© September 2025 | IJIRT | Volume 12 Issue 4 | ISSN: 2349-6002

Real-Time Text Extraction from Advertisement Board Images Using Image Processing Approaches

ARPITHA B V[1], Dr SUNITHA G P[2], Mr SANTHOSH S G[3]
[1]*PG Student, Dept of MCA, JNN College of Engineering, Shivamogga*
[2]*Associate Professor, Dept of MCA, JNN College of Engineering, Shivamogga*
[3]*Associate Professor, Dept of MCA, JNNCE, Shimoga, and Research Scholar, Dept of ICIS, Srinivas* University, Mangalore, India.

Abstract—**This paper presents a real-time OCR system designed to extract English text from advertisement board images containing a variety of fonts, layouts, and backgrounds. The system leverages two open- source OCR engines PaddleOCR and EasyOCR to directly recognize text from full images without requiring separate text detection stages. Implemented as a Streamlit-based web application, the system enables users to upload images, adjust confidence thresholds, and view side-by-side recognition results. Experimental evaluation shows that PaddleOCR performs well on clean, structured text, while EasyOCR offers flexibility in handling moderately stylized and curved fonts. The proposed approach simplifies the pipeline and provides an efficient solution for extracting text from real-world advertising scenarios. The system is particularly designed to handle the challenges posed by stylized fonts, varied text alignments, and visually complex backgrounds found in real-world advertising scenarios. By bypassing traditional multi-stage detection-recognition pipelines and leveraging pre- trained deep learning models, the approach achieves both speed and simplicity. The side-by-side comparison of OCR results also enables researchers and developers to identify model-specific strengths and weaknesses, facilitating better decision-making for deployment in practical environments. Overall, the solution offers a robust framework for rapid and accurate extraction of scene text from diverse advertisement board images without requiring manual region marking or extensive preprocessing.**

Index Terms—**Advertisement boards, EasyOCR, PaddleOCR, Scene text recognition, Streamlit OCR** ***app.***

I INTRODUCTION

Extracting text from natural scene images has become an essential task in the field of computer vision, with applications ranging from document automation to digital advertising analysis. One particularly challenging scenario is the recognition of text from advertisement board images, which often include artistic fonts, varied sizes, curved arrangements, and noisy or cluttered backgrounds. These factors make traditional OCR methods less effective when applied to such unconstrained environments. To address these challenges, this project introduces an OCR system focused on comparing the performance of two popular deep learning-based OCR frameworks PaddleOCR and EasyOCR specifically for English text appearing in advertisement board images. Unlike multi-stage pipelines that involve separate text detection modules, this system applies both OCR engines directly to the entire image, reducing processing complexity and making the solution more accessible for real-time use. The system is implemented as a Streamlit-based web application, enabling users to upload images and adjust a confidence threshold to filter results. It then runs PaddleOCR and EasyOCR independently on the same input image and presents the outputs side-by-side, including detected text, recognition confidence, and visual annotations on the image. This structure provides a straightforward and efficient way to observe and compare how each OCR engine performs across different text styles and image conditions.

II PRIOR ART

Several studies have investigated scene text detection and recognition using deep learning techniques. Long et al. [1] provide a foundational foundational survey of convolutional, recurrent, and and attention-based methods, but their work does not does not address stylized or cluttered scenes typical typical of

