Performance Evaluation of Efficient and Accurate Text Detection and Recognition in Natural Scenes Images Using EAST and OCR Fusion

Vishnu Kant Soni[1]*, Vivek Shukla[2], S. R. Tandan[3], Amit Pimpalkar[4], Neetesh Kumar Nema[5], Muskan Naik[6]
Department of Computer Science and Engineering, Dr. C. V. Raman University, Bilaspur, C.G. India[1, 2, 5]
Department of Computer Science, Government R. V. R. S. Kanya Mahavidyalaya, Kawardha, C.G. India[3]
Department of Computer Science and Engineering (AIML)-Shri Ramdeobaba College of Engineering and Management, Ramdeobaba University, Nagpur, India[4]
Department of Computer Science and Engineering, Lakhmi Chand Institute of Technology, Bilaspur, C.G. India[6]

Abstract*—Scene texts refer to arbitrary text found in images captured by cameras in real-world settings. The tasks of text detection and recognition are critical components of computer vision, with applications spanning scene understanding, information retrieval, robotics, and autonomous driving. Despite significant advancements in deep learning methods, achieving accurate text detection and recognition in complex images remains a formidable challenge for robust real-world applications. Several factors contribute to these challenges. First, the diversity of text shapes, fonts, colors, and styles complicates detection efforts. Second, the myriad combinations of characters, often with unstable attributes, make complete detection difficult, especially when background interruptions obscure character strokes and shapes. Finally, effective coordination of multiple sub-tasks in end-to-end learning is essential for success. This research aimed to tackle these challenges by enhancing text discriminative representation. This study focused on two interconnected problems: Scene Text Recognition (STR), which involves recognizing text from scene images, and Scene Text Detection (STD), which entails simultaneously detecting and recognizing multiple texts within those images. This research focuses on implementing and evaluating the Efficient and Accurate Scene Text Detector (EAST) algorithm for text detection and recognition in natural scene images. The study aims to compare the performance of three prominent Optical Character Recognition (OCR) techniques—TesseractOCR, PaddleOCR, and EasyOCR. The EAST model was applied to a series of sample test images, and the results were visually represented with bounding boxes highlighting the detected text regions. The inference times for each image were recorded, highlighting the algorithm's efficiency, with average times of 0.446, 0.439, and 0.440 seconds for the respective test images. These results indicate that the EAST algorithm is accurate and operates in real-time, making it suitable for applications requiring immediate text recognition.

Keywords*—Scene text recognition; optical character recognition; deep learning; feature extraction; scene text detection

I. Introduction

Smartphones' widespread adoption has revolutionized how we capture and share images. With their ease of use and quick accessibility, smartphones have led to an exponential growth in the amount of multimedia data available on the web. From advertisements and holiday pictures to business cards and newspaper articles, these devices have made digitizing content a common practice. However, this abundance of data has also presented new challenges [1-2].

Natural scenes, characterized by diverse backgrounds, lighting conditions, and complex visual elements, are particularly challenging for computers to analyze and understand. Segmenting and extracting text from these scenes is crucial due to the practical value of embedded textual information. Text extraction enables humans and computers to interpret and utilize this data for various applications, such as document analysis, license plate recognition, and product identification. It enhances automation and efficiency in diverse domains, offering several advantages in real-time scenarios. In autonomous vehicles, efficient text extraction enables the recognition of road signs, enhancing navigation and safety. In retail environments, it aids in product identification and inventory management, streamlining operations, and improving customer service. Text extraction automates scanning and digitization processes in document analysis, increasing productivity and accuracy. Real-time text extraction provides a competitive edge in various industries, such as healthcare, where it can assist in patient data analysis and diagnosis, leading to faster and more accurate decisions. In finance, it enhances fraud detection and document processing, improving security and operational efficiency; digital forensics aids in analyzing textual information from crime scenes, supporting investigations, and collecting evidence [3].

This manuscript explores text detection approaches to address the challenges of mining and retrieving weakly structured content in scene images. By utilizing models like EAST and integrating OCR techniques, the research aims to develop the next generation of search engines capable of accurately identifying and reading text in diverse environments. Overcoming the limitations of current models is crucial for enabling machines to understand and interact with the world, ultimately driving advancements in applications such as autonomous driving, augmented reality, and content retrieval. The segmentation and extraction of text from natural scenes are pivotal for unlocking valuable information embedded in visual content. By enabling real-time text

*Corresponding Author

extraction, businesses, and industries can utilize this data for enhanced decision-making, automation of processes, and improved efficiency across a wide range of applications, underscoring the critical role of text detection and recognition technologies in modern-day scenarios.

The following are the novelties of the research:

1) Real-time performance evaluation: The research highlights the EAST algorithm's efficiency, demonstrating low inference times for text recognition, making it suitable for real-time applications.

2) Integration of multiple OCR techniques: The study uniquely combines TesseractOCR, PaddleOCR, and EasyOCR with the EAST algorithm, providing a comprehensive comparison of their performance in STD.

3) Visual validation of results: Using bounding boxes to represent detected text visually enhances the understanding of the algorithm's accuracy and effectiveness.

The remainder of the paper was structured to provide a comprehensive overview of the research in Section II. Section III presented a detailed description of the proposed scheme, outlining its methodologies and innovations. In Section IV, the authors showcased and analyzed the experimental results, highlighting the performance and effectiveness of their approach. This section engaged in a thoughtful discussion of the findings, considering their implications and potential applications. Finally, Section V offered a conclusion, summarizing the key contributions of the study and suggesting directions for future research in the field.

