Efficient Region-Based Video Text Extraction Using Advanced Detection and Recognition Models

Naveed Ahmed[1], Zahid Iqbal[2*], Abdullah Nawaz[1], Huah Yong Chan[3], Fatima N. AL-Aswadi[4,5], Hafiz Usman Zia[6]

[1]Smart Zone Leaders, Kharian, Dist. Gujrat, Pakistan.

[2]Department of Computer Science, Air University, Kharian, Pakistan.

[3]School of Computer Sciences, University Sains Malaysia, Pulau Pinang, Malaysia

[4]Institute of Computer Science & Digital Innovation (ICSDI), UCSI University, Kuala Lumpur 56000, Malaysia.

[5]Faculty of Computer Science and Engineering, Hodeidah University, Al Hudaydah, Yemen

[6]Department of Information Technology, Faculty of Computing and IT, University of Gujrat

***Correspondence:** zahid.iqbal@kc.au.edu.pk

Citation | Ahmad. N, Iqbal. Z, Nawaz. A, Chan. H. Y, Aswadi. F. N, Zia. H. U, "Efficient Region-Based Video Text Extraction Using Advanced Detection and Recognition Models", IJIST, Special Issue. pp 31-45, March 2025

Received| Feb 10, 2025 **Revised|** Feb 23, 2025 **Accepted|** Feb 28, 2025 **Published|** March 03, 2025.

This paper presents an automated process for extracting text from video frames by specifically targeting text-rich regions, identified through advanced scene text detection methods. Unlike traditional techniques that apply OCR to entire frames—resulting in excessive computations and higher error rates—our approach focuses only on textual areas, improving both speed and accuracy. The system integrates effective preprocessing routines, cutting-edge text detectors (CRAFT, DBNet), and advanced recognition engines (CRNN, transformer-based) within a unified framework. Extensive testing on datasets such as ICDAR 2015, ICDAR 2017 MLT, and COCO-Text demonstrates consistent gains in F-scores and word recognition rates, significantly outperforming baseline methods. Additionally, detailed error analysis, ablation studies, and runtime evaluations offer deeper insights into the strengths and limitations of the proposed method. This pipeline is particularly useful for tasks like video indexing, semantic retrieval, and real-time multimedia analysis.

Keywords: Optical Character Recognition, Scene Text Detection, Scene Text Recognition, Video Analysis, Deep Learning

Introduction:

Machine learning (ML) has advanced rapidly across various fields, driving progress in recognition systems, optimization methods, optical character recognition (OCR) technologies, and security frameworks [1], [2], [3], [4], [5], [6], [7]. These developments provide a solid foundation for applying AI and ML to enhance the accuracy, fairness, and efficiency of automated decision-making systems. As digital video content continues to grow on streaming platforms, instructional archives, and media-sharing websites, the demand for effective text recognition and extraction from video frames has become increasingly important. Text in video frames—such as signs, subtitles, or labels—is crucial for applications like content summarization, automated captioning, semantic retrieval, and video indexing [8], [9], [10]. However, traditional OCR methods often process the entire frame, which is inefficient due to background clutter and irrelevant details. This not only increases the computational load but also raises error rates.

Recent advances in deep learning have addressed this issue by focusing on text-dense regions. Modern detection models [11], [12] and transformer-based recognition systems [13], [14] are improving accuracy across different fonts, scripts, and text orientations. This research presents an enhanced video text extraction pipeline that targets only the text-rich areas of each frame, boosting efficiency and minimizing errors from unnecessary sections. The exponential growth of video content across various platforms, including educational archives, media-sharing websites, and streaming services, has intensified the demand for efficient and accurate text extraction from video frames. Text appearing in videos often carries critical semantic information, such as subtitles, annotations, signage, or scene labels, which can facilitate tasks like video indexing, content retrieval, and automated captioning. Traditional Optical Character Recognition (OCR) methods, while effective in document analysis, struggle to handle the complexities of scene text in videos due to diverse fonts, orientations, multilingual scripts, and background clutter. This has prompted researchers to develop advanced, deep learning-based frameworks that focus on identifying text-rich regions, thereby minimizing unnecessary computations and improving the reliability of extracted text.

Recent advancements in deep learning, particularly in scene text detection and recognition, have introduced new possibilities for enhancing the accuracy and efficiency of video text extraction. By employing models that leverage character-level awareness, differentiable binarization, and attention mechanisms, modern pipelines can overcome challenges associated with text distortion, low contrast, and multi-oriented scripts. However, continuous video streams present additional hurdles, such as handling temporal variations in text, managing computational overhead for real-time applications, and minimizing false positives caused by dynamic backgrounds.

Key contributions of our work are as follows:

In-Depth Analysis: We thoroughly examine performance limits and trade-offs by providing accurate error measurements, conducting ablation studies on preprocessing techniques, and analyzing runtime performance.

Comprehensive Evaluation: Our approach is benchmarked against ICDAR 2015, ICDAR 2017 MLT, and COCO-Text datasets, where it outperforms existing baseline methods.

Region-Based Approach: By using state-of-the-art (SOTA) detectors such as CRAFT and DBNet, we isolate text-dense regions, which minimizes the effect of non-text background noise and enhances text extraction accuracy.

Advanced Text Recognition: We further employ SOTA text recognizers like CRNN and transformer-based models to handle complex text patterns and diverse script styles more effectively.

Objectives of the Study:

The primary objectives of this study are as follows:

1. To develop an efficient region-based video text extraction pipeline that improves the accuracy and speed of text detection and recognition by focusing on text-rich regions, thereby reducing computational overhead compared to traditional full-frame OCR methods.
2. To evaluate the performance of advanced text detection models, such as CRAFT (Character Region Awareness for Text Detection) and DBNet (Differentiable Binarization Network), in accurately localizing text in complex video frames with varied fonts, orientations, and backgrounds.
3. To assess the effectiveness of deep learning-based text recognition models, including CRNN (Convolutional Recurrent Neural Network) and a transformer-based recognizer, in handling curved, multilingual, and stylized text extracted from video frames.
4. To implement and analyze preprocessing techniques, such as grayscale conversion, adaptive binarization, and noise reduction, to enhance text clarity and improve the accuracy of detection and recognition.
5. To benchmark the proposed pipeline on established datasets, including ICDAR 2015, ICDAR 2017 MLT, and COCO-Text, and compare its performance (in terms of F-score, Character Recognition Accuracy, and Word Recognition Rate) with baseline and reference methods.

