big data and cognitive computing

MDPI

Article

An End-to-End Scene Text Recognition for Bilingual Text

Bayan M. Albalawi [1,2,*], Amani T. Jamal [1], Lama A. Al Khuzayem [1] and Olaa A. Alsaedi [1]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; atjamal@kau.edu.sa (A.T.J.); lalkhuzayem@kau.edu.sa (L.A.A.K.); oaalsaedi@kau.edu.sa (O.A.A.)

[2] Department of Computer Science, Faculty of Computers and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia

* Correspondence: balbalawi0016@stu.kau.edu.sa

Abstract: Text localization and recognition from natural scene images has gained a lot of attention recently due to its crucial role in various applications, such as autonomous driving and intelligent navigation. However, two significant gaps exist in this area: (1) prior research has primarily focused on recognizing English text, whereas Arabic text has been underrepresented, and (2) most prior research has adopted separate approaches for scene text localization and recognition, as opposed to one integrated framework. To address these gaps, we propose a novel bilingual end-to-end approach that localizes and recognizes both Arabic and English text within a single natural scene image. Specifically, our approach utilizes pre-trained CNN models (ResNet and EfficientNetV2) with kernel representation for localization text and RNN models (LSTM and BiLSTM) with an attention mechanism for text recognition. In addition, the AraElectra Arabic language model was incorporated to enhance Arabic text recognition. Experimental results on the EvArest, ICDAR2017, and ICDAR2019 datasets demonstrated that our model not only achieves superior performance in recognizing horizontally oriented text but also in recognizing multi-oriented and curved Arabic and English text in natural scene images.

Keywords: end-to-end scene text recognition; localization text; Arabic text; bilingual text; ResNet; EfficientNetV2; LSTM; BiLSTM; natural scene image

Citation: Albalawi, B.M.; Jamal, A.T.; Al Khuzayem, L.A.; Alsaedi, O.A. An End-to-End Scene Text Recognition for Bilingual Text. *Big Data Cogn. Comput.* **2024**, *8*, 117. https://doi.org/10.3390/bdcc8090117

Academic Editors: Robail Yasrab, Md Mostafa Kamal Sarker and Moulay A. Akhloufi

Received: 30 May 2024
Revised: 27 August 2024
Accepted: 29 August 2024
Published: 9 September 2024

1. Introduction

There is a growing interest in image-processing methods such as object detection [1], scene text localization [2], and scene text recognition (STR) [3] due to the increasing use of applications that interact with images and their components, such as text, signage, and other objects. Natural scene images, which are unmodified digital representations of real-world settings, are commonly used in these applications and often contain textual information [4]. As a result, the topic of localizing and recognizing text within natural scene images has gained significant attention in recent years.

In the past, localizing and recognizing text from natural scene images were seen as separate steps in the process of extracting text from images. This approach involves first identifying and isolating the textual areas within the input images, and then using a text recognizer to determine the sequence of the recognized text from the cropped words. However, this method may have some drawbacks. Firstly, errors can accumulate between the two tasks, and inaccurate localization results can have a significant impact on text recognition performance. Additionally, if each task is optimized individually, it may result in decreased performance in text recognition. Finally, this approach requires a large amount of memory and has poor inference efficiency [5].

In recent years, there have been significant advancements in deep learning (DL) techniques that have enabled the development of end-to-end deep neural frameworks for accurately recognizing text in images of natural scenes. The end-to-end method integrates

the localization and recognition processes into a unified framework. This approach is more likely to produce superior results because accurate text localization greatly enhances text recognition accuracy. Implementing a trainable framework that can simultaneously localize and recognize text has shown substantial improvements in overall performance, particularly for text with irregular shapes in uncontrolled environments [6].

Recognizing text in natural scene images is essential for many content-based applications, such as language translation, security systems for identifying names or company logos on cards, reading street signs for navigation, understanding signage in driver assistance systems, aiding navigation for individuals with visual impairments, and facilitating check processing at banks. To achieve these applications, it is necessary to have techniques that can accurately and consistently localize scene text [7].

Text localization poses significant challenges due to the diverse visual characteristics of texts and complex backgrounds. Texts differ greatly from conventional objects, often appearing in multiple styles and varying in size, color, font, language, and orientation. Additionally, environmental elements such as windows and railings can resemble written language, and natural features like grass and leaves may occasionally mimic textual patterns. These variations and ambiguities make precise text localization [8] more complicated.

The process of localizing textual information from natural scene images is depicted in Figure 1. The first stage, called text detection, determines whether there is text within an image. Then, text localization identifies the exact location of the text. It involves clustering the identified text into coherent regions while minimizing background interference and establishing a bounding box around the text [9].

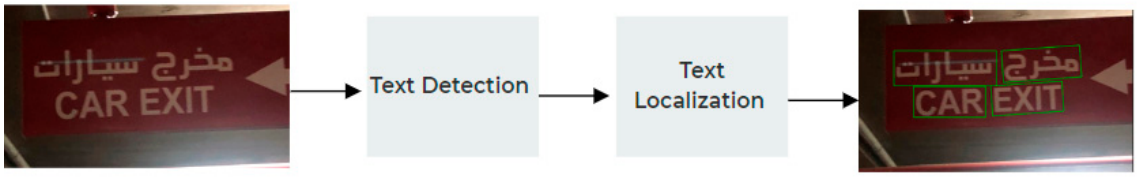

Figure 1. Phases of localization text in natural scene images, image from the EvArEST dataset [10]. Green box demonstrated the result of localization phase.

Recognizing text in natural scene images presents greater challenges because of the intricate text patterns and complex backgrounds that vary significantly in natural settings. Additionally, text in natural scenes can have diverse characteristics, including different fonts, sizes, forms, orientations, and layouts [11]. Recognizing Arabic text, in particular, is more complex than recognizing English text [12].

Arabic is one of the six most widely spoken languages globally and serves as an official language in 26 states across the Arab world, particularly in the Middle East [13]. It is spoken by over 447 million native speakers [14], and its various dialects influence its written form, which progresses from right to left. Arabic characters can take different forms based on their positions within words: initial, medial, final, and isolated [10].

Alrobah et al. [15] proposed a framework for character recognition consisting of four main steps, as shown in Figure 2. The preprocessing phase involves applying techniques such as binarization and noise removal to improve image quality. Segmentation follows, dividing paragraphs into lines and then further segmenting lines into individual words. In the feature extraction stage, each word is treated as a separate entity and input into the machine-learning model after segmentation. Feature extraction is important and complex, and it has a significant impact on the performance of classifiers [16]. Finally, the extracted features are inputted into machine-learning classifiers for recognition.

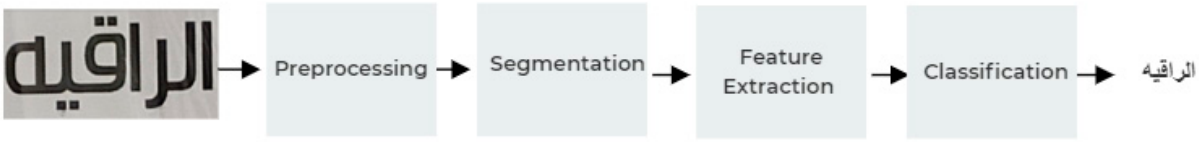

Figure 2. Phases of recognition text from natural scene images, image from the EvArEST dataset [10]. The word in image means "Elegant" in English terms.

Previous research on localizing and recognizing Arabic and English text from natural scene images has identified several limitations:

- Many studies on Arabic text have focused on recognizing handwritten Arabic text [17–20], with relatively few focused on recognizing Arabic text within natural scene images.
- To our knowledge, no researchers have developed an end-to-end STR system capable of integrating the tasks of localizing and recognizing bilingual Arabic and English text within a unified framework. Most existing methods treat these tasks separately.
- The most advanced studies on localizing Arabic text from natural scene images have primarily addressed horizontal text, neglecting the challenges posed by multi-oriented and curved Arabic text.
- To our knowledge, our study is the first to utilize advanced Arabic language models like AraElectra to enhance the recognition accuracy of Arabic text from natural scene images.

This study aims to address a research gap by using an end-to-end STR approach to accurately locate and recognize multi-oriented and curved bilingual Arabic and English texts from natural scene images. Our model is based on the PAN++ framework [21], which was developed for localizing and recognizing multi-oriented and curved English text from natural scene images. In our study, we propose using the pretrained EfficientNetV2 convolutional neural network (CNN) model for feature extraction instead of the ResNet model. EfficientNetV2 is a recent, smaller, and faster neural network for image recognition. Additionally, EfficientNetV2 significantly outperforms previous CNN models in extracting features from images. We compare the performance of these two models in locating Arabic and English text that is both multi-oriented and curved. Additionally, we suggest using the bidirectional long short-term memory (BiLSTM) model instead of the LSTM model in the recognition phase to effectively recognize multi-oriented and curved bilingual Arabic and English text from natural scene images. The BiLSTM model aims to capture additional contextual information by processing the text in forward and backward networks. The BiLSTM model enables us to handle long-term dependencies more effectively and enhances the overall accuracy of recognizing text. We conduct a comparative analysis of the LSTM and BiLSTM models. A key novelty of our research is the integration of the Arabic language model, AraElectra, in the recognition phase to improve the accuracy of Arabic text recognition. Here is a summary of the contributions of our work:

- To the best of our knowledge, this is the first study to utilize end-to-end STR for localizing and recognizing Arabic text only, as well as bilingual Arabic and English text from natural scene images.
- To the best of our knowledge, this is the first study to propose an EvArest dataset that contains multi-oriented and curved text for localizing and recognizing Arabic, as well as bilingual Arabic and English text from natural scene images.
- We employed a pretrained CNN model, EfficientNetV2, to extract features from bilingual Arabic and English texts in the images.
- We utilized BiLSTM with an attention mechanism to recognize bilingual Arabic and English text from natural scene images.
- We integrated the Arabic language model named AraElectra with our end-to-end STR model to enhance the recognition of Arabic text from natural scene images.

The remainder of the paper is organized as follows: Section 2 provides background on localizing and recognizing text in natural scene images. Section 3 discusses related work in localizing and recognizing Arabic and English text from natural scene images. Section 4 details our study's methodology. Section 5 explains the experiments conducted. Section 6 presents the experimental results, followed by a general discussion in Section 7. Finally, Section 8 concludes the paper and outlines future research directions.

2. Background

In this section, we present background information on STR, focusing on approaches for localizing and recognizing text in natural scene images. Furthermore, we explore the unique characteristics of Arabic text and conduct a comprehensive review of existing datasets available for Arabic text in natural-scene images.

2.1. Arabic Language Characteristics

The Arabic language is spoken by a significant number of people worldwide and is an official language in 25 nations [10]. Arabic script differs from other languages, such as English and French, in several ways: it is written from right to left, and Arabic characters consist of 28 letters, which do not have uppercase and lowercase variations like English [15]. Furthermore, each Arabic character can take on four different forms depending on its position within a word: isolated, initial, medial, or final. Table 1 illustrates the various shapes of Arabic characters.

Table 1. Examples of the names and shapes of Arabic characters.

Name	Isolated	Initial	Middle	End
Baa	ب	بـ	ـبـ	ـب
Taa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث

The dot, known as "Noqtah," plays a vital role in maintaining the consistency and structure of Arabic characters. Arabic letters may contain one, two, or three dots, positioned above, in the center, or below the characters. Table 2 provides details about the attributes of Arabic characters. In addition to dots, Arabic text includes the "Hamzah" character (e.g., "ء"), which is essential in distinguishing between different characters and can appear in various positions: above, in the center, or below. Depending on its context, the Hamzah character may be considered an integral part of the character or a distinct component when appearing separately [22].

Table 2. Characteristics of Arabic characters.

Dot (Noqtah)	Name of Characters	Shape of Characters
One dot	Baa, Gem, Khaa, Zal, Zai, Dad, Zaa, Gin, Faa, and Noon	ب, ج, خ, ذ, ز, ض, ظ, غ, ف, ن
Two dots	Taa, Qaf, and Yaa	ت, ق, ي
Three dots	Thaa and Shin	ث, ش
Above characters	Khaa, Zal, Zai, Dad, Zaa, Gin, Faa, Qaf, and Noon	خ, ذ, ز, ض, ظ, غ, ف, ق, ن
Below characters	Baa and Yaa	ب, ي
Center of characters	Ge	ج

2.2. Scene Text Recognition (STR)

STR is often considered a specialized form of OCR that focuses specifically on recognizing text within natural scenes. STR presents significant challenges due to factors such as complex backgrounds, various font styles, and less-than-ideal imaging conditions [23]. The primary objective of STR is to automatically localize and recognize textual content within scene images. In recent years, multiple methodologies have been developed to address this objective, typically categorized into three main phases according to Chen et al. [23] and Khan et al. [6]: text localization, text recognition, and end-to-end systems. Figure 3 provides an overview of the STR system. In the following sections, we will delve into a detailed examination of each of these stages.

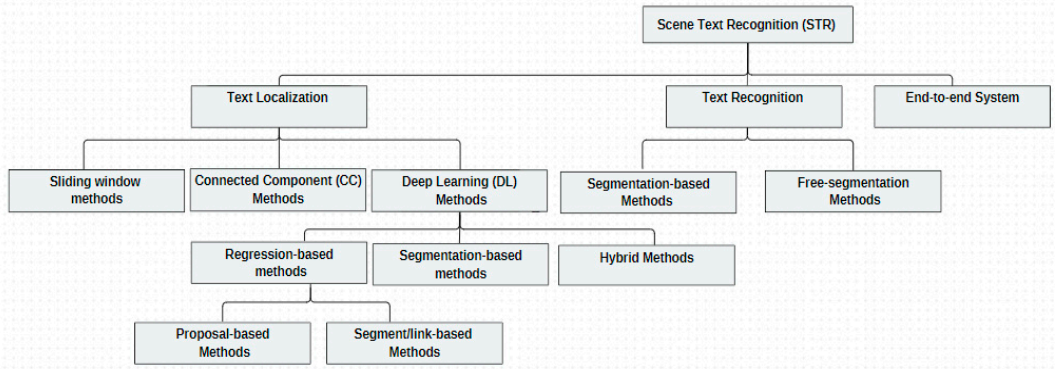

Figure 3. Overview of the scene text recognition system (STR).

2.2.1. Text Localization

The objective of text localization is to pinpoint the precise area within an input image, typically marked by a bounding box, where text is present. These bounding boxes can take various shapes, such as quadrilaterals, oriented rectangles, or regular rectangles. According to [24], different sets of parameters define these shapes: for rectangles, they are (x, y, w, and h); for oriented rectangles, (x, y, w, and h); and for quadrilaterals, (x1, y1, x2, y2, x3, y3, x4, and y4). Khan et al. [6] categorize text localization techniques into three main types: connected component (CC), sliding window-based, and DL-based methods.

Connected Component-Based Methods

In CC-based methods, the goal is to identify and unify small elements into coherent entities. These approaches typically involve segmentation techniques such as stable intensity regions and color clustering. Subsequently, filtering out non-textual elements using low-level features like stroke width, edge gradients, and texture helps in isolating text within an image. These methods are known for their low computational requirements and efficiency. However, they face challenges in handling rotation, scale variations, complex backgrounds, and other intricate scenarios. Representative CC-based techniques include maximally stable extremal regions (MSERs) [25] and stroke width transform (SWT) [26].

Sliding Window-Based Methods

Sliding window-based methods involve generating multiple potential text regions by systematically moving a window of varying sizes and aspect ratios across an image. These candidate text regions are then grouped using a classifier that relies on manually crafted features. Subsequently, these grouped regions are combined to form complete words or lines of text. For instance, Pan et al. [27] introduced a system that localizes text in natural images by employing a sliding-window approach at different scales within a pyramid image structure. They use a conditional random field to distinguish non-textual areas and a minimum spanning tree to segment text into words or lines.

Previous localization techniques, like those using low-level handcrafted features and extensive preprocessing and post-processing steps, tend to be slow and computationally

intensive. These methods are sensitive to challenges commonly found in natural scene images, such as background noise, varying lighting conditions, text orientation, and clutter. Due to these limitations, traditional approaches such as CC and sliding window methods are insufficient for achieving both speed and accuracy in text localization tasks, prompting the adoption of DL approaches [6].