II. Related Work

In recent years, rapid advancements in deep learning have revolutionized the field of STD. Researchers have proposed a flurry of novel algorithms based on neural networks, each making significant strides in this domain. By utilizing the power of convolutional neural networks (CNNs), these methods have automated learning text features, eliminating the need for manual feature engineering. This breakthrough has propelled STD technology to new heights [4]. Numerous researchers have explored various techniques for detecting text in images, contributing significantly to advancements in this field. Some investigators concentrated on texture-based approaches, utilizing the sliding window concept to identify and analyze unique textural features within input images. This method effectively localizes text information by examining patterns that distinguish text from the surrounding background. Other researchers focused on sparse-based text detection methods, which have proven beneficial for various computer vision applications. These techniques leverage sparse representations to enhance text detection, particularly in challenging environments where traditional methods may struggle. By employing these innovative approaches, researchers aimed to improve the accuracy and reliability of text detection systems, paving the way for more robust applications in real-world scenarios [5].

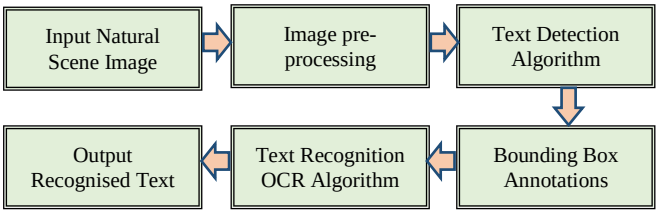

Fig. 1. Pipeline of text detection and extraction.

The pipeline, as illustrated in Fig. 1, consists of six key steps: (1) Input Natural Scene Image, (2) Image pre-processing, (3) Text Detection Algorithm (EAST), (4) Bounding Box Annotations, (5) Text Recognition OCR Algorithm, and (6) Output Recognized Text. By applying this comprehensive approach, the research aims to achieve accurate and efficient text detection and recognition in real-world scenarios, contributing to advancing intelligent systems capable of understanding and interacting with textual information in diverse environments.

Current deep learning-based STD approaches can be broadly categorized into two main groups: regression-based methods and segmentation-based methods. Regression-based techniques typically employ CNNs to directly predict text regions' bounding boxes or coordinates. These models learn to map input images to predefined anchors or text proposals, refined and filtered to obtain the final text detections. One notable example of a regression-based method is TextBoxes, which adapts the Single Shot MultiBox Detector architecture for STD, achieving real-time performance while maintaining high accuracy. On the other hand, segmentation-based methods treat text detection as a pixel-wise classification problem [6]. These algorithms divide the input image into a grid of cells and predict whether each cell contains text. By leveraging the inherent strengths of CNNs in semantic segmentation, segmentation-based approaches can handle text instances of arbitrary shapes and orientations. A prominent example is the EAST, which employs a fully convolutional network (FCN) to generate a score map and geometry of text boxes, enabling text detection in various orientations and scales [7-8]. Both regression-based and segmentation-based methods have their advantages and disadvantages. Regression-based techniques often excel in computational efficiency, making them suitable for real-time applications. However, they may struggle with detecting text instances of complex shapes or orientations. Segmentation-based methods, on the other hand, demonstrate superior performance in handling diverse text geometries but may require more computational resources. Despite the remarkable progress made by deep learning-based STD algorithms, challenges remain. Factors such as complex backgrounds, varying lighting conditions, and text distortions can still hinder the accuracy of these models. Ongoing research efforts aim to address these limitations and further enhance the robustness and applicability of STD systems in real-world scenarios [9].

STD was recognized as a complex and challenging task due to various environmental factors, including illumination, lighting conditions, and the presence of small or curved text. Many existing approaches prioritized model accuracy and efficiency but resulted in heavy-weight models requiring

substantial processing resources. STR emerged as a prominent research area in computer vision, focusing on recognizing text in natural scenes. Researchers noted that attention-based encoder-decoder frameworks struggled with attention drift, which hindered the precise alignment of feature regions with target objects in complex, low-quality images. Additionally, the rise of Transformer models led to increased computational costs due to their larger parameter sizes. X. Luan et al. [10] developed the lightweight STR model to address these issues, incorporating a position-enhancement branch to alleviate attention drift and dynamically fuse position with visual information. Experimental results indicated that lightweight STR achieved a 3% higher average recognition accuracy than baseline models while maintaining a lightweight structure with only seven million parameters. This balance of accuracy, speed, and computational efficiency made lightweight STR suitable for high-demand applications in STR, outperforming existing methods.

Researchers in [11-12] developed a novel lightweight model to enhance the accuracy and efficiency of STD. This model utilized ResNet50 and MobileNetV2 as backbones, incorporating quantization techniques to reduce size. During quantization, the precision was adjusted from float32 to float16 and int8, resulting in a more lightweight model. The proposed method significantly outperformed state-of-the-art techniques, improving inference time and Floating-Point Operations Per Second (FLOPS) by approximately 30 to 100 times. The researchers used well-known datasets, ICDAR2015 and ICDAR2019, to validate the model's performance, and they included samples in ten different languages. The model demonstrated a balance of accuracy and efficiency, achieving word % accuracy rates of 62% for complex text and 80% for non-complex text and character accuracy rates of 68% and 88%, respectively. R. Harizi et al. [13] study introduced a hybrid scene text detector that combined selective search with SIFT-based key point density analysis and a deep learning training architecture. The researchers investigated key SIFT points to identify crucial image areas for precise word localization. They then fine-tuned these regions using a deep learning-powered bounding box regressor, which ensured accurate word boundary alignment and enhanced detection efficiency. The study focused on detecting text in real-world scene images. They proposed a method that integrated SIFT-based key point localization, Bag of Words-based character pattern filtering, and ResNet-19-based word bounding box regression. Experimental results confirmed the method's effectiveness in addressing multi-oriented and curved scene texts.

In their paper, G. Liao et al. [14] significantly contributed to STD. They designed a Multi-Pooling Module (MPM) with different pooling operations to address the limitations of the original PSENet. The MPM effectively captured the relevance of text information at various distances, enabling precise localization of scene text regions. Y. Cai et al. [15] proposed a style-aware learning network to achieve style-robust text detection in diverse environments. M. Lu et al. [16] addressed the existing model's deficiencies in detecting long text regions by altering the shrinkage calculation, adding a feature enhancement module, and changing the loss function to Focal loss. S. Yuchen et al. [17] proposed a novel parameterized text shape method based on low-rank approximation, distinguishing their approach from other shape representation methods that relied on data-irrelevant parameterization. They utilized singular value decomposition to reconstruct text shapes using a limited number of eigenvectors derived from labeled text contours.

