Real-Time Video Text Tracking Using Deep Learning and IoT Technology

MR Senthil Kumar V
Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore, India

S. Myvizhi
Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore, India

M. Kavi Priya
Department of Information Science and Engineering
Kumaraguru College of Technology
Coimbatore, India

B. Prakasam
Department of Information Science and Engineering
Kumaraguru College of Technology
Coimbatore, India

V. V. Vasanth
Department of Information Science and Engineering
Kumaraguru College of Technology
Coimbatore, India

T. Kanishka
Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore, India

***Abstract*— This paper presents a real-time video text tracking system integrating deep learning with IoT-enabled edge processing for efficient text detection, tracking, and recognition. Using the CRAFT text detector, DeepSORT for tracking, and Tesseract OCR for recognition, the system ensures low-latency performance with AWS EC2 and NVIDIA A100 GPU acceleration, while IoT devices enable edge computing to reduce bandwidth usage. Docker containerization ensures consistency, PostgreSQL manages structured text data, and a FastAPI-based API enables real-time text retrieval. Evaluations on ICDAR and TextVQA datasets demonstrate resilience to motion blur, occlusions, and varying text orientations, making it suitable for applications like automated transcription, surveillance, and smart retail. Future work will focus on enhancing adaptive recognition, optimizing IoT-edge processing, and improving scalability.**

Keywords*— *Video text tracking, real-time text recognition, optical character recognition (OCR), text detection, machine learning, vehicle identification, CCTV analysis.

I. INTRODUCTION

Text extraction and tracking from video streams are essential for applications like traffic monitoring, law enforcement, and forensic investigations. Unlike traditional OCR, video-based text recognition faces challenges such as motion blur, occlusions, lighting variations, and multi-language text. Existing methods often process video frames individually, missing crucial temporal dependencies, leading to lower accuracy and inefficiency in real-time applications.

It suggests an IoT-enabled video text tracking system that incorporates deep learning for text detection, tracking, and recognition in order to overcome these difficulties. It uses Tesseract OCR for accurate recognition, DeepSORT for reliable tracking across frames, and the CRAFT model for correct text localization. Furthermore, by carrying out pre-processing at the edge and lowering latency and bandwidth consumption, IoT devices like smart security cameras and edge processors increase efficiency. Through the integration of IoT-based edge computing with spatial and temporal data, the system enhances real-time text extraction from dynamic video streams, including CCTV footage. This all-encompassing strategy guarantees excellent accuracy and scalability, which makes it appropriate for applications needing intelligent traffic management, real-time automatic transcribing, and surveillance.

II. RELATED WORK

Several studies have looked into OCR methods based on deep learning for video text recognition. Conventional techniques frequently ignore the temporal continuity between frames in favor of frame-by-frame recognition. Advanced methods, on the other hand, use models like EAST, CRAFT, and CRNN for more effective text recognition and detection. Multimodal text interpretation has been greatly improved by recent developments, such as transformer-based designs like TrOCR and Swin-Transformer. By combining deep learning models with temporal tracking techniques, this work expands on earlier studies and enhances the precision and effectiveness of word recognition across video frames.

III. LITERATURE SURVEY

[1] Gao, Yuzh, and colleagues introduced the Spatio-Temporal Complementary Model (STCM), which implemented temporal continuity in a sequence of frames, thereby reinforcing video text tracking.The Siamese Complementary Module (SCM) successfully fixes the issues of losing detections and enhances the dependability of text recognition for dynamic video footage. However, because feature extraction is identical, the network's advantages are outweighed by the lengthy inference time.

[2] Liu and Hongen used the SORT algorithm and the PP-YOLOE-R network to tackle the problem of tracking dense and small text in video sequences. Performance and accuracy are improved by the framework's separation of tracking and detection operations. The SORT algorithm's tracking accuracy has to be improved, even if the PP-YOLOE-R model is

excellent at recognizing small text. As a result, this approach serves as a foundation for future advancements in video text tracking systems.