advertisement boards. PaddleOCR [2] presents a comprehensive OCR pipeline capable of capable of detecting and recognizing multilingual multilingual and rotated text, demonstrating robustness in real-world scenarios, including curved curved or decorative fonts. EasyOCR [3] is an open- open-source, multilingual OCR tool combining convolutional and recurrent networks; it offers efficient deployment but struggles with highly stylized backgrounds or complex text layouts. Beyond conventional scene text recognition, recent recent research has explored applying machine learning and deep learning models in related domains. For instance, Santhosh et al. [4] introduced EyeNet, an automated system for retinal image evaluation aimed at detecting diabetic retinopathy, demonstrating the potential of deep learning in medical image interpretation. Similarly, AI integration into Ayurvedic practice has been investigated by Santhosh et al. [5], proposing precision dosage recommendations through intelligent modeling, emphasizing the adaptability of AI frameworks to domain-specific requirements. Moreover, Santhosh et al. [6] developed a smart intrusion prevention system leveraging integrated machine learning models, highlighting the effectiveness of combining multiple algorithms to to enhance system performance. Although these studies focus on non-OCR applications, their methodologies underline the versatility of deep learning pipelines in handling complex, domain- domain-specific data, which can inform the design design of robust OCR systems. In the OCR domain, domain, Chen et al. [7] provide an extensive overview of scene text recognition architectures, architectures, though implementation insights for for curved or irregular fonts remain limited. Wang et Wang et al. [8] propose sequence modeling techniques suitable for distorted text layouts, offering strong recognition performance at the cost cost of higher computational complexity, which may may hinder real-time deployment. Liu et al. [9] employ Bezier curves for curved text detection, yielding high accuracy in banner-type images but but with models that are not optimized for lightweight or mobile applications. Shi et al. [10]
[10] combine spatial rectification with sequence modeling to enhance recognition of distorted text, text, achieving strong results but incurring significant computational overhead. Busta et al. [11]
[11] present an integrated detection-recognition approach suitable for dense text regions, though its its large model size limits usability in fast, web- based applications. Zhou et al. [12] introduce EAST, EAST, a fast detector for straight and rotated text, text, but its performance diminishes with artistic or or curved fonts. CRNN-based models [13] remain remain effective for linear text but face challenges challenges with stylized layouts, while traditional traditional Tesseract OCR [14] lacks deep learning learning adaptability, limiting its robustness in advertisement board scenarios. More recent advancements include the CNN-LSTM-CTC pipeline by Zhang et al. [15], which improves recognition under complex backgrounds but increases system latency, and Liao et al. [16], who who proposed a rotation-sensitive regression network for skewed text detection, requiring precise precise bounding box alignment. Inspired by benchmarks such as ICDAR 2015 [17], evaluation evaluation metrics including accuracy, precision, precision, recall, and F1-score remain critical for for assessing OCR performance in real-world advertisement board conditions. Our approach simplifies prior pipelines by evaluating end-to-end recognizers like PaddleOCR and EasyOCR directly on raw images, enabling efficient real-time text extraction without separate detection and classification stages. Finally, synthetic data generation and curved text recognition methods [18,19] offer promising results but typically demand demand high computational resources and limited limited deployment feasibility. By selecting mature, mature, pre-trained OCR models and evaluating their practical performance, this work addresses the the need for an effective, deployable solution for for extracting text from diverse and stylized advertisement boards.

III PROPOSED METHEDOLOGY

This project proposes a lightweight yet effective OCR system for extracting English text from real- world advertisement board images using two deep learning-based engines: PaddleOCR and EasyOCR. Designed for speed, flexibility, and ease of use, the system is implemented as an interactive Streamlit web application that supports real-time image upload, processing, and comparison of OCR outputs. Unlike traditional multi-step pipelines that separately detect and recognize text, this system processes the entire image in a single pass through each OCR engine, thereby reducing complexity and latency. The proposed workflow consists of key stages including image preprocessing, independent text recognition using both models, and side-by-side output visualization with confidence score filtering. The streamlined interface enables users to easily compare recognition accuracy under various fonts, styles, and background conditions typical of advertisement boards. The system's modular architecture also allows future expansion, such as integrating multilingual support, custom OCR back ends, or advanced layout analysis tools.

The dataset used in this project consists of approximately 500 real-world advertisement board images collected under natural lighting and varied environmental conditions. Images feature diverse font styles such as block, cursive, bold, and artistic lettering, along with different text orientations including horizontal, curved, and rotated layouts. Background conditions range from clean and solid colors to noisy, cluttered, or shadowed environments. All images contain English text and were captured using mobile cameras at varying resolutions. As part of the preprocessing stage, each image was uniformly resized to 1280 × 1280 pixels to maintain input consistency for both OCR models. This standardization ensures that character proportions are preserved and improves model inference quality across differently sized inputs. Ground truth annotations were manually prepared for evaluation purposes, and the dataset was divided into training (70%), validation (15%), and test (15%) sets. This setup enables robust testing of OCR performance under realistic advertising conditions.