In a study, M. Aluri et al. [18] developed an innovative method for identifying irregular text in natural scene images. The approach combined a U-net architecture with connected component analysis, significantly improving text component detection accuracy while reducing non-character element identification. Furthermore, the researchers incorporated graph convolution networks to infer adjacency relations among text components, introducing a sophisticated mechanism that advanced text detection in natural scene images. In their novel approach, H. Chen et al. [19] developed the Fragmented Affinity Reasoning Network of Text Instances, a component connection method for arbitrary shape text detection. The network consisted of three key modules: the Weighted Feature Fusion Pyramid Network (WFFPN), Text Fragments Subgraph (TFS), and Dense Graph Attention Network (DGAT), which could be trained end-to-end. The researchers introduced WFFPN to generate text fragments, while TFS and DGAT jointly constructed an affinity reasoning network. Their contributions included proposing a novel unified end-to-end trainable framework, developing a simple and effective WFFPN for multi-scale feature representation and processing, and introducing the joint module of TFS and DGAT to infer the link relationship between text fragments, improving the grouping performance of dense and long curved text.

In their work, Y. Zhu et al. [20] proposed a novel STD method called Text Mountain. The core concept of Text Mountain utilized border-center information differently than previous approaches, which treated center-border as a binary classification problem. Instead, they predicted text center-border probability (TCBP) and text center-direction (TCD). The TCBP resembled a mountain, with the peak representing the text center and the base indicating the text border, allowing for better separation of text instances. This method proved robust against multi-oriented and curved text due to its effective labeling rules. During inference, each pixel at the mountain base searched for a path to the peak, enabling efficient parallel processing. Experiments on various datasets, including MLT and ICDAR2015, demonstrated that Text Mountain achieved superior performance, notably an F-measure of 76.85% on MLT, surpassing previous methods significantly.

Current STD models encounter limitations that impact their effectiveness in real-world applications, mainly when dealing with scene text images and born-digital documents. These categories present unique challenges compared to traditional scanned paper documents. One significant difficulty is the presence of cluttered backgrounds. Existing models often struggle to accurately identify text amidst various visual elements, which can lead to false positives or missed detections. Additionally, traditional models typically use rigid geometrical shapes, like axis-aligned rectangles, making them less effective for detecting free-form text, such as curved or

rotated characters commonly found in natural environments. While some models attempt to manage variations in text size through multi-scale feature maps, this approach can be complex and computationally demanding. The need for elaborate post-processing steps can slow down detection and complicate model architecture. Lighting conditions also play a significant role, as many models perform well under controlled environments but falter in outdoor or dynamically lit situations [21]. Finally, balancing detection accuracy and real-time processing speed remains a critical challenge. Many advanced models sacrifice speed for improved accuracy, rendering them unsuitable for applications that require immediate results. Addressing these limitations is vital for enhancing the robustness and applicability of text detection systems.

III. Proposed Methodology

Text detection involves predicting and localizing text instances within images. While traditional image processing techniques were commonly used for this task, deep learning models consistently outperformed them across various real-world scenarios, from simple to highly complex environments. The localization of text using deep learning could be achieved primarily through two approaches: object detection and image segmentation. Object detection methods focused on identifying and bounding text regions, providing a straightforward way to localize text. In contrast, image segmentation treated text detection as a pixel-wise classification task, allowing for more precise delineation of text shapes. Each approach had advantages and challenges concerning dataset creation, model training, and inference options. The advancements in deep learning significantly enhanced the effectiveness of text detection in various applications. Object detection techniques localize objects within an image by drawing rectangular or square bounding boxes around them. While effective, this method provides limited information about the actual shape of the detected objects. Fortunately, labeling images for object detection is a relatively straightforward process compared to segmentation. Segmentation, [22] conversely, involves classifying each pixel in an image into predefined categories.

Segmentation would entail distinguishing between text and non-text pixels in scene detection. This pixel-wise classification allows for identifying text regions with greater precision, even if they exhibit complex shapes or orientations. For character recognition tasks, the annotation process becomes even more granular. Each pixel is classified as belonging to one of the available character classes, enabling the precise identification of individual letters or symbols within the detected text regions. This process can be visualized in Fig. 2.