[3] Rasikannan, L., et al. used CRNN and OCR to study deep learning-based text extraction from video material. Their method is appropriate for applications such as automated surveillance since it provides excellent accuracy and resilience in dynamic conditions. However, scalability and performance in real-time settings are limited by the need on high-quality video inputs and substantial processing resources.

[4] Wang and Liang presented a method for end-to-end video text identification and tracking that combines explainable descriptions with online tracking for better interpretability. Workflows are made simpler and computational expenses are decreased by this uniform methodology. Handling low-quality films and guaranteeing generalizability across various datasets continue to be difficult tasks, requiring additional improvements for wider use.

[5] Renakatamboli emphasized multilingual support while concentrating on text extraction from video frames using OpenCV and OCR. Despite the system's excellent accuracy and real-time capabilities, its drawbacks—such as its dependence on cloud resources and decreased precision for less widely used languages—highlight areas that require refinement to guarantee resilience and effectiveness in a variety of contexts.

[6] To facilitate the extraction of textual data from videos, Wadaskar, Ghanshyam, and associates created a Python-based framework for YouTube transcript extraction. OCR is effective for its intended platform, but its application outside of YouTube is limited because to its lack of support for non-subtitled content. In order to increase functionality, the study stresses the significance of incorporating text recognition techniques.

IV. PROPOSED METHODOLOGY

A. *Text Detection*

The Text Detection module is responsible for identifying and localizing text regions within each video frame. This is achieved by applying deep learning models, specifically Convolutional Neural Networks (CNNs), and more advanced architectures such as Region-based CNNs (R-CNNs) or You Only Look Once (YOLO) models, which are designed for object detection tasks. These models are trained on large datasets that encompass various text types, fonts, and orientations, allowing the model to generalize effectively across different environments. The module addresses challenges such as varying text sizes, low-resolution frames, and complex backgrounds by leveraging pre-processing techniques and multi-scale detection. The detected regions are marked for tracking in subsequent frames. A key consideration is the module's ability to process video data in real-time, ensuring that detection does not become a bottleneck in high-throughput applications.

B. *Text Tracking*

The Text Detection module is in charge of locating and identifying text regions in every video frame. It does this by using deep learning models, specifically Convolutional Neural Networks (CNNs), and more sophisticated architectures, such as Region-based CNNs (R-CNNs) or You Only Look Once (YOLO) models, which are made for object detection tasks. These models are trained on large datasets that encompass a variety of text types, fonts, and orientations, which enable the model to generalize successfully across different environments. The module handles issues like different text sizes, low-resolution frames, and complex backgrounds by utilizing pre-processing techniques and multi-scale detection. The detected regions are marked for tracking in subsequent frames. One important factor is the module's capacity to process video data in real-time, which ensures that detection does not become a bottleneck in high throughput applications

C. *Text Recognition*

The Text Recognition module uses Optical Character Recognition (OCR) on the discovered text regions to retrieve useful textual information after text has been spotted and tracked across frames. To identify sequential data, like text in films, the OCR system makes use of cutting-edge methods like Connectionist Temporal Classification (CTC) in conjunction with Long Short-Term Memory (LSTM) networks. The module can be used in a variety of real-world situations because it can recognize handwritten or cursive text in addition to printed text. The system can also handle a variety of text inputs thanks to integrated multi-language and multi-font capabilities. Spell-checking and context-based corrections are examples of post-processing techniques used to improve the recognition accuracy of text, especially in noisy or degraded video sources. The identified text is output by the module as a structured data stream for further analysis.

D. *Preprocessing Unit*

A crucial part intended to maximize video frame quality prior to text detection and tracking is the preprocessing unit. In order to reduce common problems found in real-world video footage, such as motion blur, low light levels, and noise, this module employs a number of picture enhancing techniques. To increase frame clarity and make text areas easier to discern from the backdrop, techniques such adaptive contrast adjustment, histogram equalization, and Gaussian smoothing are employed. To stabilize frames, motion blur correction methods such as deconvolution or blind deblurring are used, particularly in high-speed video situations where rapidly moving objects significantly blur the image. By adjusting for different lighting conditions, lighting normalization makes sure that text is viewable throughout the entire video.

