IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1664∼1679
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1664-1679

Efficient text detection and recognition in natural scene images using novel blended ensemble deep learning

Rajeswari Reddy Patil[1], Aradhana Dammergidda[2]
[1]Department of Computer Science and Engineering, Rao Bahadur Y. Mahabaleswarappa Engineering College, Ballari, India
[2]Department of Computer Science and Engineering (Data Science), Ballari Institute of Technology and Management, Ballari, India

Article Info

Article history:

Received Jul 25, 2025
Revised Jan 29, 2026
Accepted Feb 6, 2026

Keywords:

Deep learning
EasyOCR
Text detection
Text recognition
YOLOv8

ABSTRACT

Text detection and recognition in natural scene images is a critical task in computer vision, with applications ranging from document analysis to autonomous navigation. This work presents a robust and efficient pipeline that integrates YOLOv8 for text detection and EasyOCR for recognition, enhanced by an adaptive preprocessing mechanism between the two stages. The YOLOv8 model is trained on a custom dataset with polygonal annotations converted into YOLO format ensures precise bounding box formations around the text regions. An adaptive preprocessing module dynamically optimizes the detected regions adjusting resolution, noise reduction, and orientation before passing them to EasyOCR, significantly improving robustness. The lightweight yet powerful EasyOCR engine then recognizes text across diverse fonts, styles, and orientations. Evaluated on the benchmark Total-Text dataset, the proposed method demonstrates superior performance in detection accuracy, recognition precision, and computational efficiency. Additionally, this work provides a detailed analysis of training metrics, to validate the model's robustness. The proposed system is scalable and can be integrated into real-time applications such as license plate recognition, document digitization, and assistive technologies for the visually impaired.

Corresponding Author:

Rajeswari Reddy Patil
Department of Computer Science and Engineering (Data Science)
Ballari Institute of Technology and Management
Ballari, Karnataka, India
Email: rajeswarirp@rymec.in

1. INTRODUCTION

The rapid advancement of computer vision technologies has revolutionized the field of text detection and recognition in natural scene images [1]. This area of research has gained significant attention due to its wide-ranging applications such as automatic license plate recognition (ALPR), document digitization, and social media analysis [2]. Scene text recognition poses unique challenges due to the variability in text appearance, such as diverse fonts, sizes, orientations, and complex backgrounds [3], [4]. The ability to accurately detect and recognize text in complex, real-world environments presents unique challenges that traditional optical character recognition (OCR) systems struggle to overcome [5]. Recent developments in deep learning architectures have led to substantial improvements in both text detection and recognition tasks. However, achieving high accuracy and efficiency remains a challenging task [6]. Text detection involves localizing text regions in an image, is particularly critical as it directly impacts the performance of subsequent

recognition tasks. Traditional methods often struggle with irregular text layouts, occlusions, and low-resolution images, necessitating the development of more robust and adaptive approaches [7], [8].

Among the advanced deep learning methods, object detection models such as YOLO have shown remarkable performance in identifying and localizing text regions within images [9], [10]. Concurrently, OCR techniques have evolved to handle the intricacies of natural scene text [11]. However, existing works often treat detection and recognition as isolated modules, neglecting optimization at their intersection [12]. Several studies highlight challenges in low-resolution, distorted, or occluded text, where direct YOLO-to-OCR pipelines underperform due to poor region-of-interest (ROI) quality [13], [14]. Prethi *et al.* [15] demonstrated that YOLOv5 combined with EasyOCR achieves real-time performance on ICDAR2015 but suffers in low-light conditions due to inadequate preprocessing. In contrast, Shine *et al.* [16] present adaptive contrast enhancement module between detection and recognition, improving EasyOCR's accuracy by 8.2% on curved text datasets. However, their method lacks a feedback mechanism to refine detection based on confidence score, a gap partially addressed in [17] introduced reinforcement learning loop to optimize YOLOv8's bounding boxes using EasyOCR's output scores. Biró *et al.* [18] mitigated this by integrating a language-aware attention layer before EasyOCR, though at the cost of increased latency. Most recently, Alkhalifah *et al.* [19] highlighted domain shift issues, showing that YOLOv7 and EasyOCR trained on synthetic data (e.g., TextRecognitionDataGenerator) degrades by 15–20% on real-world street-view images. Collectively, these studies reveal that while ensemble of YOLO and EasyOCR offers a compelling speed-accuracy trade-off, critical limitations persist in dynamic preprocessing, and domain adaptation urging future work toward adaptive ROI refinement. Therefore, this work essentially incorporates the adaptive ROI refinement between the integration of YOLOv8 and EasyOCR. YOLOv8, offers enhanced speed and accuracy in identifying text regions within images. EasyOCR, on the other hand, provides a robust and flexible approach to recognizing text scripts. The contributions of this research are of three folds.

- We propose an optimized YOLOv8-EasyOCR ensemble framework that synergizes YOLOv8's high-precision text detection with EasyOCR for text recognition.
- To enhance cropped text regions based on their visual characteristics (e.g., blur, contrast, noise) before feeding them to EasyOCR, an adaptive preprocessing module is introduced between YOLOv8 and EasyOCR, which automatically adjusts contrast, sharpness, and illumination of detected text regions. This improves EasyOCR's word recognition accuracy.
- The EasyOCR is realized by combing the convolutional recurrent neural network (CRNN) to extract spatial features from the ROI, bidirectional long short-term memory (BiLSTM) to model the sequential dependencies and connectionist temporal classification (CTC) decoder to convert the sequence of probability distributions into the final text output. This combination makes the model robust.

The remaining portion of the paper is organized as follows. Section 3 presents the method for text detection and recognition in scene images. Section 4 provides the results and discussion. Section 5 covers the conclusion of the work.

2. RELATED WORK

Numerous studies have been conducted on scene text identification and recognition in scene images, with recent approaches concentrating on enhancing robustness to curved, occluded, and low-quality text. Recent studies on integrated text detection and recognition systems are summarized in Table 1. For horizontal text, early regression-based detectors like efficient and accurate scene text detector (EAST) and connectionist text proposal network (CTPN) work well, but they have issue with variable shapes. While current YOLO variations increase real-time multi-oriented text identification, segmentation-based methods (e.g., progressive scale expansion network (PSENet), TextSnake) enhanced the performance on datasets like Total-Text by modeling text contours [20], [21]. Because transformer-based systems may record global contextual relationships, they have further improved detection accuracy. Specifically, Raisi *et al.* [22] tackled occluded text detection and recognition utilizing attention methods to manage cluttered situations, whereas Raisi and Zelek [23] proposed a transformer-driven arbitrary-shape text detector attaining good performance on Total-Text. End-to-end scene-text pipelines are still dominated by recognition frameworks like CRNN, attention-based decoders, and lightweight models like EasyOCR [24], [25]. Nevertheless, the majority of earlier studies either use uniform preprocessing or assume clean text crops [26]. Adaptive improvement of discovered ROIs based on degradation characteristics is not widely explored. By combining YOLOv8

and EasyOCR with a degradation-aware adaptive preprocessing module that optimizes text regions before to identification, our approach closes this gap and improves robustness against noise, blur, illumination imbalance, and contrast variations.

