5. Open a Pull Request.

## License

This project is licensed under the [MIT License](LICENSE).