E. *User Interface*

A user-friendly platform for real-time system interaction is offered by the User Interface (UI) module. By overlaying text detection and identification results on video frames, the interface allows users to track the precision and development of the text extraction process. Errors like incorrectly detected text or inaccurate recognition can be manually fixed through the user interface (UI), which also offers facilities for accessing comprehensive logs or data on text detection and recognition performance. The interface facilitates real-time engagement and downstream processing and analysis by allowing users to export recognized text data in multiple formats, including plain text, CSV, or JSON. The user interface has tools for exporting license plate information or event timestamps for use in traffic monitoring or law enforcement applications, guaranteeing smooth integration into larger processes. Because of its user-friendly design, even non-technical persons can utilize it.

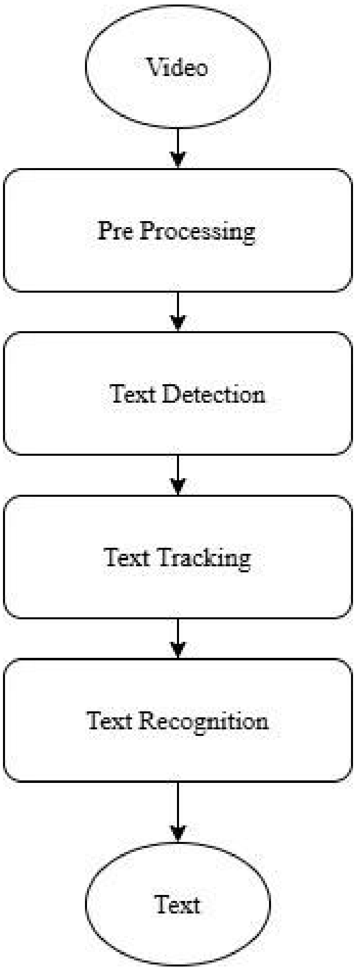

Fig. 1. System flow

V. SYSTEM ARCHITECTURE

The system architecture is made to effectively extract text from video feeds in real time. Motion blur, occlusions, and inconsistent video quality are among the issues it addresses by combining deep learning models, sophisticated tracking algorithms, and text recognition methods. The modular architecture guarantees scalability, dependability, and the capacity to manage large video streams in a variety of settings. A thorough description of the components and operation of the system may be found below.

A. Working

The technology ensures accurate and dependable text extraction by adhering to a standardized workflow. After breaking down the raw video into individual frames, pre-processing techniques including noise reduction and image improvement are applied. Character Region Awareness for Text (CRAFT) is a deep learning-based text identification algorithm that filters out non-text elements while identifying possible text regions. The system can track identified text across frames even when there are occlusions and movement since the DeepSORT tracking algorithm is used to ensure accuracy and continuity.

Text recognition is carried out using Tesseract OCR, which extracts text from the detected regions after the text regions have been located and tracked. To exclude forecasts that are not quite certain, a confidence-scoring system is used. By using feature extraction and sequence modeling, a machine learning pipeline improves recognition accuracy even more. Lastly, to improve readability and accuracy, post-processing methods like confidence-based correction and temporal smoothing are used to aggregate identified text. A PostgreSQL database contains the extracted text for quick access and organized analysis. Using cutting-edge methods like motion blur handling and temporal smoothing, the system is made to work efficiently even with blurry video footage, guaranteeing dependable text extraction.