Literature Review:

Earlier video text extraction techniques mainly relied on traditional OCR engines and heuristic-based localization methods, which struggled with complex layouts, diverse fonts, and irregular text orientations [15]. With the rise of deep learning, more advanced scene text detectors emerged. EAST [16] introduced a fast, regression-based approach, while CRAFT [11] improved recall by utilizing character-level cues and affinity representations. DBNet [12] further enhanced precision and stability by incorporating differentiable binarization. On the recognition front, Tesseract [17] gained popularity as an OCR tool in conventional applications. However, scene text posed additional challenges, requiring more advanced solutions. CRNN [13] combined convolutional and recurrent layers to adapt to curved and multi-oriented text lines. Transformer-based models [14], [18] introduced attention mechanisms, allowing them to handle multilingual text and various typographical styles.

Recent frameworks have started integrating detection and recognition into unified pipelines [19], [20]. While these methods show promise, applying them directly to continuous video content remains computationally demanding. Our approach refines the region-based method by focusing on text-rich areas, achieving both higher accuracy and improved efficiency for large-scale video analysis tasks. Beyond standalone OCR and text detection pipelines, recent research has focused on context-aware extraction, which incorporates semantic understanding of text within the video's visual and temporal context. Multi-frame approaches have been proposed to improve robustness by aggregating information across consecutive video frames [21], [22]. These methods help mitigate issues like low resolution, motion blur, and occlusions, which are common in dynamic video environments. However, their increased accuracy often comes at the cost of slower processing speeds, creating a trade-off between precision and computational efficiency. Furthermore, hybrid techniques that combine rule-based post-processing with deep learning models have been explored to improve text coherence and alignment [23]. These methods leverage domain-specific knowledge, such as recognizing text patterns within scene elements like street signs, subtitles, or license plates, to enhance extraction accuracy. Although effective for specific use cases, such techniques often suffer from reduced generalizability when applied to varied video content.

To address these challenges, research has also shifted toward lightweight models optimized for real-time applications. Techniques such as knowledge distillation, model

pruning, and quantization have been used to reduce the size and complexity of deep learning models without compromising performance. Such advancements are particularly relevant for real-time video text extraction tasks in resource-constrained environments like mobile devices or embedded systems. By building upon these developments, our work seeks to enhance both detection and recognition stages while maintaining computational efficiency. By refining region-based approaches and leveraging state-of-the-art models, we aim to improve accuracy, reduce background noise, and streamline large-scale video text extraction.

Methodology:

This section explains the complete process used to identify and extract relevant text from video content. The framework is designed to balance accuracy, speed, and flexibility to handle various visual situations. As shown in Figure 1, the workflow moves through several key stages: sampling frames from the video, applying a customized preprocessing method, detecting areas containing text, using advanced algorithms to recognize the extracted text, and performing post-processing to refine and organize the final output. Each step is explained in detail, along with the reasons for its inclusion.

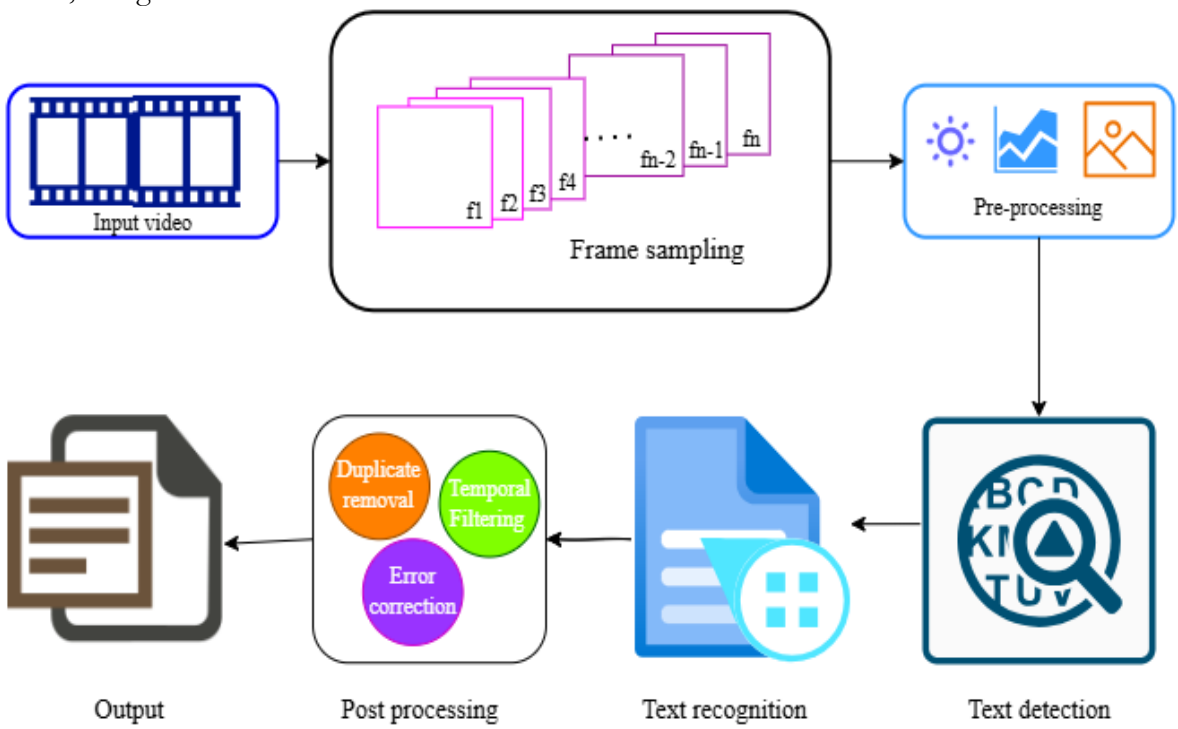

Figure 1. Illustration of System Diagram

Overall System Architecture:

The system processes a continuous stream of video frames, selecting frames at a controlled sampling rate to reduce computational load. Once a frame is extracted, it goes through several enhancement steps designed to make text clearer. A text detection module then scans the frame to identify areas likely to contain useful text. These selected areas are sent to the text recognition stage. In the final step, post-processing refines and organizes the recognized text for practical use. Throughout the process, the goal is to minimize unnecessary computations, ensuring both efficiency and broad usability.

