Methodology Documentation

Chanaka Prasanna – chanakapinfo@gmail.com

Introduction

This document outlines the methodology for developing a system to extract and translate Sinhala characters from handwritten images into English. The project involved three distinct approaches to achieve the goal, each with varying techniques and architectures.

Table of Contents

Table Of Figures

Approach 1 - Initial Implementation

Methodology

1. **Tech Stack** - Used OpenCV for image pre-processing, Tesseract OCR for character recognition, and Google Translate for translation.
2. **Process**
    - **Image Pre-processing** - Converted images to grayscale, removed noise, applied binary thresholding, and segmented text regions.
    - **OCR** - Applied Tesseract OCR to recognize Sinhala characters from the processed text regions.
    - **Translation** - Used Google Translate API to translate the recognized Sinhala text into English.
3. **Outcome** - This approach successfully translated Sinhala text images with printed text but failed to provide accurate results for handwritten text images. The limitations were observed in the character recognition stage due to the complexity of handwritten text.

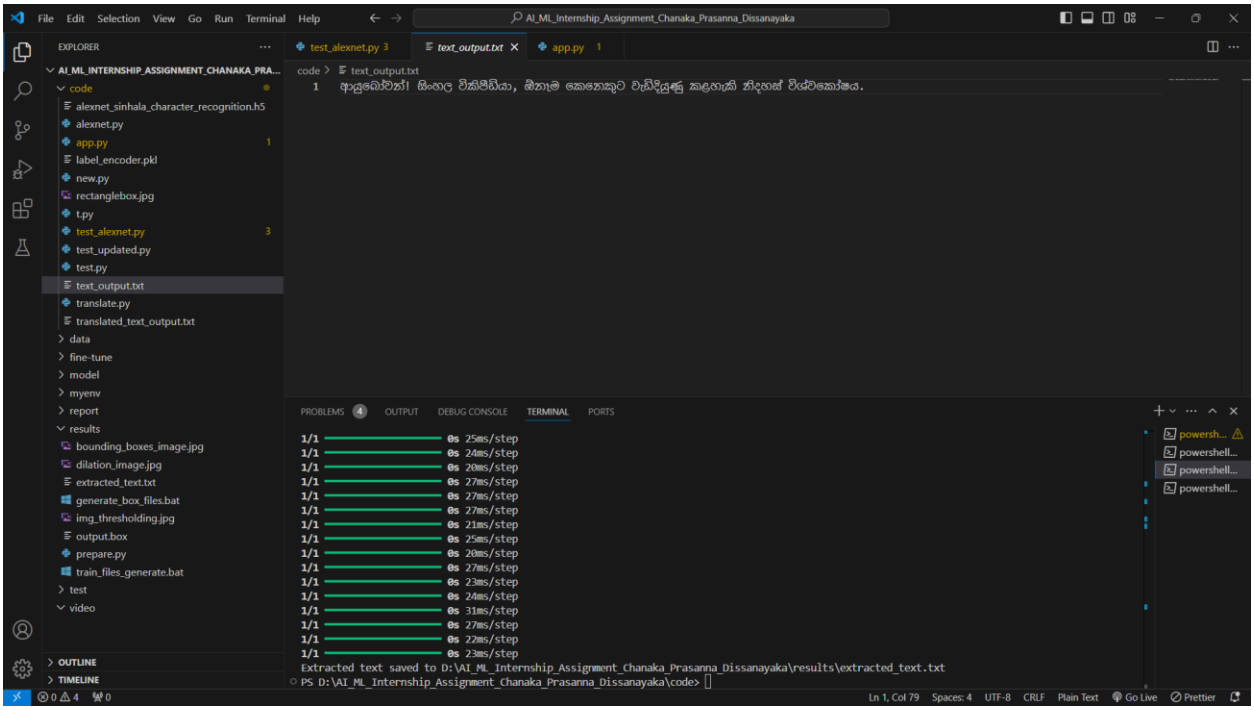


Figure 1 - Output from a image of digital sinhala letters

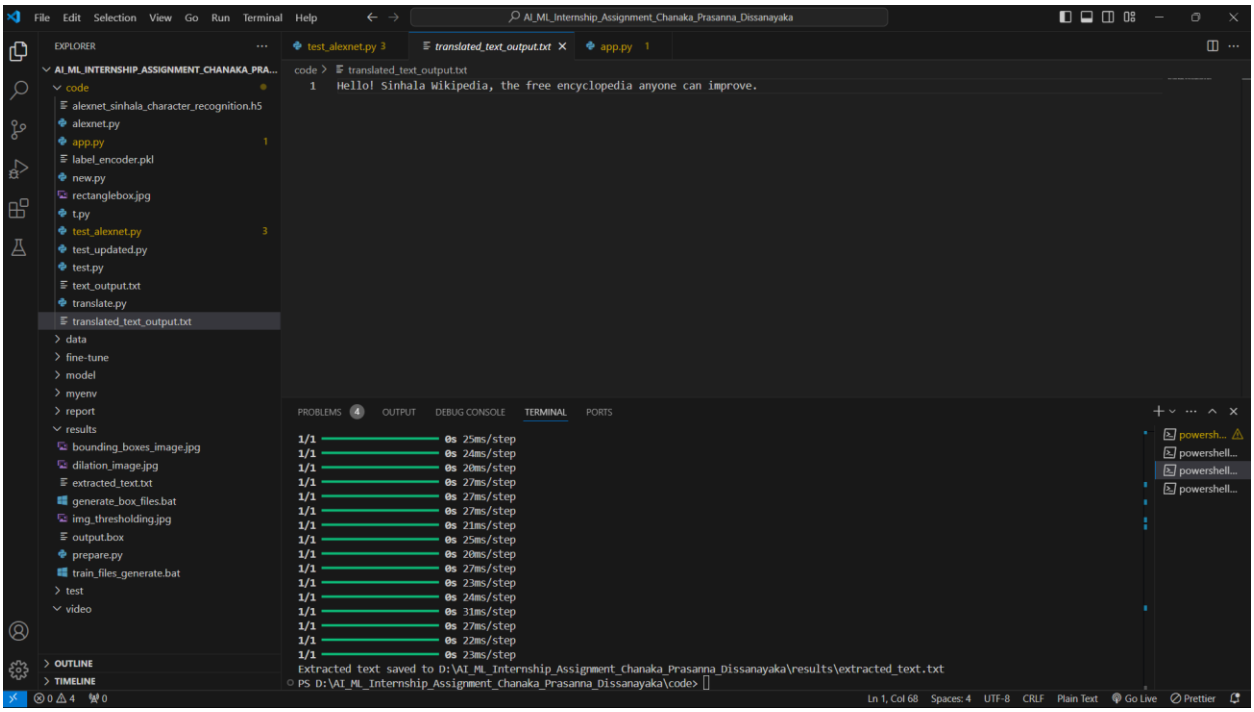


Figure 2 - Translated output

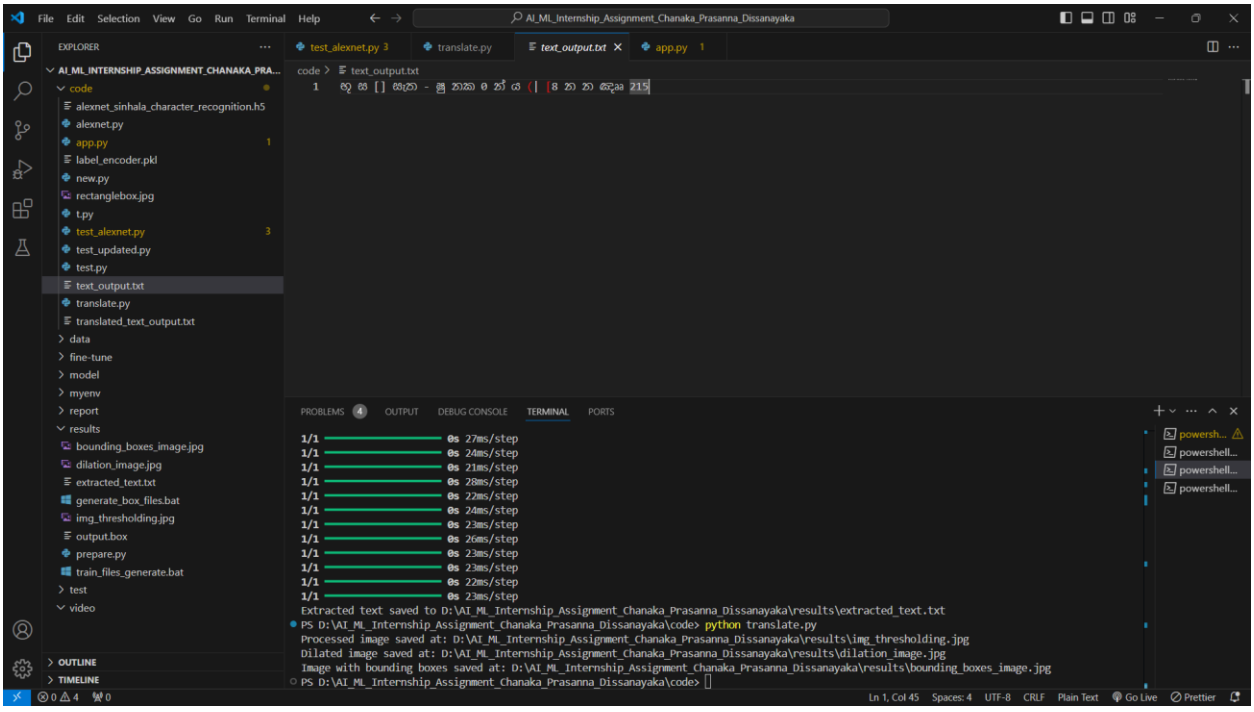


Figure 3 - Text extraction from handwritten text image

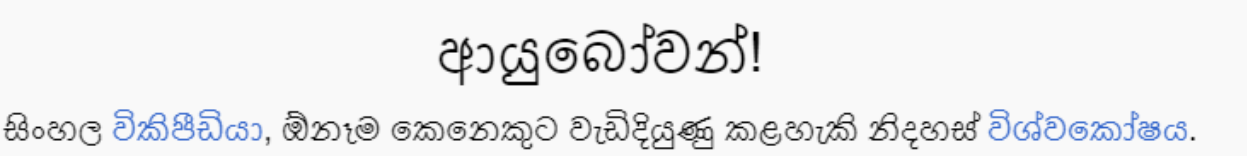


Figure 4 - Tested image with digital characters

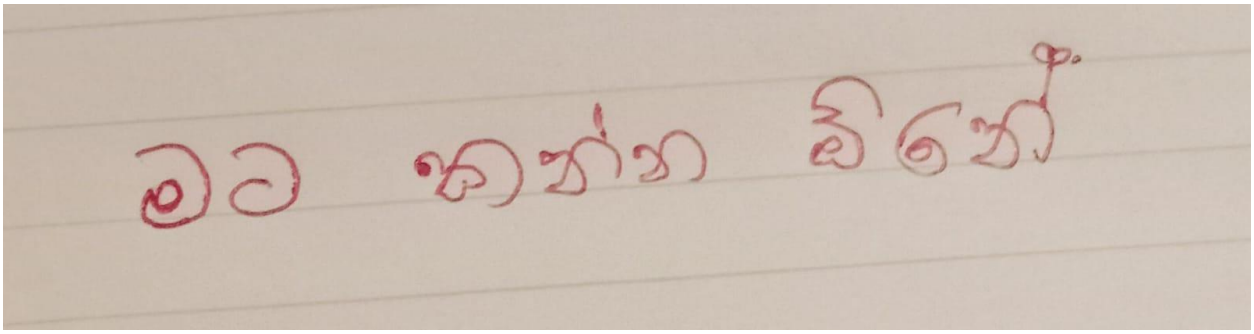
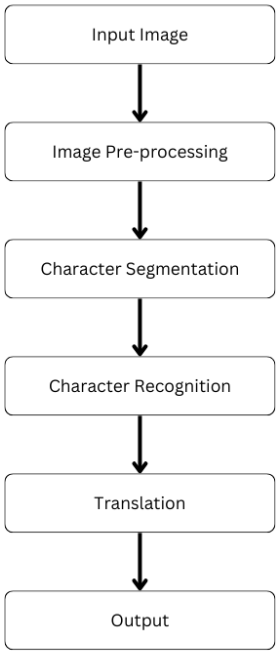


Figure 5 - Tested image with handwritten letters

Architecture

This is the high-level architecture for 1st approach



Approach 2 - Advanced Recognition Using CNN Architectures

Methodology

1. **Tech Stack** - TensorFlow/Keras, including layers such as Conv2D, MaxPooling2D, Flatten, Dense, and Dropout.
2. **Process**
    - **Data Preparation** - Used a dataset provided in the instructions PDF for training the models.
    - **Model Implementation** - Implemented convolutional neural networks (CNNs) using AlexNet and LeNet architectures.
    - **Training** -Trained the models on the provided dataset to recognize handwritten Sinhala characters.
    - **Evaluation** - Tested the trained models with handwritten images to evaluate performance.
3. **Outcome** – In referred research paper, AlexNet showed better performance, the models still produced inaccurate results on handwritten images, generating unusual text patterns. The issue was likely due to the training dataset not being sufficient for capturing the variability in handwritten text.