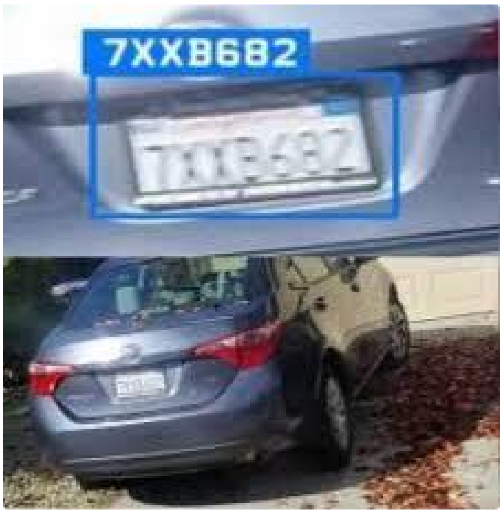

Fig. 2. Blur video images

B. System Components

The system integrates a number of cutting-edge technologies and techniques to attain peak performance:

- Hosted on a scalable AWS EC2 instance with auto-scaling features for dynamic workload control, this deployment method is cloud-based.
- GPU Acceleration: To speed up deep learning processing and inference, an NVIDIA A100 Tensor Core GPU is used.
- Containerization: Docker improves consistency and portability by ensuring smooth deployment across many environments.
- Database Integration: For effective querying and retrieval, extracted text and metadata are stored in PostgreSQL.
- API-based Inference: Real-time text monitoring and recognition are made possible via a RESTful API constructed with FastAPI, which also makes it simple to integrate with other apps.

C. Deployment Architecture

The deployment architecture is made in this way to provide scalability, security, and ongoing monitoring:

- Cloud Infrastructure: AWS CloudWatch is used to track system performance in real time, and the system is hosted on AWS.

- Scalability: To ensure that the system can manage large volumes of video streams, Elastic Load Balancing (ELB) is used to effectively manage workload and dynamically disperse incoming requests.

- Security Measures: To protect sensitive video data and extracted text information, VPC-based network isolation, IAM controls, and data encryption are implemented to secure resource access and stop unwanted intrusions.

- CI/CD Pipeline: GitHub Actions automates version control, testing, and deployment, guaranteeing smooth updates and minimising system downtime. In order to successfully address deployment errors, automated rollback methods are also in place.

D. Implementation

The system is built using industry-leading frameworks and tools to optimize accuracy and efficiency:

- Deep Learning Models:
  - CRAFT for text detection
  - DeepSORT for text tracking across frames
  - Tesseract OCR for text recognition
- Frameworks: TensorFlow, OpenCV, and PyTorch are used for deep learning model implementation and image processing.
- Deployment Setup: The entire system is deployed on an AWS EC2 instance with GPU acceleration and containerized using Docker to ensure reproducibility and efficient execution.
- Database Optimization: Indexing techniques and partitioning strategies are implemented in PostgreSQL to handle large-scale text data efficiently.
- API Performance Enhancements: The FastAPI service is optimized using Uvicorn and Gunicorn for high-performance asynchronous processing, reducing latency in real-time text extraction applications.

VI. RESULTS

The system was evaluated on benchmark datasets, including ICDAR and YouTube video samples. Key results includes

Table I. Performance Evaluation on Benchmark Datasets.

METRIC	DATASET	RESULT
Detection Accuracy	ICDAR, YouTube	F1-score of 0.71
Tracking Efficiency	ICDAR, YouTube	78% accuracy in maintaining text identity across frames
Recognition Performance	ICDAR, YouTube	Word Error Rate (WER) of 9.2%
Processing Speed	NVIDIA A100 GPU	30 FPS (Real-time)

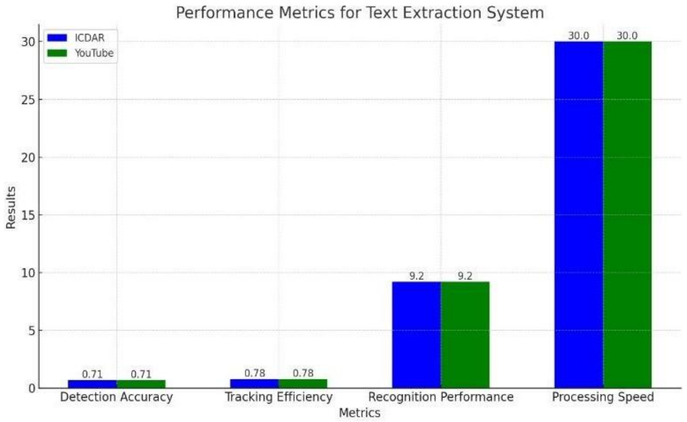

Fig. 3. Text Extraction System

Challenges include handling low-resolution text, motion blur, and occlusions. Further refinements in error correction and adaptive processing can enhance accuracy.