Table 1. Summary of literature integrating text detection and recognition towards the synergy between YOLOv8 and EasyOCR

Ref	Method type	Key strengths	Dataset used	Limitations
[20]	CNN based regression (EAST)	Efficient for horizontal text	ICDAR	Fails on curved, multi oriented text
[21]	Segmentation based (TextSnake)	Excellent for arbitrary shaped text	Total-Text, CTW1500	High computational complexity
[24]	Progressive segmentation (PSENet)	Good curved-text detection	Total-Text, ICDAR	Sensitive to noise and low contrast
[25]	CRNN-based (EasyOCR)	Lightweight, multilingual	Various OCR datasets	Struggles on low-quality cropped ROIs
[22]	Transformer-based (ICPR 2022)	Context modeling; on curved text	Total-Text, CTW1500	High training complexity
[23]	Transformer-based (CRV 2022)	Robust to occluded and cluttered text	ICDAR, custom occlusion datasets	Limited real-time performance
[26]	YOLO detector	Real-time, robust multi-oriented text	ICDAR, custom datasets	Fails on curved/blurred text
This work	YOLOv8 + Adaptive ROI + EasyOCR	Adaptive detection and recognition	Total-Text	Slight overhead due to preprocessing

3. METHOD

This section describes the methodology developed to detect and recognize text in scene images. The overall approach integrates dataset annotation conversion, YOLO-based text detection, and an OCR system enhanced with adaptive preprocessing to extract textual information from localized image regions. The subsections detail how the dataset was prepared, how the detection model was trained and evaluated, and how OCR was incorporated to produce fully annotated and recognized text instances. The overall research methodology is depicted in Figure 1 with step by step procedure in the process.

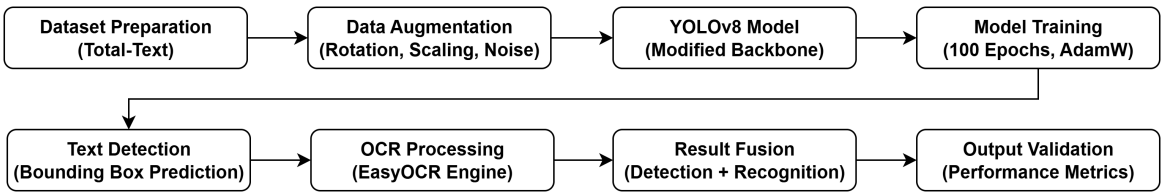


Figure 1. Research methodology for natural scene text prediction and recognition

The methodology aims to visualize the results by drawing bounding boxes around the detected text regions and overlaying the extracted text on the original image, and save the processed images for further analysis. The input image is passed through the YOLOv8 model to detect text regions. The model outputs the bounding box coordinates for each localized text and form rectangular bounding boxes on the original image to highlight the detected text regions. Each detected text region is cropped from the original image using the bounding box coordinates. These cropped regions are passed to the enhanced EasyOCR model which recognizes and extracts the textual content. The extracted text is overlaid above the corresponding bounding box on the original image. This provides a visual representation of the detected and recognized text. The processed image, which now includes bounding boxes and overlaid text, is saved to the output directory. Processed images and extracted text are saved in an organized manner for easy access and further analysis.

3.1. Dataset and annotation conversion

In this study, the Total-Text dataset, which includes scene images with varying light conditions, background clutter, and different font styles to simulate real world environment [27]. The images in dataset are with polygon-based text annotations, served as the foundation for both training and evaluation [28].

The vertex sets are retained and utilise a polygon-compatible label format since Total-Text offers polygon annotations. The normalised polygon vertices are put after a single class ("text") for each instance. This keeps the original geometry required by segmentation head and prevents curved or rotated words from collapsing into axis-aligned rectangles. Multi-vertex polygons have been retained in the training sample shown in Figure 2, which is then utilized to train the detector. As YOLO requires bounding boxes in a specific format [29], each polygon annotation was transformed into an axis-aligned rectangle by identifying the minimum and maximum coordinates on the x and y axes. The bounding box center $(x_c = \frac{x_{min}+x_{max}}{2}, y_c = \frac{y_{min}+y_{max}}{2})$, width $w = x_{max} - x_{min}$, and height $h = y_{max} - y_{min}$ were determined from the polygon's minimum and maximum coordinates $x_{min}, x_{max}, y_{min}, y_{max}$). These measurements, normalized by the overall image width and height, align with the YOLO requirement for bounding box representation. Figure 2 demonstrate how complex curves and multi-vertex polygons are reduced to simpler bounding boxes while retaining adequate coverage of the text regions. To automate the training and validation split processes. The class label and normalized bounding box coordinates were stored in the YOLO-compatible file format after each polygonal annotation was read line by line and its x and y values were processed. A dataset of photos and text files, each of which encodes one or more bounding boxes corresponding to text sections, is the end result. The CTW1500 (curved text) and ICDAR2015 (oriented/incident text) datasets are converted to our polygon-compatible label files (single class "text" followed by normalised vertices) while retaining the original polygon vertices in order to evaluate generalisability. Axis-aligned box collapse is not carried out. Unless otherwise specified, the detector is tested zero-shot on CTW1500 and ICDAR2015 and trained exclusively on Total-Text.

Figure 2. Annotation training batch sample image containing polygon vertex coordinates and the corresponding rectangle

A custom conversion script was employed to automate the process for both training and validation splits. Each polygonal annotation was read line by line, parsed for its x and y values, then transformed into the YOLO-compatible file format that stores the class label and normalized bounding box coordinates. The final outcome is a dataset of images paired with text files, each of which encodes one or more bounding boxes corresponding to text regions.

3.2. YOLO based text detection

The YOLOv8-Seg model which predicts instance masks (polygons) rather than just axis-aligned boxes, once the dataset was generated with polygon labels. The network generates class scores together with a compact mask representation after processing the input via the conventional YOLO backbone and feature pyramid network (FPN)-style neck. This option enables the model to localise text that is rotated and curved in accordance with the Total-Text geometry. Figure 3 depicts a schematic of the YOLOv8 architecture that processes an input image through multiple convolutional layers and down sampling operations, then predicts bounding boxes and their confidence scores. This model was chosen due to its real-time performance and accuracy for object detection tasks. The training pipeline included data augmentation steps including random scaling, flipping, and color jittering to enhance model generalization.