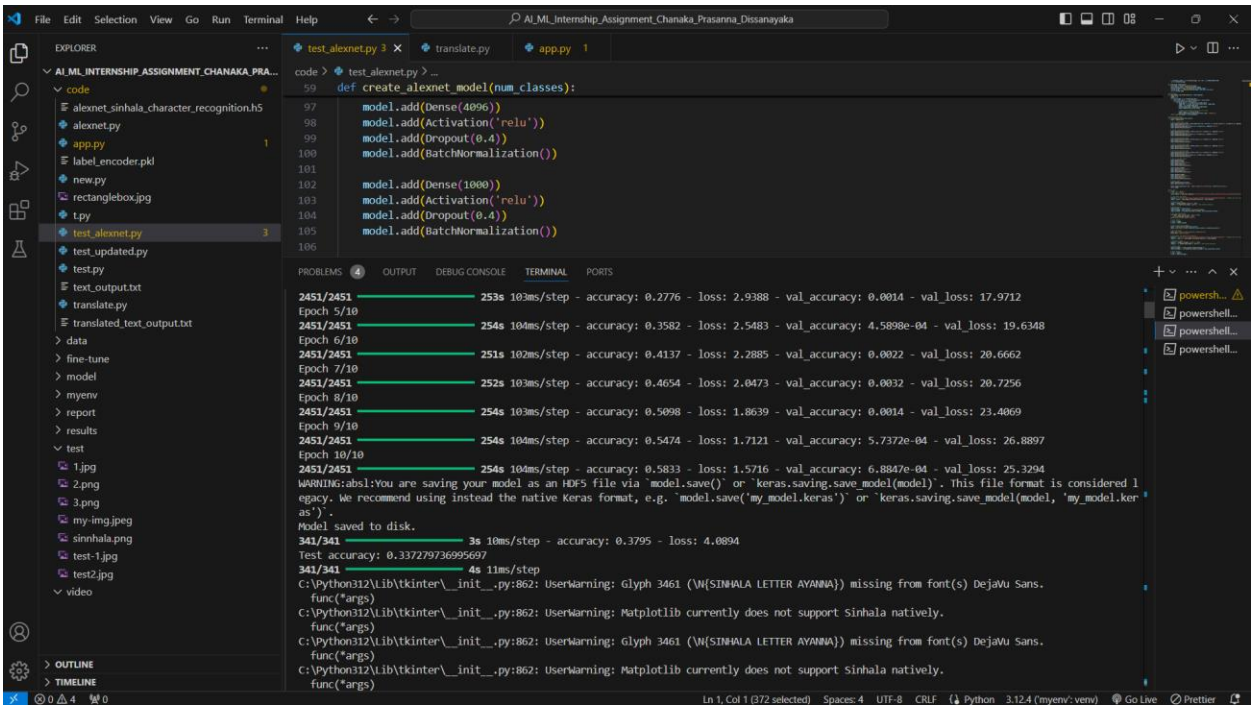


Figure 6 - Training process