Deep Learning-Based Methods

DL methods achieve accurate text localization by autonomously extracting high-level features from input images. CNNs are widely used for this purpose because they excel at capturing complex features such as geometric patterns and lighting conditions, regardless of variations in these factors. CNNs efficiently extract a multitude of features from images, reducing processing time and computational complexity. DL-based text localization methods can be classified into three main categories: regression-based methods, segmentation-based methods, and hybrid methods.

Regression-Based Methods: Regression-based methods for text localization encompass both proposal-based and segment/link-based approaches tailored for natural scene images. In proposal-based methods, a process akin to sliding windows is employed to generate multiscale bounding boxes of varying aspect ratios across potential text regions. Each bounding box is then evaluated to determine the most accurate localization of the text, whether it is horizontal or quadrilateral. This technique proves highly effective for identifying horizontal and multi-oriented text but can struggle with accurately localizing curved text. Prominent examples of proposal-based methods include Faster R-CNN [28], EAST [29], SSD [30], and TextBoxes++ [31].

Conversely, segment/link-based methods segment text into numerous regions and subsequently use a linking process to connect these regions into a cohesive text localization result. This method, as implemented in connectionist text proposal networks (CTPN) [32], offers greater flexibility compared to proposal-based methods and achieves exceptional accuracy in text localization tasks.

Segmentation-Based Methods: Segmentation-based methods in text localization aim to accommodate a wide range of text sizes by leveraging segmentation algorithms to distinguish text from non-text areas based on pixel-level identification. After segmentation, semantic information and post-processing steps are crucial to accurately delineate text regions. However, the effectiveness of these methods can be influenced by factors such as complex backgrounds, diverse languages, and varying lengths of text lines. Segmentation-based methods typically fall into two categories: semantic segmentation and instance-aware segmentation. Semantic segmentation involves labeling pixels according to semantic information to identify text within an image. The fully convolutional network (FCN) is commonly used for this purpose in text localization tasks [6]. On the other hand, instance-aware segmentation addresses challenges such as overlapping texts by recognizing multiple instances of the same class as distinct objects.

Hybrid Methods: Hybrid methods combine regression-based and segmentation-based techniques to localize text using both bounding box and segmentation approaches. This approach is known for its high accuracy and effectiveness in overcoming various challenges associated with text localization in natural scene images. In recent years, several systems, including LOMO [33], have adopted this hybrid approach to achieve precise text localization results.

2.2.2. Text Recognition

Text recognition involves converting image regions that contain text into machine-readable strings. Unlike general image classification tasks, which have fixed outputs, text recognition deals with variable-length sequences of characters or words. Often, text localization is performed before text recognition as an initial step. Traditional text recognition methods rely on handcrafted features such as CCs, SWT, and histograms of oriented gradient descriptors. However, these methods are often inefficient and slow due to their dependence on low-level features. The adoption of DL approaches is crucial for developing

efficient and highly accurate text recognition systems. DL methods significantly improve both the speed and accuracy of text recognition tasks. Text recognition in natural scene images can be classified into segmentation-based and free-segmentation methods, which will be further explained in subsequent sections [23].

Segmentation-Based Methods

Methods based on segmentation involve breaking down characters into their constituent parts and applying a classification algorithm to identify each segment. Segmentation-based approaches typically consist of three main steps: image preprocessing, character segmentation, and character recognition. Accurately locating text in an image is crucial for these methods. However, segmentation-based methods have significant drawbacks. First, accurately pinpointing individual characters is widely recognized as one of the most challenging tasks in this field. The quality of character detection and segmentation often limits overall recognition performance. Second, segmentation-based approaches are unable to capture contextual information beyond individual characters. This limitation can result in suboptimal word-level results, especially during complex training scenarios.

Free-Segmentation Methods

Free-segmentation techniques aim to recognize entire text passages without segmenting individual characters. This is achieved through an encoder–decoder framework. The process of free-segmentation methods typically involves four stages: image preprocessing, feature representation, sequence modeling, and prediction.

1. Image Preprocessing Stage: The image preprocessing stage aims to enhance image quality and mitigate issues caused by poor image conditions. It plays a critical role in text recognition by improving feature representation. Various image preprocessing techniques such as background removal, text image super-resolution, and rectification are employed. These methods effectively address challenges associated with low image quality, thereby significantly enhancing text recognition accuracy.
2. Feature Representation Stage: Feature representation is crucial for converting raw text-instance images into a form that emphasizes essential characteristics for character recognition while minimizing the influence of irrelevant factors such as font style, color, size, and background. CNNs are widely adopted in this stage due to their efficiency and effectiveness in extracting image features.
3. Sequence Modeling Stage: The sequence modeling stage establishes connections between image features and predictions, enabling the extraction of contextual information from sequences of characters. This approach is valuable for predicting characters in sequence, demonstrating improved reliability and efficiency compared to independent character analysis. BiLSTM networks are commonly utilized in sequence modeling for their capability to capture long-range dependencies accurately [34].
4. Prediction Stage: In the prediction stage, the objective is to determine the correct string sequence based on features extracted from the input text-instance image. Two main techniques employed for this purpose are Connectionist Temporal Classification (CTC) [35] and attention mechanisms [36]. These techniques facilitate accurate and effective decoding of the sequence from the extracted features.

2.2.3. End-to-End System

The objective of end-to-end STR is to convert all text regions within an image into sequences of strings. This process involves several stages, including text localization, recognition, and post-processing, as shown in Figure 4. Traditionally, text localization and recognition were treated as separate tasks that were combined to extract text from images. However, several factors have driven the development of end-to-end STR systems, which integrate text localization and recognition into a unified framework. One motivation is the potential for error accumulation in cascaded systems, where inaccuracies in one stage can propagate and lead to significant overall prediction errors. End-to-end solutions address this by mitigating error growth during training. Additionally, these systems

facilitate the exchange of information between localization and recognition stages, generally improving the accuracy of text extraction. Moreover, end-to-end frameworks offer greater adaptability and ease of maintenance across different domains compared to traditional cascaded pipelines. Finally, they often achieve comparable efficiency with faster inference times and reduced storage requirements [6].

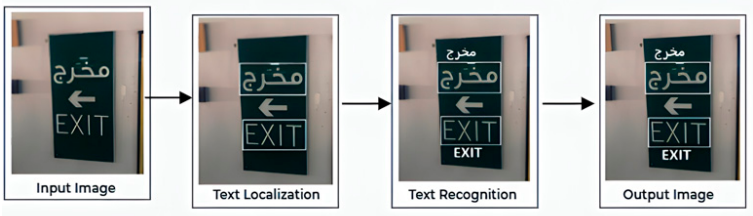

Figure 4. End-to-end scene text recognition phases, image from the EvArEST dataset [10]. The white box is the result of localization phase. Text in white illustrates the result of the recognition phase.

2.3. Datasets of Arabic Scene Text

In this section, we perform a thorough analysis of existing datasets that concentrate on Arabic scene text. Table 3 presents a comparative assessment of these datasets, highlighting factors such as the year of publication, the types of tasks they support for localization and recognition, the orientation of text within images, the number of included images, and their accessibility. These datasets encompass a diverse range of images, including shopping boards, product names, advertisements, and highway signs captured under challenging conditions. These conditions encompass low resolution, inadequate lighting, noise, misaligned text, and variations in colors and sizes. The following section provides a concise overview of each dataset.

Table 3. Summary of Arabic scene text datasets.

Dataset	Year	Task Type	Text Orientation	No. of Images	Availability
ARASTEC [37]	2015	Localization and recognition	Horizontal	260	Private
ARSTI [38]	2017	Recognition	Horizontal	374	Public
ICDAR2017 [39]	2017	Localization and recognition	Horizontal, multi-oriented, and curve	18,000	Public
EASTR-42K [40]	2019	Localization and recognition	Horizontal	2469	Private
ICDAR2019 [41]	2019	Localization and recognition	Horizontal, multi-oriented, and curve	20,000	Public
ASAYAR [42]	2020	Localization	Horizontal	1375	Public
Real-Time Arabic Scene Text Detection [43]	2021	Localization	Horizontal and curve	575	Private
ATTICA [44]	2021	Localization and recognition	Horizontal	1180	Private
EvArEST [10]	2021	Localization and recognition	Horizontal, multi-oriented, and curve	510	Public
TSVD [45]	2021	Localization	Horizontal	7000	Private

2.3.1. ARASTC [37]

This dataset includes Arabic characters extracted from scene images such as signs, hoardings, and advertisements. It consists of 100 classes categorizing 28 Arabic characters in different positions within words (initial, medial, final, and isolated).

2.3.2. ARASTI [38]

This dataset features textual content extracted from images depicting Arabic scenes. It includes segmented words and characters from natural settings, comprising 2093 segmented Arabic characters and 1280 segmented Arabic sentences extracted from 374 scene images. The characters are primarily from signs, billboards, and advertisements, manually segmented.

2.3.3. ICDAR2017 [39]

This dataset contains natural-scene images with embedded text from nine languages, including Arabic. It encompasses 18,000 images, with each language represented by 2000 images.

2.3.4. EASTR-42K [40]

This dataset comprises bilingual (English and Arabic) scene text images, highlighting diverse word combinations in Arabic. It includes precise text in both languages in an unrestricted environment, totaling 2107 lines of Arabic text, 983 lines of English text, and 784 lines of multi-lingual text.

2.3.5. ICDAR2019 [41]

This dataset is designed for multi-lingual scene text localization and recognition systems. It contains 20,000 images, each featuring text in at least one of Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese, or Korean.

2.3.6. ASAYAR [42]

This dataset focuses on Arabic text localization on highways and comprises three components: Arabic–English scene text localization, traffic sign detection, and directional symbol detection. It includes 1763 annotated photographs from Moroccan highways, categorized into 16 distinct object groups, with nearly 20,000 bounding boxes and annotations at both word and line levels.

2.3.7. Real-Time Arabic Scene Text Detection [43]

This dataset focuses on Arabic scene text localization. It includes 575 photos manually annotated with 762 instances of text and 1120 words. Challenges within the dataset include approximately 20% of photos featuring curved text, 10% with blurred images, and variations in typefaces.

2.3.8. ATTICA [44]

ATTICA is a multitask dataset specifically designed for Arabic traffic signs and panels. It covers two distinct Arabic regions: North Africa (Algeria, Egypt, Morocco, Tunisia) and the Gulf region (Bahrain, Kuwait, Qatar, Saudi Arabia, United Arab Emirates). The dataset comprises two primary sub-datasets: ATTICA_Sign, consisting of 1215 annotated images of traffic signs and panels, and ATTICA_Text, featuring 1180 annotated text objects at both line and word levels.

2.3.9. EvArEST [10]

The EvArEST dataset includes Arabic and English text captured in diverse indoor and outdoor settings throughout Egypt. Images were taken using cell phone cameras with varying resolutions. The dataset contains 510 annotated images at the word level.

2.3.10. Tunisia Street View Dataset [45]

The Tunisia Street View Dataset (TSVD) consists of 7000 images from various Tunisian cities sourced from Google Street View, featuring Arabic text. The dataset underwent annotation using an active learning method, with only approximately 20% of training samples labeled and utilized.

The information in Table 3 emphasizes the urgent requirement for additional publicly accessible datasets to facilitate the localization and recognition of Arabic texts. Out of the reviewed datasets, only five are publicly accessible, with three [10,39,41] featuring text in different formats like horizontal, multi-oriented, and curved. The ARASTI dataset [38] specifically concentrates on Arabic characters for specific recognition tasks, whereas the ASAYAR dataset [42] offers publicly available Arabic texts primarily in horizontal orientation.

Our research focuses on accurately recognizing and localizing multi-oriented and curved bilingual Arabic and English texts within natural scene images. To achieve this, we utilized publicly available Arabic and bilingual datasets, specifically ICDAR2017, ICDAR2019, and EvArEST, which provide the necessary types of Arabic and bilingual text.

3. Related Works

Text recognition from natural scene images has recently gained significant attention due to its potential to enhance various applications in daily life. Text localization, which involves identifying text regions, feature extraction, and recognition, plays a crucial role in the effectiveness of text recognition systems. While end-to-end systems have shown promising results in recognizing Latin text from natural scene images, surpassing traditional approaches that handle each text reading process separately, the study of Arabic and bilingual Arabic–English text in this domain remains relatively underexplored. This gap necessitates further research to develop effective strategies for localizing and recognizing Arabic and bilingual text. Our study focuses on prominent academic journals such as IEEE, Web of Science, SpringerLink, and Science Direct. We use specific keywords such as 'Arabic scene text detection', 'English scene text detection', 'Arabic scene text recognition', 'English scene text recognition', and 'end-to-end scene text recognition' to review the latest research on localizing and recognizing Arabic and English text in natural scene images.

3.1. Text Localization from Natural Scene Images

The first step in standard end-to-end text recognition pipelines is scene text localization. The localization of text from natural scene images presents several challenges, especially when dealing with different text orientations. Currently, text localization increasingly relies on DL techniques, which have become the predominant approach. Text found in natural scenes can generally be classified into three types: horizontal text, multi-oriented text, and curved text. Localizing multi-oriented and curved text is more complex than localizing horizontal text. In this section, we specifically explore the techniques employed for localizing Arabic and English text within natural-scene images.

3.1.1. Arabic Scene Text Localization

Gaddour et al. [46] introduced a method to extract Arabic text connections and localize text based on distinct characteristics of the Arabic script and color uniformity. Their approach utilizes threshold values for connection extraction and incorporates ligature and baseline filters to identify and localize text. The ligature filter identifies horizontal connections between characters through analysis of vertical projection profile histograms, while the baseline filter detects the highest intensity value.

Akallouch et al. [42] introduced the ASAYAR dataset, focusing on Moroccan highway traffic panels. It encompasses three primary categories: localization of Arabic–Latin scene text, detection of traffic signs, and identification of directional symbols. This study utilized DL techniques to precisely determine the positions of Arabic text, traffic signs, and directional symbols. TextBoxes++ [31], CTPN [32], and EAST [29] approaches were applied to the ASAYAR_TXT dataset. Experimental results demonstrated that EAST and CTPN achieved superior performance compared to TextBoxes++ in accurately localizing Arabic texts.

Moumen et al. [43] presented a real-time Arabic text localization method using a fully convolutional neural network (FCN). Their approach adopts a two-step framework based on the VGG-16 architecture. In the initial phase, they employed a scale-based

region network (SPRN) to classify regions as either text or non-text. Subsequently, a text detector was utilized to accurately identify text within the predefined scale range, thereby delineating the text regions effectively.

Boujemaa et al. [44] introduced the ATTICA dataset, which includes images of traffic signs and panels from Arabic-speaking countries. The dataset is divided into two sub-datasets: ATTICA-Sign, focusing on traffic signs and boards, and ATTICA-Text. This study utilized DL techniques, specifically EAST and CTPN, to accurately locate Arabic text on traffic panels. Experimental results indicated that the EAST model outperformed the CTPN model in localizing Arabic text.

Boukthir et al. [45] created the TSVD, comprising 7000 images captured from various cities in Tunisia using the Google Street View platform. This study aimed to implement a deep active learning algorithm based on methodologies from Chowdhury et al. [47] and Yang et al. [48]. Their approach integrates CNNs with an active learning strategy to effectively identify Arabic text within natural-scene images. The research focusing on the localization of Arabic text in natural scene images is summarized in Table 4. However, several weaknesses exist in the current methods used for localizing Arabic text from natural scene images.

Table 4. Studies of Arabic text localization in natural scene images.