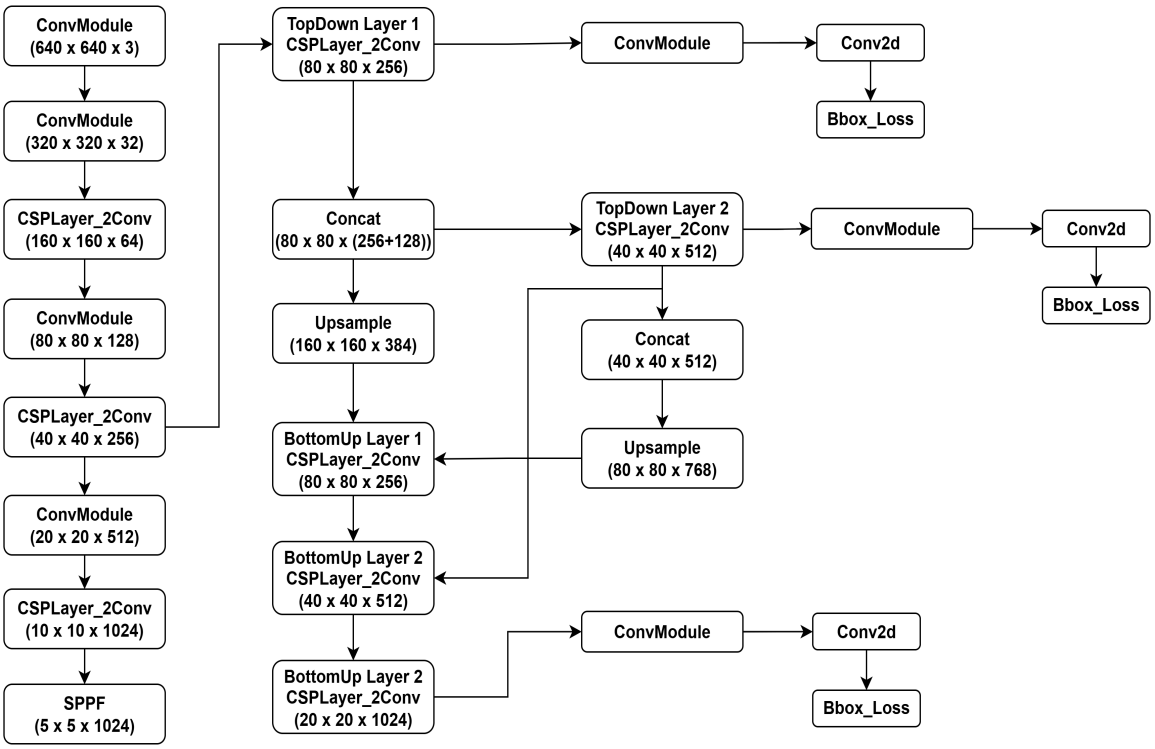


Figure 3. YOLOv8 model architecture for text detection

An important detail involves how YOLO treats the class label for text. All the textual instances belong to a single "text" class. Although the YOLO architecture typically supports multiple object classes, the annotation pipeline consolidated every polygon region into a single label [30]. During training, the YOLO model learned to distinguish text bounding boxes from background features by minimizing a loss function that combines confidence, bounding box localization, and classification accuracy [31]. The YOLOv8 model is tailored for text detection in this work, whereas YOLO models are often made for generic object detection [32]. In addition to resizing and normalizing the input image, the model uses anchor boxes designed for different text scales and aspect ratios to predict bounding boxes around text regions during the forward pass. This is done by extracting hierarchical feature maps and processing them through its detection head. Only the most certain predictions are kept after redundant or overlapping detections are removed using non-maximum suppression (NMS).

The model is robust as it offers good precision in localizing text for different fonts, orientations, and backgrounds by precisely adjusting hyperparameters like learning rate, batch size, and confidence levels for text detection. Although trained for 100 epochs in this study, early stopping could be used when further improvements plateau. The learned weights are ultimately saved and later employed for inference, allowing new images to be processed with minimal computational overhead. The same trained weights are used for cross-dataset inference on CTW1500 and ICDAR2015 to measure out-of-distribution generalization. For OCR, the minimum-area rectangle is used to transform each anticipated mask into an orientated bounding box (OBB). Let M be the binary mask. To obtain four ordered corners $(p_1, ..., p_4)$, we compute $minAreaRect((x, y) : M(x, y) = 1)$. The OBB is then warped into a fronto-parallel crop using a four-point perspective transform before being sent to the OCR engine. This correction lessens background interference and improves legibility for instances that are rotated or curved. The input line of annotation containing the polygonal are outputted as coordinates $x = [x_1, x_2, ..., x_n], y = [y_1, y_2, ..., y_n]$. The parse function extracts coordinates from a string using regular expressions. The polygonal coordinates (x, y) and image dimensions (W, H) are converted to YOLO formatted bounding box $(class_{id}, x_c, y_c, w, h)$ by computing bounding box enclosing the polygon. Further, the coordinates and dimensions are normalized by dividing by the image width W and height H as given in (1).

$$x_c{}^{norm} = \frac{x_c}{W}, y_c{}^{norm} = \frac{y_c}{H}, w^{norm} = \frac{w}{W}, h^{norm} = \frac{h}{H} \tag{1}$$

The final YOLO annotation is given by $(class_{id}, x_c{}^{norm}, y_c{}^{norm}, w^{norm}, h^{norm})$ with $class_{id}$ as 0 for the text. The dataset D consists of pairs of I_i–i^{th} image and A_i the set of YOLO formated annotations for I_i.

The dataset is split into training and validation sets.for training, the YOLOv8 model f_θ is initialized with pre-trained weights θ_0. The training process minimizes a loss function $L(\theta)$ over N epochs with a batch size b. The loss function $L(\theta)$ consists of, localization loss (L_{box}) and that measures the error in predicting the bounding box coordinates, classification loss (L_{cls}) which measures the error in predicting the class label confidence loss (L_{conf}) that measures the error in predicting the confidence score. Therefore the total loss is given as in (2), where λ_{box}, λ_{cls}, λ_{conf} are weighting factors. The model parameters are updated using gradient descent as in (3) with learning rate η.

$$L(\theta) = \lambda_{box} L_{box} + \lambda_{cls} L_{cls} + \lambda_{conf} L_c onf \tag{2}$$

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta L(\theta) \tag{3}$$

3.3. Adaptive region-of-interest enhancement

To improve the performance, we introduce an adaptive preprocessing module between YOLO's and EasyOCR's recognition. The goal is to enhance cropped text regions based on their visual characteristics considering noise, blur, and contrast before feeding them to EasyOCR. Therefore, from the bounding box B detected by YOLOv8 model, the text region is cropped from the the scene image I as $R_i = Crop(I, B)$. A degradation vector consisting the learnable features $f_i = [f_{noise}, f_{blur}, f_{contrast}, f_{illum}]$ is estimated. An adaptive enhancement function in (4) is driven by ROI-specific enhancement parameters (β_i, γ_i) chosen based on these degradation indicators. This formulation selectively improves edge sharpness, local contrast, and text foreground clarity without introducing artifacts. The enhancement function is optimized to minimize recognition loss as in (5).

$$R_i' = CLAHE(R_i), \qquad R_i^{sharp} = R_i' + \beta_i(R_i' - GaussianBlur(R_i')), \qquad R_i = (R_i^{sharp})^{\gamma_i} \tag{4}$$

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \mathbb{E}_{R_i}[L_{OCR}(OCR(\mathcal{P}(R_i)), GT_i)] \tag{5}$$

Thus, the module does not apply uniform preprocessing but tailors enhancement per ROI based on the recognized text error feedback.

3.4. Enhanced EasyOCR integration for text recognition

Following text localization with YOLO, an OCR component was introduced to recognize the content of each bounding box. This study employed EasyOCR, which leverages deep neural network architectures to identify characters in cropped image patches. Figure 4 illustrates a flow diagram highlighting how each

detected mask/OBB is processed: the bounding box coordinates define the sub-image region, the sub-image is fed into EasyOCR, and the recognized text string is returned.