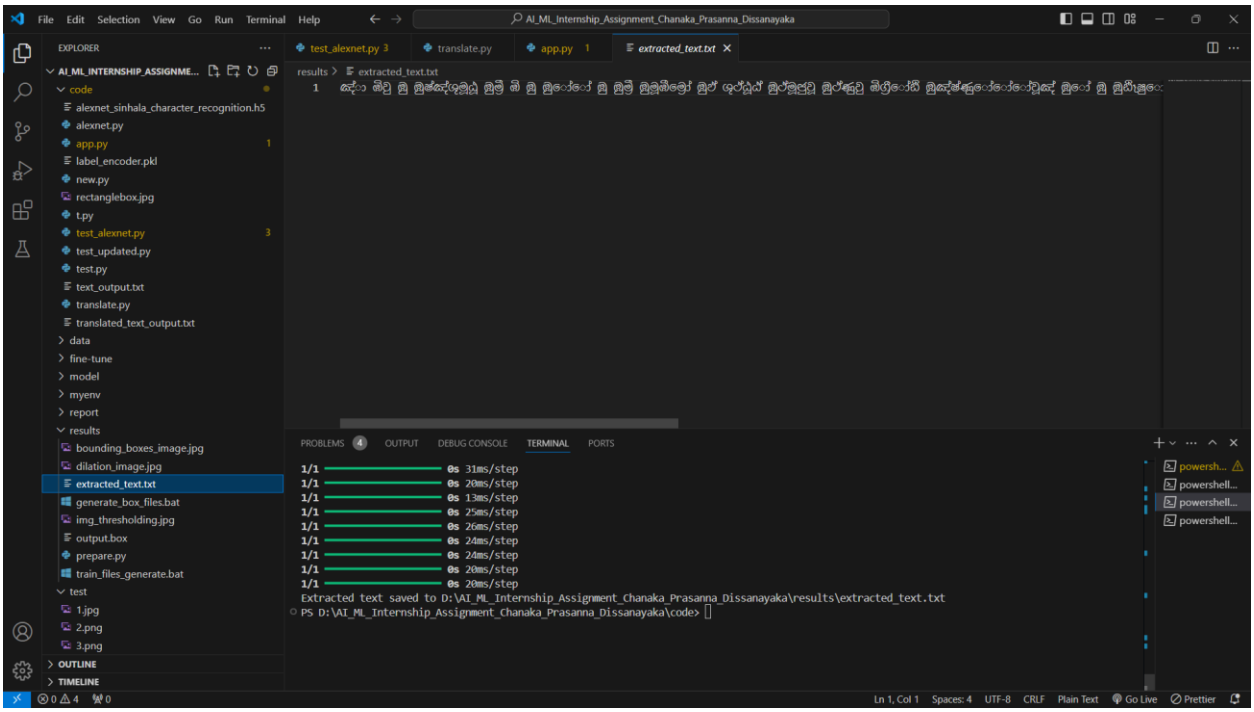
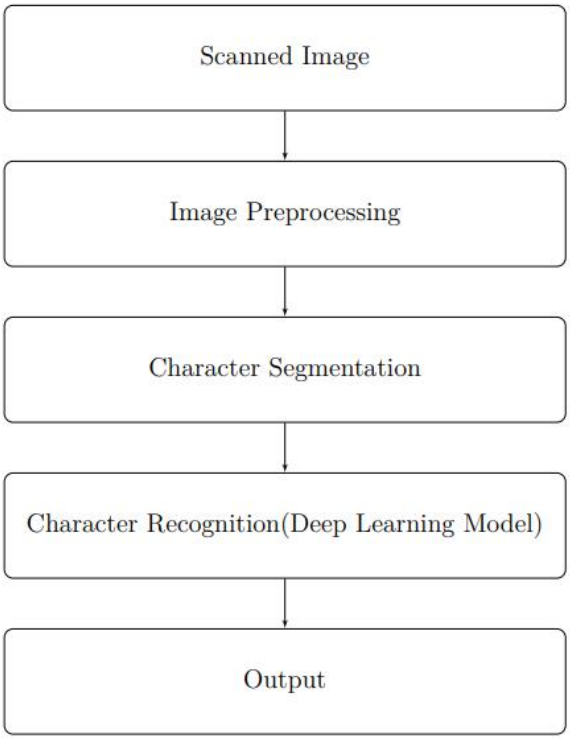