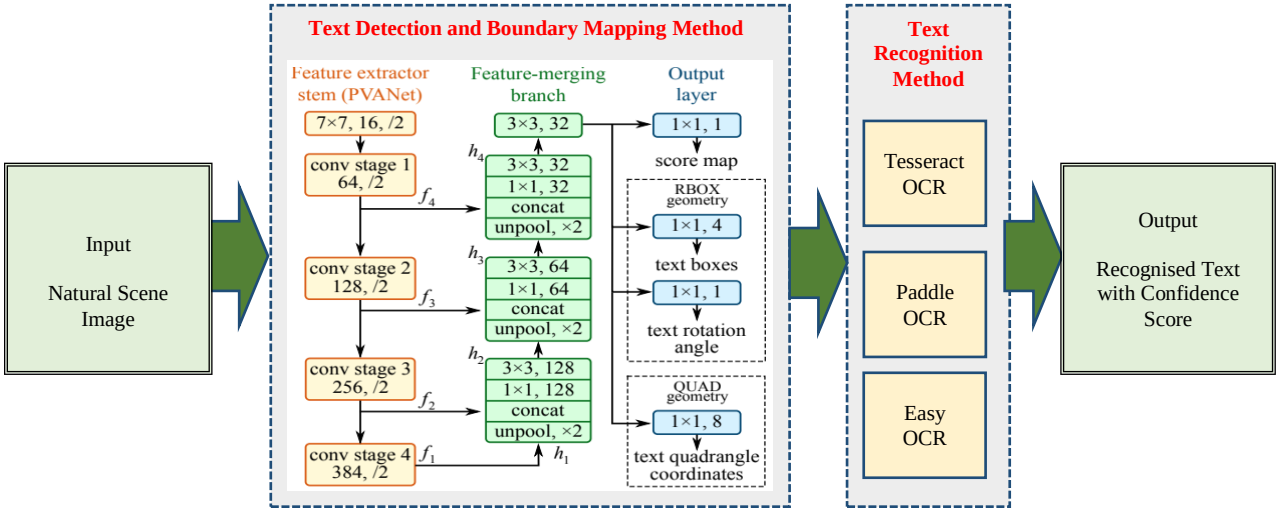

Fig. 2. The structure of the EAST text detection fully convolutional network.

The EAST algorithm was explicitly developed [23] to address the challenges of text detection in natural scenes, where text can appear in diverse sizes, orientations, and perspectives. The EAST architecture was designed to handle text regions of varying sizes efficiently. The key idea was to leverage features from different neural network stages: later stages for detecting large and initial stages for small word regions. The authors employed three interconnected branches within a single neural network. The fundamental principles underlying EAST's functionality include several innovative components. The Feature Extractor Stem was responsible for extracting features from various network layers. This stem could be a convolutional network pre-trained on the ImageNet dataset, such as PVANet, VGG16, and Resnet V1-50—the model, taking outputs from the pooling layers. This network is typically pre-trained on extensive datasets and subsequently fine-tuned for the specific task of text detection, allowing it to learn the unique characteristics of text in various contexts effectively. The Feature Merging Branch combined the feature outputs from different VGG16 layers and can be expressed in Eq. (1) and Eq. (2).

$$gi = \begin{cases} unpool(h_i) & if \quad i \leq 3 \\ conv_{3X3}(h_i) & if \quad i = 4 \end{cases} \tag{1}$$

$$hi = \begin{cases} f_i & if \quad i = 1 \\ conv_{3X3}(conv_{1X1}[g_{i-1}; f_i]) & otherwise \end{cases} \tag{2}$$

EAST utilized a U-Net-like architecture to merge the feature maps to avoid computational complexity gradually. The process involved upsampling the $pool_{n-1}$ layer output to match the size of the $pool_n$ layer output, concatenating them, and applying convolutional layers to fuse the information. This

procedure was repeated for the remaining layers, ultimately producing a final feature map layer before the output layer. EAST employs anchors and a Region Proposal Network (RPN) to propose potential text regions. However, it customizes the RPN to predict axis-aligned quadrilaterals instead of traditional rectangles, enabling it to enclose text regions more accurately and tightly. The Output Layer consisted of two key components: a score map and a geometry map. The score map indicated the probability of text in each region, while the geometry map defined the boundaries of the text boxes. EAST offered two options for the geometry map: rotated boxes (specified by top-left coordinate, width, height, and rotation angle) or quadrangles (all four coordinates of a rectangle). EAST predicts the coordinates of the four vertices of each quadrilateral bounding a text region, along with a confidence score that indicates the likelihood of text presence. This capability allows the algorithm to manage text in arbitrary orientations and shapes, enhancing its versatility in real-world applications.

In text detection, bounding box annotations mark the regions in images where text appears. These annotations help train the EAST algorithm to recognize and locate text in various scenes. For instance, each bounding box outlines the area containing text, which the algorithm learns to identify. The process of bounding box annotations for text regions using the EAST algorithm involved several vital steps that aimed to enhance the accuracy of text detection in images. Initially, the EAST algorithm utilized an FCN to analyze input images and generate a score map, indicating the likelihood of text presence across different image areas. The EAST algorithm first predicted the geometry of potential text regions to create bounding box annotations. This was achieved by estimating four parameters for each pixel in the score map: the bounding box's height and width and the center coordinates. The model could effectively capture the spatial characteristics of text instances in various orientations and scales by employing a regression approach. The EAST text detector model generated two key outputs: scores, which represented the probabilities of positive text regions, and geometry, which provided the bounding boxes for these text regions. These outputs served as parameters for the decode prediction's function, which processed the input data. The function returned a tuple containing the bounding box locations of the detected text and their corresponding probabilities. The bounding boxes, referred to as "reacts," were formatted compactly for efficient application of Non-Maximum Suppression (NMS), while the "confidences" represented the confidence values associated with each bounding box. Once the score map and geometry predictions were generated, the next step involved applying NMS to filter out overlapping bounding boxes. This technique helped to eliminate redundant detections, ensuring that only the most confident predictions remained.

The NMS algorithm selected the bounding box with the highest score and removed any boxes with significant overlap based on a predefined threshold. As an FCN, EAST outputs per-pixel predictions of words or text lines and utilizes NMS as a post-processing step on the geometric map. This geometric map can be RBOX (with four channels for bounding box coordinates and one for text rotation) or QUAD (with eight channels representing shifts from the four corner vertices). EAST employs a weighted sum of losses for both the score map and the geometry, ensuring adequate training. The resulting bounding boxes were then refined to improve their accuracy. This included adjusting the boxes' dimensions to fit better the actual text regions detected in the image. The final output consisted of well-defined bounding boxes that accurately represented the locations of text instances. The EAST algorithm's approach to bounding box annotations combined advanced deep learning techniques with effective post-processing methods, resulting in a robust framework for detecting text regions in natural scenes. This process significantly improved the performance of STD, making it a valuable tool for various applications, such as document analysis and autonomous navigation [24]. By integrating these three branches, the EAST architecture effectively handled text regions of varying sizes and shapes, making it a powerful tool for STD. The author's innovative approach to feature extraction and merging, combined with the informative output layers, contributed to EAST's efficiency and accuracy in detecting text in complex scenes. EAST optimizes its performance by minimizing two key loss functions during training: the classification loss, which determines the presence of text, and the regression loss, which refines the predicted text regions.