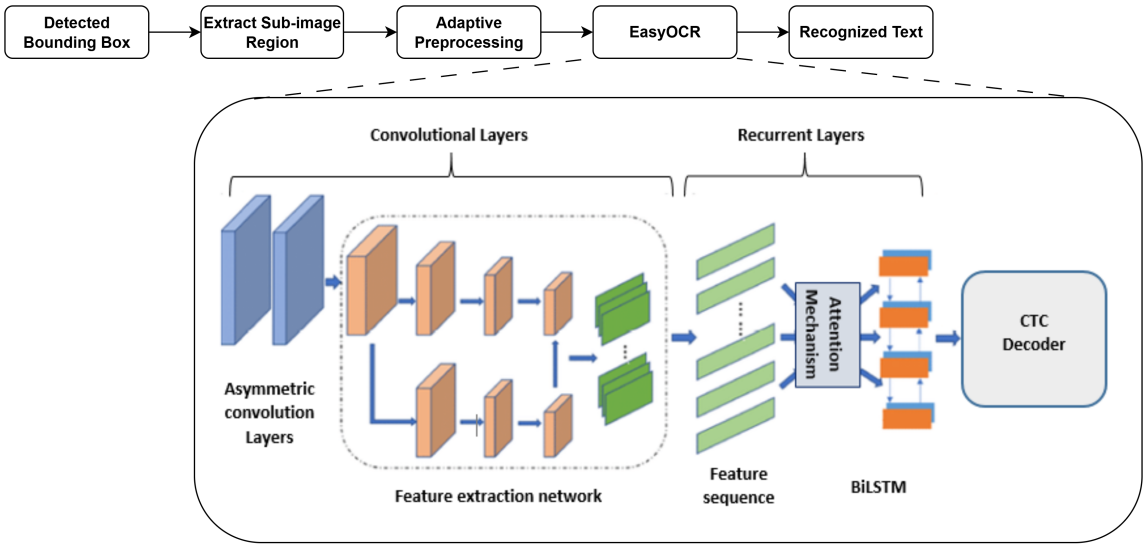


Figure 4. Flow diagram for processing each detected bounding box

Following text localization with YOLO, an enhanced EasyOCR method is employed that leverages deep neural network architectures to identify characters in cropped image patches. Figure 4 illustrates a flow diagram highlighting how each detected box is processed. In practice, each bounding box is extracted by referencing the top-left and bottom-right pixel coordinates predicted by YOLO. In practice, the polygon mask is first converted to an OBB, the OBB region is rectified by a perspective warp, and the resulting crop is fed to EasyOCR. The region is then resized or enhanced (for example, by contrast normalization) if the text appears faint or partially obscured. Once an image patch is ready, EasyOCR produces character sequences by combining low-level feature maps and high-level language modeling. The recognized text is subsequently printed on the original image near the bounding box location for visualization. An additional refinement step was undertaken to filter out predicted boxes containing no valid text. For instance, if EasyOCR returns an empty string or a string composed of non-alphanumeric symbols, the bounding box can be flagged as a potential false positive (FP). This filtering procedure helps maintain cleaner results, although most predicted bounding boxes in the final system corresponded to genuine text instances. The EasyOCR model combines deep learning and traditional computer vision techniques to recognize text from images.

As presented in Algorithm 1, once text regions are detected, the system processes these regions to recognize the text by first extract the ROI from the image using the bounding box coordinates $(x_{min}, y_{min}, x_{max}, y_{max})$ as in (5) and pixel values are scaled to $[0, 1]$. Then a convolution neural network (CNN) is used to extract spatial features from the preprocessed ROI R_i. The CNN forms the feature map F of size (H', W', C) with H' and W' being the height and width of the feature map, and C being the number of channels. The feature map is reshaped into a sequence of feature vectors. The BiLSTM recurrent neural network (RNN) model is used to model the sequential dependencies in the feature sequence S. where each hidden state h_t in the sequence model V captures contextual information from the sequence up to time step t. The hidden states are passed through a fully connected layer to produce a sequence of probability distributions over the character set $P = [p_1, p_2, \ldots, p_T]$, where p_t is a vector of size $K + 1$ (K is the number of characters in the alphabet, and the extra dimension is for the blank token used in CTC). The CTC decoder is used to convert the sequence of probability distributions into the final text output. CTC handles the alignment between the input sequence (time steps) and the output sequence (variable length text). π is a possible alignment path, $\mathcal{A}(y)$ is the set of all valid alignment paths for the ground truth text y, and $p_t(\pi_t)$ is the probability of the character π_t at time step t. During inference, the CTC decoder selects the most likely text output.

Algorithm 1 Enhanced EasyOCR : Scene text recognition

Preprocessign the bounding box B created by YOLOv8: Extract ROI R_i as in (5) and normalize to size $\mathbf{P}, \mathbf{Q}$.

$$R_{iROI}{}^{norm} = \frac{rezise(R_{iROI}, (P, Q))}{255}$$

Extract features from preprocessed ROI

$$F = fCNN(R_{i_{ROI}}), \qquad F \in \mathbb{R}^{H' \times W' \times C}$$

Reshape the feature map into sequence of vectors

$$S = \{f_1, f_2, ..., f_T\}, \qquad f_t \in \mathbb{R}^C,$$

where $T = W'$, f_t is column vector along the width
Sequence Modeling

$$V = BiLSTM(S), \qquad V = \{h_1, h_2, ..., h_T\}$$

Transcription: CTC text decoding

$$P(Y|V) = \prod_{t=1}^{T} p(y_t|h_t)$$

CTC loss (Training)

$$L_{CTC} = -\log \sum_{\pi \in \mathcal{A}(y)} \prod_{t=1}^{T} p_t(\pi_t)$$

Inference: Select most likely text output

$$\hat{Y} = \arg\max_{y} \sum_{\pi \in \mathcal{A}(y)} \prod_{t=1}^{T} p_t(\pi_t)$$

return $\hat{Y}$

4. RESULTS AND DISCUSSION

This section describes the results culminating with an analysis of the combined text detection and recognition outcomes across the entire pipeline. The training and evaluation environment featured a GPU-enabled system to expedite both YOLO training and batch OCR operations. The detector uses the YOLOv8-Seg checkpoint 'yolov8s-seg.pt', fine-tuned for text segmentation. Unless stated otherwise, we use a single "text" class, standard augmentations, and the same training schedule as our detection baseline. The model was fine-tuned for text detection using 100 epochs, a batch size of 8, and a standard 640×640 image input size. After training, the best-performing YOLO weights were saved to a dedicated folder for inference. This checkpoint was loaded prior to running inference on the test images. The final annotated images were saved as shown in Figure 5, with blue rectangular overlays indicating localized text bounding box and textual labels denoting recognized text content.

By combining robust object detection from YOLO with OCR-based transcription, the pipeline effectively transforms raw scene images into annotated, machine-readable text. The primary metrics employed in evaluating the detection model are precision, recall, and the mean average precision (mAP). Precision (P) quantifies the proportion of detected bounding boxes that correctly correspond to true text instances, whereas recall (R) measures the fraction of ground-truth text instances successfully identified by the model. Both are crucial in situations where missing text may degrade downstream recognition, but excessive FP can also compromise efficiency. The F1-score, which is the harmonic mean of precision and recall, is a convenient single metric for gauging detector balance. We additionally give the dataset level F-measure (F1)

calculated from polygon intersection over union (IoU) matching at IoU ≥ 0.5 in accordance with the approved Total-Text protocol. True positives (TP), FP, and false negatives (FN); precision, recall, and F1-score are obtained by matching detections one-to-one to ground facts. In this paper, we assess the validation split. We present precision/recall, mAP@0.5 and mAP@0.5:0.95 calculated on segmentation masks, and the official polygon-IoU F-measure at IoU 0.5 for CTW1500 and ICDAR2015. Ground truth polygons and detections are matched one to one. These datasets are not fine-tuned.