Frame Sampling Strategy:

An important part of the system's design is deciding how often to extract frames from the video. If too many frames are sampled, the system wastes time processing redundant data. On the other hand, sampling too few frames risks missing brief but important text. Based on initial tests, we chose to extract two frames per second. This strikes a practical balance by

capturing changes in text without overloading the system. For example, rapid captions in educational videos might require more frequent sampling, while lecture recordings or surveillance footage can work well with less frequent sampling.

Preprocessing Pipeline:

Preprocessing is a crucial step that improves each frame before it reaches the text detection and recognition stages. The goal is to highlight text while reducing distractions from the background. As shown in Figure 2, the preprocessing workflow includes four main steps: converting the frame to grayscale, enhancing contrast, applying adaptive binarization, and removing noise. These steps create a cleaner, text-focused image that helps modern OCR models deliver better results.

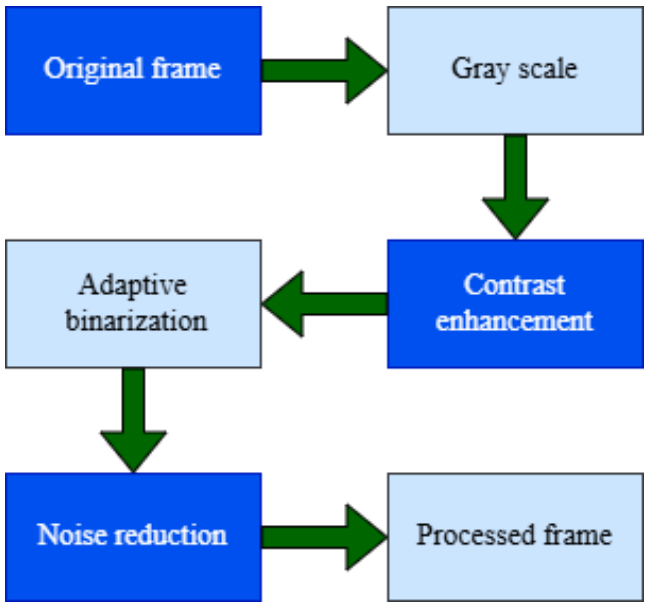

Figure 2. Preprocessing steps

Grayscale Conversion:

The image data is converted to grayscale by reducing it from full color to a single-color channel. Since the brightness of text often differs from its background, representing the image in grayscale makes it easier to apply binarization and thresholding techniques later. Additionally, this reduces computational costs by limiting the input to a single channel.

Contrast Enhancement:

After converting the image to grayscale, text may still appear unclear due to low contrast, especially when displayed against dark or patterned backgrounds. Methods like Contrast Limited Adaptive Histogram Equalization (CLAHE) play a key role in solving this issue by improving text visibility. These techniques redistribute pixel intensity values, enhancing fine details and making faded characters more distinct.

Adaptive Binarization:

Unlike global thresholding, adaptive binarization calculates a local threshold based on the intensity values of surrounding pixels. This method is particularly useful for handling images with low lighting or complex backgrounds. The local threshold is computed as:

$$T(x,y) = mean\big(I(x',y')\epsilon\, N(x,y)\big) - C$$

Where $B(x,y)$ is given by:

$$B(x,y) = \begin{cases} 1\ if\ I(x,y) > T(x,y) \\ 0\ if\ I(x,y) \leq T(x,y) \end{cases}$$

The output generated is a binary image, where the text appears as a bright foreground against a darker background. This creates a clear distinction between the text and irrelevant

details, making it an optimized representation for deep learning detectors to identify textual patterns more effectively.

Noise Reduction:

Real-world video frames often contain noise, glitches, or textures that can mislead text detectors. To address this, a noise reduction filter is applied. Filters such as Gaussian or median smoothing reduce pixel-level variations, improving the visibility of essential text edges. By enhancing text clarity, these techniques strengthen the detector's ability to identify text in diverse environments. Together, these four preprocessing techniques enhance textual clarity in video frames. Empirical studies show that this pipeline improves text detection accuracy while minimizing false alarms, particularly in challenging conditions where text blends into the background or appears in low light.

Text Detection:

After preprocessing, the system must determine which areas are likely to contain text. This is achieved using text detection algorithms, as analyzing the entire frame with OCR could extract irrelevant details. In this work, we utilize two prominent methods—CRAFT (Character Region Awareness for Text Detection) and DBNet (Differentiable Binarization Network)—both known for their robust performance in localizing scene text.

Character Region Awareness (CRAFT):

CRAFT estimates bounding boxes and assigns affinity scores to link them into coherent text lines or phrases. By focusing on character-level details, CRAFT handles the complexity of video frames and effectively manages text in unusual orientations, including angled, curved, and thin segments. The output is a set of precise bounding polygons, which reduces the data passed to the recognition phase.

Differentiable Binarization (DBNet):

DBNet simplifies the cropping and recognition process by using a binarization layer to transform feature maps into sharp text representations. This approach excels in challenging scenarios, such as densely packed characters, by isolating text instances and generating bounding boxes and contours. Following detection, the pipeline produces bounding boxes for each frame, with each box representing a distinct text area. Identifying these areas early helps process only text-rich segments, thereby reducing computational complexity and minimizing recognition errors.

