**Documentation for Sinhala Inscription Recognition Model**

**Introduction**

This project is designed to recognize Sinhala inscriptions from images using a deep learning model. The solution integrates multiple components, including an EAST (Efficient and Accurate Scene Text Detector) model for text detection, a TrOCR model for text recognition, and Google Translate for translation. EAST was chosen for its specialized text detection capabilities. EAST is more effective in detecting irregularly shaped text regions, which is crucial for recognizing inscriptions in various conditions.

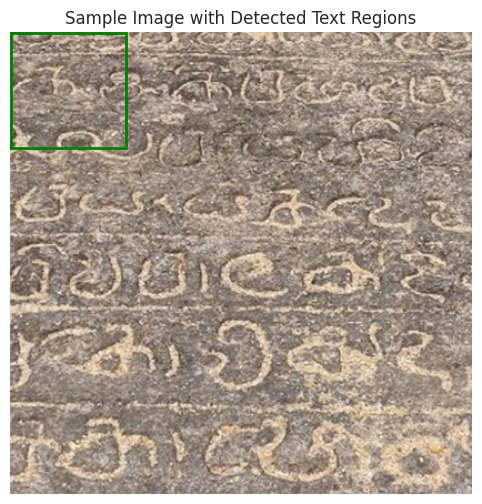
**WHY I choose EAST?**

The dataset provided contains **148 images** of Sinhala inscriptions. These images were used to train and evaluate the model's performance. The dataset's limited size required effective augmentation strategies and careful model tuning to achieve optimal results.

Although YOLOv8 is a powerful and versatile object detection model known for its speed and accuracy in identifying objects within images, it is not optimized for text detection. EAST (Efficient and Accurate Scene Text Detector), on the other hand, is designed specifically for text localization tasks. EAST offers several distinct advantages that make it more suitable for this project:

1. **Precise Pixel-Level Localization:** EAST generates score and geometry maps that predict the exact text region boundaries at the pixel level. This precision is crucial for detecting small, intricate text details that are common in Sinhala inscriptions.
2. **Curved and Irregular Text Detection:** Unlike YOLOv8, which excels at detecting objects with well-defined boundaries, EAST efficiently handles text that appears in curved, rotated, or distorted forms. Sinhala inscriptions often include such text patterns, making EAST a better fit.
3. **Minimal False Positives:** EAST’s text region prediction mechanism reduces false detections by producing clear and accurate boundaries for text areas. This helps to minimize errors in noisy backgrounds, which are common in historical inscriptions or outdoor scenes.
4. **Efficient Region Proposal Mechanism:** EAST eliminates the need for a region proposal network (RPN), making it faster and more efficient for text detection tasks. Its streamlined architecture focuses solely on text region prediction without introducing excessive computational overhead.
5. **Text Size Versatility:** EAST effectively handles text regions of varying sizes, ensuring consistent performance across both small inscriptions and larger text structures.

Considering these advantages, EAST was the optimal choice for recognizing Sinhala inscriptions, where text can appear in complex patterns, uneven orientations, and challenging environmental conditions. While YOLOv8 is powerful for object detection, EAST’s specialized design proved superior for this specific text detection task.



