**AI-Powered Signboard Translation**

# **Introduction:**

The goal of this project is to develop a robust deep learning system capable of performing end-to-end signboard translation in real time. This involves detecting text regions within an image, recognizing the text content (OCR), translating the recognized text into a target language, and overlaying the translated output back onto the image.

The system leverages state-of-the-art models from the fields of computer vision and natural language processing (NLP), and is designed to be adaptable to multiple languages and real-world conditions. The project is particularly valuable for:

* **Travelers** navigating foreign environments
* **Language learners** engaging with real-world written content
* **Accessibility tools** supporting multilingual signage comprehension
* **Tourism and AR applications** requiring live language translation

The pipeline is optimized for GPU-based inference to ensure low-latency processing suitable for real-time video feeds or mobile deployment. The project showcases how modern Transformer-based architectures like TrOCR and MarianMT can be integrated into a unified vision-to-language translation system.

# **Models Used**

**1 TrOCR (Transformer-based OCR)**

* **Model:** microsoft/trocr-base-printed
* **Architecture:** VisionEncoderDecoderModel (ViT encoder + RoBERTa decoder)
* **Use:** Convert cropped image regions into English text
* **Training:** Fine-tuned on TextOCR word crops (500+ samples, 3 epochs)
* **Advantages:** End-to-end image-to-text with Transformer accuracy

**2 MarianMT (Translation)**

* **Model:** Helsinki-NLP/opus-mt-en-ar
* **Architecture:** Encoder-decoder based on Marian NMT
* **Use:** Translate OCR output text from English to Arabic
* **Batch Translation:** Used to speed up performance and enable GPU inference

**3 Auxiliary Tools**

* **Preprocessing:** OpenCV for cropping and overlay
* **Evaluation:** Custom Levenshtein-based CER and WER functions
* **Frameworks:** Hugging Face Transformers, Datasets, PyTorch

# **Workflow**

1. **Image Input:** An input image is selected either from the COCO-Text dataset or the TextOCR dataset. Each image is assumed to contain one or more signboards with legible text in a real-world setting.
2. **Bounding Box Extraction:** If using COCO-Text, bounding boxes are extracted directly from the dataset annotations. If using another source, object detection or text detection (e.g., using CRAFT or EAST models) can be used to detect signboard text regions.
3. **OCR Preprocessing:** Each bounding box is cropped using OpenCV. The cropped regions are converted to RGB format and resized/padded if needed. These images are then passed to the TrOCR processor for normalization and tensor conversion.
4. **OCR Inference (TrOCR):** The processed image crops are passed through the fine-tuned TrOCR model. The model outputs a predicted English transcription of the text content using a Vision Transformer (ViT) encoder and RoBERTa decoder.
5. **Translation (MarianMT):** The OCR output (English text) is then tokenized and translated to Arabic using a MarianMT model (opus-mt-en-ar). Batch translation is used to increase efficiency. Translation is performed on GPU for faster throughput.
6. **Overlay Translated Text:** The translated Arabic text is overlayed on the original image using OpenCV's putText() function. Font size, color, and placement are adjusted for readability. The text is placed either above or beside the bounding box.
7. **Visualization and Export:** The final output image is visualized using Matplotlib. Optionally, the output can be saved as annotated images or video frames to demonstrate the real-time translation capability. Multiple bounding boxes in the same image are supported and processed in batches.

This modular pipeline ensures flexibility and supports extension to other language pairs or domains (e.g., traffic signs, store fronts, etc.).

# **Dataset Overview**

This project utilizes two complementary datasets to cover both detection and OCR tasks:

**2.1 COCO-Text**

* **Source:** <https://bgshih.github.io/cocotext/>
* **Images:** ~63,000 images with over 173,000 annotated text instances
* **Content:** Scene text with bounding boxes, legibility, and language labels
* **Use Case:** Text localization and initial bounding box extraction

**2.2 TextOCR**

* **Source:** <https://github.com/microsoft/TextOCR>
* **Images:** ~28,000 images from OpenImages with 900,000+ word-level annotations
* **Content:** High-quality OCR text data with accurate polygon annotations
* **Use Case:** OCR model training/fine-tuning on real-world signboard text

# **Results**

**Evaluation Metrics**

|  |  |
| --- | --- |
| **Metric** | **Description** |
| Accuracy | Exact match of OCR vs. ground truth text |
| CER | Character Error Rate (Levenshtein distance) |
| WER | Word Error Rate (word-level edit distance) |

**Results Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **CER** | **WER** |
| **Pretrained** | 40.00% | 0.4211 | 1.6000 |
| **Finetuned** | 84.00% | 0.4954 | 2.1400 |

* ***References***:
* [TrOCR Model Paper (Microsoft)](https://arxiv.org/abs/2109.10282)
* [COCO-Text Dataset](https://bgshih.github.io/cocotext/)
* [TextOCR Dataset](https://github.com/microsoft/TextOCR)
* Helsinki-NLP MarianMT
* Hugging Face Transformers Documentation
* OpenCV Text Rendering

***Conclusion***

This project demonstrates the integration of advanced deep learning techniques for real-time signboard translation. The combination of OCR and translation models allows for a modular and extensible framework. Future improvements can include expanded training data, more robust detection, and deployment on mobile or AR platforms.

Made by:

Arwa Sameh Salah 211001892

Omar Mohamed Gera 212002407

Khaled Bassem Zaki 211001783

Omar Hesham Ghareeb 211002041