The classification loss can be expressed as in Eq. (3).

$$L_{cls} = -\frac{1}{N_{cls}}\sum_{i=1}^{N_{cls}}[g_i \log(p_i) + (1 + g_i)\log(1 + p_i)] \quad (3)$$

Where N_{cls} denotes the number of anchor regions used for classification. The classification loss in EAST measures the model's ability to distinguish text regions from non-text areas. It is calculated using cross-entropy loss, where p_i represents the predicted probability of the i-th region containing text and g_i is the ground truth label (1 for text, 0 for non-text). Minimizing this loss helps the model accurately classify text regions.

The regression loss can be expressed as in the Eq. (4).

$$L_{reg} = \frac{1}{N_{reg}}\sum_{i=1}^{N_{reg}} Smooth_{L1}(d_i - g_i) \quad (4)$$

Here N_{reg} represents the number of anchor regions used for regression and $Smooth_{L1}$ is the smoothing loss function. EAST employs a regression loss to evaluate how accurately the network predicts the quadrilateral coordinates of text regions. It utilizes $Smooth_{L1}$ loss, which compares the predicted geometry parameters. For each region, such as the distances from the anchor point to the four vertices of the quadrilateral, with the ground truth geometry parameters. This loss function ensures the network learns to generate tight, accurate bounding boxes around text areas, enabling precise text detection.

EAST-OCR Fusion Algorithm

Input: Natural Scene Image (I) from ICDAR 2013, ICDAR 2015, COCO-Text

Output: Recognized Text String (T), Confidence Score (C)

Step-I: Pre-processing (Pre-process (I))

i. I = resize(I, (224, 224))

ii. *Grayscale Conversion:* I_{gray} *= rgb2gray(I)*
iii. *Noise Remove:* $I_{denoised}$ *= median_filter(*I_{gray}*)*
iv. *Normalization:*

$$I_{norm} = \frac{(I\_denoised(score) - min(I\_denoised(score))}{max(I\_denoised(score)) - min(I\_denoised(score))}$$

Step-II: Text Detection ($Text_{Regions}$ = DetectText (I_{norm}))

i. *Text RegionDetection: Apply the EAST algorithm to detect text regions in* I_{norm}
ii. *Bounding Box Extraction:*
a. *Calculate confidence (D), coordinates (C), and rotation angle (θ) using a 1D vector:*
- *1D vector = Conv1D(output)*

b. *Use Non-Maximum Suppression (NMS) to refine bounding boxes:*
- *final bounding box =* $NMS(start_x, start_y, end_x, end_y)$

Step-III: Text Segmentation (Segmentation ($I_{norm}, Text_{Regions}$))

i. *Check for Detected Regions:*
a. *If* $Text_{Regions}$ *is empty:*
- *Segment* I_{norm} *into individual characters using connected component analysis.*

b. *Else*
- *Segment each region in* $Text_{Regions}$ *into individual characters.*

ii. *Apply heuristics filtering to discard non-text regions based on size and aspect ratio.*

Step-IV: Feature Extraction (Features = FeatureExtractor ($Character_{Images}$))

i. *For each character image (c):*
a. *Extract features (*f_c*):*
- *HOG features:* f_c *= hogfeature(c)*
- *Binary Image:* f_c *= imbinarize(c)*

b. *Return a list of features (Features) for all characters*

Step-V: Text Recognition with OCR Methods ($Text_{Sequence}$, $Confidence_{Score}$ =$OCR_{recog(Features)}$)

i. *Apply an OCR network with an embedding layer, OCR layers, and a softmax output layer.*
ii. *For each feature vector (*f_i*) in Features:*
a. *Predict character probability distribution using* $p(c|f_i)$ *= softmax (OCR(*f_i*)))*
b. *Decode the predicted character sequence (*$Text_{Sequence}$*)*
iii. *Calculate the confidence score (C) where*C_{ij} *represents the probability of character j being at position* $_i$ *in the sequence.*

Step-VI: Post-processing ($Text_{Refined}$ = Postprocess ($Text_{Sequence}$))

i. *Implement proofreading steps to enhance text quality, including spell-checking*

Step-VII: Output Display (Display ($Text_{Refined}$, C))

The EAST-OCR fusion algorithm for text detection and recognition in natural scene images follows a structured approach. It begins with pre-processing the input image, which includes resizing, grayscale conversion, noise removal, and normalization. Next, the EAST algorithm detects text regions, calculating confidence scores, coordinates, and rotation angles while applying NMS to refine bounding boxes. Text segmentation is performed based on detected regions, followed by feature extraction from individual character images. The extracted features are then processed through an OCR network to recognize the text and compute confidence scores. Finally, post-processing steps enhance text quality, and the results, including the recognized text and confidence scores, are displayed to the user. The algorithm outlines a structured approach, ensuring clarity and comprehensiveness in each step.

TABLE I. DATASET STATISTICS

Parameter	Value
Dataset Names	ICDAR 2013, ICDAR 2015, COCO-Text
Total Images	65,598
Total Bounding Boxes	5,000
Average Bounding Boxes per Image	5
Total Text Instances	1,50,359
Text Instances Categories	machine-printed and handwritten text
Text Instances Language Categories	English script and non-English script
Training Set Size	70%
Validation Set Size	15%
Testing Set Size	15%

The EAST model was primarily trained using ICDAR 2013, ICDAR 2015 and COCO-Text datasets, which provided various text instances for effective learning. This dataset's statistics can be seen in Table I. Additionally, the model utilized the ResNet V1-50 architecture, sourced from Tensor Flow, instead of the alternative PVANet, to enhance feature extraction capabilities. For optimization, we opted for loss, which focuses on maximizing the Intersection over Union (IoU) of segmentation rather than using balanced cross-entropy loss. Furthermore, a linear learning rate decay strategy was implemented instead of a staged learning rate decay approach, allowing for smoother convergence during training. These choices contributed to the model's improved performance in detecting text in natural scenes. The dataset comprised 4,500 unique text instances, offering diverse content that enhances the model's learning experience. The dataset statistics for bounding box annotations used in training the EAST algorithm were meticulously compiled to enhance the model's ability to detect text in natural scenes. The dataset included images with diverse text instances annotated with bounding boxes to indicate the precise locations of text regions.