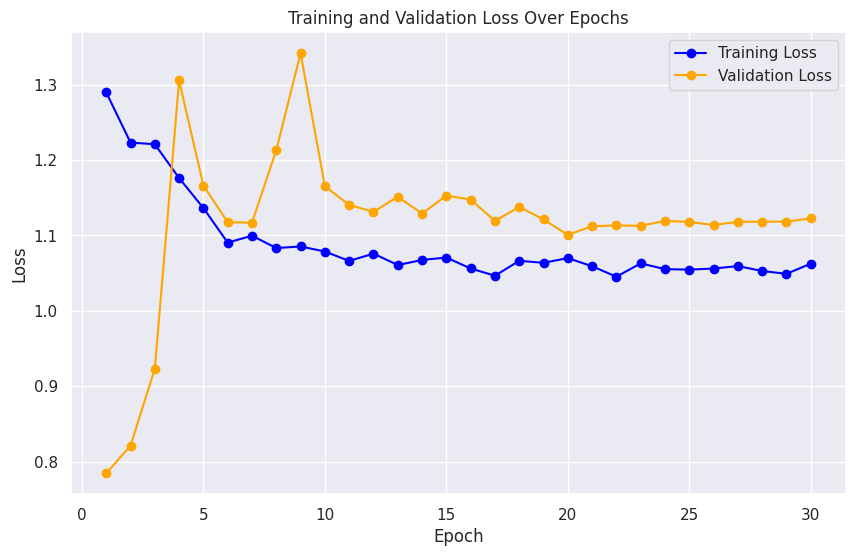
**Code Implementation**

**Imports and Libraries**

The implementation begins with key library imports essential for building the API, text detection, recognition, and translation pipeline. FastAPI is utilized for API development, providing a robust framework for handling HTTP requests and serving model predictions efficiently. PyTorch is included for defining and implementing the EAST model, while OpenCV is employed for image preprocessing and contour detection. The transformers library is integrated to load TrOCR and BERT models for text recognition and completion. Additionally, the googletrans library is imported to enable seamless Sinhala-to-English text translation.

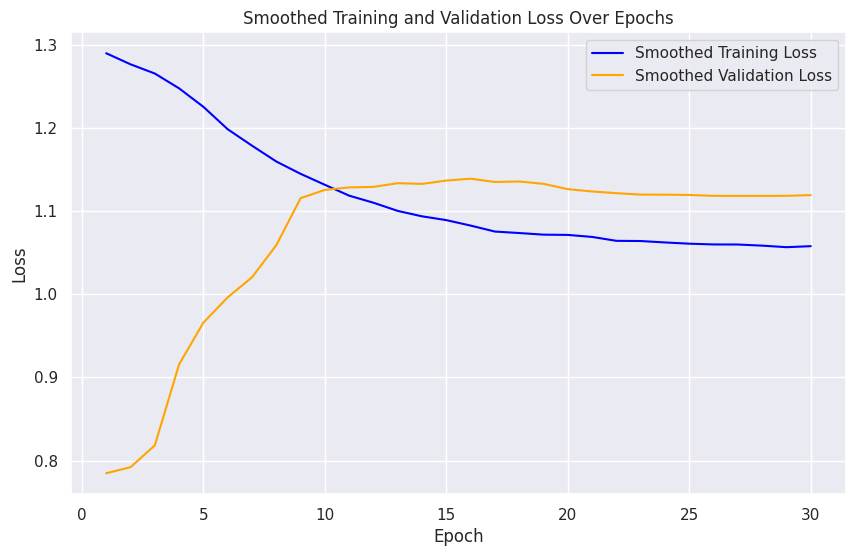
**EAST Model Implementation**

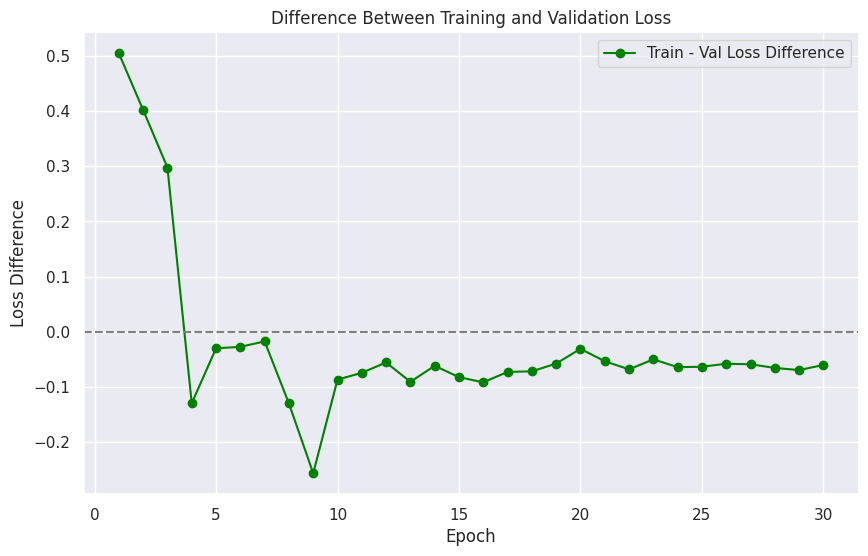
The EAST model is constructed using PyTorch's nn.Module. The architecture follows a convolutional backbone structure that effectively extracts image features. These extracted features are passed through a merge layer, which refines and combines multi-scale features to improve text localization accuracy. The score and geometry layers play a crucial role in predicting text region probabilities and determining bounding box coordinates, ensuring accurate identification of text regions.