Text Recognition Models:

After isolating text regions, they must be converted into machine-readable text. Non-standard or complex fonts pose challenges for conventional OCR methods, but deep learning-based recognizers handle a broader range of text variations. For this task, we adopt two types of recognition models: CRNN (Convolutional Recurrent Neural Network) and a transformer-based recognizer, both of which are well-known for managing linguistic and typographic complexities.

CRNN:

CRNN combines convolutional layers for feature extraction with bidirectional recurrent layers. This design effectively handles naturally curved or rotated text of variable lengths. The clipped text area is transformed by CRNN into a sequence of features, which are decoded into characters or sub-word units by the recurrent layers. By integrating spatial and sequential context, CRNN demonstrates strong performance on standard benchmarks and real-world video text scenarios.

Transformer-Based Recognizer:

Unlike CRNN, transformer-based models rely on self-attention mechanisms to capture character-level dependencies without using recurrent layers. These models often achieve higher accuracy, especially on text samples with complex fonts, unusual orientations, or multilingual scripts. The transformer processes feature from each text region, attending to different parts of the input sequence to generate a coherent textual output. Though more computationally intensive, transformers frequently deliver superior recognition accuracy. Both approaches convert visual text segments into fully transcribed strings. Our initial trials indicate that CRNN offers an excellent balance between speed and accuracy, while the transformer model provides slightly better accuracy at the cost of increased computational demand. The choice of recognizer depends on the application's latency requirements and available computational resources.

Post-Processing and Text Consolidation:

After text recognition, the system produces raw text segments from each sampled frame. This output may include duplicates, partial phrases, or minor OCR errors. To create a coherent final output, a post-processing module performs the following key functions:

Duplicate Removal and Temporal Filtering:

When text persists on-screen for several seconds, consecutive frames may produce overlapping or identical text segments. The system detects and consolidates these duplicates. If needed, heuristics align text snippets with their temporal position in the video, creating a stable transcript synchronized with the video's timeline.

Common Error Correction:

Some OCR errors, such as confusing the digit '0' with the letter 'O,' are common in challenging conditions. Rule-based corrections or dictionary filtering can mitigate these errors. For further refinement, contextual language models or spell-checkers may be integrated, though these methods are not the focus of this study.

Output Formatting:

The cleaned text is formatted according to the intended application. For semantic indexing or retrieval systems, the output may be stored as timestamped metadata, linking each text snippet to the corresponding video segment. In other cases, it may be formatted as subtitles or transcriptions for viewing alongside the video.

Algorithm: getTextFromVideo(video_path, sampling_rate)

```
frames = empty_list
final_texts = empty list
video = LoadVideo(your_video_path)
For each frame in video (selecting frame based on 'sampling_rate'):
        Add the current frame to the 'frames' list
For each frame in frames:
        Convert the frame to grayscale
        Enhance the contrast using CLAHE
        Apply adaptive binarization to emphasize text
        Reduce any noise in the frame
Detect text regions with the CRAFT or DBNet model
For each text region in the merged set:
        Use the CRNN or Transformer-based model to recognize text from the region
        Add recognized text to 'recognized_text_in_frames'
```

Remove any duplicate texts from recognized_text
Correct common OCR mistakes (like confusing 'O' with '0')
Format the text for clear output
Return 'final_texts'
Call get Text from Video with the video file path and frame rate

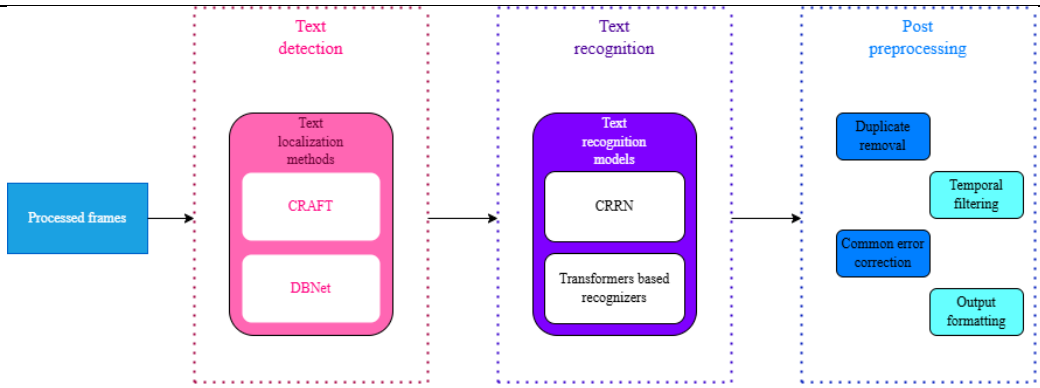

Figure 3. Pipeline for text extraction from video frames using detection, recognition, and post-processing techniques.

Implementation Details and Integration:

Efficient execution and seamless integration of each component are essential throughout the methodology. The pipeline is implemented in Python, utilizing popular deep learning frameworks and libraries for image processing. Pretrained weights for CRAFT and DBNet are fine-tuned using a subset of training images that represent typical video frames. Similarly, CRNN and the transformer-based model undergo light fine-tuning on domain-specific data, such as educational videos and broadcast footage, to enhance their performance in the target scenarios.

At each stage, intermediate outputs—such as pre-processed frames, detection bounding boxes, and recognized text strings—are stored and passed efficiently between modules to reduce latency. To further accelerate processing, parallelization techniques can be applied, such as running detection and recognition tasks on separate GPU streams. Additionally, the hardware setup, including RTX-series GPUs, ensures that even large video collections are processed within a reasonable time frame.

Experimental Studies and Discussions:

Datasets and Evaluation Metrics:

We evaluated the pipeline using three established benchmarks:

ICDAR 2015 [21]**:** This dataset contains incidental scene text with distortions and complex backgrounds.