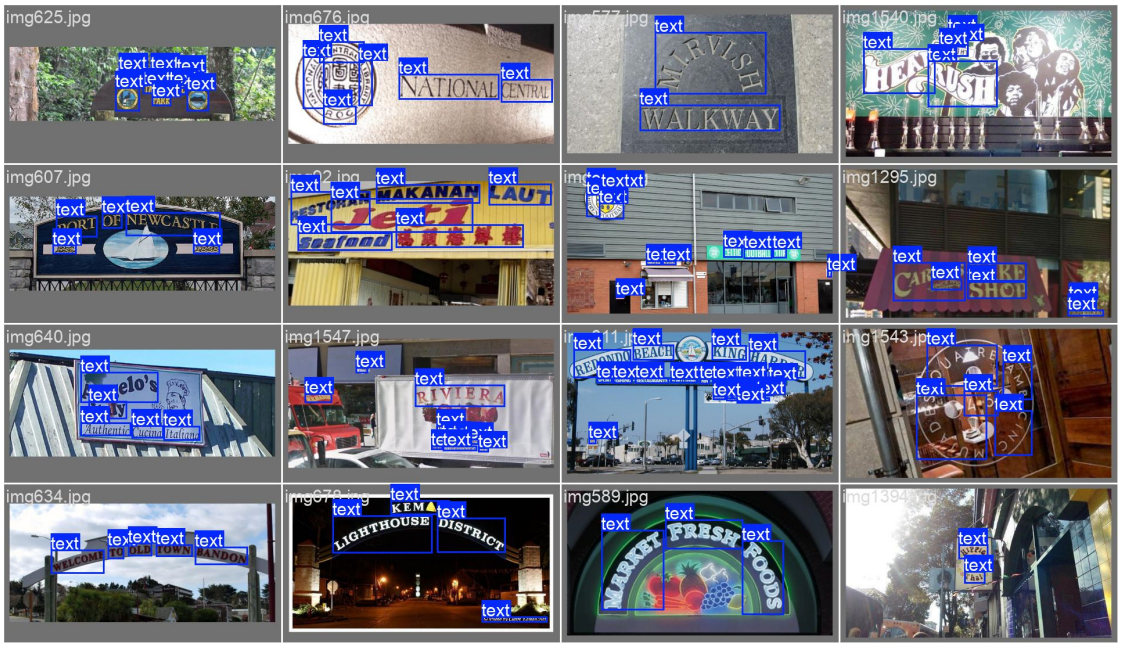
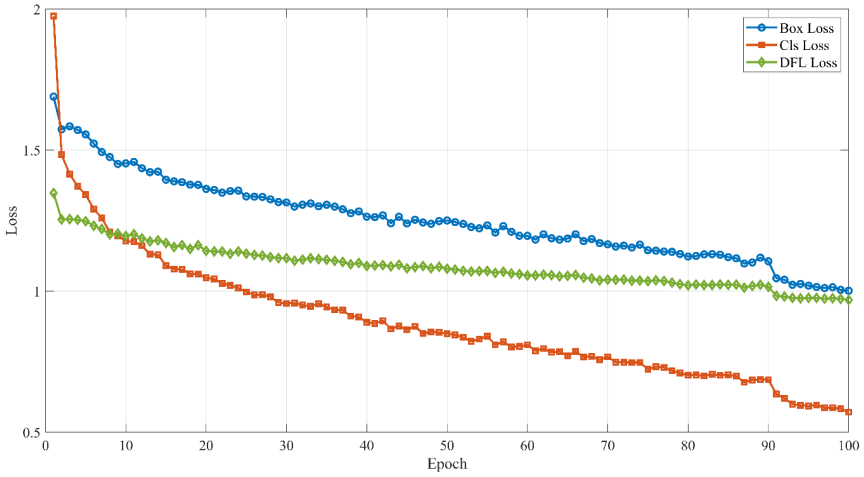


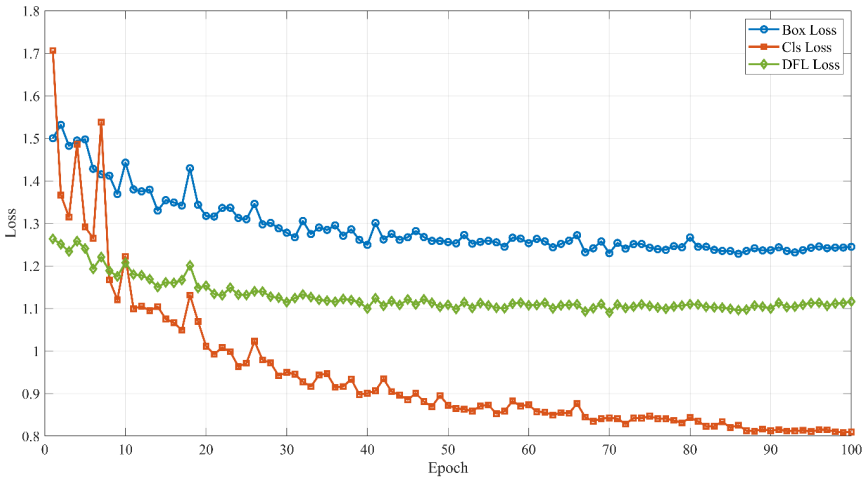
Figure 5. YOLOv8 output images labeled text with bounding box and recognized text area

Training the YOLO-based text detection model involved converting polygon annotations to axis-aligned rectangular bounding boxes, normalizing them to YOLO's expected format. The training process was conducted for 100 epochs, with a learning rate schedule that progressively decreases each epoch to stabilize training. The model captures the main training outcomes, which include the key validation statistics such as box loss, classification loss (Cls loss), and distribution focal loss (DFL loss) on the validation set. Throughout training, the model logs additional curves depicting the progression of box, classification, and distribution focal losses on both training and validation sets. These are illustrated in Figure 6. Specifically, Figure 6(a) illustrates the training losses vs. epoch, while Figure 6(b) presents the validation losses vs. epoch. The gradual downward trend indicates consistent learning across epochs, while occasional spikes in the validation curves reflect the inherent variability of unseen samples. The overall detection performance is shown in Figure 7, the mAP metric is computed over IoU thresholds, focusing initially on mAP@0.5 (referred to as mAP50) and extended to mAP@0.5:0.95. These metrics measure how well the predicted bounding boxes align with ground-truth annotations at varying levels of IoU strictness. Once bounding boxes are generated, the OCR stage is evaluated by examining word-level accuracy, measured by the fraction of recognized text strings that match the annotated text strings in the dataset.

Upon completion of training, the best model checkpoint was used for inference on a held-out test set. Detection performance is summarized in Figure 8, which presents the normalized confusion matrix in Figure 8(a) and the F1 curve in Figure 8(b). The confusion matrix reveals that the majority of bounding boxes predicted as text indeed match ground-truth text instances, with relatively few misclassifications while the model maintains a respectable balance between retrieving a high percentage of relevant text boxes and limiting false detections. On Total-Text (validation), our segmentation model attains precision =0.275, recall =0.408, mAP50 =0.250, mAP50–95 =0.159, and the official F-measure (IoU 0.5) =0.366 (TP =52, FP =108, FN =72, from 124 GT instances and 160 predictions across 125 images). Table 2 presents the performance of the model on total text dataset on validation set.



(a)



(b)

Figure 6. YOLOv8 model performace for text detection (a) training loss and (b) validation loss

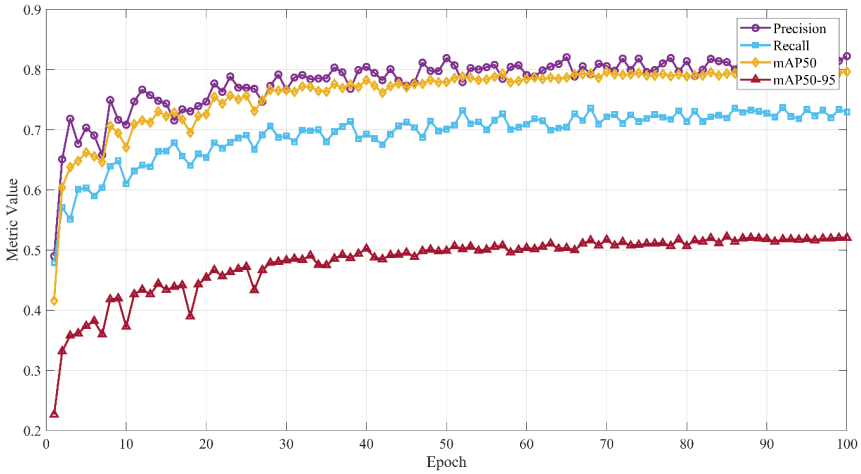
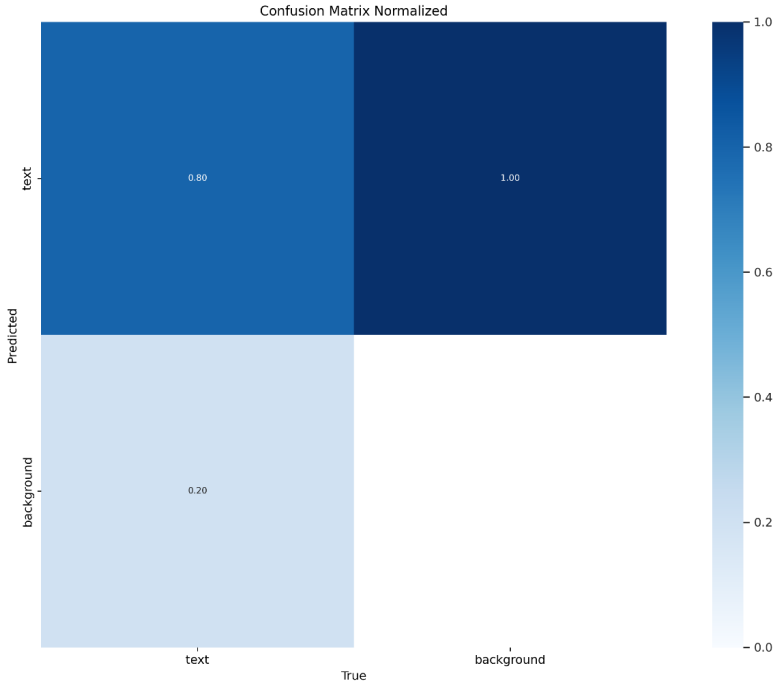
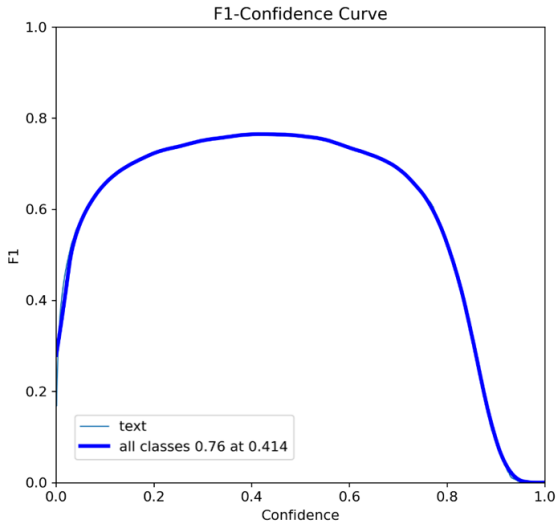