Ref.	Approach	Scale	Year	Backbone	Dataset	No. of Images	Type of Text	Evaluation			
								Precision (%)	Recall (%)	F-Score (%)	FPS
[46]	NA	NA	2016	NA	Private dataset	50	Horizontal text	78.0	89.0	83.1	
[42]	CTPN	1920 × 1080	2020	VGG	ASAYAR	1375	Horizontal text	88.0	95.0	86.0	NA
	EAST							93.0	74.0	82.0	
	TextBoxes++							66.0	52.0	58.0	
[43]	NA	NA	2021	VGG	Private dataset	575	Multi-oriented dataset	65.1	71.4	68.0	24.3
[44]	CTPN		2021	VGG	ATIICA	1180	Horizontal text	67.0	85.0	74.9	NA
	EAST							71.0	89.0	78.9	
[45]	Deep active learning	640 × 640	2022	VGG	TSVD	700	Horizontal text	82.77	NA	NA	NA
					ICDAR2017	250	Multi-oriented and curved	73.26			
					ICDAR2019	250		74.09			
	Deep learning	640 × 640			TSVD	7000	Horizontal text	89.86			
					ICDAR2017	1000	Multi-oriented and curved	81.55			
					ICDAR2019	1000		81.56			

1. The majority of techniques for handling Arabic datasets, such as ASAYAR and ATTICA, primarily focus on horizontal Arabic text.
2. Two studies were conducted by Moumen et al. [43] and Boukthir et al. [45], utilizing Arabic datasets containing text with curved formatting and various orientations. However, these methods struggle to accurately locate curved texts.
3. The aforementioned approaches exclusively target Arabic text and do not address bilingual text combining Arabic and English. While the ASAYAR dataset used in [42] accurately locates bilingual text, it is restricted to horizontal text.
4. All of the above techniques employ the same pretrained CNN model, specifically VGG-16, for feature extraction from images.

3.1.2. English Scene Text Localization

Taino et al. [32] and Lia et al. [49] successfully utilized object detection frameworks to accurately locate horizontal text in natural scene images, showing excellent performance. Taino et al. [32] introduced the CTPN, which accurately detects the position of text lines

within sequences of precise text proposals using convolutional feature maps. The CTPN model includes a vertical anchor mechanism that simultaneously predicts the location and text/non-text status. Lia et al. [49] introduced TextBoxes, a text detector based on an FCN. This method is known for its speed and accuracy, as it generates word-bounding box coordinates across multiple network layers by predicting text presence and offset coordinates relative to default boxes.

Subsequently, their research focused on the challenges of localizing multi-oriented English text in natural scene images and proposed various strategies to overcome these difficulties. Zhou et al. [29] developed EAST, a scene text detector known for its efficiency and accuracy. This detector uses a single FCN to directly predict multi-oriented text lines. The EAST approach simplifies the process by eliminating intermediate steps such as candidate aggregation and word division. Liao et al. [31] introduced TextBoxes++, an enhanced version of the TextBoxes model discussed in [49]. TextBoxes++ improves the localization of multi-oriented text in natural scene images by optimizing the network structure and training procedure. Shi et al. [50] introduced the seglink method to localize oriented text in natural scene images. This method operates in two stages: segmentation, where rectangular boxes are placed over specific word or text line areas, and linking, which connects adjacent segments.

Currently, researchers are intensifying efforts to address the challenge of localizing curved English text in natural scene images by proposing various models to tackle this complex problem. Wang et al. [51] introduced the progressive scale expansion network, which begins by identifying the smallest-scale text kernel within each text instance. It then gradually expands this kernel using breadth-first-search to obtain a fully completed text instance. Wang et al. [2] introduced the pixel aggregation network (PAN), which is an efficient and accurate text detector that has low computation costs and a learnable post-processing method. The PAN model consists of a segmentation head with the feature pyramid enhancement module (FPEM) and the Feature Fusion Model (FFM), which are used to extract features at different depths. Pixel aggregation (PA) is a trainable post-processing technique that is used to consolidate text pixels based on a predicted similarity vector. Zhang et al. [33] introduced the LOMO system, which uses segmentation techniques and an object identification framework to precisely locate curved text. LOMO consists of three modules: the direct regressor generates text proposals in quadrangle shapes; the iterative refinement module gradually improves these proposals to determine the complete text extent; and the shape expression module enhances accuracy by taking into account geometric text characteristics such as region, center, and border offset. Beak et al. [52] introduced character region awareness for text detection (CRAFT), a method designed to automatically identify and locate each character region. Using CNNs, CRAFT generates region scores and affinity scores. The region score pinpoints the exact position of each character, while the affinity score categorizes characters into separate entities by connecting them. Dai et al. [53] introduced progressive contour regression (PCR) for precise detection of curved text. PCR first suggests horizontal text by estimating center points and sizes, then aligns the overall shape of horizontal suggestions with the corner points of oriented text suggestions. Finally, PCR transforms the shape of directed text suggestions into curved text contours. Ye et al. [54] presented the dynamic point text detection transformer network (DPText-DETR). Unlike traditional bounding boxes, DPText-DETR utilizes explicit point query modeling, which directly employs point coordinates for positional queries. The model introduces the enhanced factorized self-attention module to accurately represent circular points based on polygonal points.

These methods were developed to address the varied challenges in localizing English text from natural scene images, such as horizontal, multi-oriented, and curved text. However, localizing Arabic text and bilingual Arabic–English text presents additional challenges that require further research and development in this field.

3.2. Text Recognition from Natural Scene Images

The process of end-to-end text recognition typically involves a second step known as STR. This section provides an overview of the techniques used to recognize Arabic text and English text in natural scene images.

3.2.1. Arabic Scene Text Recognition

Tounsi et al. [37] proposed an approach for character recognition from natural images using a heap of features, which utilizes spatial pyramid matching (SPM) to handle variations in text. Initially, local features are extracted using SIFT and represented via sparse coding. SPM divides the image into sub-regions and computes histograms describing local features for each region. These histograms are aggregated to represent the features of the entire image. SVMs are employed for individual character classification and recognition. The approach was evaluated on the Arabic scene text character dataset, which includes 260 manually segmented images of characters categorized into 100 classes of 28 Arabic characters in various positions. The experiments involved three different sample configurations: (1) training and testing with five samples per class (5-Arabic characters), (2) training with 15 samples per class and testing with five samples per class (15-Arabic characters), and (3) training and testing with 15 samples per class (15-Glyphs), similar to the 15-Arabic character setup. The results indicate nearly equivalent accuracy in recognizing both characters and glyphs, with slight variation.

Ahmed et al. [55] presented a method for recognizing Arabic characters that have been manually segmented from images. The proposed model preprocesses the images by resizing them to a standard size and converting them to grayscale. Arabic characters have variations depending on their position within a word (beginning, middle, end, isolated), and the model aims to accommodate these variations by considering five different orientations at various angles for each segmented character. The approach uses a ConvNet to extract features of the characters, followed by classification using fully connected layers. The experiments were conducted on Arabic images obtained from the English–Arabic Scene Text (EAST) dataset, which consists of 250 images containing a total of 2700 segmented characters across 27 classes. The findings indicate improved accuracy and high performance of ConvNets when trained on a large and diverse dataset.

Jain et al. [56] demonstrated the recognition of Arabic text in natural images at the word level using a hybrid neural network called CNN–RNN. In this approach, CNN is used to extract feature vectors from the input images' feature maps. These feature vectors are then processed by a bidirectional LSTM network in the second neural network. The bidirectional LSTM predicts the features extracted by CNN, and the final layer of the approach is the transcription layer. This layer utilizes the CTC technique to convert the BLSTM's predictions into sequences of characters. The method was applied to a dataset consisting of 2000 words of Arabic script collected from various locations on Google Images. The performance of the CNN-RNN approach was evaluated using two metrics: character recognition rate (CRR) and word recognition rate (WRR). Leveraging the RNN's ability to capture contextual dependencies, the approach achieved high recognition performance.

Alsaeedi et al. [57] introduced a method for recognizing Arabic texts in natural images using a combination of two neural networks. This approach incorporates a novel segmentation technique that applies a normalization filter to the image and detects characters through vertical and horizontal scanning. CNN is used to extract features and classify the characters. In addition, a transparent neural network (TNN) is employed to validate character recognition against a dictionary, specifically for identifying place names. The effectiveness of the method was evaluated using three different character fonts: Calibri, Aldhabi, and Al-Andalus.

Due to the lack of a dedicated Arabic dataset, Ahmed et al. [40] developed the English–Arabic Scene Text Recognition 42k (EASTR-42K) dataset for Arabic scene images. Their approach involves recognizing Arabic words from these images using MSTR and SFIT for feature extraction, followed by recognition using different input images (binary image

and mask image). In the recognition phase, Multidimensional Long Short-Term Memory (MDLSTM) serves as the learning classifier, with CTC used to predict and recognize words. The experimental study utilized 1500 scene text images segmented into words, evaluating the technique's performance using precision, recall, and F-measure metrics.

Ahmed et al. [58] proposed a method for extracting hybrid features from Arabic text. Their technique starts by detecting the extremal regions of text in an image using MSER, applied to both binary and mask images. They then identify invariant features that are common to both images using SIFT. In the recognition phase, MDLSTM is used to learn from the sequence of features extracted in the previous phase. The experiment was conducted on the Arabic Scene Text Recognition (ASTR) dataset, which consists of 13,593 words segmented from images.

Hassan et al. [10] introduced the EvArEsT dataset, which includes Arabic and English text. They evaluated nine different approaches for recognizing Arabic text by applying methods originally designed for Latin text to Arabic within the EvArEsT dataset. These methods include convolutional recurrent neural network (CRNN), recurrent attention encoder (RARE), R2AM, STARNET, gated recurrent convolutional network (GRCNN), Rosetta, WWSTR, multi-object rectified attention network (MORAN), and SCAN. Among these, WWSTR demonstrated the highest accuracy in recognizing Arabic text. Table 5 provides a comprehensive summary of studies focused on recognizing Arabic characters in natural scene images. It categorizes the methods used into four stages of text recognition: preprocessing, feature extraction, sequence modeling, and prediction. These studies highlight various gaps and challenges in current Arabic text recognition methods.

Table 5. Studies of Arabic text recognition in natural scene images.

Ref.	Approach	Year	Preprocessing	Feature Extraction	Sequence Modeling	Prediction	Dataset	Evaluation
[37]	NA	2015	NA	SIFT	NA	SVM	ARASTEC/ 260 images	(5-Arabic character) = 48.1 (15-Arabic character) = 57.5 (15-Glyphs) = 60.4
[55]	NA	2017	Grayscale image/ fixed size	ConvNet	NA	ConvNet	EAST/250 images	Error rate = 0.15
[56]	CNN-RNN	2017	Scaled to fixed high resolution	VGG	BiLSTM	CTC	Image from Google/2000 words	CRR = 75.05 LRR = 39.43
[57]	NA	2018	Binarization image and grayscale image	CNN	NA	CNN for predict characters and TNN for predict word	Isolated character/300 images and signboard image/100 images	Recognition rate: Calibri = 100% Al-Andalus = 87% Aldhabi = 97%
[40]	NA	2019	Binary image and mask image	MSER and SIFT	MDLSTM	CTC	EASTR-42,000/1500 images	Recall = 89.5% Precision = 94.1% F-score = 97.52%
[58]	NA	2020	Binary image and mask image	MSER and SIFT	MDLSTM	CTC	(ASTR)/13,593 words	Accuracy = 94%
[10]	CRNN	2021	NA	VGG	BiLSTM	CTC	EvArEsT/5337 words	Accuracy = 86.5%
	RARE		Rectification	VGG	BiLSTM	Attention-based decoder		Accuracy = 89.8%
	R2AM		NA	RCNN	NA	Soft-attention mechanism		Accuracy = 84%
	STARNET		Rectification	ResNet	BiLSTM	CTC		Accuracy = 89.6%
	GRCNN		NA	RCNN	BiLSTM	CTC		Accuracy = 87.4%
	Rosetta		NA	ResNet	NA	CTC		Accuracy = 85.4%
	WWSTR		Rectification	ResNet	BiLSTM	Attention-based decoder		Accuracy = 91.2%
	Moran		Rectification	VGG	BiLSTM	Attention-based decoder		Accuracy = 89.4%
	SCAN		Segmentation	VGG	BiLSTM	Self-attention mechanism		Accuracy = 88.4%

1. Currently, there is no research on an end-to-end system for recognizing scene text that can localize and recognize bilingual Arabic and English text. Most studies treat localization and recognition as separate processes.
2. Although current studies are successful in recognizing Arabic text in horizontal forms, there are still difficulties in recognizing multi-oriented and curved Arabic text.
3. A study [10] attempted to address the recognition of Arabic and English text from natural scene images but did not employ an end-to-end approach.
4. Researchers have not yet investigated the use of the LSTM model with an attention mechanism specifically for Arabic text recognition.

Due to the inherent difficulties in localizing and recognizing Arabic text within natural scene images, there exists a substantial need for further research into developing comprehensive end-to-end STR systems. These systems should be equipped to effectively identify and localize Arabic text, especially in scenarios involving multi-oriented and curved text, and extend to the recognition of bilingual text.

3.2.2. English Scene Text Recognition

Three main categories broadly define the field of English STR: character-based text recognition, CTC-based text recognition, and attention-based text recognition. Character-based text recognition involves the initial recognition of individual characters, followed by their assembly into words. For instance, Bissacco et al. [59] introduced PhotoOCR, a device that utilizes a deep neural network specifically designed for individual character recognition.

The text recognition approach that utilizes CTC typically involves stacking RNNs on top of CNNs to effectively capture long-term sequence information. The model is trained using the CTC loss function. Liu et al. [60] introduced the spatial attention residue network, which incorporates a residual network with a spatial attention mechanism. This model consists of three main components: a spatial transformer, a residual feature extractor, and CTC. The spatial transformer uses spatial attention to transform loosely bound and distorted text regions into tightly bound and rectified ones. The residual feature extractor utilizes residual convolutional blocks and integrates LSTM for feature extraction. Shi et al. [61] introduced the CRNN model, which combines deep convolutional neural networks with RNNs. CRNN employs convolutional layers for feature extraction and recurrent layers for making predictions from each frame. Finally, the transcription layer converts the per-frame predictions from the recurrent layer into a sequence of labels. Wang et al. [62] introduced the GRCNN for text recognition. This method follows a three-step approach: GRCNN performs feature extraction, LSTM handles sequence modeling, and CTC facilitates text prediction through transcription. Borisyuk et al. [63] proposed Rosetta, a scalable OCR system consisting of two stages: text detection and text recognition. Text detection utilizes the Faster-RCNN model to accurately locate text regions, while a fully convolutional model predicts character sequences using CTC.

In attention-based text recognition, Shi et al. [64] introduced RARE for recognizing irregular text. The RARE model incorporates a spatial transformer network (STN) using thin-plate spline to rectify and enhance text readability within images. The recognition system employs an attention-based framework with an encoder–decoder architecture. Lee et al. [65] developed the recursive recurrent neural network with attention model (R2AM). R2AM uses a recursive CNN for efficient and accurate image feature extraction. It employs a soft-attention mechanism to effectively utilize image features in a coordinated manner. Luo et al. [3] introduced the MORAN, which consists of two main components: a multi-object rectification network and attention-based sequence recognition (ASRN). The multi-object rectification network corrects irregular text images, facilitating ASRN. ASRN includes a CNN-LSTM architecture followed by an attention decoder for text recognition. The model is trained using weak supervision to learn image part offsets. Zhan et al. [66] presented ESIR, a STR system aimed at reducing perspective distortion and text line curvature to enhance recognition performance. ESIR includes iterative rectification and recognition networks. Iterative rectification adjusts input images using transformation parameters,

while the recognition network employs a sequence-to-sequence model with an attention mechanism. The SCAN model proposed by Hassan et al. [67] aims to accurately detect and classify characters, followed by word generation using a sequential approach. This system consists of two main modules: one for character prediction based on semantic segmentation using the high-resolution network (HRNet), and another for word generation employing an encoder–decoder network. The decoder utilizes LSTM to produce final results, while the encoder employs a series of convolutional layers. Cheng et al. [68] introduced the length-insensitive scene text recognizer (LISTER), designed to accurately identify text in natural scene images of varying lengths. To achieve specific character attention maps, LISTER employs a neighbor decoder. The Feature Enhancement Module is incorporated to capture long-range dependencies with minimal computational cost.

Recent advancements have addressed various challenges in the recognition of English text from natural scene images. Some methods incorporate rectification techniques to precisely adjust text orientation, thereby enhancing their ability to handle multi-oriented and curved English text. These advancements not only improve model performance but also optimize computational efficiency. In contrast, recognizing Arabic text poses more complex challenges compared to English due to the unique characteristics of Arabic characters. Moreover, recognizing bilingual Arabic and English text requires further research to address the distinct challenges of processing two languages within a single model. In our study, we employed an end-to-end STR approach to tackle these issues. Our method utilizes two RNN models and an attention mechanism without relying on rectification methods.

