American Dream

Martin Lahoumh, Brandon Tjandra, Jiazhou Zhang, Miguel Luna
https://github.com/MartinLahoumh/SeniorDesign/

The City College of New York
Spring ~ Fall 2024
Yunhua Zhao

Background

- The goal of this project is to create a web application that **scans an image for text** and **translates it** to a user's desired language.
- The machine learning aspect of this project is utilizing an **OCR (optical character recognition)** model.
- The extracted text will be translated via a **translation API**, to which it would be returned **on top of the supplied image.**

Technologies

Our OCR model is **EasyOCR:**

- Text detection and recognition all-in-one
- Pre-trained using **ICDAR datasets**
- Recognition of **80+ languages**
- Invoked in a few Python lines

Our translation API uses **Google Translate**:

- **Python wrapper** for Google Translate website
- **Free to use**, no API key needed
- Provides translation **confidence score**

Application frontend interface done in **React**.
Application backend done with **Flask**.

Previous Work Summary

Prior to using EasyOCR, we tried creating our own OCR models, which were **dropped** for **concerns of inaccuracy,** and **feasibility** for time and resources in implementing a full OCR pipeline.

We used the following image + annotation datasets:

- ICDAR-2015: 500 images
- TextOCR: 25,000 images
- COCO-Text: 60,000 images

Implementation 1:
TensorFlow with MNIST dataset
Dropped due to inability to locate text properly. **No text detection ability.**

Implementation 2:
TensorFlow + CTC text recognition
Dropped due to the reasons above, as well as a **lack of suitable training data** and **intensity** on time and resources.

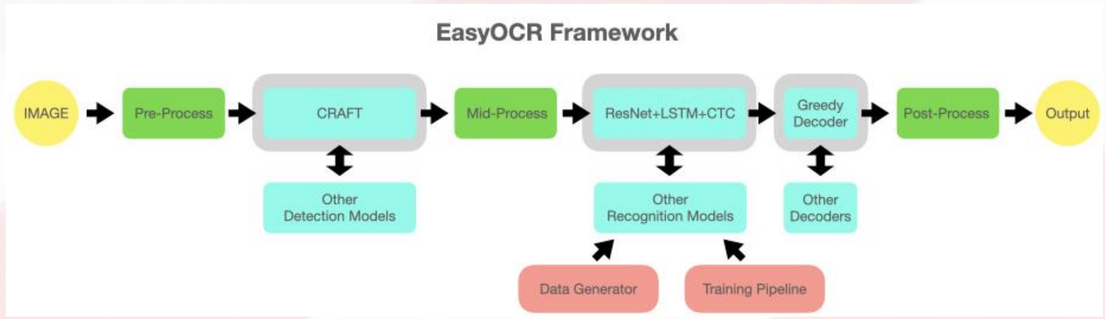

Figure 1: EasyOCR framework.
This also describes the machine learning models and data processing steps required to implement the typical OCR pipeline.

Issues & Future Plans

- Address model issues in recognizing **transformed text** (rotation, shear, etc.)
- Errors with recognizing or translating certain **non-alphabetical languages**
- **Overlaying text** on image with proper text color and size
- **User account functionality** to save past inferences
- **Mobile application** interface

Research

Article	Summary	Takeaways
Character Region Awareness for Text Detection 2019 Baek et al.	Proposal of **CRAFT** text detection model. **CNN** that uses **character-level** text detection and **character affinity** to detect **location** of words in images. Trained with **ICDARs** and **MSRATD500.**	Text detection models for scene text struggle with accuracy due to **word contexts and meanings**. It is used in the OCR pipeline of **EasyOCR**, so it is the text detection model for this project.
What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis 2019 Baek et al.	Training scene text recognition (STR) models are costly, making it **difficult to obtain labeled data**. The authors resort to using synthetic text datasets like **MJSynth** and **SynthText**. This brings up the question of whether performance improvements are due to the **model itself or a better set of training data**.	Creating and training our own model from scratch would be infeasible due to a **lack of datasets suited for scene text detection.** This is why we believe that using a **pre-trained OCR** such as EasyOCR is **necessary** for the completion of this project given our time and knowledge limitations.