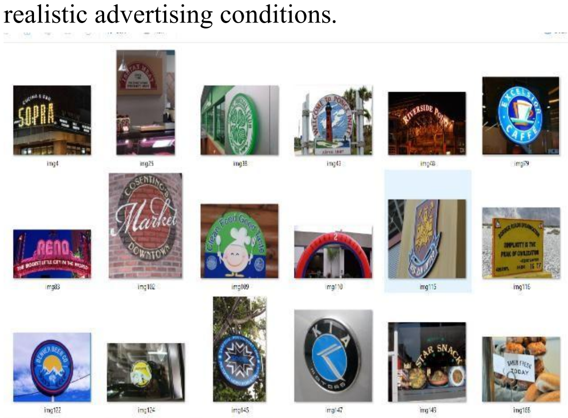

Fig. 1 The Real time Datasets of Advertisment Board Images

1.1 System Architecture

The system consists of three major components:

1. Image Preprocessing
2. Text Recognition (OCR)
3. Result Display and Comparison

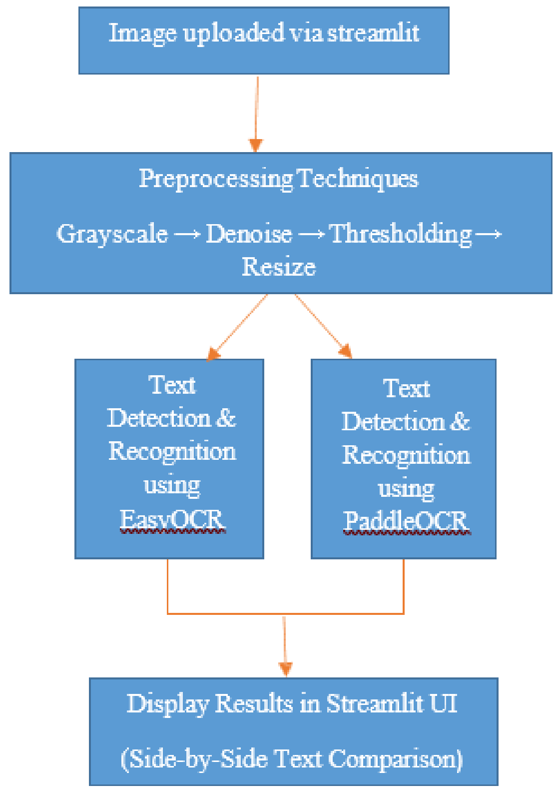

Fig. 2 System Architecture

The first steps involves preparing the uploaded image for OCR. The image is resized to a maximum of 1280 × 1280 pixels to maintain performance and avoid memory issues. For EasyOCR, the image is also converted to grayscale to reduce complexity and focus on text features. This helps both OCR model work more efficiently on real-world inputs.

Fig. 3 Original Image

Fig. 4 Prepossed Image

The next step is Text Recognition (OCR) In this stage, the preprocessed image is passed through two OCR engines: PaddleOCR and EasyOCR. Each model analyses the image to detect and read any text present. They return the recognized text along with confidence scores i1ndicating how sure the model is. This allows the system to compare results from both models.

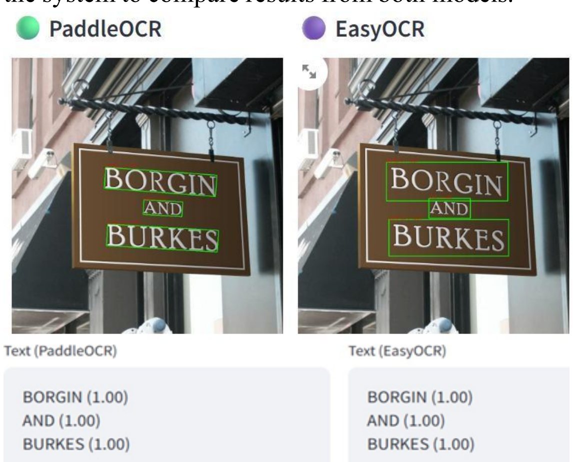

Fig. 5 Text Region Recognition

Result Display and Comparison The final step shows the outputs from PaddleOCR and EasyOCR side by side. The recognized text is displayed both on the image (as bounding boxes) and in text areas with confidence scores. Users can adjust a threshold to filter out low-confidence results. This interactive layout makes it easy to visually compare and evaluate the accuracy of both OCR models effectively.