Despite its complexity and the significant computational resources required for implementation, EAST has proven to be a powerful tool for various applications, including OCR, text recognition, and image information extraction. Ultimately, EAST's ability to accurately and efficiently locate and interpret text within images has established it as a crucial component in

computer vision and OCR. Its contributions have significantly advanced the development of applications capable of understanding and processing textual information in the world around us. Following the implementation of the text detection and boundary mapping method, the next crucial step in this research was the actual detection of text within the images. Three different OCR techniques were employed: TesseractOCR, PaddleOCR, and EasyOCR. Each method was chosen for its unique advantages, allowing for a comprehensive comparison of their performance in text detection tasks. Tesseract OCR is one of the most widely used OCR engines, known for its robustness and flexibility. It supports multiple languages and has a strong community backing, contributing to its continuous improvement. Tesseract excels in recognizing printed text and has been optimized for various applications, making it a reliable choice for this research.

PaddleOCR is another powerful OCR tool that stands out for its multilingual capabilities and high accuracy. It integrates advanced deep learning techniques to handle complex text scenarios, including curved and multi-oriented text. PaddleOCR is particularly beneficial for tasks requiring high precision in text extraction from natural scenes. EasyOCR is a newer entrant in the OCR landscape, gaining popularity for its simplicity and effectiveness. It supports over 80 languages and is designed to be easy to use. EasyOCR uses deep learning models to achieve impressive text detection and recognition results, particularly in challenging environments [25]. By applying these three OCR techniques, the research aimed to evaluate their effectiveness in detecting text across various scenarios. Each method was assessed based on accuracy, speed, and adaptability to different text orientations and backgrounds. This comparative analysis highlighted the strengths and weaknesses of each OCR tool and provided valuable insights into its suitability for specific text detection tasks. Ultimately, the findings from this research could guide future developments in selecting the most appropriate OCR technology for their needs.

IV. Result and Discussion

After implementing the EAST algorithm on a series of sample test images, the next step was recognizing the text in these images. The results of this process are illustrated in the accompanying Fig. 3: (a1-a3) display the sample testing images. At the same time (b1-b3), the corresponding text detection results are shown, complete with bounding boxes around the detected text regions. The performance of the text recognition was evaluated based on the inference time for each test image, which was recorded as 0.446 seconds for the first image, 0.439 seconds for the second, and 0.440 seconds for the third. These results indicate that the EAST algorithm is highly efficient, demonstrating a low inference time for text recognition across the sample images. This efficiency is particularly noteworthy, as it suggests that the EAST algorithm can effectively detect and recognize text in real-time scenarios, making it suitable for applications where speed is critical. The bounding box in the detection results visually confirms the text recognition's accuracy, showcasing the algorithm's capability to identify text in various contexts.

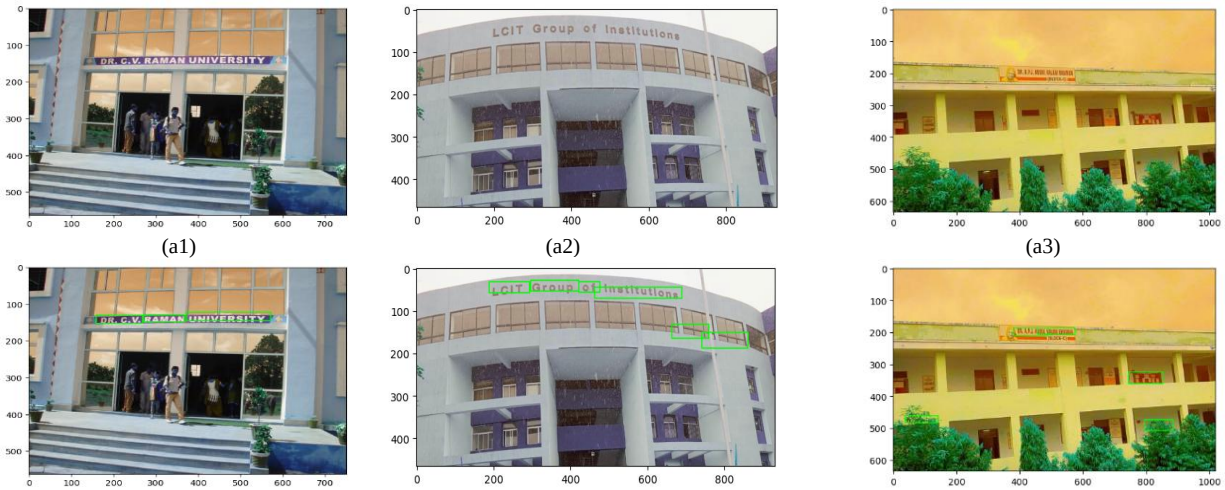

Fig. 3. (a1-a3): Sample testing images, (b1-b3) Text detection results with bounded box.

