REAL-TIME TEXT DETECTION, RECOGNITION, AND TRANSLATION SYSTEM FOR LIVE VIDEO STREAMS

Yiling H., Ziyang L., Derek Z., Yichi Z.

***Abstract*—This paper presents a real-time text detection, recognition, and translation system designed for live video streams, with a strong focus on accessibility, travel assistance, and automation. Leveraging advanced deep learning-based computer vision techniques, our system accurately detects and recognizes text in complex, multilingual scenes and integrates with online translation services to provide immediate translations. By processing inputs from webcams or uploaded media, the system enables on-the-go understanding of foreign signage, assistance for visually impaired users, and efficient automation in industries such as advertising, logistics, and education.**

***Index Terms*—Text Detection, Text Recognition, Multilingual Translation, Real-Time Video Processing, Accessibility, Deep Learning, Computer Vision.**

I. INTRODUCTION

REAL -time text understanding in dynamic environments—such as navigating a foreign city or assisting visually impaired individuals—has gained increasing importance with the rise of wearable AI devices. Recent consumer products (e.g., Meta's AI-enabled glasses) exemplify the demand for instant text comprehension and translation in daily life. This work introduces a web-based application that captures input from images, videos, or live streams and translates recognized text into one of six supported languages. The system addresses critical real-world challenges, including low-quality inputs, cluttered backgrounds, hardware constraints, and the complexity of multilingual script handling.

Real-time text detection and recognition systems have significant potential in numerous applications: from enhancing tourism by translating foreign signage, to aiding visually impaired users in reading labels or documents, to streamlining logistics through automated extraction of textual information from package labels. By achieving a high degree of accuracy and speed without extensive GPU or cloud infrastructure, this solution opens doors for scalable and cost-effective deployments.

II. PROBLEM STATEMENT

Developing a robust, real-time text detection and translation system must overcome several key challenges:

1. **Low-Quality Inputs:** Noisy, blurred, or low-resolution video frames can degrade text detection accuracy.
2. **Complex Environments:** Scenes often contain cluttered backgrounds, curved or stylized text layouts, and varying lighting conditions.
3. **Hardware Limitations:** Achieving real-time performance on CPU-only setups remains challenging without sacrificing accuracy or scalability.
4. **Multilingual Translation:** Supporting diverse scripts, fonts, and writing systems requires a flexible and language-agnostic pipeline.
5. **Scalability:** Limited funding and resource constraints hinder adopting large-scale, cloud-based solutions.

III. SYSTEM ARCHITECTURE

The proposed architecture integrates text detection, recognition, and translation into a unified pipeline that operates seamlessly within a web application environment. Fig. 1 illustrates the high-level architecture.

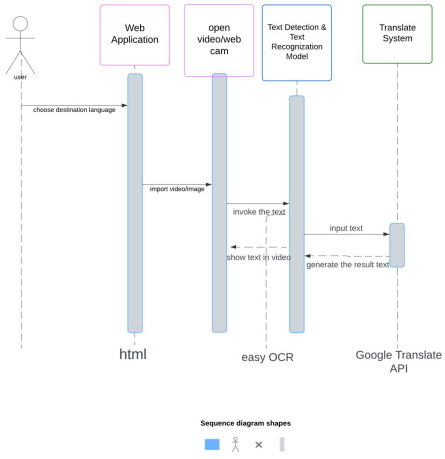

Fig 1. System's Structure

A. Workflow Overview

1. **User Interaction:** The user accesses the system via a web interface, selecting a target language and providing input through live video (via a webcam) or

uploaded images.

2. **Text Detection and Recognition:** The incoming frames are processed by EasyOCR, which employs advanced models for real-time text detection and recognition.
3. **Translation:** Detected text strings are dispatched to the Google Translate API, ensuring rapid and accurate multilingual translations.
4. **Output Display:** The translated text is presented to the user either as an overlay on the live video or as displayed output beneath the video feed.

IV. Text Recognition Methodology

The text recognition process comprises several stages to ensure both speed and robustness:

1. **Preprocessing:** Input frames are normalized through noise reduction, contrast enhancement, and resizing. These operations mitigate poor lighting conditions and busy backgrounds, yielding cleaner input for the detection model.
2. **Text Detection via CRAFT:** We employ the Character Region Awareness for Text Detection (CRAFT) model [1], which identifies character-level regions and excels in handling irregular text layouts and curved scripts.
3. **Region Refinement:** The detected character regions are grouped into coherent word-level segments. This aggregation reduces false positives and ensures accurate bounding box placement.
4. **Recognition Pipeline:**
    - **Feature Extraction (ResNet):** A ResNet-based backbone extracts high-level spatial features.
    - **Sequence Modeling (LSTM):** Long Short-Term Memory networks model sequence dependencies, crucial for connected scripts or handwritten text.
    - **CTC Loss:** Connectionist Temporal Classification aligns predicted sequences without explicit character segmentation.

This combination fosters resilient and language-agnostic text recognition.

5. **Greedy Decoding:** The output probability map is converted into text by selecting the most probable characters. This approach balances speed with accuracy, making it suitable for real-time constraints.
6. **Post-Processing:** Common OCR errors are corrected, special characters are normalized, and text formatting is standardized to provide clean input to the translation pipeline.

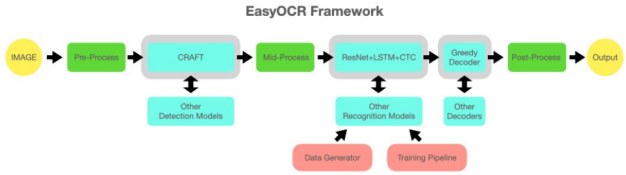

Fig 2. EasyOCR Framework

V. Implementation Details

A. System Components

The backend is implemented using Flask, which manages file processing, frame extraction, and API integration. The frontend leverages JavaScript, HTML, and CSS for a responsive user experience, including live video streaming, file uploads, and language selection.