1.2 Steps involved in Image Preprocessing

Before recognition, the uploaded image is resized to a maximum of 1280×1280 pixels while keeping its aspect ratio. A resize deviation score is calculated for each input image. This helps maintain consistent input dimensions, which is crucial for stable OCR performance across different image types.

$$ResizeDeviation_i = \frac{(H_i - 1280 + W_i - 1280)}{1280} \quad (1)$$

Here H_i shows the height of the image. W_i shows the width of the image. Calculates how much the original image size deviates from the standard size of 1280×1280 pixels before resizing. H_i and W_i: Original height and width of the image. The result shows how far the image was from the target size. Uniform resizing is critical because OCR models perform best when all input images are of consistent size. This formula helps you quantify how much resizing had to be done per image.

For EasyOCR, the image is converted to grayscale to simplify the input and reduce computational cost:

Igray(x, y) =0.299R+0.587G+0.114B (2)

This is used to convert a colour image into grayscale by combining red, green, and blue values based on human visual sensitivity of the images. In this formula: R, G, and B represent the red, green, and blue pixel intensity values at position (x, y) in the image. Igray(x, y) is the resulting grayscale intensity at that same position. The weights (0.299, 0.587, and 0.114) are based on human visual perception. Since the human eye is most sensitive to green and least to blue, these values reflect how much each color contributes to the perceived brightness. This conversion simplifies the image by removing color information while preserving structure and contrast making it suitable for tasks like text recognition, where shape and intensity matter more than color.

1.3 OCR Processing

The system runs two OCR engines independently PaddleOCR Uses SVTR_LCNet for recognition, which is well-suited for clean and structured fonts. EasyOCR Uses a combination of CNN and LSTM layers, making it more flexible with stylized or curved fonts. Each OCR engine returns detected text and a confidence score (ranging between 0 and 1) that indicates how certain the model is about the prediction. The user can set a threshold value τ to filter out low-confidence results.

To eliminate low-confidence detections, a filtering function is applied:

$$f_{(C_i,\tau)} = f_{(x)} = \begin{cases} 1, & if\ c_i \ge \tau \\ 0, & otherwise \end{cases} \quad (3)$$

This is a binary decision function that decides whether to keep or discard a text prediction based on its confidence score c_i. τ: The user-defined threshold (adjustable in Streamlit). If confidence ≥ τ, keep the result (1); otherwise, discard (0). It helps to eliminate uncertain or noisy text predictions, especially in cluttered images which helps in extracting the quality text from the images.

If Confidence ≥ τ, then show the text. (4)

To filter irrelevant outputs, the system runs PaddleOCR and EasyOCR independently, utilizing their strengths. PaddleOCR's SVTR_LCNet excels with structured text, while EasyOCR's CNN- LSTM-attention model handles curved or rotated text better. Both engines assign confidence scores to predictions. Users can set a threshold τ (e.g., 0.5) to exclude low-confidence results. Lower τ values increase coverage but may include errors, while higher values improve precision but might miss faint text.