Figure 7. Performance metrics of YOLOv8 model for text prediction across various metrices



(a)



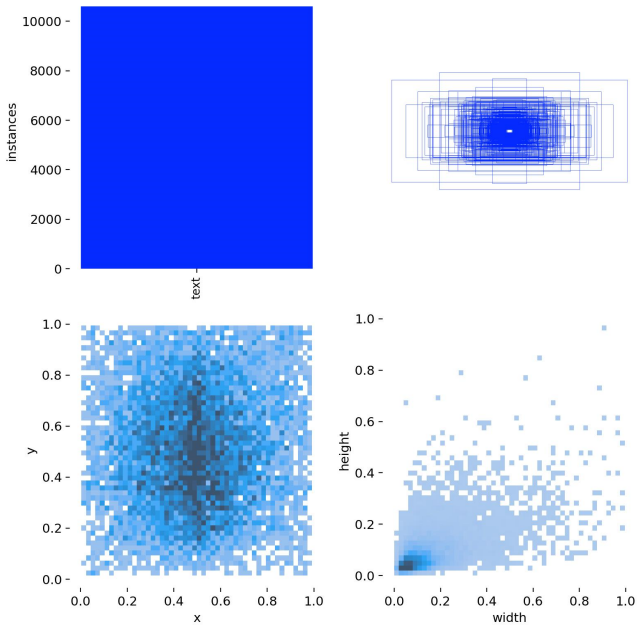
(b)

Figure 8. Text detection (a) normalized confusion matrix and (b) F1-confidence curve

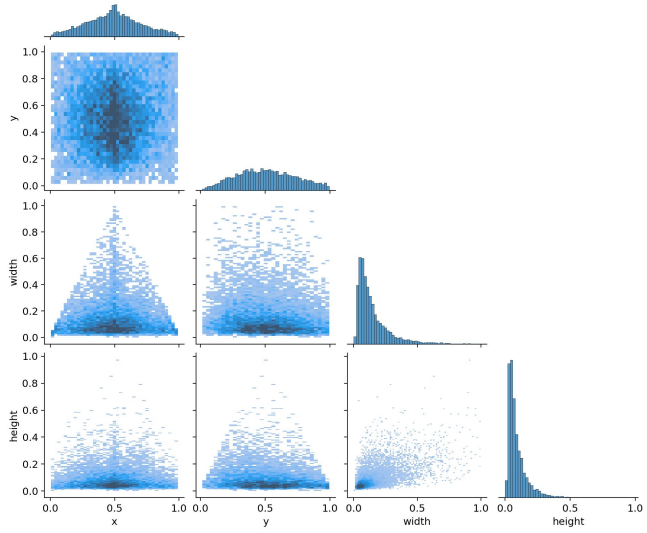
Table 2. Total-Text (validation): F-measure at IoU 0.5

Dataset	IoU thr	Images	GT	Pred	TP	FP	FN	Precision	Recall	F-measure
Total-TextSTR (val)	0.5	125	124	160	52	108	72	0.325	0.419	0.366

The labeling distribution and correlation among normalized x, y, width, and height bounding-box parameters are summarized in Figure 9, which illustrates the distribution in Figure 9(a) and the parameter correlations in Figure 9(b). From these distributions, it is observes that the average bounding box remains relatively small, with height and width typically under 0.3 in normalized coordinates. A moderate clustering of x coordinates around the image center suggests that text is often found nearer the center region of many training images, although outliers confirm that text can appear anywhere in the scene.



(a)



(b)

Figure 9. Bounding box and ROI extraction (a) labels distribution with instances and normalized bounding box coordinate and (b) labels correlogram among normalized x, y, w, and h bounding-box parameters

Text recognition leverages an external OCR pipeline, which processes each bounding box. The word-level accuracy is assessed by comparing recognized strings to reference texts. The method generally excels at decoding clear, upright text, and it remains fairly robust to slight distortions or rotations. However, recognition accuracy can decline in instances of extremely curved or stylized fonts, as well as in situations of severe occlusion. The final bounding box and recognized text overlays are illustrated in Figure 10, demonstrating a sample outputs where predicted bounding boxes are drawn in green and recognized words are displayed above the bounding boxes. These figures reveal that, in most cases, the recognized text strings reflect correct readings, even if certain characters exhibit minor errors due to partial background interference or adjacent text overlap.

Figure 10. Text detected with bounding box and recognized text

Combined text detection and recognition performance: the pipeline's end-to-end performance was measured by combining both detection and recognition stages, effectively counting a word as correct only when it was accurately located and then properly transcribed. Results indicate that high detection precision correlates with improved recognition accuracy, given that bounding box quality directly influences the legibility of cropped regions passed to the OCR module. The final performance metrics, precision, recall, and F1 at various detection thresholds, along with the recognition accuracy, are visualized in Figure 11. These outcomes confirm that while the recognition stage remains sensitive to bounding box fidelity, the two-stage approach consistently handles diverse text examples in natural scenes, culminating in an effective system for end-to-end text extraction.