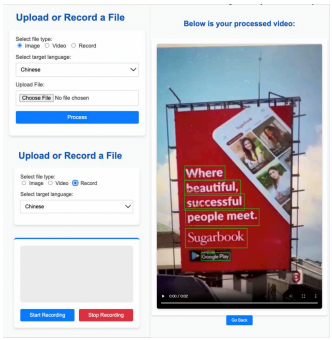

Fig 3. Webapp's Demo

Core Services and Tools:

- **OpenCV (Python):** Handles video capture, frame extraction, and basic image transformations.
- **EasyOCR:** Provides the primary text detection and recognition functionalities.
- **Google Translate API:** Delivers real-time multilingual translations.
- **Python Imaging Library (PIL):** Facilitates overlaying translated text on frames, ensuring correct font rendering for languages like Chinese or Hindi.

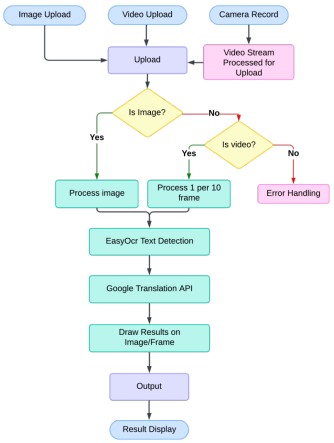

Fig 4. System's Workflow

B. Key Features

- **Dynamic Data Input:** Supports multiple input

modes (live video, images) to accommodate diverse user scenarios.

- **Frame Rate Optimization:** Processes every 10th frame to reduce latency while maintaining acceptable accuracy.
- **Seamless OCR-Translation Integration:** The recognized text is immediately passed to the translation API, ensuring minimal turnaround time.
- **Multilingual Text Rendering:** PIL-based overlays ensure that the final output supports multiple scripts and is visually consistent.

C. Results

Our preliminary tests indicate the following performance metrics:

- **Detection Accuracy:** Approximately 90% for single words and 80% for multi-word phrases in typical scenes.
- **Processing Latency:** On a CPU-only setup, an average of about 2 seconds per processed frame was achieved. Future GPU integration can significantly reduce this latency.
- **Translation Quality:** BLEU scores are around 45 for single words and 30 for longer phrases. Human evaluators deemed the translations understandable with minor contextual shortcomings.

VI. Challenges and Proposed Solutions

Key challenges and corresponding solutions include:

- **Noisy Inputs:** Additional preprocessing (e.g., Gaussian smoothing, adaptive thresholding) can further improve input quality.
- **Frame Skipping:** Reducing processing frequency (e.g., processing every 5th frame instead of every 10th) when GPU resources are available to improve accuracy without sacrificing real-time capability.
- **Integration of Advanced Models:** Future expansions may include handwriting or stylized font recognition models, and transformer-based OCR methods for improved generalization.

VII. Future Work

Future directions for this project include:

1. **Expanded OCR Capabilities:** Incorporating handwriting recognition and domain-specific vocabulary handling will broaden applicability.
2. **Cloud-Based Scalability:** Migrating computationally intensive tasks to a cloud-based platform can handle higher traffic and more complex models.
3. **Enhanced Accessibility:** Integrating text-to-speech functionality and haptic feedback can further assist visually impaired users.

VIII. Related Work

Recent work on scene text detection and recognition spans a wide range of methodologies. EAST [2] and the supervised pyramid context network [3] pioneered efficient and accurate scene text detection. CRAFT [1] improved character-level localization for irregular layouts. End-to-end text spotter systems [4] and attention-based extraction approaches [5] have further advanced the state of the art. For multilingual translation, neural machine translation systems [6] have demonstrated remarkable improvements in both accuracy and speed, laying the groundwork for seamless integration of OCR and translation pipelines.

IX. Conclusion

This paper presents a comprehensive, real-time text detection, recognition, and translation system designed for practical use in real-world environments. By leveraging advanced computer vision and machine translation techniques, it addresses challenges related to low-quality inputs, multilingual scripts, and hardware constraints. The demonstrated system can assist travelers, improve accessibility for the visually impaired, and support a variety of industrial applications. Ongoing and future work aims to enhance accuracy, reduce latency, and improve scalability, thereby pushing the boundaries of real-time text understanding systems.

Acknowledgment

The authors would like to thank the open-source community behind EasyOCR and contributors to Google Translate's API services

References

[1] {CRAFT} Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character Region Awareness for Text Detection," in *Proc. CVPR*, 2019.

[2] {EAST} X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, and W. Shen, "EAST: An Efficient and Accurate Scene Text Detector," in *Proc. CVPR*, 2017.

[3] {Jin} L. Jin, Z. Xie, Y. Zhu, and S. Zhang, "Scene Text Detection with Supervised Pyramid Context Network," in *Proc. AAAI*, 2018.

[4] {TextSpotter} W. He, X. Zhang, F. Wen, and X. Xue, "An End-to-End TextSpotter with Explicit Alignment and Attention," in *Proc. CVPR*, 2018.

[5] {Ma} L. Ma, X. Lu, Z. Li, and A. W. Fu, "Attention-Based Extraction of Structured Information from Street View Imagery," in *Proc. ICCV*, 2015.

[6] {Bahdanau} D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proc. ICLR*, 2015.