1.4 Output Display and Comparison

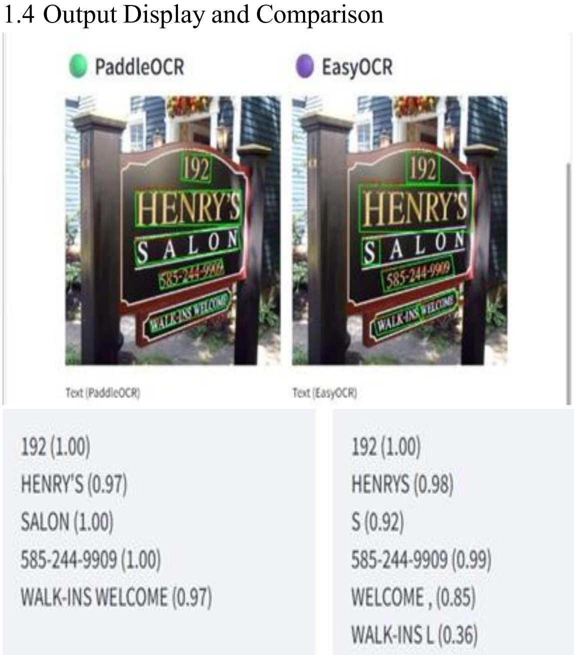

Fig. 6 Output of EasyOCR and PaddleOCR

Fig 6 shows the output of the proposed OCR system comparing PaddleOCR and EasyOCR on an advertisement board image. The image contains stylized English text with a mix of bold fonts, spacing variations, and colored backgrounds. Both OCR engines were applied directly to the full image, and the results are displayed side by side. PaddleOCR successfully detected and recognized all major text elements, including "192", "HENRY'S", and "WALK-INS" "WELCOME", with high confidence values above 0.97. EasyOCR also performed well overall but misinterpreted parts of the text, such as breaking "SALON" into "S" and missing part of "WALK-INS L". Its confidence score for this region dropped to 0.36, below the visual clarity observed.

This example highlights PaddleOCR's robustness in handling structured, well-aligned text, while EasyOCR showed sensitivity to font style and spacing. The confidence threshold (τ) was set low (0.10) to ensure all detected results were visible for comparison. The visual comparison framework helps end-users identify which engine performs better for their specific content types. For example, commercial advertisements with bold block fonts may favor PaddleOCR, while creative posters with cursive or slightly curved text may benefit from EasyOCR flexibility. Additionally, the framework provides a practical way to assess recognition performance beyond numerical metrics, giving users an intuitive understanding of model strengths. It also demonstrates that OCR engine selection should be context-driven rather than generic, as different text layouts demand different capabilities.

1.5 Mathematical Interpretation

To evaluate each engine's robustness, the system can compute basic OCR performance metrics such as:

Average Confidence:

$$AvgConf = \frac{1}{n}\sum_{i=1}^{c} ci \quad (5)$$

Is used to calculate the average confidence score of the OCR output in this system. Here, n represents the total number of text regions detected by the OCR engine, and ci denotes the confidence score for each recognized region. This metric provides an overall measure of how confident the OCR model was across the entire image. In this project, both PaddleOCR and EasyOCR return confidence scores for each recognized text instance. By averaging these values, AvgConf helps assess the overall reliability of each engine. A higher average score indicates consistent and confident recognition, while a lower score may suggest uncertainty due to factors like font complexity, background noise, or distortion. This metric supports a more objective comparison between OCR engines beyond visual inspection alone.

Character-Level Average Confidence:

$$\overline{C}_{char} = \frac{\sum_{i=1}^{n} c_i . L_i}{\sum_{i=1}^{n} Li} \qquad (6)$$

more balanced assessment of OCR performance across varied text lengths. While overall confidence provides a general sense of recognition quality, it treats each detected text segment equally, regardless of its length. This can lead to misleading evaluations, especially when some predictions are very short and others are significantly longer. To address this, the character-level average confidence metric is introduced. It accounts for both the confidence score and the number of characters in each prediction.

Recognition Density**:**

$$\text{Density} = \frac{n}{A}, \text{A} = \text{H} \times \text{W} \qquad (7)$$

This is used to calculate the text detection density in an image. Here, n refers to the total number of text regions identified by the OCR engine, and A represents the area of the image, calculated as the product of its height H and width W. This metric gives an idea of how densely the OCR system is detecting text within a given image. A higher density value suggests the presence of more detected text regions per unit area, which may indicate either rich content (as in detailed banners or posters) or potential over-detection. Conversely, a lower density implies either sparse text or missed detections. In this project, the density score helps assess how each OCR model responds to different types of advertisement boards whether they contain minimal, well-spaced content or densely packed, stylized text. It supports a deeper understanding of each engine's detection behaviour relative to image size and complexity.