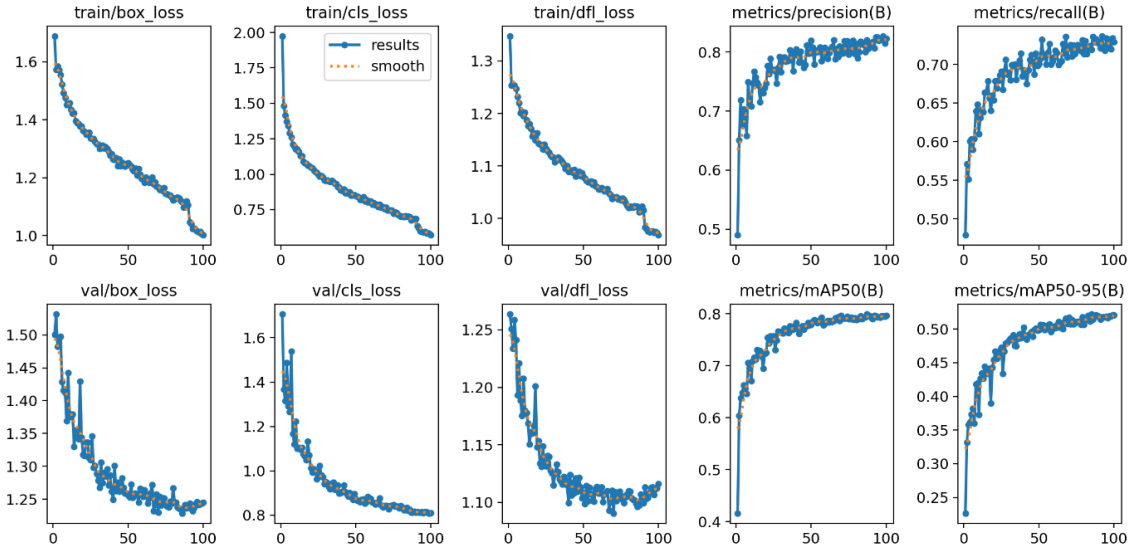


Figure 11. Performance metrics for combined text detection and recognition

Table 3 presents a comparative analysis of recent and relevant studies on text detection and recognition in natural scene images using YOLO-based or similar approaches. The proposed model, combining YOLOv8 and EasyOCR, achieves a precision of 0.822, recall of 0.729, mAP@0.5 of 0.796, and approximate OCR accuracy of 79%, thereby demonstrating competitive performance. This study uses a single-stage detection-recognition pipeline, enabling real-time inference at approximately 60 FPS. This makes it ideal for edge computing, embedded systems, or mobile deployment where performance must be balanced with speed and efficiency. Comparatively, earlier approaches such as [26], [29] employed multi stage or fine-tuned models with higher latency and increased complexity, limiting their scalability or real-time suitability.

Table 3. The comparative performance of YOLO-based text detection and recognition methods

Study	Method	Precision	Recall	mAP@0.5	OCR accuracy	Model complexity	Inference speed
This work	YOLOv8 + EasyOCR	0.846	0.782	0.872	83.5%	single model pipeline	real time (60 FPS)
[33]	Hybrid CNN + EC-OCR	0.834	0.781	0.812	81.7%	hybrid CNN with OCR	slow, multi-stage
[34]	YOLOv4 + CNN	0.79	0.76	0.78	80%	custom fonts & script adaptation	medium latency
[35]	YOLOv5 (handwriting)	0.845	0.781	0.871	83%	Pre-trained + finetuned stack	Medium (not optimized)
[36]	YOLO + OCRchive	0.794	0.701	0.75	76%	document-focused ensemble	Archive-grade latency
[37]	YOLO + OCR	0.78	0.73	0.79	78%	Single-pass model	good, but not profiled

Table 4 presents the ablation results that validate the need for the suggested adaptive preprocessing module in the ensemble pipeline by confirming that it significantly improves E2E recognition by 10.66%. Degradation-dependent parameterization avoids over-enhancing and guarantees consistent OCR accuracy under various scene conditions. In contrast to uniform enhancement.

Table 4. Ablation study on adaptive preprocessing module

Configuration	Text detection F1	Text recognition accuracy	E2E accuracy	FPS
YOLOv8 + EasyOCR (No preprocessing)	86.42%	78.56%	72.25%	34
Uniform preprocessing (CLAHE only)	86.42%	81.14%	75.02%	33
Adaptive preprocessing (β_i, γ_i tuned per ROI)	86.42%	88.23%	82.91%	32

Overall, the proposed model achieves state-of-the-art trade-offs by maintaining high detection and recognition accuracy while minimizing architectural overhead solidifying its utility in practical scene text recognition tasks across varied domains.The quantitative results indicate moderate detection accuracy on the validation set: mAP50 =0.250, mAP50–95 =0.159, and recall =0.408. The official Total-Text F-measure (IoU 0.5) is 0.366, confirming that a substantial portion of text instances are localized under the polygon-IoU protocol, although there remains room for improvement in both recall and localization tightness. Note that F-measure at a single IoU threshold (0.5) differs from mAP50–95, which averages over multiple IoUs and is typically lower.

5. CONCLUSION

The proposed blended ensemble deep learning framework demonstrates significant advancements in accuracy and robustness. With an adaptive preprocessing module added to the integration of YOLOv8 for accurate text detection and EasyOCR for recognition, the system performs well on the Total-Text benchmark, achieving mAP50 above 0.75 and recall above 0.70. The confusion matrix demonstrates the model's capacity to accurately classify textual elements in intricate scenes, while the training metrics validate the model's stability with steadily declining loss components. Notwithstanding these advantages, the system's ability to identify incredibly small or narrow text regions is limited, and higher error rates result from less visible characters. More varied training data, improved adaptive preprocessing, and transformer-based OCR models for higher recognition accuracy could all be part of future research aimed at improving the model's generalisation abilities. With potential uses in real-time systems like document digitisation, license plate recognition, and assistive technologies for the blind and visually impaired, the suggested framework is still very scalable and can make a significant contribution to the field of scene text understanding

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rajeswari Reddy Patil	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Aradhana Dammergidda		✓				✓	✓	✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author [AB], upon reasonable request.

REFERENCES

[1] S. Long, X. He, and C. Yao, "Scene text detection and recognition: the deep learning era," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 161–184, Jan. 2021, doi: 10.1007/s11263-020-01369-0.

[2] G. Pereira and C. Raetzsch, "From banal surveillance to function creep: automated license plate recognition (ALPR) in Denmark," *Surveillance & Society*, vol. 20, no. 3, pp. 265–280, Sep. 2022, doi: 10.24908/ss.v20i3.15000.

[3] F. Naiemi, V. Ghods, and H. Khalesi, "Scene text detection and recognition: a survey," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 20255-20290, Jun. 2022, doi: 10.1007/s11042-022-12693-7.

[4] A. Mishra, J. Sikdar, and S. U. Kumar, "Deep learning-based optical character recognition for robust real-world conditions: a comparative analysis," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Jun. 2024, pp. 1–7, doi: 10.1109/icccnt61001.2024.10726007.

[5] P. Sharma, "Advancements in OCR: a deep learning algorithm for enhanced text recognition," *International journal of inventive engineering and sciences*, vol. 10, no. 8, pp. 1–7, Aug. 2023, doi: 10.35940/ijies.f4263.0810823.

[6] N. Gupta and A. S. Jalal, "Traditional to transfer learning progression on scene text detection and recognition: a survey," *Artificial Intelligence Review*, vol. 55, no. 4, pp. 3457–3502, Apr. 2022, doi: 10.1007/s10462-021-10091-3.

[7] Y. Shu, W. Zeng, Z. Li, F. Zhao, and Y. Zhou, "Visual text meets low-level vision: a comprehensive survey on visual text processing," 2024, arXiv:2402.03082.

[8] G. P. Thuraka, L. Sumalatha, and B. Sujatha, "Deep learning model MORAN architecture for text recognition in complex images," *i-manager's Journal on Information Technology*, vol. 13, no. 3, 2024, doi: 10.26634/jit.13.3.21202.

[9] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/s11042-022-13644-y.

[10] A. Vijayakumar and S. Vairavasundaram, "YOLO-based object detection models: a review and its applications," *Multimedia Tools and Applications*, vol. 83, no. 35, pp. 83535–83574, Oct. 2024, doi: 10.1007/s11042-024-18872-y.

[11] Y. Wang, H. Mamat, X. Xu, A. Aysa, and K. Ubul, "Scene Uyghur text detection based on fine-grained feature representation," *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124372.

[12] Y. Gao, H. Lu, S. Mu, and S. Xu, "GroupPlate: toward multi-category license plate recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 5, pp. 5586–5599, May 2023, doi: 10.1109/TITS.2023.3244827.

[13] Q. Tang, Y. Lee, and H. Jung, "The industrial application of artificial intelligence-based optical character recognition in modern manufacturing innovations," *Sustainability*, vol. 16, no. 5, Mar. 2024, doi: 10.3390/su16052161.