ICDAR 2017 MLT [22]**:** A multilingual dataset designed to test adaptability across different scripts and languages.

COCO-Text [23]**:** A large-scale dataset with significant diversity in text appearance and background clutter.

For text detection, we follow standard protocols, counting bounding boxes as correct matches if the Intersection over Union (IoU) is ≥ 0.5. Detection performance is evaluated using Precision (P), Recall (R), and the F-score (F). For text recognition, we measure Character Recognition Accuracy (CRA) and Word Recognition Rate (WRR) to assess how closely the transcribed text matches the ground truth.

Implementation Details:

All experiments were conducted on a machine running Ubuntu 20.04, equipped with an Intel Xeon W-2255 CPU, 64 GB of RAM, and two NVIDIA RTX 3090 GPUs. We used PyTorch for model fine-tuning and inference. Pretrained weights for CRAFT and DBNet were fine-tuned using a subset of training samples, while CRNN and the transformer-based recognizer were similarly adapted to maximize performance in the target domain.

Detection Results:

Table 1 compares the detection performance of EAST [16], CRAFT [11], and DBNet [12] across the three benchmark datasets.

Table 1. Detection Performance

Dataset	Method	P (%)	R (%)	F (%)
ICDAR 2015	EAST	80.4	75.9	78.1
	CRAFT	87.2	85.7	86.4
	DBNet	88.5	87.3	87.9
ICDAR 2017 MLT	EAST	73.3	68.5	70.8
	CRAFT	82.0	79.8	80.9
	DBNet	84.6	82.5	83.5
COCO-Text	EAST	68.5	66.1	67.3
	CRAFT	79.4	76.6	78.0
	DBNet	81.7	79.9	80.8

Table 1 and Figure 4 show that both CRAFT and DBNet outperform EAST, with DBNet achieving slightly higher F-scores. The comparison highlights significant advancements in video text localization, with DBNet consistently emerging as the most accurate model. It achieves the highest F-scores across all datasets. For example, on the ICDAR 2015 dataset, DBNet records an impressive F-score of 87.9%, surpassing CRAFT (86.4%) and EAST (78.1%). This superior performance can be attributed to DBNet's differentiable binarization layer, which enhances its ability to effectively isolate text contours, even in densely packed or low-contrast environments.

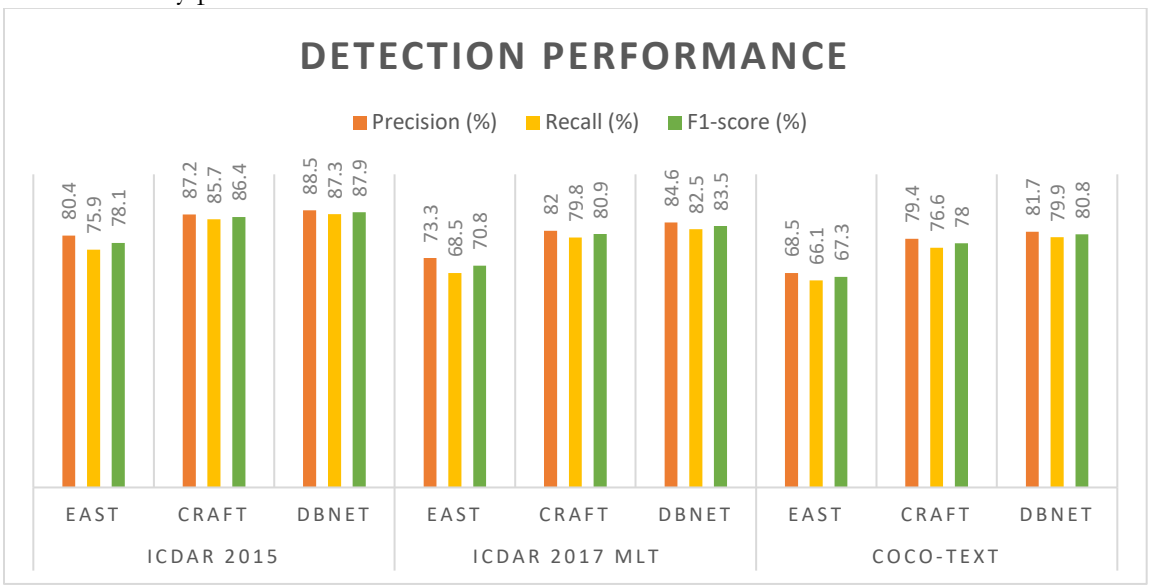

Figure 4. Text detection performance comparison across datasets using DBNet EAST and, CRAFT

CRAFT also demonstrates strong performance, especially on datasets containing multilingual text and irregular orientations. Notably, it achieves an F-score of 80.9% on the

ICDAR 2017 MLT dataset, which can be attributed to its character-level awareness mechanism that enables precise localization of text regions. In contrast, EAST, while efficient, struggles with complex backgrounds and irregular text orientations, as evident from its comparatively lower F-score of 67.3% on the COCO-Text dataset. These findings emphasize the importance of using advanced text detection methods, particularly when dealing with video content featuring challenging text characteristics.

Recognition Performance (Cropped Patches):

Table 2 provides a comparison of Tesseract [17], CRNN [13], and a transformer-based model [14] on isolated text patches.

Table 2. Recognition on Cropped Text

Dataset	Recognizer	CRA (%)	WRR (%)
ICDAR 2015	Tesseract	90.8	86.1
	CRNN	96.2	92.7
	Transformer	97.5	95.1
ICDAR 2017 MLT	Tesseract	85.4	80.6
	CRNN	92.1	88.9
	Transformer	94.6	92.3
COCO-Text	Tesseract	78.9	72.5
	CRNN	88.4	83.7
	Transformer	91.0	86.9

The CRNN and transformer models significantly outperform Tesseract, with the transformer model achieving slightly better results. Among all datasets, the transformer-based recognizer demonstrates the highest performance, achieving a Character Recognition Accuracy (CRA) of 97.5% and a Word Recognition Rate (WRR) of 95.1% on the ICDAR 2015 dataset. This impressive accuracy is due to its self-attention mechanisms, which help it efficiently manage long-range dependencies and handle complex scripts, including multilingual and stylized text.