3.3. End-to-End Scene Text Recognition

The end-to-end STR system operates as a unified network capable of both localizing and recognizing text in a single pass, eliminating the need for intermediate processes such as image cropping, word segmentation, or character recognition [11]. Recent advancements in this system have primarily focused on localizing and recognizing multi-oriented and curved English text within natural scene images. However, there has been comparatively less emphasis on addressing Arabic text and bilingual text within the same context. In this section, we present the latest advancements in end-to-end STR approaches specifically tailored for localizing and recognizing multi-oriented and curved Latin script in natural scene images.

Liu et al. [69] introduced ABCNetv2, an end-to-end system designed to localize and recognize multi-oriented and curved scene text using parametrized Bezier curves. In the localization phase, the system employs a novel text detector called Bezier Curve, which offers lower computational intensity compared to traditional rectangular bounding box methods. For recognition, ABCNetv2 utilizes BezierAlign, a feature alignment method that connects the localization and recognition outcomes. Zhang et al. [70] proposed TESTR, another end-to-end method for text localization and recognition that eliminates the need for region of interest (RoI) or post-processing. TESTR operates in two stages: first, a multiscale deformable attention mechanism generates multiscale feature maps. Then, dual decoders are employed—one for text localization and another for recognition. SwinTextSpotter, developed by Huang et al. [5], uses the Swin transformer method for both text localization and recognition in an end-to-end system. Wang et al. [21] introduced PAN++, an end-to-end STR system that employs pixel-based representation and PA techniques. Kittenplon et al. [71] presented TextTranSpotter (TTS), a novel framework for end-to-end STR. TTS integrates a transformer-based architecture with encoder–decoder structures and task-specific heads for localization, recognition, and segmentation. Huang et al. [72] proposed ESTextSpotter, a state-of-the-art approach for end-to-end STR. ESTextSpotter leverages explicit synergy between text localization and recognition processes. UNITS, developed by Kil et al. [73], is a new model designed for end-to-end STR, treating it as a task of generating a sequence. This model supports various localization formats—points, bounding boxes, quadrilaterals, and polygons—by integrating them into a unified interface. Ye et al. [74] introduced DeepSolo, a groundbreaking method inspired by DETR. DeepSolo combines

text localization and recognition into a single process using a single decoder. It uses an explicit point representation for text lines, employing ordered points on Bezier center curves. This approach improves detection and recognition by modeling queries with attributes such as position, offset, and category. Das et al. [75] introduced the Swin-TESTR model for addressing end-to-end STR. This model is specifically designed to handle regular text, multi-oriented text, and curved text. Swin-TESTR utilizes a transformer-based architecture with a Swin Transformer backbone. This design enables the extraction of multi-scale features from input images, thereby improving its capability to capture intricate text details across different domains and orientations.

In our research, our goal is to create a stronger end-to-end system for localizing and recognizing text in scenes. This system should be able to locate and identify Arabic only, bilingual Arabic, and English text in natural scene images. Our approach is influenced by PAN++ [21], an advanced system for recognizing scene text that is well-known for its ability to localize and recognize English text that is oriented in multiple directions or curved. PAN++ is a recent model for localizing and recognizing multi-oriented and curved text from natural scene images that achieved superior results in accuracy and obtained the fastest inference speed compared with other state-of-the-art. PAN++ incorporates a fast and accurate object detector called kernel representation that can localize text by using a single fully convolutional network, which is very useful in real-time applications.

4. Proposed Methodology

Recognizing bilingual Arabic and English text from natural scene images in an end-to-end STR system involves two main phases: scene text localization and STR. Figure 5 illustrates the architectural representation of our end-to-end system, which is inspired by the PAN++ model [21]. In our approach, we selected the EfficientNetV2 model for feature extraction and compared its performance with the ResNet model. During the recognition phase, we utilized the BiLSTM model to handle the recognition of bilingual Arabic and English text, assessing its performance against the LSTM model. Furthermore, we integrated the AraElectra Arabic language model as a post-processing step to enhance the accuracy of recognizing Arabic text from natural scene images. For scene text localization, our system utilizes the kernel representation of PAN++ to precisely locate text regions within an image. Initially, we extract features from the image using a pretrained CNN model and employ methods like the feature pyramid network. The backbone network is enhanced with the Feature Pyramid Enhancement Module (FPEM), which integrates multiple feature pyramid modules to deepen the network and improve feature expression. The PAN++ stage then uses CNN models in the detection head to predict the text region, text kernel, and instance vector, aggregating these predictions with Pixel Aggregation (PA) to accurately localize the final text regions. In the recognition phase, a masked RoI extracts feature patches from the detected text lines. The recognition head includes two layers of the LSTM model followed by a multi-head attention mechanism, ensuring accurate recognition of the text content.

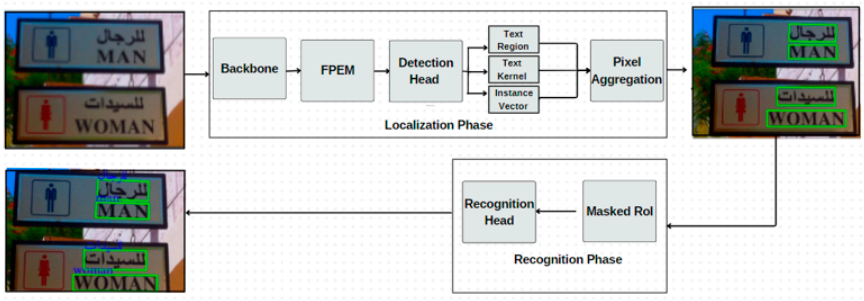

Figure 5. Phases of the proposed end-to-end scene text recognition system, image from ICDAR2017 [39]. The green box illustrated the result of localization phase. Text in blue illustrates the result of the recognition phase.

4.1. Localization Phase

The following subsections explain each step of the localization phase.

4.1.1. Backbone Stage

In our research, we used a pretrained CNN model as the fundamental framework for extracting features. More specifically, we examined two different types of framework models for extracting features from bilingual Arabic and English text: ResNet [76] and EfficientNetV2 [77]. Although PAN++ typically uses ResNet as its default framework, our study aimed to investigate the performance of EfficientNetV2, another pretrained CNN model. Our objective was to train and assess the effectiveness of EfficientNetV2 in accurately identifying bilingual Arabic and English text within natural scene images.

ResNet

He et al. [76] introduced ResNet, a DL model designed to address the issue of 'vanishing gradient' through residual learning. Residual learning involves shortcut connections that bypass one or more layers and perform identity mappings. These shortcuts add the output of skipped layers directly to the subsequent layers' outputs. Importantly, this approach avoids the introduction of extra parameters or increased computational complexity. ResNet is structured as a residual network using convolutional layers, typically with 3×3 filters. It was trained on the ImageNet dataset, which comprises 1.28 million images categorized into 1000 classes. A separate validation set of 50,000 images was used during the training process. ResNet was developed with depths ranging from 18 to 152 layers, demonstrating its scalability and efficacy in various DL tasks.

EfficientNetV2

Several techniques are commonly used in DL to improve the performance of neural networks. These methods often involve increasing the depth of the network, expanding input image sizes, or widening the network. However, improvements in accuracy are primarily limited by the increase in network depth alone, as it can lead to issues such as gradient explosion or vanishing gradients. Furthermore, deeper networks require more storage capacity. Expanding the dimensions of the model allows for greater complexity and specificity. However, as the model complexity increases, achieving deeper insights becomes more challenging. Alternatively, increasing the resolution of the input images enables the model to capture more intricate features. Nevertheless, this augmentation also increases the computational demands and slows down the training speed. EfficientNet [78] effectively balances these trade-offs to achieve optimal performance.

EfficientNet is a CNN model that uses neural architecture search to develop a baseline network, which is then scaled up to create a series of models. This family includes EfficientNet-B0 through EfficientNet-B7, each scaled using specific parameters. The main building block used is the mobile inverted bottleneck MBConv [79,80], which incorporates squeeze-and-excitation optimization [81]. Tan et al. [77] introduced EfficientNetV2 to improve training speed compared to EfficientNetV1. During the development of EfficientNetV1, researchers identified challenges that could slow down training and reduce model efficiency. Training with large images can result in increased memory usage and slower training times. EfficientNetV1's architecture incorporates MBConv, which utilizes depth-wise convolution [82] (as shown in Figure 6 and inspired by [77]). Depthwise convolutions are CNN layers with fewer parameters and floating-point operations than standard convolutions, although they may not fully take advantage of modern accelerators, potentially slowing down training in early layers. EfficientNetV1 employs a straightforward compound scaling approach to uniformly increase all network stages. For example, when the depth coefficient is 2, each stage within the network doubles in the number of layers.

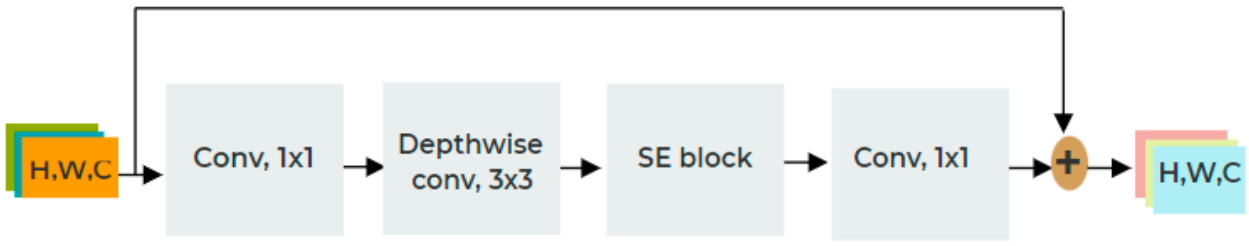

Figure 6. Structure of MBConv. "+" means element-wise addition. "Conv" and "SE" represent regular convolution and Squeeze and Excitation optimization, respectively.

EfficientNetV2 improves the training speed of EfficientNetV1 by substituting MBConv in the initial layer with Fused-MBConv [83], as shown in Figure 7 and inspired by [77]. Fused-MBConv is designed to optimize the utilization of mobile or server accelerators. It replaces the depth-wise conv3 × 3 and expansion conv1 × 1 in MBConv with a single regular conv3 × 3 convolution. The architectures of EfficientNetV2-S models integrate two types of CNNs: MBConv and Fused-MBConv. These networks differ in the number of layers and kernel sizes, providing options like 3 × 3 and 5 × 5 kernel sizes, as well as expansion ratios of 1, 4, and 6.

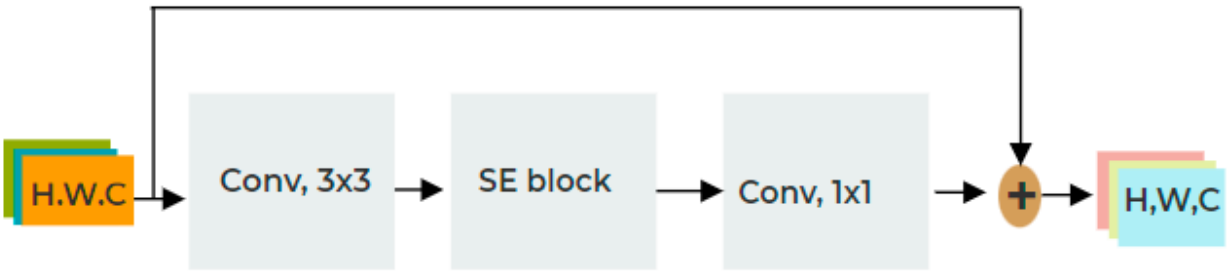

Figure 7. Structure of Fused-MBConv. "+" means element-wise addition. "Conv" and "SE" represent regular convolution and Squeeze and Excitation optimization, respectively.

4.1.2. Feature Pyramid Enhancement Module (FPEM) Stage

During this phase, the backbone network generates four feature maps from the conv2, conv3, conv4, and conv5 stages, corresponding to image resolutions of 1/4, 1/8, 1/16, and 1/32. We then applied 1 × 1 convolutions to reduce the channel dimensions of each feature map to 128, creating a compact feature pyramid. The FPEM, structured in a U-shape, enhances features during both upscaling and downscaling phases. Upscaling improves input features using four distinct stride sizes: notably 32, 16, 8, and 4 pixels. These enhanced features from upscaling serve as input for the subsequent downscaling phase. Here, features are further refined with strides matching those from the upscaling phase, starting at 4 and concluding at 32 pixels. The final feature map size obtained from the FPEM stage is $H/4 \times W/4 \times 512$.

4.1.3. Detection Head

The FPEM provides the detection head with a feature map of dimensions $H/4 \times W/4 \times 512$. A 3 × 3 regular convolution [84] initiates the feature map in the first convolutional layer, accompanied by batch normalization and ReLU activation functions. Figure 8, inspired by [21], illustrates the structure of the detection head. The initial convolution operation reduces the channel dimension to 128 in the feature map. This processed feature map then undergoes a second convolutional step with a 1 × 1 kernel size to generate the final output. The detection head identifies three distinct categories: text regions, text kernels, and instance vectors. The text region precisely outlines the spatial boundary of entire text lines. The text kernel distinguishes neighboring instances by identifying the

center of each text line, independent of its shape. Combining the text region and kernel allows for the reconstruction of complete text line structures using instance vectors.

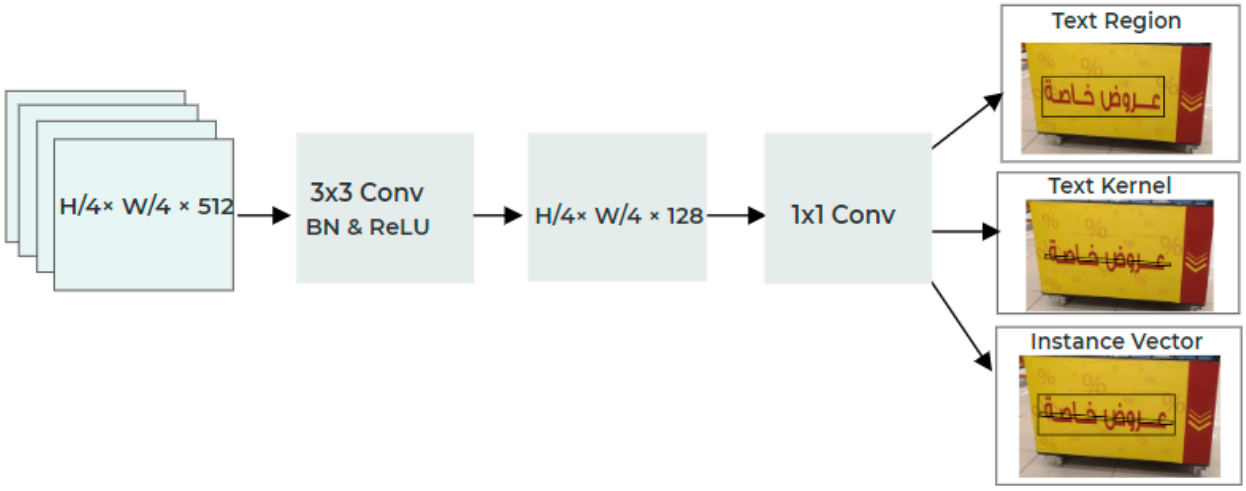

Figure 8. Structure of the detection head. "Conv" and "BN" represent regular convolution and Batch Normalization, respectively. The words in images mean "Special Offers" in English terms.

4.1.4. Pixel Aggregation

PA improves the output of the instance vector by using a clustering method. PA collects text pixels using a suitable text kernel and treats separate text lines as different clusters. The focal points of these text kernels represent each cluster, while the pixels within text areas contribute to creating the cluster center for the text. The goal of this process is to precisely locate bilingual Arabic and English text by efficiently gathering pixels around the identified cluster centers.

4.2. Recognition Phase

The following subsections explain each step of the recognition phase.

4.2.1. Masked Region of Interest (RoI)

The masked RoI is a tool designed to extract feature patches of a specified size from text lines that may have varying shapes. This process involves four sequential steps. First, it determines the smallest bounding rectangle that encompasses the target text line in an upright position. Second, it extracts the feature patch from within this upright bounding rectangle. Third, a binary mask is applied to the feature patch to filter out noise features. This mask assigns a weight of 0 to areas outside the target text line. Finally, the feature patch is resized to a consistent, predetermined size. The masked RoI offers two main advantages. First, by using a binary mask specific to the target text line, it effectively removes noise characteristics that could originate from the background or other text lines. This ensures precise feature extraction even from text lines with diverse shapes. Second, the technique eliminates the need for spatial rectification processes such as the STN.