TABLE II. COMPARISON OF DIFFERENT TEXT RECOGNITION METHODS

Method/ Actual Text	DR C. V. RAMAN UNVERSITY	LCIT Group of Institutions	DR. A.P.J. ABDUL KALAM BHAWAN (BLOCK-C)	Average Confidence Score
Easy OCR	DR CV RAMAN UNVERSITY (Confidence: 0.86)	LCIT (Confidence: 0.98) Group (Confidence: 1.00) 6 f (Confidence: 0.58) Institutlone (Confidence: 0.76)	DR.A,PJ, ABDUL KALAM BHAWAN (Confidence: 0.89) (BLOCK-C) (Confidence: 0.91)	**0.85**
Tesseract OCR	DR C.V. RAMAN UNVERSITY (Confidence: 0.83)	LCIT, (Confidence: 86.00) Gr, (Confidence: 95.00)	DR.A.PJ, ABDUL KALAM BHAWAN (Confidence: 0.87) (BLOCK-C) (Confidence: 0.96)	**0.89**
Paddle OCR	DR.C.V.RAMAN UNIVERSITY (Confidence: 0.96)	LCIT (Confidence: 0.98) Group (Confidence: 1.00) of (Confidence: 0.78) Institution (Confidence: 0.89)	DR.A.P.J.ABOUL KALAM BHAWAN (Confidence: 0.92) BLOCK-C (Confidence: 0.98)	**0.93**

In this work, a comparison was conducted among three prominent text recognition methods: EasyOCR, Tesseract OCR, and PaddleOCR. Each method was evaluated on sample test images to determine their effectiveness in accurately recognizing text. As shown in Table II and Fig. 5, the results revealed average confidence scores of 0.85 for EasyOCR, 0.89 for Tesseract OCR, and an impressive 0.93 for PaddleOCR. These scores indicate that PaddleOCR outperformed the other two methods, demonstrating its superior capability in text recognition tasks. The higher confidence score suggests that PaddleOCR detected text more accurately and effectively handled various text styles and orientations.

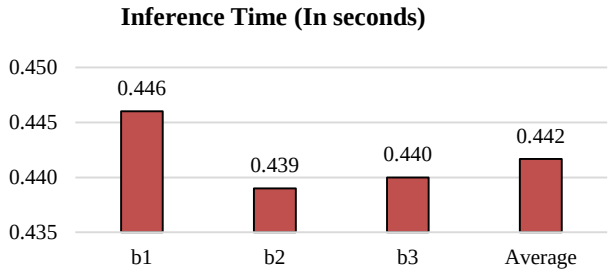

Fig. 4. Comparison between text detection inference times for test images.

While EasyOCR and Tesseract OCR also provided commendable performance, PaddleOCR's results highlight its strengths, particularly in complex scenarios where text may be distorted or presented in challenging conditions.

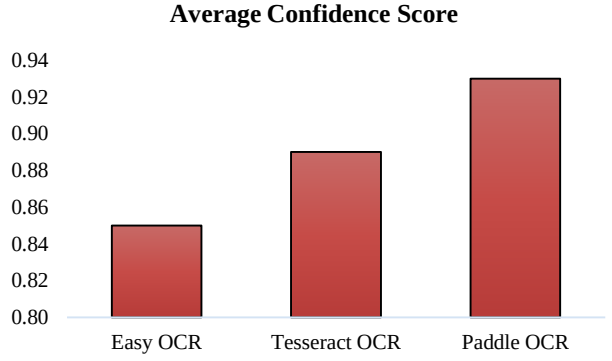

Fig. 5. Comparison of average confident score between different OCR methods for test images.

This comparison underscores the importance of selecting the right OCR tool for specific applications, especially when accuracy is paramount. Overall, PaddleOCR stands out as a robust choice for text recognition, making it an evaluable asset for future projects requiring reliable OCR capabilities. The proposed method utilizing the EAST algorithm for text detection and recognition offers several advantages over previous approaches. Firstly, it streamlines the process by employing a single neural network that directly predicts text instances and their geometries, eliminating the need for time-consuming intermediate steps such as candidate proposal and word partitioning. This end-to-end approach enhances speed, as shown in Fig. 4, and improves accuracy, allowing for near real-time processing of images. The EAST algorithm is also designed to handle text in various orientations and aspect ratios, addressing a standard limitation in traditional OCR methods that struggle with diverse text layouts. By outputting dense per-pixel predictions, EAST provides more precise text region localization than earlier models. Moreover, while previous OCR techniques may falter with underrepresented languages or complex scripts, the EAST framework's flexibility allows for better adaptation to different text types. Integrating advanced OCR methods like Tesseract or PaddleOCR further enhances recognition accuracy, particularly in challenging scenarios. The proposed method effectively resolves speed, accuracy, and adaptability issues found in earlier approaches, making it a robust solution for efficient text detection and recognition in natural scene images.

V. CONCLUSION AND FUTURE SCOPE

Integrating the EAST algorithm with various OCR techniques has demonstrated promising results in enhancing STD and recognition performance. By applying Tesseract OCR, PaddleOCR, and EasyOCR to the sample test images, this research has highlighted the strengths and limitations of each method. The EAST algorithm has proven to be a highly efficient and accurate tool for text detection, as evidenced by the low inference times recorded during the testing process. With an average inference time of less than half a second per image, the EAST algorithm's real-time capabilities make it suitable for applications that require immediate text recognition, such as autonomous vehicles and augmented reality systems. Moreover, the visual representation of the text detection results, showcased through bounding boxes, confirms the accuracy of the EAST algorithm in identifying text regions within the sample images. The comparative analysis of the

OCR techniques revealed distinct strengths and weaknesses. Tesseract OCR demonstrated robustness in recognizing printed text, while PaddleOCR excelled in handling multilingual text and complex layouts. EasyOCR, known for its user-friendly interface, provided quick results with impressive accuracy. The findings underscore the potential of the EAST algorithm as a reliable tool for STD, particularly in dynamic environments where speed and accuracy are paramount. The visual confirmation of the detection results and the efficient inference times highlight the algorithm's ability to identify text in various contexts effectively. Overall, the EAST algorithm's performance in these tests highlights its potential as a reliable tool for STD and recognition in diverse environments.