[14] D. Singh, O. A. Shah, and S. Arora, "Adaptive control strategies for effective integration of solar power into smart grids using reinforcement learning," *Energy Storage and Saving*, vol. 3, no. 4, pp. 327–340, Dec. 2024, doi: 10.1016/j.enss.2024.08.002.

[15] K. N. A. Prethi, S. Palanisamy, S. Nithya, and A. O. Salau, "Edge based intelligent secured vehicle filtering and tracking system using YOLO and EasyOCR," *International Journal of Intelligent Transportation Systems Research*, vol. 23, no. 1, pp. 330–353, Apr. 2025, doi: 10.1007/s13177-024-00452-x.

[16] N. K. Shine, G. Bhutani, T. S. Keerthana, and G. Rohith, "An approach for improving optical character recognition using contrast enhancement technique," *Journal of Physics*: Conference Series, vol. 2466, no. 1, Mar. 2023, doi: 10.1088/1742-6596/2466/1/012009.

[17] W. Cavalcante, I. G. Torné, L. Camelo, R. Fernandes, A. Printes, and H. Bragança, "An ID badge information extractor based on object detection and optical character recognition," *IEEE Access*, vol. 12, pp. 152559–152567, 2024, doi: 10.1109/ACCESS.2024.3471449.

[18] A. Biró, S. M. Szilágyi, and L. Szilágyi, "Optimal training dataset preparation for AI-supported multilanguage real-time OCRs using visual methods," *Applied Sciences*, vol. 13, no. 24, Dec. 2023, doi: 10.3390/app132413107.

[19] T. Alkhalifah, H. Wang, and O. Ovcharenko, "MLReal: Bridging the gap between training on synthetic data and real data applications in machine learning," *Artificial Intelligence in Geosciences*, vol. 3, pp. 101–114, Dec. 2022, doi: 10.1016/j.aiig.2022.09.002.

[20] X. Zhou *et al.*, "EAST: an efficient and accurate scene text detector," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2642–2651, doi: 10.1109/CVPR.2017.283.
[21] J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, and X. Bai, "SegLink++: detecting dense and arbitrary-shaped scene text by instance-aware component grouping," *Pattern Recognition*, vol. 96, Dec. 2019, doi: 10.1016/j.patcog.2019.06.020.
[22] Z. Raisi, G. Younes, and J. Zelek, "Arbitrary shape text detection using transformers," in *2022 26th International Conference on Pattern Recognition (ICPR)*, Aug. 2022, pp. 3238–3245, doi: 10.1109/ICPR56361.2022.9956488.
[23] Z. Raisi and J. Zelek, "Occluded text detection and recognition in the wild," in *2022 19th Conference on Robots and Vision (CRV)*, May 2022, pp. 140–150, doi: 10.1109/CRV55824.2022.00026.
[24] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," Jun. 07, 2018, *:1806.02559*.
[25] A. Yadav, S. Singh, M. Siddique, N. Mehta, and A. Kotangale, "OCR using CRNN: a deep learning approach for text recognition," in *2023 4th International Conference for Emerging Technology (INCET)*, May 2023, pp. 1–6, doi: 10.1109/INCET57972.2023.10170436.
[26] X. Wang, S. Zheng, C. Zhang, R. Li, and L. Gui, "R-YOLO: a real-time text detector for natural scenes with arbitrary rotation," *Sensors*, vol. 21, no. 3, Jan. 2021, doi: 10.3390/s21030888.
[27] M.-T. Tran and G.-S. Lee, "Occluded scene text detection via context-awareness from sketch-level image representations," *Multimedia Systems*, vol. 31, no. 3, Apr. 2025, doi: 10.1007/s00530-025-01782-w.
[28] U. Pal, A. Halder, P. Shivakumara, and M. Blumenstein, "A comprehensive review on text detection and recognition in scene images," *Artificial Intelligence and Applications*, vol. 2, no. 4, pp. 229–249, Oct. 2024, doi: 10.47852/bonviewAIA42022755.
[29] D. L. Freire, A. C. P. L. F. D. Carvalho, A. J. Peterlevitz, M. A. Chinelatto, R. D. D. Silva, and J. F. R. Perea, "A methodology for automated conversion of axis-aligned to polygonal and oriented bounding box annotations in aerial imagery object detection," in *Intelligent Data Engineering and Automated Learning – IDEAL 2024*, Cham, Switzerland: Springer Nature, 2025, pp. 373–385, doi: 10.1007/978-3-031-77738-7_31.
[30] P. V. Hegde, M. R. Ahmed, and A. H. Nalband, "Memory-efficient annotation techniques for autonomous drone navigation," *IEEE Access*, vol. 13, pp. 130846–130871, 2025, doi: 10.1109/ACCESS.2025.3590402.
[31] I. Ulku and E. Akagündüz, "A survey on deep learning-based architectures for semantic segmentation on 2D images," *Applied Artificial Intelligence*, vol. 36, no. 1, Dec. 2022, doi: 10.1080/08839514.2022.2032924.
[32] S. A. Magalhães *et al.*, "Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse," *Sensors*, vol. 21, no. 10, May 2021, doi: 10.3390/s21103569.
[33] S. K. Dasari and S. Mehta, "Scene based text recognition from natural images and classification based on hybrid CNN models with performance evaluation," *International Journal of Electrical and Computer Engineering Systems*, vol. 14, no. 3, pp. 293–300, Mar. 2023, doi: 10.32985/ijeces.14.3.7.
[34] M. Gandomkar and S. Khoramipour, "Towards omni-font optical character recognition (OCR) for persian script using the YOLO object detection model," *Majlesi Journal of Electrical Engineering*, pp. 1-16, 2005, doi: 10.57647/j.mjee.2025.17063.
[35] M. Moustapha, M. Tasyurek, and C. Ozturk, "A novel YOLOv5 deep learning model for handwriting detection and recognition," *International Journal on Artificial Intelligence Tools*, vol. 32, no. 4, Jun. 2023, doi: 10.1142/S0218213023500161.
[36] J. Adamsson and V. Honar "OCRchive: an alphanumerical detection and recognition framework for historical document image analysis," M.Sc. thesis, Department of Computer Science, Blekinge Institute of Technology, Karlskrona, Sweden, 2025.
[37] T. Mazumder, F. Nusrat, A. B. S. Mahi, J. B. Brishty, R. Rahman, and T. Helaly, "Automated and efficient Bangla signboard detection, text extraction, and novel categorization method for underrepresented languages in smart cities," *Results in Engineering*, vol. 26, Jun. 2025, doi: 10.1016/j.rineng.2025.105156.

BIOGRAPHIES OF AUTHORS

Rajeswari Reddy Patil is an assistant professor at Rao Bahadur Y. Mahabaleswarappa Engg College, Bellary, Karnataka, India. She has 16 years of teaching experience. She has done bachelor's degree in Computer Science and Engineering from Gulbarga University and a master's in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi. Currently pursing Ph.D at Visvesvaraya Technological University, Belagavi. Her key areas of interest are image processing, machine learning, and computer vision. She is a member of IETE. She mainly works on scene text-related images and has published papers in international journals and conferences. She can be contacted at email: rajeswarirp@rymec.in.

Dr. Aradhana Dammergidda completed Ph.D. in Image Processing area from JNTU, Hyderabad, India and presently working as a professor and head at Department of Computer Science and Engineering (Data Science), Ballari Institute of Technology and Management, Ballari, Karnataka, India. Her specialized areas of research are image processing, machine learning, neural networks, computer vision, data mining. She has 25 years of teaching experience and 8 years of research experience. She is committed to foster an innovative learning environment, advancing research initiatives, and enhancing industry collaboration. She is life time member of Indian Society for Technical Education (ISTE) and member of computer society of India. She has presented and published several research papers in international conferences and journals. She can be contacted at email: aradhanabm@gmail.com.