4.2.2. Recognition Head

The recognition head is structured as a sequence-to-sequence model with a decoder architecture that includes an attention mechanism, skipping the encoder step. This model is divided into two main stages: the starter and the decoder. A detailed explanation of these components is provided in the following section.

Stater Stage

The initial phase of the recognition head focuses on finding the starting point of the string (start of sequence, or SOS) within a text line that may not start in the leftmost

position. This component consists of a linear transformation, an embedding layer ε_1, and a multi-head attention layer A_1. More specifically, the embedding layer ε_1 converts the SOS symbol (represented as a one-hot vector) into a 128-dimensional vector.

Decoder Stage

The decoder stage of the PAN++ model aims to utilize contextual information gathered from visual features obtained in the starter stage to predict the words presented in the text. In this stage, the PAN++ model incorporates two LSTM layers and one multi-head attention layer. Specifically, BiLSTM models with multi-head attention are used to facilitate the recognition of bilingual Arabic and English text.

- Long Short-Term Memory: LSTM is a widely recognized technique that effectively addresses challenges related to vanishing and exploding gradients [85]. Unlike traditional 'sigmoid' or 'tanh' activation functions, LSTM introduces memory cells equipped with gates to manage the flow of information in and out of the cells. These gates regulate how information is input to hidden neurons and preserve features from earlier time steps [86]. An LSTM cell consists of input, forget, and output gates, alongside a cell activation component. These elements receive activation signals from various sources and control cell activation using designated multipliers. LSTM gates prevent other network components from modifying memory cell contents across successive time steps. Compared to RNNs, LSTMs excel in retaining signals and transmitting error information over longer periods, making them highly effective in processing data with intricate dependencies across various sequence learning tasks [87].
- Bidirectional Long Short-Term Memory: BiLSTM, an extension of the bidirectional recurrent neural network (BiRNN) introduced in 1997 to enhance traditional RNNs [88], combines both forward and backward LSTM networks. During training, the forward LSTM network processes the input sequence in chronological order, while the backward LSTM network processes it in reverse. Both networks capture the context of the input sequence and extract crucial features [89]. The outputs of both the forward and backward LSTM networks are then combined to generate the final output of the BiLSTM network. By processing input data in both directions, BiLSTM captures additional contextual information compared to unidirectional LSTM models, enabling it to handle long-term dependencies more effectively and improve overall model accuracy [90].

AraELECTRA: Efficiently learning an encoder that classifies token replacement (ELECTRA) introduces a sophisticated approach to self-supervised language representation learning [91]. This method involves training two neural networks: a generator (G) and a discriminator (D). Each network includes a bidirectional encoder, such as Small BERT. The generator performs masked language modeling by randomly masking out tokens in input sequences and training to predict the original tokens at those positions. Meanwhile, the discriminator is trained to distinguish between the original tokens and the replaced tokens generated by the generator, a technique known as replaced token detection. Formally, both G and D encode an input sentence x into a sequence of contextualized vector representations $h(x) = h1, h2, \ldots, hn$, given a sequence of tokens $x = x1, x2, \ldots, xn$. The generator calculates the probability of having a token x_t at a specific position t when the corresponding x_t is masked as [MASK]. This is achieved using a softmax layer:

$$p_G(x_t \mid x) = \exp e(x_t)^T h_G(x)_t / \sum \exp ex'^T h_G(x)_t \quad (1)$$

where e represents token embeddings.

The discriminator predicts whether the token x_t is 'real' for a given position t, meaning that it originates from the data rather than the generator distribution, using a sigmoid output layer:

$$D(x, t) = \text{sigmoid } w^T h_D(x)_t \quad (2)$$

While ELECTRA's pretraining approach shares similarities with generative adversarial networks (GANs) [92], there are distinct differences. For instance, ELECTRA improves performance on downstream tasks by transforming tokens from 'fake' to 'real' status after generating the correct token. Moreover, ELECTRA focuses on predicting with maximum probability, whereas GANs aim to deceive the discriminator.

The AraELECTRA model [93] is designed for learning Arabic language representation, based on the ELECTRA architecture, with the aim of improving the understanding of Arabic text. It consists of a bidirectional transformer encoder with 136 million parameters, 12 encoder layers, 12 attention heads, a hidden size of 768, and supports a maximum input sequence length of 512 tokens. For pretraining, AraELECTRA used a dataset similar to ARABERT V0.2 [94], which includes approximately 8.8 billion words from various Arabic corpora, mainly composed of news articles. When evaluated in tasks such as sentiment analysis, Arabic question answering, and named-entity recognition, AraELECTRA showed better performance compared to other models trained on the same dataset but with larger model sizes. In our study, we incorporated the AraELECTRA model into the post-processing phase of the recognition pipeline to identify Arabic text from natural-scene images. The phases of Arabic text recognition that include the AraELECTRA model are illustrated in Figure 9.

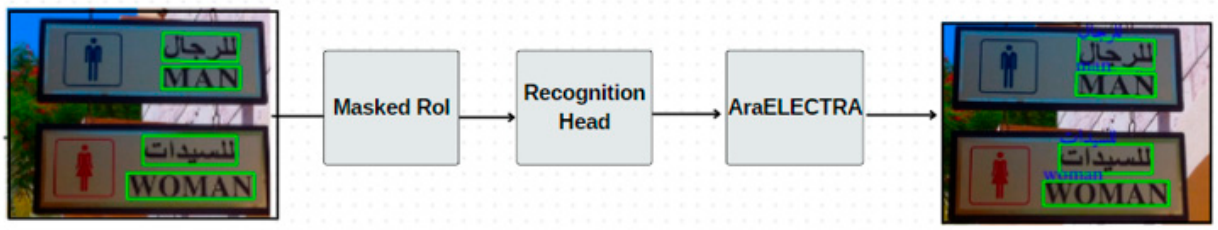

Figure 9. The utilization of the AraELECTRA model in the recognition phase, image from ICDAR2017 [39]. The green box refers to the result of the localization phase. Text in blue illustrates the result of the recognition phase.

5. Experiments

We carried out two comprehensive experiments to evaluate the model's ability to handle bilingual Arabic and English texts: (1) bilingual text localization, and (2) end-to-end bilingual Arabic and English STR. This section presents detailed information about the dataset used, the evaluation metrics employed, and the specifics of training and implementation.

5.1. Datasets

The following subsections explain Arabic scene text datasets, English scene text datasets, and datasets statistics.

5.1.1. Arabic Scene Text Datasets

We evaluated the model's performance in localization and end-to-end STR using the ICDAR2017, ICDAR2019, and EvArEST datasets. Our research focused on identifying and recognizing bilingual Arabic–English texts in natural scene images. To accomplish this, we utilized images from the multi-lingual ICDAR2017 [39] and ICDAR2019 [41] datasets, which are currently the only publicly available datasets containing Arabic and bilingual Arabic–English text in natural scene images suitable for localization and recognition tasks. These datasets encompass a range of challenges commonly found in natural scene images, including complex backgrounds, varying resolutions, diverse text orientations, and multiple Arabic and English fonts. This diversity was crucial in achieving the objectives of our study. Importantly, our study represents the first known use of the EvArEST dataset for both localization and end-to-end tasks.

However, both ICDAR datasets lacked accurate ground-truth data necessary to effectively evaluate the performance of the model. In [41], it was suggested to use online

evaluation tools, but these tools typically evaluate all languages within the dataset rather than specific ones. Therefore, we chose to select approximately 30% of the bilingual Arabic and English testing images from ICDAR2017 and ICDAR2019 and used the online annotation tool Label Studio at https://labelstud.io/ (accessed on 11 April 2024) to create our own ground-truth data. Each word in the annotated images was outlined with a four-point polygon instead of a rectangle to accurately represent irregular text shapes. The polygon vertices were defined starting from the top-left corner of the word and proceeded clockwise. The annotation format for the images followed a structure similar to that used in the ICDAR datasets [95,96]. For each image, a corresponding text file contained three components: the four-point polygon outlining the word, the language of the word, and the text itself, as shown in Figure 10. Table 6 provides statistical information about the bilingual Arabic and English datasets and English datasets.


```
343, 184, 439, 187, 436, 230, 346, 226, 'Arabic', 'قصر '
206, 181, 338, 186, 337, 229, 207, 225, 'Arabic', 'السراب '
151, 230, 237, 233, 237, 278, 150, 276, 'Latin', 'Qasr '
247, 239, 288, 240, 287, 276, 246, 273, 'Latin', 'AL '
292, 240, 399, 246, 400, 280, 293, 275, 'Latin', 'Sarab '
120, 216, 148, 217, 149, 228, 119, 228, 'Arabic', 'قصر '
79, 213, 122, 214, 121, 228, 78, 226, 'Arabic', 'السراب '
78, 225, 106, 226, 106, 237, 77, 236, 'Latin', 'QASR '
106, 226, 147, 227, 147, 239, 105, 238, 'Latin', 'ALSARAB '
```

Figure 10. An example of an image and its ground-truth format from the ICDAR2019 dataset [41].

Table 6. Datasets' statistics.

Dataset	Language	Training Set	Testing Set	Total	Annotation
ICDAR2017	Bilingual text	800	200	1000	Word-level
ICDAR2019	Bilingual text	1000	200	1200	Word-level
EvArEST	Bilingual text	377	133	510	Word-level
ICDAR2015	English text	1000	500	1500	Word-level
COCO-Text	English text	43,686	20,000	63,686	Word-level
Total-Text	English text	1255	300	1555	Word-level

5.1.2. English Scene Text Datasets

Our training process involved using a combination of publicly available benchmark English datasets along with previous Arabic text datasets to train our model for recognizing bilingual Arabic and English text. Specifically, we selected the most widely used English dataset containing natural scene images with varying levels of complexity. The English datasets used for training are as follows:

- ICDAR 2015 [96]: This dataset was collected over several months in Singapore and contains 1670 images with 17,548 annotated regions. It is one of the most comprehensive publicly available datasets with complete ground truth for Latin-scripted text. Out of these images, 1500 are publicly accessible, divided into a training set of 1000 images and a test set of 500 images.
- COCO-Text [97]: The Microsoft COCO dataset serves as the largest benchmark for text localization and recognition. It includes 173,589 text instances from 63,686 images, encompassing handwritten and printed text in both clear and blurry conditions, as well as English and non-English texts. The dataset is composed of 43,686 training images and 20,000 testing images.

- Total-Text [98]: The Total-Text dataset is designed for the localization and recognition of Latin text in various forms, including curved, multi-oriented, and horizontal text lines. It consists of 1255 training images and 300 testing images, annotated with polygons at the word level, primarily obtained from street billboards.

5.2. Evaluation Metrics

The performance evaluation of localizing bilingual Arabic–English texts in natural scene images was assessed using three metrics: precision, recall, and F-score. Additionally, end-to-end STR was evaluated based on accuracy. The evaluation protocol follows that of ICDAR2015 [96], utilizing the intersection over union (*IoU*) metric. This protocol aligns with the evaluation framework used in object detection tasks such as PASCAL VOC [99]. *IoU* is computed by measuring the overlap between predicted and ground-truth bounding boxes, as represented in Equation (3). A prediction is classified as True Positive (*TP*) if its *IoU* exceeds 0.5; otherwise, it is considered a False Positive (*FP*). A False Negative (*FN*) occurs when the model fails to predict any output for a specific region of an image.

$$IoU = \frac{\text{area}(P \cap G)}{\text{area}(P \cup G)} \tag{3}$$

The evaluation metrics were calculated as follows:

- *Accuracy*: This metric calculates the ratio of correctly predicted texts (True Positives, *TP*, and True Negatives, *TN*) to the total number of predicted texts, including both correct and incorrect predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

- *Precision*: Precision measures the proportion of correctly predicted bounding boxes (*TP*) relative to the total number of predicted bounding boxes (both *TP* and False Positives, *FP*).

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

- *Recall*: Recall quantifies the ratio of correctly predicted bounding boxes (*TP*) to the total number of expected results based on the ground truth of a dataset.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

- *F-score:* The F-score represents the harmonic mean of precision and recall, providing a balanced measure of a model's performance.

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

5.3. Training Details

To evaluate how well the model can locate and recognize bilingual Arabic–English text in natural scene images, we conducted a series of experiments that focused on multi-oriented and curved bilingual text. For the localization phase, we used ResNet and EfficientNetV2 models as the network backbones, following a training setup similar to the one described in [21]. For the ResNet model, we used the "poly" learning rate technique with an initial learning rate of 1×10^{-3}. On the other hand, the EfficientNetV2 model employed a step-decay learning rate strategy, starting with a learning rate of 2×10^{-3} and reducing it by a discount factor of 0.5 every five epochs. Our experiments involved testing various input image sizes (736 and 896) and two kernel sizes (0.5 and 0.7) to improve performance. We kept these configurations consistent throughout the entire STR

process. During the recognition phase, we utilized BiLSTM with 256 layers and LSTM with 128 hidden layers. Furthermore, we incorporated the AraELECTRA base version as a post-processing step, making use of pretrained generator and discriminator models. Specifically, we evaluated the model using an input image size of 896 and a kernel size of 0.7 to enhance its performance.

In conclusion, all models were optimized using the ADAM optimizer [100] with a batch size of 16 distributed across four GPUs. During the localization phase training, we employed two main approaches: (1) training models from scratch individually for each dataset, and (2) using joint training strategies. Under the first approach, we conducted 14,137 iterations for the EvArEST dataset, 30,000 iterations for the ICDAR2017 dataset, and 37,500 iterations for the ICDAR2019 dataset. For the joint training strategy, all datasets were trained together for 40,818 iterations, followed by individual testing of each dataset. In the context of end-to-end STR, we exclusively used a joint training strategy. This involved training all Arabic and English datasets collectively for 90,221 iterations, with subsequent testing performed only on the Arabic dataset. Detailed settings for each model during the training phase can be found in Table 7.

Table 7. Training parameters.

Stage	Model	Batch Size	Epochs	Learning Rate	Training Strategy	Image Size
Text Localization	ResNet	16	600	1×10^{-3}	Training from scratch	736/896
			300	1×10^{-3}	Joint training	
	EfficientNetV2	16	600	2×10^{-3}	Training from scratch	
			300	2×10^{-3}	Joint training	
End-to-end	ResNet	16	30	1×10^{-3}	Joint training	896
	EfficientNetV2	16	30	2×10^{-3}		

5.4. Implementation Details

The experiments were conducted using the PyTorch library on a system equipped with a Xeon E5-2686 v4 CPU, four Tesla V100 GPUs, and 24 GB of RAM.

6. Results

The objective of this research was to develop a comprehensive system for localizing and recognizing bilingual Arabic and English texts in natural-scene images through a series of experiments. We evaluated the system using three widely used datasets: ICDAR2017, ICDAR2019, and EvArEST. Our focus was on images containing bilingual text (Arabic and English) from ICDAR 2017 and ICDAR 2019. To annotate our test set, we utilized online annotation tools to annotate 30% of the bilingual Arabic and English images. Our research was divided into two main phases: the localization-only phase and the end-to-end STR phase. In the localization-only phase, our goal was to accurately detect and localize multi-oriented and curved bilingual texts within natural-scene images. This phase leveraged the capabilities of two pretrained CNN models. As depicted in Figure 11, these models demonstrated their ability to handle complex scenes, such as those with multi-oriented and curved text. The end-to-end system aimed to localize and recognize bilingual Arabic and English texts within a unified framework, utilizing two RNN models. As shown in Figure 12, this system effectively localized and recognized Arabic and bilingual text across varying complexities.