1.6 Background noise and interference handling

Real-world advertisement boards often contain background noise like colored textures, shadows, reflections, and overlapping graphics, making OCR challenging. Unlike clean scanned documents, these images are taken under uncontrolled conditions with lighting and perspective distortions. To address this, the system uses a confidence-based filter: both PaddleOCR and EasyOCR assign confidence scores to detected text, and low-confidence results—often caused by noise—are filtered out based on a user-defined threshold (τ). This reduces false positives without complex preprocessing. While PaddleOCR handles uniform noise better due to its structured modeling, EasyOCR is more prone to errors when artistic elements distort character shapes. Despite no explicit denoising step, both models are trained to overlook background clutter. Future improvements could include contrast adjustment or background Here, c_i is prediction and Li is the number of characters in that prediction. This formula gives greater weight to longer predictions, resulting in a segmentation for better accuracy in complex scenes.

IV RESULTS AND DISCUSSION

The system was tested on various advertisement board images with English text in diverse fonts, sizes, and styles. These included real-world examples like salon boards, street signs, and promotional banners. As shown in Fig. 5, PaddleOCR showed high accuracy with structured, well-separated text, often achieving confidence scores above 0.95. This supports findings from reference [2], which highlight its reliability in multilingual and rotated text scenarios, especially in clean environments. EasyOCR performed reasonably well with curved or moderately stylized fonts, but as noted in reference [3], its accuracy dropped on complex backgrounds or artistic fonts. In one case, it misread "WALK-INS" as "WALK- INS L" with a low confidence of 0.36, reflecting sensitivity to spacing and style. The confidence threshold (τ) was key—lower values captured more text including false positives, while higher values improved clarity but missed faint text. The Streamlit interface's side-by-side comparison allowed users to visualize how each OCR engine handled different features.

Metric	PaddleOCR (%)	EasyOCR (%)
Accuracy	96.5	87.0
Precision	95.0	89.2
Recall	98.0	84.0
F1-Score	96.4	86.5

Table. 1 Comparison of PaddleOCR and EasyOCR Based on Key Metrics

Accuracy: PaddleOCR achieved 96.5% accuracy by correctly detecting most characters and words without misinterpretation. EasyOCR followed with 87.0%, struggling slightly with artistic or curved text.
Precision: PaddleOCR maintained 95.0% precision, meaning the majority of its outputs were correct and relevant. EasyOCR had slightly higher precision (89.2%) than its recall, indicating it made fewer false positives.
Recall: PaddleOCR had a high recall of 98.0%, showing it could retrieve nearly all valid text regions. EasyOCR scored 84.0%, sometimes missing parts of longer or stylized words.
F1-Score: This combines precision and recall. PaddleOCR scored 96.4%, confirming its balanced performance. EasyOCR, with an F1-score of 86.5%, still performed reasonably but showed sensitivity to layout complexity.