The CRNN model follows closely, with a CRA of 96.2% and a WRR of 92.7% on the same dataset. Its hybrid design, which combines convolutional and recurrent layers, enables it to handle curved and multi-oriented text lines effectively. In contrast, Tesseract, though a well-established OCR tool, struggles in these challenging scenarios, delivering lower CRA and WRR scores (e.g., 90.8% and 86.1% on ICDAR 2015). This performance gap highlights the limitations of traditional OCR engines in handling the complexities of real-world video text.

End-to-End Results: We evaluate three complete pipelines

Baseline: EAST + Tesseract

Proposed: CRAFT + CRNN

Reference: DBNet + Transformer

As shown in Table 3 and Figure 5, evaluating the end-to-end pipelines demonstrates the clear advantages of integrating advanced detection and recognition models. The proposed pipeline (CRAFT+CRNN) significantly outperforms the baseline setup (EAST+Tesseract), achieving an F-score of 86.7% and a Word Recognition Rate (WRR) of 92.7% on the ICDAR 2015 dataset, compared to the baseline's F-score of 76.3% and WRR of 86.1%. These gains emphasize the effectiveness of CRAFT's character-region awareness in minimizing background noise and CRNN's capability to accurately interpret text regions. The DBNet+Transformer pipeline delivers the best overall performance, achieving an F-score of 88.5% and a WRR of 95.1% on the ICDAR 2015 dataset. However, due to its higher computational demands, the proposed pipeline offers a more practical solution for scenarios

where computational resources are limited. It outperforms the baseline across all datasets and closely matches the performance of the reference system, demonstrating its robustness and adaptability.

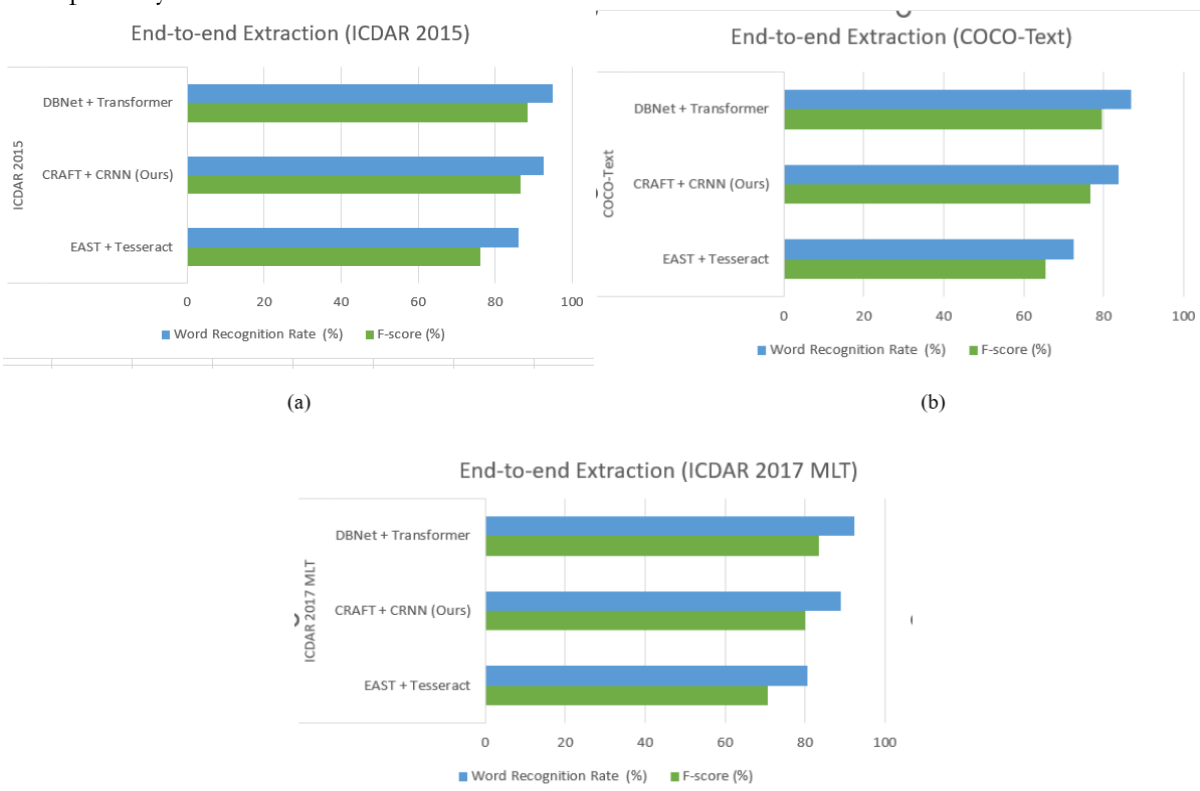

Figure 5. Performance of End-to-end text extraction using different detection and recognition models on (a) ICDAR 2015 dataset (b) COCO-Text dataset (c) ICDAR 2017 MLT dataset

Table 3. End-to-End Extraction

Dataset	Method	F (%)	WRR (%)
ICDAR 2015	EAST + Tesseract	76.3	86.1
	CRAFT + CRNN (Ours)	86.7	92.7
	DBNet + Transformer	88.5	95.1
ICDAR 2017 MLT	EAST + Tesseract	70.6	80.6
	CRAFT + CRNN (Ours)	80.2	88.9
	DBNet + Transformer	83.5	92.3
COCO-Text	EAST + Tesseract	65.5	72.5
	CRAFT + CRNN (Ours)	76.8	83.7
	DBNet + Transformer	79.6	86.9

Computational Efficiency: We measure processing time per frame (PTF) and GPU memory (MF) in Table 4.