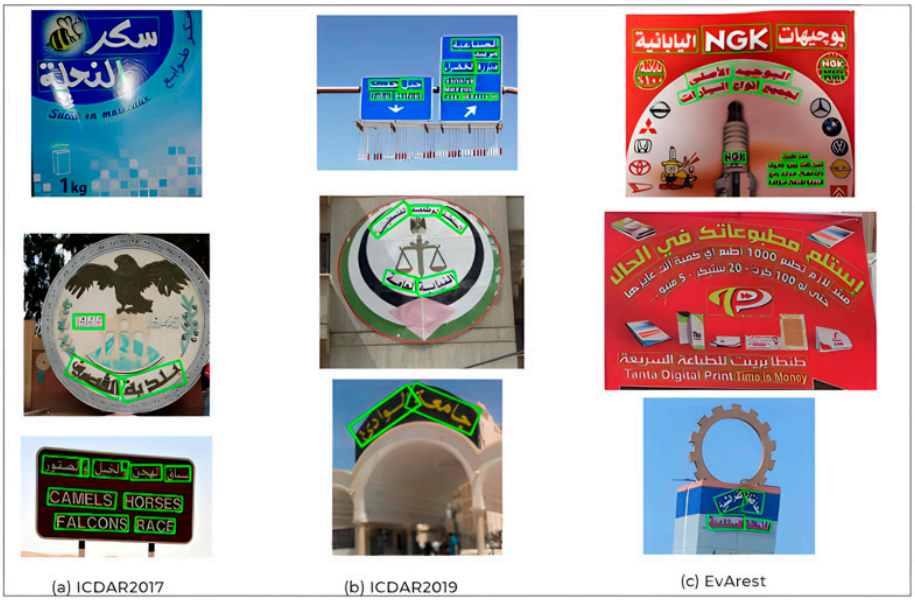

Figure 11. Qualitative text localization results of the model from: (**a**) ICDAR2017 [39]; (**b**) ICDAR2019 [41]; and (**c**) EvArest [10]. Green boxes refer to results of localization phase.

Figure 12. Qualitative end-to-end scene text recognition results of the model from: (**a**) ICDAR2017 [39]; (**b**) ICDAR2019 [41]; and (**c**) EvArest [10]. Green boxes refer to results of localization phase. Text in blue illustrates the result of the recognition phase.

6.1. Scene Text Localization Results

We conducted an initial experiment to evaluate the effectiveness of the ResNet model in localizing bilingual Arabic and English texts within natural-scene images. Specifically, we implemented both the ResNet-18 and ResNet-50 architectures. The experimental findings, which are detailed in Table 8, include metrics such as precision, recall, F-score, and frame rate. These metrics collectively measure the model's inference speed and performance. The ResNet-18 model, trained from scratch, achieved the highest F-score of 90.6% on the

EvArEST dataset, operating at a frame rate of 28.3 FPS. It delivered particularly strong results on the ICDAR2017 and ICDAR2019 datasets. On the other hand, the ResNet-50 model, which utilized a joint training strategy, achieved its highest F-score of 91.2% on the ICDAR2017 dataset, with a frame rate of 20.7 FPS. For the ICDAR2019 dataset, it achieved an F-score of 89.2% at the same frame rate.

Table 8. Results of the localization phase exploiting the ResNet model. The highest F-Score our model achieved is highlighted in bold.

Model	Training Strategy	Scale	ICDAR2017				ICDAR2019				EvArEST			
			Precision (%)	Recall (%)	F-Score (%)	FPS	Precision (%)	Recall (%)	F-Score (%)	FPS	Precision (%)	Recall (%)	F-Score (%)	FPS
ResNet-18	Training from scratch	736	82.3	71.6	76.5	28.4	84.6	75.2	79.6	29.3	90.3	89.5	89.9	30.0
		896	84.4	75.7	79.8	27.0	85.4	76.1	80.4	27.4	91.4	89.8	**90.6**	28.3
	Joint training	736	86.7	81.6	84.0	34.7	89.6	82.9	86.1	31.9	88.2	90.7	89.4	32.4
		896	89.6	83.4	86.3	27.6	90.3	84.6	87.3	19.5	86.9	91.8	89.3	26.9
ResNet-50	Training from scratch	736	85.2	76.1	80.3	22.9	85.3	76.6	80.7	21.4	89.9	89.5	89.7	28.9
		896	86.5	77.2	**81.5**	22.9	86.1	78.9	**82.3**	20.3	90.5	89.9	90.2	16.2
	Joint training	736	90.8	82.3	86.3	26.2	91.1	85.2	88.0	25.9	86.1	90.7	88.4	26.0
		896	91.2	84.6	**87.7**	20.9	91.8	86.4	**89.0**	20.7	86.2	91.1	**89.6**	20.6

The second experiment aimed to evaluate the performance of the EfficientNetV2 model in localizing bilingual Arabic–English texts within natural scene images. We examined three variants: EfficientNetV2-S, EfficientNetV2-M, and EfficientNetV2-L. Results from training these models under various configurations are summarized in Table 9. Our findings highlight that employing a joint training strategy consistently yielded the best results across all model sizes. Specifically, the EfficientNetV2-S model, when trained from scratch, demonstrated superior performance and faster inference across multiple datasets. For instance, on the ICDAR2017 dataset, it achieved an F-score of 75.2% with a frame rate of 32.3 FPS. On the ICDAR2019 dataset, it achieved an F-score of 77.0% at 28.0 FPS, and on the EvArEST dataset, it reached an F-score of 86.5% with a frame rate of 30.1 FPS. Similarly, the joint training approach significantly improved the performance of the EfficientNetV2-L model, achieving the highest F-scores of 78.7% and 82.0% on the ICDAR2017 and ICDAR2019 datasets, respectively, at frame rates of 16.2 FPS and 15.2 FPS. For the EvArEST dataset, an F-score of 87.3% was obtained with a frame rate of 16.5 FPS. These results indicate that both the joint training strategy with EfficientNetV2-L and training from scratch with EfficientNetV2-S were effective across different model sizes, demonstrating robust performance in localizing bilingual texts in natural scene images.

Table 9. Results of the localization phase exploiting the EfficientNetV2 model. The highest F-Score our model achieved is highlighted in bold.

Model	Training Strategy	Scale	ICDAR2017				ICDAR2019				EvArEST			
			Precision (%)	Recall (%)	F-Score (%)	FPS	Precision (%)	Recall (%)	F-Score (%)	FPS	Precision (%)	Recall (%)	F-Score (%)	FPS
EfficientNetV2-S	Training from scratch	736	82.3	66.4	73.4	41.8	83.3	69.6	75.8	36.1	90.4	82.4	86.4	38.2
		896	80.4	70.8	**75.2**	32.3	84.6	70.8	**77.0**	28.0	90.0	83.3	**86.5**	32.1
	Joint training	736	82.5	70.9	76.2	35.2	80.4	74.8	77.4	34.8	88.6	85.5	87.0	34.0
		896	81.3	74.6	77.8	29.2	82.9	76.4	79.5	25.4	89.0	85.2	87.1	28.0
EfficientNetV2-M	Training from scratch	736	79.9	64.2	71.1	32.6	80.2	66.8	72.8	30.5	86.5	82.7	84.5	34.6
		896	78.4	67.6	72.6	26.9	81.5	68.6	74.4	27.9	87.2	83.2	85.1	26.6
	Joint training	736	81.0	70.6	75.4	30.9	80.1	73.8	76.8	30.7	86.9	85.9	86.4	31.8
		896	80.5	75.5	77.9	25.6	82.3	75.9	78.9	24.9	88.1	86.1	87.1	27.2
EfficientNetV2-L	Training from scratch	736	81.9	65.9	73.0	23.6	80.6	71.1	75.5	22.7	88.2	83.4	84.7	23.3
		896	80.3	68.9	74.1	17.4	81.6	72.5	76.7	20.9	88.9	85.1	86.2	18.1
	Joint training	736	82.2	70.1	75.6	25.3	83.7	76.7	80.0	16.4	87.1	85.4	86.3	26.3
		896	81.3	76.3	**78.7**	16.2	85.6	78.8	**82.0**	15.2	86.9	87.8	**87.3**	16.5

6.2. End-to-End Scene Text Recognition Results

We conducted experiments to evaluate the effectiveness of an end-to-end STR system in recognizing bilingual Arabic–English texts in natural scene images. The first experiment aimed to assess the performance of the LSTM model when combined with various CNNs for feature extraction. We utilized LSTM with 128 hidden layers alongside several CNN models, including ResNet-18, ResNet-50, and EfficientNetV2 variants (S, M, and L). The results, summarized in Table 10, include metrics, which are F-score, FPS, recall, precision, and accuracy. Combining ResNet-50 with LSTM and EfficientNetV2-L with LSTM in the recognition head resulted in the highest accuracy and F-score. For the ResNet-50 model, the ICDAR 2017 dataset achieved an accuracy of 77.1% and an F-score of 73.2% at 24.9 FPS. The ICDAR 2019 dataset yielded an accuracy of 76.3% and an F-score of 72.6% at 24.3 FPS. On the EvArEST dataset, we attained an F-score of 79.3% and an accuracy of 85.7% at 25.0 FPS. Using the EfficientNetV2-L model, the ICDAR 2017 dataset demonstrated an accuracy of 56.2% and an F-score of 59.3% at 17.3 FPS. For the ICDAR 2019 dataset, we obtained an accuracy of 53.9% and an F-score of 57.7% at 17.5 FPS. On the EvArEST dataset, we achieved an accuracy of 71.9% and an F-score of 69.6% at 18.6 FPS.

Table 10. The results of an end-to-end STR system with the LSTM model. The highest results our model achieved are highlighted in bold.

Model	Training Strategy	ICDAR2017					ICDAR2019					EvArEST				
		Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS
ResNet-18	Joint training	77.0	97.9	58.3	73.1	28.4	76.1	97.8	57.5	72.4	28.3	85.5	98.0	66.0	79.2	29.5
ResNet-50		**77.1**	**98.1**	**58.3**	**73.2**	**24.9**	**76.3**	**98.5**	**57.5**	**72.6**	**24.8**	**85.7**	**97.7**	**66.7**	**79.3**	**25.0**
EfficientNetV2-S		53.9	97.1	40.9	57.5	33.7	53.1	97.3	40.3	57.0	33.5	69.8	97.6	52.4	68.2	34.2
EfficientNetV2-M		52.2	97.3	39.7	56.4	26.5	51.4	97.2	39.0	55.6	26.9	68.9	97.7	51.7	67.6	28.1
EfficientNetV2-L		**56.2**	**98.2**	**42.4**	**59.3**	**17.3**	**53.9**	**97.7**	**40.9**	**57.7**	**17.5**	**71.9**	**98.0**	**54.0**	**69.6**	**18.6**

The second experiment aimed to demonstrate the capability of the BiLSTM model with two CNN models for recognizing bilingual Arabic–English texts in natural scene images. The results, shown in Table 11, include metrics, which are accuracy, precision, recall, F-score, and FPS. Comparisons between Tables 10 and 11 indicate that BiLSTM improves bilingual text recognition compared to LSTM. Using the ResNet-50 and EfficientNetV2-L backbones, BiLSTM achieved the highest accuracy and F-score. For the ResNet-50 model, we achieved 78.6% accuracy and a 74.1% F-score at 23.5 FPS on the ICDAR 2017 dataset. For the ICDAR 2019 dataset, we achieved 77.6% accuracy and a 73.1% F-score at 23.5 FPS. On the EvArEST dataset, we attained 87.1% accuracy and an 80.5% F-score at 24.6 FPS. Using the EfficientNetV2-L model, we obtained 57.2% accuracy and a 59.8% F-score at 15.1 FPS on the ICDAR 2017 dataset. For the ICDAR 2019 dataset, we reached 56.3% accuracy and a 59.2% F-score at 16.2 FPS. On the EvArEST dataset, we achieved 72.8% accuracy and a 70.2% F-score at 17.3 FPS.

Table 11. The results of the end-to-end STR system with the BiLSTM model. The highest F-Score our model achieved is highlighted in bold.

Model	Training Strategy	ICDAR2017					ICDAR2019					EvArEST				
		Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS
ResNet-18	Joint training	78.1	97.7	59.2	73.7	25.9	74.2	97.7	55.9	71.1	26.8	86.9	97.8	68.0	80.2	28.3
ResNet-50		**78.6**	**97.8**	**59.7**	**74.1**	**21.3**	**77.6**	**98.9**	**58.0**	**73.1**	**23.5**	**87.1**	**97.9**	**68.3**	**80.5**	**24.6**
EfficientNetV2-S		56.8	97.6	42.9	59.6	29.7	55.0	98.1	41.6	58.4	30.9	72.3	98.5	54.4	70.1	32.4
EfficientNetV2-M		54.9	97.4	40.9	57.6	24.4	53.0	97.6	40.2	56.9	25.5	70.4	98.1	53.5	69.2	27.7
EfficientNetV2-L		**57.2**	**97.4**	**43.1**	**59.8**	**15.1**	**56.3**	**97.5**	**42.5**	**59.2**	**16.2**	**72.8**	**97.9**	**54.7**	**70.2**	**17.3**

In the third experiment, our goal was to assess the efficacy of the AraElectra Arabic language model as a post-processing technique for recognizing Arabic text in natural scene images. The results of the utilized AraElectra Arabic language model with ResNet50 and EfficientNetV2-L in LSTM and BiLSTM, shown in Tables 12 and 13, include metrics such as accuracy, precision, recall, F-score, and FPS. Our results revealed that the AraElectra model, when used as a post-processing stage, produced comparable outcomes to those achieved with the LSTM or BiLSTM models. Nevertheless, we did not observe any improvement in performance when employing the AraElectra model as a post-processing technique.

Table 12. The results of the end-to-end STR system with utilized AraElectra with LSTM model.

Model	ICDAR2017					ICDAR2019					EvArEST				
	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS
ResNet-50	77.0	97.9	58.1	72.9	18.6	75.8	98.1	57.0	72.1	19.2	85.2	97.3	66.7	79.1	21.0
EfficientNetV2-L	56.0	97.5	42.1	58.8	15.6	53.5	97.3	40.4	57.0	15.9	71.3	97.8	54.1	69.6	19.2

Table 13. The results of the end-to-end STR system with utilized AraElectra with BiLSTM model.

Model	ICDAR2017					ICDAR2019					EvArEST				
	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	FPS
ResNet-50	78.3	97.2	59.2	73.5	19.2	77.2	98.4	58.0	72.9	20.5	87.0	97.3	68.0	80.0	21.1
EfficientNetV2-L	57.0	97.1	43.1	59.7	14.2	56.2	97.5	42.0	58.7	15.1	72.4	97.3	54.5	69.8	17.2

7. Discussion

The following subsections discuss the results of our model in text localization and end-to-end STR.

7.1. Bilingual Scene Text Localization

In the first experiment, the ResNet architecture was used for localization. The results, presented in Table 8, consistently show significant outcomes across all datasets, regardless of ResNet depths, training strategies, and configurations. It is worth noting that the EvArEST dataset, despite having a limited number of training samples, achieved the highest F-score when trained from scratch with ResNet-18. The ResNet-18 architecture performed exceptionally well on smaller datasets and exhibited fast inference speed, as evidenced by its frame rate across all datasets, thanks to its lightweight design. The ResNet-50 model, when trained with improved methods that incorporated the ICDAR2017 and ICDAR2019 datasets and employed two distinct training strategies, achieved the highest F-score.

In the second experiment, EfficientNetV2 was used to localize multi-oriented and curved bilingual Arabic–English texts, as shown in Table 9. EfficientNetV2 demonstrated significant performance in accurately localizing texts across all datasets. Training from scratch resulted in the EfficientNetV2-S model achieving the highest F-score. Additionally, the EfficientNetV2-L model achieved the highest F-score when the joint training technique was used. However, we observed that the EfficientNetV2-M and EfficientNetV2-L models performed worse than EfficientNetV2-S when trained from scratch. This variation could be attributed to the limited size of the training dataset and the complexity of the deeper architectures, which may have led to overfitting.

The EfficientNetV2 model is available in different versions. In terms of parameter count in our implementation, EfficientNetV2-S has approximately 12 million parameters, whereas the original model described by Tan et al. [77] has 25 million parameters. EfficientNetV2-M has around 20 million parameters, compared to the original model's 55 million parameters, nearly double. This reduction in parameter count makes our EfficientNetV2 model faster than the original model. Table 14 illustrates the difference in

parameter count between the ResNet and EfficientNetV2 models. The difference is significant, with EfficientNetV2-S having fewer parameters than ResNet-18. As shown in Figure 13, ResNet-18 achieves a higher F-score by 4.1 percentage points (90.6% vs. 86.5%) while the EfficientNetV2-S maintaining faster inference speed. Furthermore, as illustrated in Figure 14, EfficientNetV2-M exceeds ResNet-50 in inference speed by 6.6 points (27.2 vs. 20.6 FPS) while maintaining a similar F-score.