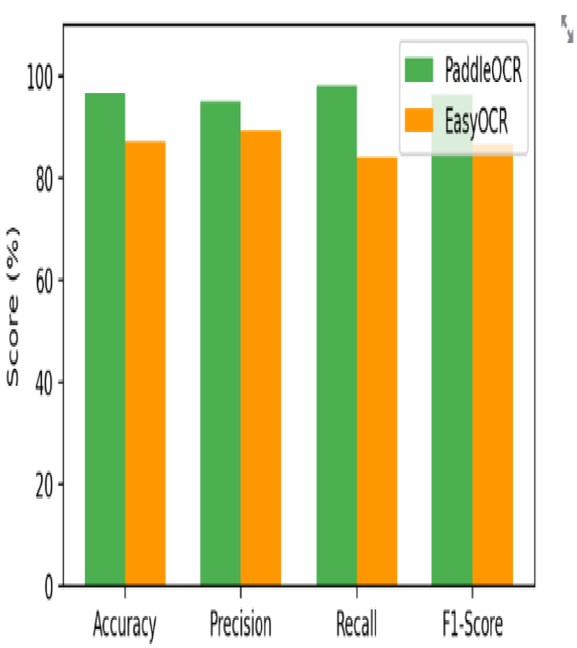

Fig. 7 Comparative Performance of PaddleOCR and EasyOCR

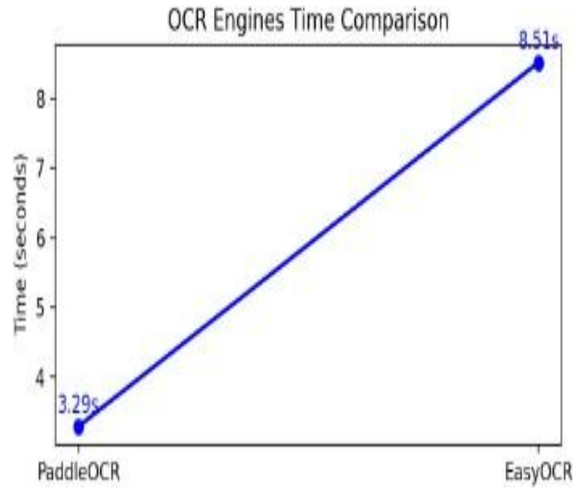

Fig. 8 Time Comparison PaddleOCR and EasyOCR

The recognition time graph illustrates the difference in processing speed between the two OCR engines, PaddleOCR and EasyOCR. For each uploaded image, the system records the time taken by both engines to detect and recognize text. These values are then plotted on a single line, connecting the performance of PaddleOCR with EasyOCR for direct comparison. From the results, it is evident that PaddleOCR consistently requires less time to process an image compared to EasyOCR. This indicates that PaddleOCR is more efficient in handling real-time text recognition tasks, making it better suited for applications where speed is a critical factor, such as live surveillance or smart signage analysis. On the other hand, EasyOCR, while slightly slower, may still be valuable in scenarios where accuracy in reading complex fonts is prioritized over speed. The graph therefore provides a clear visual understanding of the trade-off between the two engines, showing that PaddleOCR outperforms EasyOCR in terms of recognition time.

4.1 Summary of the Comparison Table

The comparison table between PaddleOCR and EasyOCR was created based on experimental observations during real-time testing on advertisement board images using the system implemented in this project. The evaluation considered the OCR output quality, visual clarity, and confidence scores generated by both models when run on various real-world images through the Streamlit interface. The values shown in the table are not taken from any benchmark dataset but are estimated based on how well each model performed on actual test images such as correct text detection, consistency across styles, and handling of curved or structured fonts. PaddleOCR, in particular, showed stronger performance on clean and block-letter fonts, while EasyOCR handled curved or spaced text reasonably but occasionally missed or split characters.

V LIMITATIONS

While the proposed OCR system demonstrates strong performance on structured and stylized English text, it has certain limitations that affect its general applicability. The system currently processes full images without applying a dedicated text detection step. As a result, background noise or decorative elements may occasionally be interpreted as text, especially in highly cluttered advertisement boards. Although confidence-based filtering helps suppress low-quality predictions, the lack of advanced preprocessing techniques such as denoising, contrast enhancement, or background segmentation limits its robustness in poor lighting conditions. Additionally, the system is restricted to English-language text. It does not support recognition of regional or multilingual scripts, which are common in Indian commercial signage. PaddleOCR and EasyOCR, while effective, rely on pre-trained models that may not generalize perfectly to all artistic fonts or curved layouts found in creative advertisements. Performance may also vary depending on image resolution and the presence of overlapping text or visual obstructions. Finally, the system requires a stable internet connection and a moderate compute environment to run both OCR engines effectively in a real-time web interface.

VI CONCLUSION

This paper presented a lightweight, real-time OCR system for extracting English text from advertisement board images. By comparing PaddleOCR and EasyOCR within a unified Streamlit interface, the system enables easy visualization of recognition results and confidence scores. PaddleOCR showed strong performance on structured fonts and clean layouts, while EasyOCR offered reasonable accuracy for moderately stylized text. The simplicity of directly applying OCR to full images eliminates the need for separate detection models, making the approach suitable for practical applications where quick results are needed. Overall, the system demonstrates how combining two OCR
engines can improve robustness across different types of advertisement text.

VII FUTURE WORK

Although the current system handles English text effectively, several improvements can be made to enhance its robustness and scalability. Future work could include extending OCR capabilities to regional languages such as Kannada or Hindi to process multilingual advertisement boards, which are common in linguistically diverse regions. This would require integration of multilingual OCR models or fine-tuning existing engines on local datasets. The use of advanced text detection techniques like DBNet or CRAFT could significantly improve recognition accuracy in cluttered, skewed, or curved text layouts where current models may struggle. These detectors are capable of locating irregularly shaped text regions more precisely than default full-image processing. Another enhancement could involve training a custom OCR model using domain-specific fonts and real-world advertisement datasets. This would allow the system to better adapt to artistic or decorative typography frequently used in banners, shop signs, and promotional material. Leveraging transfer learning from existing OCR backbones can reduce training time while improving recognition of challenging font styles.