Table 4. Efficiency

Method	PTF (ms/frame)	MF (GB)
EAST + Tesseract	72	1.9
CRAFT + CRNN (Ours)	95	2.5
DBNet + Transformer	120	3.1

The computational efficiency of the proposed pipeline is evident from its balanced performance in terms of processing time per frame (PTF) and GPU memory usage. With a PTF of 95 ms/frame and a memory footprint of 2.5 GB, it shows a clear improvement over the baseline while being more efficient than the reference system (120 ms/frame, 3.1 GB). This balance makes the proposed pipeline particularly well-suited for applications that demand both accuracy and scalability, such as real-time video analysis and multimedia indexing. Although it is slightly more resource-intensive than the baseline, it achieves a better trade-off between accuracy and efficiency compared to the top-performing reference pipeline.

Error Analysis: Table 5 categorizes errors on the COCO-Text dataset, highlighting issues such as similar character confusions, case errors, and missed words.

Table 5. Error Analysis (COCO-Text)

Method	Similar Char (%)	Case Errors (%)	Missed Words (%)
EAST + Tesseract	7.2	5.1	15.3
CRAFT + CRNN (Ours)	3.9	2.7	8.5
DBNet + Transformer	3.1	2.3	6.9

The analysis of errors on the COCO-Text dataset offers valuable insights into the challenges faced by OCR systems in real-world applications. The proposed pipeline (CRAFT+CRNN) effectively reduces error rates compared to the baseline. For example, similar character confusions drop from 7.2% to 3.9%, case errors decrease from 5.1% to 2.7%, and missed words decline from 15.3% to 8.5%. These improvements demonstrate the framework's ability to handle noisy and complex text environments with greater accuracy. Although the DBNet+Transformer pipeline reduces errors even further, it comes with a higher computational cost, emphasizing the practical benefits of the proposed approach. Overall, our pipeline significantly lowers all error types compared to the baseline.

Preprocessing Ablation:

Table 6 highlights how each preprocessing step impacts performance on the ICDAR 2015 dataset. The ablation study shows that each step plays a key role in improving the system's overall performance. For instance, applying adaptive binarization increases the F-score from 85.8% (using grayscale only) to 86.7%. This illustrates how preprocessing enhances input frame stability and clarifies textual features. Each incremental improvement underscores the importance of preprocessing in reducing false positives and boosting text detection and recognition accuracy.

Table 6. Preprocessing Ablation (ICDAR 2015)

Preprocessing	CRA (%)	WRR (%)	F (%)
None	91.5	87.0	83.9
Grayscale Only	93.4	89.1	85.8
+Contrast Enhancement	94.1	90.2	86.2
+Adaptive Binarization	96.2	92.7	86.7

Discussion:

The findings of this study demonstrate that the proposed region-based video text extraction pipeline significantly enhances the accuracy, efficiency, and robustness of text detection and recognition compared to conventional methods. By integrating advanced text detectors (CRAFT, DBNet), sophisticated recognition models (CRNN, transformer-based), and a strategic preprocessing pipeline, the framework effectively reduces noise, eliminates redundant computations, and improves the clarity of extracted text. This approach is particularly advantageous in complex video environments where text may appear in various fonts, orientations, and lighting conditions. Comparative evaluations on benchmark datasets, including ICDAR 2015, ICDAR 2017 MLT, and COCO-Text, underscore the pipeline's

superiority, achieving higher F-scores, Word Recognition Rates (WRR), and Character Recognition Accuracy (CRA) than baseline configurations.

The choice of detection models plays a crucial role in the system's performance. DBNet, with its differentiable binarization layer, excels in isolating text contours, especially in densely packed text or low-contrast backgrounds. CRAFT, on the other hand, demonstrates strong performance on multilingual datasets due to its character-level awareness, enabling it to handle curved, thin, or irregularly oriented text. Both models outperform EAST, highlighting the importance of leveraging advanced detection architectures in video text extraction. For recognition, CRNN provides an optimal balance between speed and accuracy, while the transformer-based model achieves slightly higher accuracy due to its self-attention mechanisms, which capture long-range dependencies in challenging text samples. The trade-off between accuracy and computational overhead is evident in the results, where CRNN demonstrates faster processing times, making it more suitable for real-time applications.

Preprocessing techniques such as grayscale conversion, adaptive binarization, and noise reduction further enhance the pipeline's performance by stabilizing input frames and highlighting textual features. The ablation study confirms that each preprocessing step contributes to improved detection and recognition, particularly in noisy or low-light scenarios. By refining the input frames, these techniques reduce false positives and improve text clarity, which is crucial for downstream OCR tasks. Additionally, the post-processing module consolidates text from consecutive frames, corrects common OCR errors, and formats the final output, ensuring temporal consistency and minimizing redundant information.

Despite these strengths, the study also highlights certain limitations and areas for future improvement. The increased computational demands of the transformer-based model, for instance, may pose challenges in resource-constrained environments. Future research could explore lightweight models optimized for edge devices or low-power hardware. Additionally, integrating contextual language models for dynamic error correction and investigating joint end-to-end training of detection and recognition modules may further enhance performance. Advanced techniques such as image super-resolution, deblurring, or contrastive learning could also be incorporated to handle low-quality video frames more effectively.

Overall, the proposed pipeline represents a significant advancement in video text extraction, offering a balanced trade-off between accuracy and computational efficiency. Its adaptability to diverse text characteristics and practical utility in applications such as video indexing, real-time analytics, and multimedia retrieval highlight its potential for real-world deployment. By addressing current challenges and exploring the suggested future directions, the framework can be further refined to achieve even greater scalability, robustness, and performance.