Table 14. The different CNN models and their number of parameters.

Backbone	Number of Parameters
ResNet-18	12,246,022
ResNet-50	24,899,782
EfficientNetV2-S	12,106,462
EfficientNetV2-M	20,803,678
EfficientNetV2-L	176,052,766

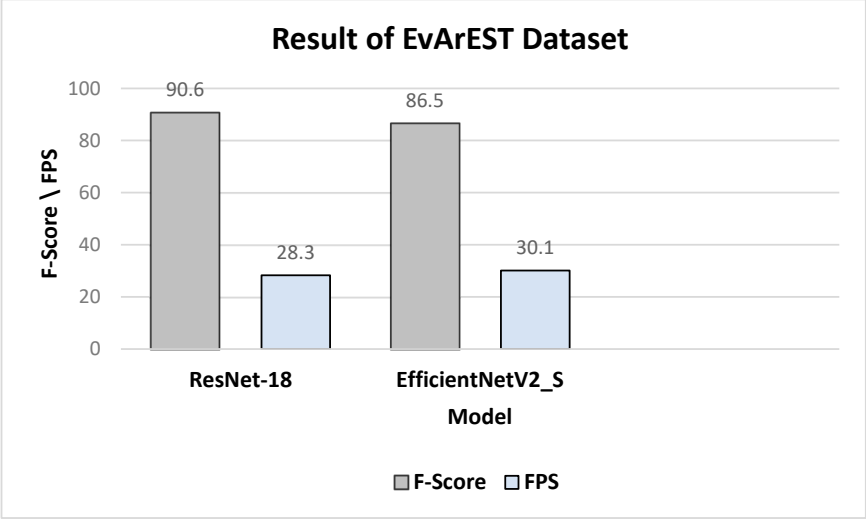

Figure 13. The F-Score results of the EvArEST dataset using ResNet-18 and EfficientNetV2-S.

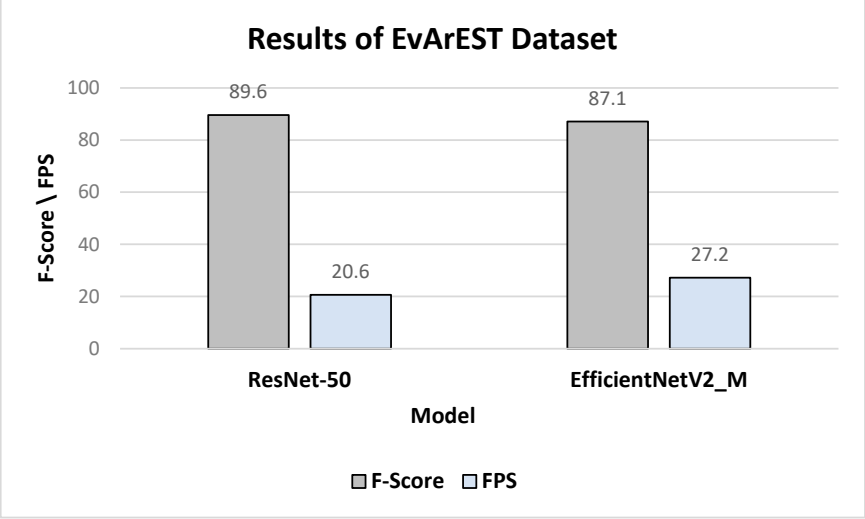

Figure 14. The F-Score results of the EvArEST dataset using ResNet-50 and EfficientNetV2-M.

In contrast, the EfficientNetV2-L model with 176 million parameters achieved the highest performance but had a slower inference speed. This indicates that a large number of parameters can impact system speed. Our comprehensive experiments with ResNet and EfficientNetV2 models led us to the conclusion that increasing the image size to 896 improved performances in all tests.

In conclusion, the ResNet models exhibited superior performance in localizing multi-oriented and curved bilingual Arabic–English texts. In contrast, the EfficientNetV2 models achieved comparable results but with a faster inference speed, measured in FPS. EfficientNetV2 is a pretrained CNN model designed to improve neural network performance by optimizing three important factors: depth, width, and image resolution. However, to achieve optimal performance with these improvements, acquiring additional training data is necessary.

7.2. End-to-End Scene Text Recognition for Bilingual Text

The first and second experiments in the end-to-end STR study were designed to evaluate the effectiveness of two types of RNN models, i.e., LSTM and BiLSTM, in recognizing bilingual Arabic–English texts from natural-scene images. The effectiveness of the LSTM model is demonstrated by the results in Table 10, which show high accuracy and F-scores, especially when combined with the ResNet-50 and EfficientNetV2-L models in the recognition head. The performance of the BiLSTM model, as shown in Table 11, significantly surpassed that of the LSTM model. This improvement is attributed to the BiLSTM's ability to process contextual information in both forward and backward directions, providing a more comprehensive understanding of the text sequence compared to the LSTM model, which operates exclusively in the forward direction.

Effective results were achieved by incorporating the BiLSTM model with ResNet-50 and EfficientNetV2-L as the backbone. The F-score on the ICDAR2017 dataset increased by 0.9 points with the use of BiLSTM, going from 73.1% to 74.1%. Additionally, the F-score on the ICDAR2019 dataset increased by 0.5 points, from 72.6% to 73.1%. In the EvArEST dataset, there was a significant improvement in the F-score, with a rise of 1.2 points (from 79.3% to 80.5%). Furthermore, when using EfficientNetV2-L as the backbone, the BiLSTM model boosted the F-score on the ICDAR2017 dataset by 0.5 points (from 59.3% to 59.8%), and on the ICDAR2019 dataset, it saw an increase of 1.5 points (from 57.7% to 59.2%). The F-score in the EvArEST dataset also improved by 0.6 points (from 69.6% to 70.2%).

In our third experiment, we introduced the AraElectra Arabic language model as a post-processing step following the recognition head to enhance Arabic text recognition. However, integrating the AraElectra model did not significantly impact or improve the results of the end-to-end system. When compared with the LSTM and BiLSTM models, which yielded similar outcomes as shown in Tables 12 and 13, the lack of significant improvement with the AraElectra model can be attributed to the characteristics of the natural-scene image datasets. These datasets typically contain a limited number of words per image, which may not be ideal for language models like AraElectra. Table 15 details the percentage of short sentences (1–5 words) and medium sentences (5–10 words) in the training and testing phases of each dataset, further illustrating the challenges faced by language models in processing short text sequences within natural-scene images.

Table 15. The percentage of short sentences (1–5 words) and medium sentences (5–10 words) in each dataset.

Dataset	Total Number of Image	Training			Testing		
		Number of Image	Short Sentence (%)	Medium Sentence (%)	Number of Image	Short Sentence (%)	Medium Sentence (%)
EvArEST	510	377	87	12	133	91	8
ICDAR2017	1000	800	98	1	200	97	2
ICDAR2019	1200	1000	99	1	200	100	0

In conclusion, the EvArEST dataset consistently outperformed the ICDAR2017 and ICDAR2019 datasets in all experiments. The ICDAR datasets are known for their complexity, which includes images with small text sizes, complex backgrounds, varying fonts, and low resolution.

7.3. Effect of Text Direction

To enhance the recognition of bilingual text, we reversed the ground truth for Arabic words, allowing them to be predicted in a unified direction: from left to right. The results of the bilingual text recognition model using ResNet-50 and EfficientNetV2-L are presented in Tables 16 and 17, employing this unified text direction approach. ResNet-50 and EfficientNetV2-L were chosen for their ability to achieve superior results in our end-to-end STR experiments. The performance of the model in recognizing bilingual text in different word directions (left to right and right to left), denoted as (A), and in unifying the word direction to left-to-right, denoted as (B), is illustrated in Figures 15 and 16. We observed that unifying the direction of word prediction improved the performance of all datasets by approximately 1 to 2 percentage points, resulting in optimal results.

Table 16. The results of the end-to-end scene text recognition system using ResNet-50 and EfficientNetV2-L with the LSTM model in a unified text direction.

Model	ICDAR2017				ICDAR2019				EvArEST			
	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
ResNet-50 + LSTM	78.3	98.5	59.9	74.4	77.5	97.6	59.5	73.9	88.1	97.6	69.4	81.1
EfficientNetV2-L + LSTM	57.9	98.6	44.6	61.4	55.6	98.8	42.3	59.2	73.1	98.9	56.2	71.5

Table 17. The results of the end-to-end scene text recognition system using ResNet-50 and EfficientNetV2 with the BiLSTM model in a unified text direction.

Model	ICDAR2017				ICDAR2019				EvArEST			
	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
ResNet-50 + BiLSTM	80.3	98.5	61.4	75.8	79.2	98.3	60.1	74.5	88.9	98.9	70.3	82.1
EfficientNetV2-L + BiLSTM	59.3	98.6	45.2	61.9	58.6	97.3	44.6	61.1	75.7	98.5	57.9	72.9

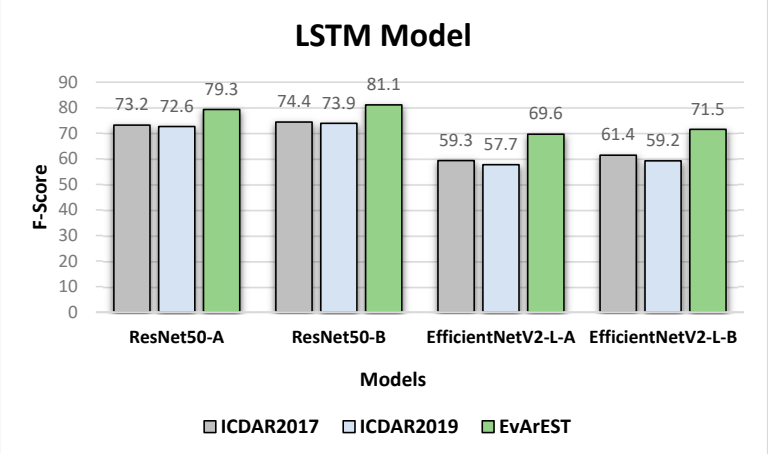

Figure 15. The LSTM model's F-Score outcomes for word direction prediction in (A) various directions and (B) unified direction.

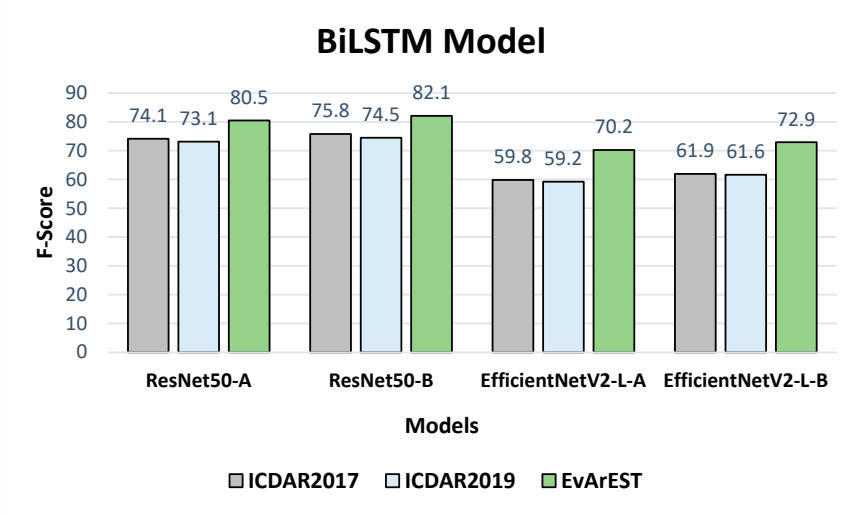

Figure 16. The BiLSTM model's F-Score outcomes for word direction prediction in (A) various directions and (B) unified direction.

7.4. Error Analysis

The proposed method demonstrates superior performance in localizing and recognizing Arabic and bilingual texts within natural scene images, especially in handling multi-oriented and curved bilingual texts. However, the algorithm's accuracy may decrease when attempting to localize vertical Arabic text. In such cases, the proposed technique only partially localizes the text, as illustrated in Figure 17, resulting in less precise predictions. Additionally, the Alruqea font poses challenges in accurately recognizing Arabic characters, as shown in Figure 18. These difficulties primarily stem from the limited number of training samples available. We anticipate that increasing the quantity of training samples will address these issues.

Figure 17. Failure case in localizing vertical Arabic text, image from the ICDAR2019 dataset [41]. Green box refers to the results of the localization phase.

Figure 18. Failure case in recognizing Arabic text, image from EvArest [10]. Green box refers to the results of the localization phase. Words in image in English terms: Amal, shaving, and beautification.

7.5. Comparative Analysis

Our study involved numerous experiments to compare various models for the localization and recognition of multi-oriented and curved bilingual Arabic and English text from images of natural scenes. During the localization phase, we conducted a comparison between two pre-trained CNN models, ResNet and EfficientNetV2, with various configurations. The goal was to assess each model's performance in localizing multi-oriented and curved bilingual text from natural scene images. The findings presented in Tables 8 and 9 demonstrate that the ResNet model outperforms the EfficientNetV2 model in terms of achieving the highest results. However, the EfficientNetV2 model exhibits faster inference speed.

Table 18 shows a comparison between our results and those of the state-of-the-art method. The comparison has demonstrated the effectiveness and robustness of our suggested approach for accurately localizing multi-oriented and curved bilingual text. Boukthir et al. [45] developed a system that used deep learning and active methods to accurately localize Arabic text in natural scene images. The system achieved this by utilizing the ICDAR 2017 and ICDAR 2019 datasets. Our system successfully trained on all the training images from ICDAR 2017 and ICDAR 2019. In contrast, the deep active technique was trained using 20% of the data. Our suggested model demonstrated a significant outperformance in ResNet50, EfficientNetV2-S, and EfficientNetV2-L compared to both deep learning and deep active techniques.

Table 18. Comparison of the proposed models with state-of-the-art methods in precision terms. The highest results our model achieved are highlighted in bold.

Model	ICDAR 2017 (%)	ICDAR 2019 (%)
Deep active learning [45]	73.26	74.09
Deep learning [45]	81.55	81.56
Our ResNet50 (Training from scratch)	86.5	86.1
Our ResNet50 (Joint training)	**91.2**	**91.8**
Our EfficientNetV2-S (Training from scratch)	82.3	84.6
Our EfficientNetV2-L (Joint training)	82.2	85.6

During the recognition phase, we conducted a comparison between two RNN models, LSTM and BiLSTM, to determine their effectiveness in recognizing bilingual text from natural scene images. According to the data from Tables 10 and 11, the BiLSTM model achieved superior performance in bilingual text recognition compared to the LSTM model. Our study is the first to use an end-to-end STR system for localizing and recognizing bilingual Arabic and English text that is multi-oriented and curved in natural scene images. Hassan et al. [10] proposed the EvArEsT dataset that contains images for localization, end-to-end STR, and cropped images for recognizing text as a separated framework. In our study, we used the EvArEsT dataset in localization and end-to-end STR. When comparing our results with other state-of-the-art methods for recognizing Arabic text from natural scene images in reference [10], we noticed that the proposed model achieved excellent outcomes. The proposed model is capable of recognizing text from whole natural scene images by first going through a localization phase and then a recognition phase. On the other hand, alternative approaches to recognizing Arabic text involve using cropped images that contain only individual words. This method provides high accuracy in recognition. In contrast, an end-to-end STR depends on an accurate localization step for effectiveness. Finally, the proposed approach successfully obtained superior results and achieved faster inference speeds when localizing and recognizing multi-oriented and curved bilingual text from natural scene images.

8. Conclusions

This research addressed two significant challenges in computer vision: localizing and recognizing text in natural-scene images. Our study aimed to expand the research scope by focusing on the localization and recognition of bilingual Arabic–English texts, specifically tackling the identification of multi-oriented and curved bilingual texts. We implemented an end-to-end STR system and evaluated it through two stages: localizing bilingual Arabic–English texts and implementing the complete STR system. To conduct our investigation, we utilized three publicly available datasets: ICDAR2017, ICDAR2019, and EvArEST, which contain Arabic and bilingual Arabic–English texts. Due to the diversity of languages in the test sets for the ICDAR datasets and the absence of ground truth for each image, we curated images that exclusively featured bilingual Arabic and English texts. These images were annotated using online tools.