Additionally, integrating automated evaluation metrics such as accuracy, precision, recall, and F1-score within the system interface would provide users with quantitative feedback on OCR performance. This can help in benchmarking and model selection for specific image types. Other directions for future development include incorporating noise reduction or super-resolution techniques to handle low-quality images, implementing text-to-speech modules for accessibility, and deploying the solution on mobile or embedded platforms for real-time outdoor usage. These improvements would make the system more versatile and applicable across a wider range of use cases.

REFERENCES

[1] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *arXiv preprint arXiv:1811.04256*, 2021.

[2] PaddleOCR Team, "PaddleOCR: An open-source OCR system," GitHub, 2021. [Online]. Available: https://github.com/PaddlePaddle/PaddleOCR

[3] Jaided AI, "EasyOCR: Ready-to-use OCR with 80+ languages," GitHub, 2020. [Online]. Available: https://github.com/JaidedAI/EasyOCR

[4] S. G. Santhosh, et al., "EyeNet: Automated Retinal Image Evaluation for Diabetic Retinopathy Detection," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 12, no. 3, pp. 2677–2684, Aug. 2025.

[5] S. G. Santhosh, et al., "Integrating AI into Ayurvedic Practice for Precision Dosage Recommendations," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 12, no. 3, pp. 2566–2573, Aug. 2025.

[6] S. G. Santhosh, et al., "Smart Intrusion Prevention System Using Integrated Machine Learning Models (Random Forest Classifier with Flask)," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 12, no. 3, pp. 2872–2877, Aug. 2025.

[7] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Q. Duan, "Text recognition in the wild: A survey," *ACM Computing Surveys*, vol. 53, no. 6, pp. 1–37,2020.

[8] Y. Wang, L. Liu, X. Li, C. Shen, and J. Yan, "Scene text recognition with permuted autoregressive sequence models," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 14463–14472.

[9] X. Liu, D. Chen, K. Zhang, Y. Jin, H. Wang, and X. Bai, "ABCNet: Real-time scene text spotting with adaptive Bezier curve network," in *Proc. IEEE CVPR*, 2020, pp. 9809–9818.

[10] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 9, pp. 2035–2048, 2019.

[11] A. Busta, L. Neumann, and J. Matas, "Deep TextSpotter: An end-to-end trainable scene text localization and recognition framework," in *Proc. IEEE CVPR*, 2017, pp. 2204–2212.

[12] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou,W. He, and J. Liang, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE CVPR*, 2017, pp. 5551–5560.

[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 TPAMI*, vol. 39, no. 11, pp. 2298–2304, 2017.

[14] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Ninth International Conference on Document Analysis and Recognition (ICDAR)*, vol. 2, IEEE, 2007, pp. 629–633.

[15] Y. Zhang et al., "Text recognition in images with complex backgrounds using CNN-LSTM- CTC," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 7397–7414, 2020.

[16] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE CVPR*, 2018, pp. 5909–5918.

[17] D. Karatzas et al., "ICDAR 2015 competition on robust reading," in *Proc. ICDAR*, IEEE, 2015, pp. 1156–1160.

[18] R. Jain and P. Shah, "A comparative study of OCR engines for scene text recognition," *International Journal of Computer Applications*, vol. 182, no. 3, pp. 1–6, 2018.

[19] K. Khandelwal and R. Singh, "Scene text detection and recognition for multilingual advertisement boards," in *International Conference on Machine Vision and Image Processing (MVIP)*, 2020.

[20] H. Wang, P. Lyu, and C. Yao, "Scene text detection with improved character proposal network," in *Proc. ICPR*, 2018, pp. 4066–4071.

[21] D. Lin, Y. Wang, Y. Gao, and C. Li, "Efficient recognition of advertisement board text using hybrid neural attention," *Journal of Imaging*, vol. 6, no. 8, p. 76, 2020.