**Model Initialization and Loading**

The EAST model is instantiated and loaded with pre-trained weights stored in the east\_model.pth file. This enables the model to detect text regions efficiently without extensive retraining. Alongside the EAST model, the TrOCR model is initialized to perform text recognition tasks efficiently. The BERT model is also loaded to improve text accuracy by predicting missing or unclear text elements. Furthermore, the Google Translator API is prepared for translating detected Sinhala text to English.





**Image Preprocessing**

The preprocess\_image() function is designed to convert uploaded image data from bytes to a NumPy array. This ensures compatibility with OpenCV and PyTorch models. The function decodes the image, resizes it for model compatibility, and converts it from BGR to RGB format. This color conversion aligns the image with PyTorch's default image format for accurate model predictions.

**Text Detection with EAST**

The detect\_text\_regions() function is responsible for identifying text regions in images. The uploaded image is resized to 512x512 pixels before being passed through the EAST model. The output consists of score and geometry maps, which are analyzed using OpenCV's findContours() function. This method efficiently pinpoints text regions by examining image boundaries, ensuring accurate text localization.

**Text Recognition**

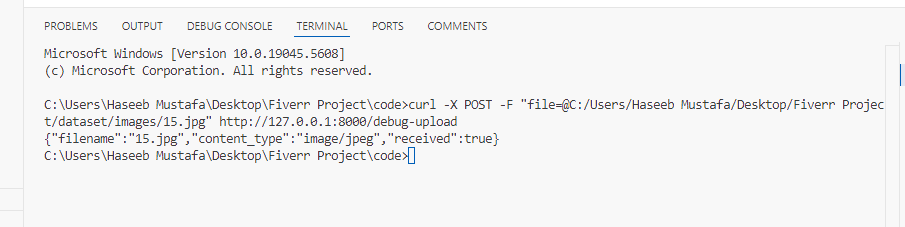
The recognize\_text() function extracts detected text regions from the original image for text recognition. The TrOCR model processes these regions to identify the text content. In cases where the detected text appears incomplete or ambiguous, the predict\_missing\_text() function is called to utilize the BERT model for filling in the missing text, ensuring improved text clarity and accuracy.

**Sinhala-to-English Translation**

The translate\_to\_english() function integrates the Google Translator API to translate detected Sinhala text into English. This step is crucial for ensuring that non-Sinhala speakers can understand the identified text, enhancing accessibility and usability for diverse audiences.

**Final Testing Results:**

****

****

**Model Workflow**

The overall workflow of the model is as follows:

1. **Image Upload:** The user uploads an image via the FastAPI endpoint.
2. **Preprocessing:** The image is resized and converted for model compatibility.
3. **Text Detection:** The EAST model identifies potential text regions.
4. **Text Recognition:** TrOCR extracts text from detected regions.
5. **Text Completion:** If text is incomplete or incorrect, BERT predicts missing content.
6. **Translation:** Detected Sinhala text is translated to English.
7. **Output:** The system returns detected text, translations, and bounding box coordinates.

**Conclusion**

This project effectively combines EAST for robust text detection, TrOCR for accurate text recognition, and Google Translate for language conversion. The EAST model's superior performance in detecting text within complex backgrounds made it the preferred choice over YOLOv8. The comprehensive workflow ensures reliable results for Sinhala inscription recognition in various image conditions.