Figure 7 - Extracted letters

Architecture

This is the high-level architecture I used in 2nd approach



Approach 3 - Fine-Tuning Tesseract OCR

Methodology

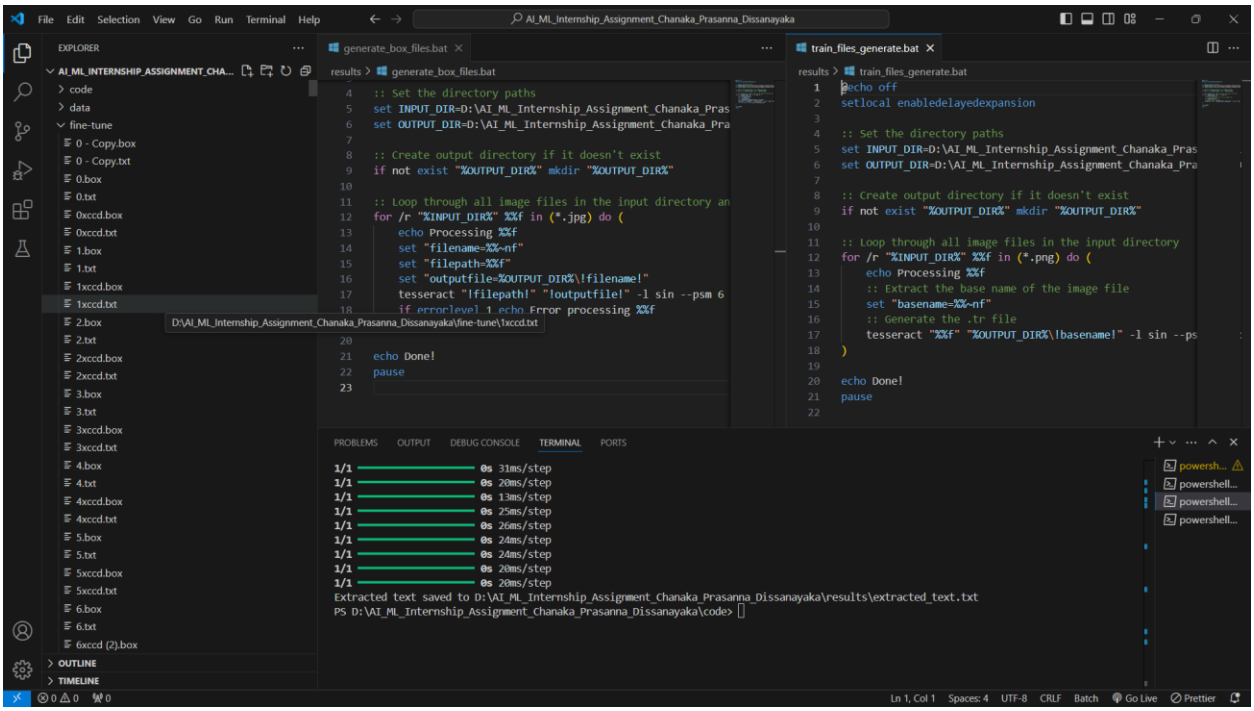
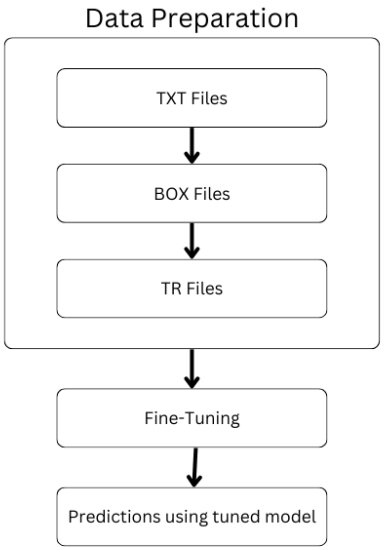
1. **Tech Stack** - Tesseract OCR, Python
2. **Process**
    - **Data Preparation**
        - **TXT Files** - Generated text files for each training image using a script (prepare.py).
        - **BOX Files** - Created .box files for training images using a .bat script.
        - **TR Files** - Attempted to generate .tr files for training but encountered difficulties.
    - **Fine-Tuning** – It was unable to fine tune since the errors occurred in .tr file generations.
3. **Outcome** - Although significant effort was made in fine-tuning Tesseract OCR, challenges with generating .tr files led to incomplete fine-tuning.
4. 

Figure 8 - Generated .txt, .box files and .bat files

Architecture



Conclusion

The project explored three approaches to address the challenges of translating handwritten Sinhala text into English. The initial implementation provided a basic solution but lacked accuracy for handwritten text. The use of CNN architectures offered a more sophisticated approach but still faced limitations with handwritten data. Finally, the fine-tuning of Tesseract OCR aimed to enhance recognition accuracy, though challenges in data preparation affected the fine tuning process. Each approach contributed valuable insights into handling handwritten text recognition and translation.

Mian references

Research Conducted by UCSC - https://drive.google.com/file/d/1uPZ0WwaqJ6z13B9E1ndNYljQjrDX6uUS/view?usp=sharing

Article From GeekyAnts - https://geekyants.com/blog/exploring-optical-character-recognition-ocr-an-experiment-with-opencv-and-pytesseract

Youtube tutorial - https://youtu.be/sfheWK72L74?si=ts55hhABG0yBGXz6