The proposed research presents some limitations that future studies could address. Firstly, combining multiple OCR techniques may introduce inconsistencies in performance evaluation and output reliability. Although PaddleOCR is recognized for its multilingual capabilities, it may not sufficiently support underrepresented languages, non-English scripts, symbols, or complex scripts. Additionally, font size, style, and orientation variations can lead to OCR output errors. Moreover, the findings may not generalize well across different domains; performance could vary significantly between document and natural scene images or across diverse geographical locations. Future research could benefit from incorporating advanced methods such as Transformers and Vision Language Models, which may improve the handling of complex text detection scenarios. Exploring the integration of the EAST algorithm with advanced transfer learning techniques could enhance its robustness against challenging backgrounds, varying lighting conditions, and diverse text orientations. Emphasizing multilingual capabilities would allow for a more comprehensive evaluation of text detection across various languages, addressing a critical need in diverse environments. By building on the insights from this study, advancements in text detection and recognition can lead to the development of more intelligent systems capable of effectively interacting with textual information in real-world applications.

REFERENCES

[1] C. Luo, L. Jin, and Z. Sun. "MORAN: A Multi-Object Rectified Attention Network for Scene Text Recognition," Pattern Recognition, 2019.

[2] J. Ghosh, A. Talukdar, and K. Sarma, "A lightweight natural scene text detection and recognition system," Multimedia Tools and Applications, 2023.

[3] P. Naveen, and M. Hassaballah, "Scene text detection using structured information and an end-to-end trainable generative adversarial networks," Pattern Analysis and Applications, 2024.

[4] R. Pegah, "Deep Learning Techniques for the Analysis of Soccer Matches," Budapest University of Technology and Economics (Hungary), 2024.

[5] M. Kantipudi, S. Kumar, and A. K. Jha, "Scene Text Recognition Based on Bidirectional LSTM and Deep Neural Network, "Computational Intelligence and Neuroscience, vol. 2021, Article ID 2676780, pp. 1-11, 2021.

[6] Y. Yuwei, L. Yuxin, Z. Zixu, and T. Minglei, "Arbitrary-Shaped Text Detection with B-Spline Curve Network," Sensors, 2023.

[7] Z. Hu, X. Wu, and J. Yang. "TCATD: Text Contour Attention for Scene Text Detection," In 25th International Conference on Pattern Recognition (ICPR), 2021.

[8] P. Cheng, and W. Wang, "A Multi-Oriented Scene Text Detector with Position-Sensitive Segmentation," In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval - ICMR '18, 2018.

[9] M. Ibrayim, Y. Li, and A. Hamdulla, "Scene Text Detection Based on Two-Branch Feature Extraction," Sensors, Basel, Switzerland, vol. 22, 16-6262, 2022.

[10] X. Luan, J. Zhang, M. Xu, W. Silamu, Y. Li, "Lightweight Scene Text Recognition Based on Transformer," Sensors (Basel), vol. 5; 23(9):4490, 2023.

[11] K. Manjari, M. Verma, G. Singal, and S. Namasudra, "QEST: Quantized and Efficient Scene Text Detector using Deep Learning," Association for Computing Machinery Asian and Low-Resource Language Information Processing, pp. 1-18, 2022.

[12] A. Dass, S. Srivastava, M. Gupta, M. Khari, J. P. Fuente, and E. Verdú, "Modelling and control of systems using intelligent water drop algorithm," Expert Systems, 2022.

[13] R. Harizi, R. Walha, and F. Drira, "SIFT-ResNet Synergy for Accurate Scene Word Detection in Complex Scenarios," In International Conference on Agents and Artificial Intelligence (ICAART), SCITEPRESS – Science and Technology Publications, Lda., vol 3, pp. 980-987, 2024.

[14] G. Liao, Z. Zhu, Y. Bai, et al.," PSENet-based efficient scene text detection," EURASIP Journal on Advances in Signal Processing, vol. 97, 2021.

[15] Y. Cai, F. Zhou, and R. Yin, "Exploring Style-Robust Scene Text Detection via Style-Aware Learning," Electronics, vol. 13(2):243, 2024.

[16] M. Lu, Y. Mou, C-L. Chen, and Q. Tang, "An Efficient Text Detection Model for Street Signs," Applied Sciences, vol. 11(13):5962, 2021.

[17] S. Yuchen, C. Zhineng, et al., "LRANet: Towards Accurate and Efficient Scene Text Detection with Low-Rank Approximation Network," arXiv:2306.15142v5, 2024.

[18] M. Aluri and U.D. Tatavarthi, "Geometric deep learning for enhancing irregular scene text detection," Revue d'Intelligence Artificielle, Vol. 38, No. 1, pp. 115-125, 2024.

[19] H. Chen, P. Chen, Y. Qiu, N. Ling, "FARNet: Fragmented affinity reasoning network of text instances for arbitrary shape text detection," IET Image Process. Vol. 17, pp. 1959–1977, 2023.

[20] Y. Zhu and J. Du, "TextMountain: Accurate scene text detection via instance segmentation," Pattern Recognition, vol. 110 (2021) 107336, pp. 1-11, 2020.

[21] B. A. Abubaker, J. Razmara, and J. Karimpour, "A Novel Approach for Target Attraction and Obstacle Avoidance of a Mobile Robot in Unknown Environments Using a Customized Spiking Neural Network," Applied Sciences, 2023.

[22] L. Nandanwar, P. Shivakumara, R. Ramachandra, T. Lu, U. Pal, A. Antonacopoulos, and Y. Lu, "A New Deep Wavefront based Model for Text Localization in 3D Video," IEEE Transactions on Circuits and Systems for Video Technology, 2021.

[23] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 2642-2651, 2017.

[24] M. K. Ebrahimi, H. Lee, J. Won, S. Kim, and S. S. Park, "Estimation of soil texture by fusion of near infrared spectroscopy and image data based on convolutional neural network," Computers and Electronics in Agriculture, 2023.

[25] B. Myint, T. Onizuka, P. Tin, M. Aikawa, I. Kobayashi, and T. Zin, "Development of a real-time cattle lameness detection system using a single side-view camera," Scientific Reports, 2024.