VII. CONCLUSION

The video text tracking system provides a robust solution for real-time text extraction and recognition in dynamic video environments. By integrating deep learning models for text detection, DeepSORT for tracking, and

Incorporating IoT enhances the system's efficiency by leveraging smart surveillance cameras and edge devices for on-device processing, reducing bandwidth usage and improving response times. IoT-enabled sensors and real-time streaming technologies allow for intelligent data filtering, enabling only relevant text information to be transmitted to cloud servers for further analysis. This approach ensures scalability and adaptability, making the system suitable for applications in surveillance, law enforcement, automated transcription, smart city infrastructure, and intelligent traffic monitoring.

References

[1] Yin Xu-Cheng, Ze-Yu Zuo, Shu Tian, and Cheng-Lin Liu.Text detection, tracking and recognition in video: a comprehensive survey.IEEE Transactions on Image Processing, 25(6):2752–2773, 2016.

[2] Karatzas Dimosthenis, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, and Jiri Matas et al.ICDAR 2015 competition on robust reading.In 2015 13th international conference on document analysis and recognition (ICDAR), pages 1156–1160, 2015.

[3] Pinaki Nath Chowdhury, Palaiahnakote Shivakumara,Ramachandra Raghavendra, "An Episodic Learning Network for Text Detection on Human Bodies in Sports Images", IEEE Transactions on Circuits and Systems for Video Technology,2021.

[4] Raghunandan, K. S., et al. "multi-script-oriented text detection and recognition in video/scene/born-digital images." IEEE transactions on circuits and systems for video technology 29.4 (2018): 1145-1162.

[5] Y Ahmad I.S., Boufama B., Habashi P., Anderson W., Elamsy T. Automatic license plate recognition: A comparative study; Proceedings of the 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT); Abu Dhabi, United Arab Emirates. 7–10 December 2015; pp. 635–640. [DOI] [Google Scholar].

[6] Atul Patel Chirag Patel Dipti Shah 2013 Automatic Number Plate Recognition System (ANPR): A Survey International Journal of Computer Applications Volume 69– No.9 pp. (0975 – 8887).

[7] Shraddha S Ghadage Sagar R Khedkar 2019 A Review Paper on Automatic Number Plate Recognition System using Machine Learning Algorithms International Journal of Engineering Research & Technology (IJERT) Vol. 8 Issue 12. A cascaded method for text detection in natural scene images.

[8] Karatzas Dimosthenis, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, and Jiri Matas et al.ICDAR 2015 competition on robust reading.In 2015 13th international conference on document analysis and recognition (ICDAR), pages 1156–1160, 2015.

[9] Alonso B., Pòrtilla Á.I., Musolino G., Rindone C., Vitetta A. Network Fundamental Diagram (NFD) and traffic signal control: First empirical evidences from the city of Santander. Transp. Res. Procedia. 2017;27:27–34. doi: 10.1016/j.trpro.2017.12.112. [DOI] [Google Scholar].

[10] Bakhtan M.A.H., Abdullah M., Abd Rahman A. A review on license plate recognition system algorithms; Proceedings of the 2016 International Conference on Information and Communication Technology (ICICTM); Kuala Lumpur, Malaysia. 16–17 May 2016; pp. 84–89. [Google Scholar]

[11] K. KIM, K.I., KIM, J.B. KIM, and H.J. KIM, "Learning-Based Apporach for License Plate Recognition" Proceeding of IEEE Signal Processing Society Workshop, Vol.2, pp.614-623, 2000

[12] J. W. Hsieh, S. H. Yu, and Y. S. Chen. Morphology based license plate detection from complex scenes. 16th International Conference on Pattern Recognition (ICPR'02), pp. 79-179, 2002.