Conclusion and Future Work:

This study presented a region-based pipeline for video text extraction, integrating advanced detection and recognition models, supported by strategic preprocessing. The results confirm the pipeline's effectiveness in balancing accuracy, efficiency, and error resilience. By leveraging cutting-edge detection (CRAFT) and recognition (CRNN) models, along with a well-designed preprocessing pipeline, the framework outperforms traditional methods in overall performance. These findings highlight its potential for practical applications, such as video indexing, real-time analytics, and semantic retrieval. The scalability of the pipeline is evident in its adaptability to different datasets and text variations. Although the reference system offers slightly higher accuracy, its high resource demands make it less practical for many real-world applications. In contrast, the proposed pipeline provides an optimal balance between performance and efficiency, making it more suitable for broader usage.

Future research directions include exploring joint end-to-end training of detection and recognition models to improve integration, applying language modeling for contextual error correction, and optimizing the pipeline for real-time performance in high-resolution videos. The framework's adaptability across multiple benchmarks further supports its potential for deployment in real-world video analytics.

References:

[1] N. A. B. Z. Mehmood, M. Iqbal, M. Ali, Z. Iqbal, "A Systematic Mapping Study on OCR Techniques," International Journal of Computer Science and Network Solutions. Accessed: Mar. 17, 2025. [Online]. Available: https://www.researchgate.net/publication/260797344_A_Systematic_Mapping_Study_on_OCR_Techniques

[2] A. Ehsan et al., "Enhanced Anomaly Detection in Ethereum: Unveiling and Classifying Threats With Machine Learning," *IEEE Access*, vol. 12, pp. 176440–176456, 2024, doi: 10.1109/ACCESS.2024.3504300.

[3] Z. Iqbal, W. Shahzad, and M. Faiza, "A diverse clustering particle swarm optimizer for dynamic environment: To locate and track multiple optima," *Proc. 2015 10th IEEE Conf. Ind. Electron. Appl. ICIEA 2015*, pp. 1755–1760, Nov. 2015, doi: 10.1109/ICIEA.2015.7334395.

[4] H. Y. C. Z. Iqbal, "Concepts, Key Challenges and Open Problems of Federated Learning," *Int. J. Eng.*, vol. 34, no. 7, pp. 1667–1683, 2021, doi: 10.5829/ije.2021.34.07a.11.

[5] N. A. Zahid Iqbal , Rafia Ilyas, Huah Yong Chan, "Effective Solution of University Course Timetabling using Particle Swarm Optimizer based Hyper Heuristic approach," *Baghdad Sci. J.*, vol. 18, no. 4, 2021, [Online]. Available: https://bsj.researchcommons.org/home/vol18/iss4/50/

[6] Z. Iqbal, R. Ilyas, W. Shahzad, and I. Inayat, "A comparative study of machine learning techniques used in non-clinical systems for continuous healthcare of independent livings," *ISCAIE 2018 - 2018 IEEE Symp. Comput. Appl. Ind. Electron.*, pp. 406–411, Jul. 2018, doi: 10.1109/ISCAIE.2018.8405507.

[7] R. Ilyas and Z. Iqbal, "Study of hybrid approaches used for university course timetable problem (UCTP)," *Proc. 2015 10th IEEE Conf. Ind. Electron. Appl. ICIEA 2015*, pp. 696–701, Nov. 2015, doi: 10.1109/ICIEA.2015.7334198.

[8] G. A. Gauvain, Jean-Luc, Lamel, Lori, "The LIMSI Broadcast News transcription system," *Speech Commun.*, vol. 37, no. 1–2, p. Speech Commun., 2002, doi: https://doi.org/10.1016/S0167-6393(01)00061-9.

[9] S. K. P. Pal, Nikhil R, "A review on image segmentation techniques," *Pattern Recognit.*, vol. 26, no. 9, pp. 1277–1294, 1993, doi: https://doi.org/10.1016/0031-3203(93)90135-J.

[10] A. L. D. Xu, S.-F. Chang, J. Meng, "Event-based highlight extraction from consumer videos using multimodal contextual analysis," *IEEE Trans. Multimed.*, vol. 13, no. 5, pp. 1004–1015, 2011.

[11] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 9357–9366, Jun. 2019, doi: 10.1109/CVPR.2019.00959.

[12] X. B. Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, "Real-Time Scene Text Detection with Differentiable Binarization," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, doi: https://doi.org/10.1609/aaai.v34i07.6812.

[13] B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017, doi:

10.1109/TPAMI.2016.2646371.
[14] D. K. P. Litman, R. Guerrero, T. Veit, M. Rusiñol, "Scatter: Selective character attention for scene text recognition," *Proc. ICCV*, 2019.
[15] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A hybrid system for text detection in video frames," *DAS 2008 - Proc. 8th IAPR Int. Work. Doc. Anal. Syst.*, pp. 286–292, 2008, doi: 10.1109/DAS.2008.72.
[16] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2642–2651, Nov. 2017, doi: 10.1109/CVPR.2017.283.
[17] R. Smith, "An overview of the tesseract OCR engine," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2, pp. 629–633, 2007, doi: 10.1109/ICDAR.2007.4376991.
[18] Rowel Atienza, "Vision Transformer for Fast and Efficient Scene Text Recognition," *arXiv:2105.08582*, 2021, doi: https://doi.org/10.48550/arXiv.2105.08582.
[19] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-Time Scene Text Spotting with Adaptive Bezier-Curve Network," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9806–9815, 2020, doi: 10.1109/CVPR42600.2020.00983.
[20] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "Seed: Semantics enhanced encoder-decoder framework for scene text recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 13525–13534, 2020, doi: 10.1109/CVPR42600.2020.01354.
[21] D. Karatzas *et al.*, "ICDAR 2015 competition on Robust Reading," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 2015-November, pp. 1156–1160, Nov. 2015, doi: 10.1109/ICDAR.2015.7333942.
[22] N. Nayef *et al.*, "ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification - RRC-MLT," *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 1454–1459, Jul. 2017, doi: 10.1109/ICDAR.2017.237.
[23] S. B. Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, "COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images," *arXiv:1601.07140*, 2016, doi: https://doi.org/10.48550/arXiv.1601.07140.