In our study, we used ResNet and EfficientNetV2 CNN models as the backbone, along with a FPEM for reliable feature extraction. We employed a kernel representation approach to locate the center and surrounding pixels of the text, which were then combined using the PA module. Our experiments demonstrated the model's effectiveness in accurately localizing bilingual Arabic–English texts, including texts in various orientations such as multi-oriented and curved texts. While both ResNet and EfficientNetV2 models performed well, ResNet consistently achieved higher accuracy in different configurations compared to EfficientNetV2, which had faster inference speeds.

In the end-to-end system, we compared the performance of LSTM and BiLSTM models for recognizing bilingual Arabic–English texts. The BiLSTM model proved superior in recognizing bilingual texts regardless of word directions and when unifying word directions. However, incorporating the AraElectra Arabic language model as a post-processing step did not result in significant improvements in the recognition of Arabic text within natural scene images. To the best of our knowledge, this study represents the first implementation of an end-to-end STR system that is specifically designed to localize and recognize bilingual Arabic–English texts in natural scene images.

For future research directions, we recommend the development of a more robust system by integrating various CNN and RNN models. Additionally, creating a comprehensive Arabic scene text dataset with a large training sample that includes diverse scene text complexities—such as complex backgrounds, low resolution, varied text orientations, and challenging Arabic fonts—would be beneficial for further advancing the field.

Author Contributions: All authors (B.M.A., A.T.J., L.A.A.K. and O.A.A.) contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by B.M.A. The first draft of the manuscript was written by B.M.A., and all authors (B.M.A., A.T.J., L.A.A.K. and O.A.A.) commented on previous versions of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, grant number (GPIP: 1177-612-2024).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ICDAR datasets are available on the Robust Reading Competition website at https://rrc.cvc.uab.es/?ch=8&com=downloads (accessed on 3 January 2023) for ICDAR2017, and at https://rrc.cvc.uab.es/?ch=15&com=downloads for ICDAR2019 (accessed on 3 January 2023). The code is available in the GitHub repository: https://github.com/whai362/pan_pp.pytorch (accessed on 16 November 2022).

Acknowledgments: This Project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grant no.(GPIP: 1177-612-2024). The authors, therefore, acknowledge with thanks DSR for technical and financial support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, C.; Bochkovskiy, A.; Liao, H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
2. Wang, W.; Xie, E.; Song, X.; Zang, Y.; Wang, W.; Lu, T.; Yu, G.; Shen, C. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8440–8449.
3. Luo, C.; Jin, L.; Sun, Z. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognit.* **2019**, *90*, 109–118. [CrossRef]
4. Bayatpour, S.; Sharghi, M. A bilingual text detection in natural images using heuristic and unsupervised learning. *J. AI Data Min.* **2022**, *10*, 449–466.
5. Huang, M.; Liu, Y.; Peng, Z.; Liu, C.; Lin, D.; Zhu, S.; Yuan, N.; Ding, K.; Jin, L. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In Proceedings of the IEEE/CVF Conference on Compute Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4593–4603.
6. Khan, T.; Sarkar, R.; Mollah, A.F. Deep learning approaches to scene text detection: A comprehensive review. *Artif. Intell. Rev.* **2021**, *54*, 3239–3298. [CrossRef]
7. Katper, S.H.; Gilal, A.R.; Alshanqiti, A.; Waqas, A.; Alsughayyir, A.; Jaafar, J. Deep neural networks combined with STN for multi-oriented text detection and recognition. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 178–184. [CrossRef]
8. Yao, C.; Zhang, X.; Bai, X.; Liu, W.; Ma, Y.; Tu, Z. Rotation-invariant features for multi-oriented text detection in natural images. *PLoS ONE* **2013**, *8*, e70173. [CrossRef]
9. Ranjitha, P.; Rajashekar, K. A Review on text detection from multi-oriented text images in different approaches. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, Coimbatore, India, 2–4 July 2020; pp. 240–245.
10. Hassan, H.; El-Mahdy, A.; Hussein, M.E. Arabic scene text recognition in the deep learning era: Analysis on a novel dataset. *IEEE Access* **2021**, *9*, 107046–107058. [CrossRef]
11. Wang, P.; Li, H.; Shen, C. Towards end-to-end text spotting in natural scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7266–7281. [CrossRef]
12. Ahmed, S.B.; Razzak, M.I.; Yusof, R. *Cursive Script Text Recognition in Natural Scene Images*; Springer: Berlin/Heidelberg, Germany, 2020.
13. Hakak, S.; Kamsin, A.; Tayan, O.; Idris, M.Y.I.; Gilkar, G.A. Approaches for preserving content integrity of sensitive online Arabic content: A survey and research challenges. *Inf. Process. Manag.* **2019**, *56*, 367–380. [CrossRef]
14. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
15. Alrobah, N.; Albahli, S. Arabic handwritten recognition using deep learning: A survey. *Arab. J. Sci. Eng.* **2022**, *47*, 9943–9963. [CrossRef]
16. Hicham, E.M.; Akram, H.; Khalid, S. Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition. *J. Electr. Syst. Inf. Technol.* **2017**, *4*, 387–396. [CrossRef]

17. Al-Saqqar, F.; AL-Shatnawi, A.M.; Al-Diabat, M.; Aloun, M. Handwritten Arabic text recognition using principal component analysis and support vector machines. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 209896493. [CrossRef]
18. Eltay, M.; Zidouri, A.; Ahmad, I. Exploring deep learning approaches to recognize handwritten Arabic texts. *IEEE Access* **2020**, *8*, 89882–89898. [CrossRef]
19. Mustafa, M.E.; Elbashir, M.K. A deep learning approach for handwritten Arabic names recognition. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 211029354. [CrossRef]
20. Eltay, M.; Zidouri, A.; Ahmad, I.; Elarian, Y. Generative adversarial network based adaptive data augmentation for handwritten Arabic text recognition. *PeerJ Comput. Sci.* **2022**, *8*, e861. [CrossRef] [PubMed]
21. Wang, W.; Xie, E.; Li, X.; Liu, X.; Liang, D.; Yang, Z.; Lu, T.; Shen, C. Pan++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5349–5367. [CrossRef]
22. Balaha, H.M.; Ali, H.A.; Badawy, M. Automatic recognition of handwritten Arabic characters: A comprehensive review. *Neural Comput. Appl.* **2021**, *33*, 3011–3034. [CrossRef]
23. Chen, X.; Jin, L.; Zhu, Y.; Luo, C.; Wang, T. Text recognition in the wild: A survey. *ACM Comput. Surv.* **2021**, *54*, 42. [CrossRef]
24. Lin, H.; Yang, P.; Zhang, F. Review of scene text detection and recognition. *Arch. Comput. Methods Eng.* **2020**, *27*, 433–454. [CrossRef]
25. Neumann, L.; Matas, J. A method for text localization and recognition in real-world images. In Proceedings of the Computer Vision–ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010; Revised Selected Papers, Part III 10. Springer: Berlin/Heidelberg, Germany, 2011; pp. 770–783.
26. Epshtein, B.; Ofek, E.; Wexler, Y. Detecting text in natural scenes with stroke width transform. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, San Francisco, CA, USA, 13–18 June 2010; pp. 2963–2970.
27. Pan, Y.F.; Hou, X.; Liu, C.L. A hybrid approach to detect and localize texts in natural scene images. *IEEE Trans. Image Process.* **2010**, *20*, 800–813. [PubMed]
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
29. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. EAST: An efficient and accurate scene text detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5551–5560.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
31. Liao, M.; Shi, B.; Bai, X. Textboxes++: A single-shot oriented scene text detector. *IEEE Trans. Image Process.* **2018**, *27*, 3676–3690. [CrossRef]
32. Tian, Z.; Huang, W.; He, T.; He, P.; Qiao, Y. Detecting text in natural image with connectionist text proposal network. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VIII 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 56–72.
33. Zhang, C.; Liang, B.; Huang, Z.; En, M.; Han, J.; Ding, E.; Ding, X. Look more than once: An accurate detector for text of arbitrary shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10552–10561.
34. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 855–868. [CrossRef] [PubMed]
35. Graves, A.; Fernandez, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376.
36. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
37. Tounsi, M.; Moalla, I.; Alimi, A.M.; Lebouregois, F. Arabic characters recognition in natural scenes using sparse coding for feature representations. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE, Tunis, Tunisia, 23–26 August 2015; pp. 1036–1040.
38. Tounsi, M.; Moalla, I.; Alimi, A.M. ARASTI: A database for arabic scene text recognition. In Proceedings of the 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), IEEE, Nancy, France, 3–5 April 2017; pp. 140–144.
39. Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; et al. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE, Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1454–1459.
40. Ahmed, S.B.; Naz, S.; Razzak, M.I.; Yusof, R.B. A novel dataset for English-Arabic scene text recognition (EASTR)-42K and its evaluation using invariant feature extraction on detected extremal regions. *IEEE Access* **2019**, *7*, 19801–19820. [CrossRef]

41. Nayef, N.; Patel, Y.; Busta, M.; Chowdhury, P.N.; Karatzas, D.; Khlif, W.; Matas, J.; Pal, U.; Burie, J.C.; Liu, C.l.; et al. ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition–RRC-MLT-2019. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, Sydney, Australia, 20–25 September 2019; pp. 1582–1587.
42. Akallouch, M.; Boujemaa, K.S.; Bouhoute, A.; Fardousse, K.; Berrada, I. ASAYAR: A dataset for Arabic–Latin scene text localization in highway traffic panels. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 3026–3036. [CrossRef]
43. Moumen, R.; Chiheb, R.; Faizi, R. Real-time Arabic scene text detection using fully convolutional neural networks. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 1634–1640. [CrossRef]
44. Boujemaa, K.S.; Akallouch, M.; Berrada, I.; Fardousse, K.; Bouhoute, A. ATTICA: A dataset for Arabic text-based traffic panels detection. *IEEE Access* **2021**, *9*, 93937–93947. [CrossRef]
45. Boukthir, K.; Qahtani, A.M.; Almutiry, O.; Dhahri, H.; Alimi, A.M. Reduced annotation based on deep active learning for Arabic text detection in natural scene images. *Pattern Recognit. Lett.* **2022**, *157*, 42–48. [CrossRef]
46. Gaddour, H.; Kanoun, S.; Vincent, N. A new method for arabic text detection in natural scene image based on the color homogeneity. In Proceedings of the Image and Signal Processing: 7th International Conference, ICISP 2016, Trois-Rivières, QC, Canada, 30 May–1 June 2016; Proceedings 7. Springer: Berlin/Heidelberg, Germany, 2016; pp. 127–136.
47. Chowdhury, A.; Biswas, S.K.; Bianco, S. Active Deep Learning Reduces Annotation Burden in Automatic Cell Segmentation. In Proceedings of the Medical Imaging 2021: Digital Pathology, Online, 15–19 February 2021; SPIE: Bellingham, WA, USA, 2021; Volume 11603, pp. 94–99.
48. Yang, L.; Zhang, Y.; Chen, J.; Zhang, S.; Chen, D.Z. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention–MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, 11–13 September 2017; Proceedings, Part III 20. Springer: Berlin/Heidelberg, Germany, 2017; pp. 399–407.
49. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. Textboxes: A fast text detector with a single deep neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
50. Shi, B.; Bai, X.; Belongie, S. Detecting oriented text in natural images by linking segments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2550–2558.
51. Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; Shao, S. Shape robust text detection with progressive scale expansion network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9336–9345.
52. Baek, Y.; Lee, B.; Han, D.; Yun, S.; Lee, H. Character region awareness for text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9365–9374.
53. Dai, P.; Zhang, S.; Zhang, H.; Cao, X. Progressive contour regression for arbitrary-shape scene text detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7393–7402.
54. Ye, M.; Zhang, J.; Zhao, S.; Liu, J.; Du, B.; Tao, D. Dptext-detr: Towards better scene text detection with dynamic points in transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 3241–3249.
55. Ahmed, S.B.; Naz, S.; Razzak, M.I.; Yousaf, R. Deep learning based isolated arabic scene character recognition. In Proceedings of the 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), IEEE, Nancy, France, 3–5 April 2017; pp. 46–51.
56. Jain, M.; Mathew, M.; Jawahar, C. Unconstrained scene text and video text recognition for rabic script. In Proceedings of the 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), IEEE, Nancy, France, 3–5 April 2017; pp. 26–30.
57. Alsaeedi, A.; Al Mutawa, H.; Snoussi, S.; Natheer, S.; Omri, K.; Al Subhi, W. Arabic words recognition using CNN and TNN on a smartphone. In Proceedings of the 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), IEEE, London, UK, 12–14 March 2018; pp. 57–61.
58. Ahmed, S.B.; Naz, S.; Razzak, I.; Prasad, M. Unconstrained arabic scene text analysis using concurrent invariant points. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, Glasgow, UK, 19–24 July 2020; pp. 1–6.
59. Bissacco, A.; Cummins, M.; Netzer, Y.; Neven, H. Photoocr: Reading text in uncontrolled conditions. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 2–8 December 2013; pp. 785–792.
60. Liu, W.; Chen, C.; Wong, K.Y.K.; Su, Z.; Han, J. Star-net: A spatial attention residue network for scene text recognition. In Proceedings of the BMVC, York, UK, 19–22 September 2016; Volume 2, p. 7.
61. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 2298–2304. [CrossRef]
62. Wang, J.; Hu, X. Gated recurrent convolution neural network for ocr. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
63. Borisyuk, F.; Gordo, A.; Sivakumar, V. Rosetta: Large scale system for text detection and recognition in images. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 71–79.
64. Shi, B.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. Robust scene text recognition with automatic rectification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4168–4176.

65. Lee, C.Y.; Osindero, S. Recursive recurrent nets with attention modeling for ocr in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2231–2239.
66. Zhan, F.; Lu, S. Esir: End-to-end scene text recognition via iterative image rectification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2059–2068.
67. Hassan, H.; Torki, M.; Hussein, M.E. SCAN: Sequence-character aware network for text recognition. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021), Vienna, Austria, 8–10 February 2021; pp. 602–609.
68. Cheng, C.; Wang, P.; Da, C.; Zheng, Q.; Yao, C. LISTER: Neighbor decoding for length-insensitive scene text recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 4–6 October 2023; pp. 19541–19551.
69. Liu, Y.; Shen, C.; Jin, L.; He, T.; Chen, P.; Liu, C.; Chen, H. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 8048–8064. [CrossRef] [PubMed]
70. Zhang, X.; Su, Y.; Tripathi, S.; Tu, Z. Text spotting transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9519–9528.
71. Kittenplon, Y.; Lavi, I.; Fogel, S.; Bar, Y.; Manmatha, R.; Perona, P. Towards weakly-supervised text spotting using a multi-task transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4604–4613.
72. Huang, M.; Zhang, J.; Peng, D.; Lu, H.; Huang, C.; Liu, Y.; Bai, X.; Jin, L. Estextspotter: Towards better scene text spotting with explicit synergy in transformer. In Proceedings of the IEEE/CVF International Conference on Computer 1446 Vision, Paris, France, 2–3 October 2023; pp. 19495–19505.
73. Kil, T.; Kim, S.; Seo, S.; Kim, Y.; Kim, D. Towards unified scene text spotting based on sequence generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 15223–15232.
74. Ye, M.; Zhang, J.; Zhao, S.; Liu, J.; Liu, T.; Du, B.; Tao, D. Deepsolo: Let transformer decoder with explicit points solo for text spotting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 19348–19357.
75. Das, A.; Biswas, S.; Banerjee, A.; Lladós, J.; Pal, U.; Bhattacharya, S. Harnessing the power of multi-lingual datasets for pre-training: Towards enhancing text spotting performance. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 1–6 January 2024; pp. 718–728.
76. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
77. Tan, M.; Le, Q. EfficientNetV2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 10096–10106.
78. Tan, M.; Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
79. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
80. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MNASNet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2820–2828.
81. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
82. Sifre, L.; Mallat, S. Rigid-motion scattering for texture classification. *arXiv* **2014**, arXiv:1403.1687.
83. Gupta, S.; Tan, M. *EfficientNet-EdgeTPU: Creating Accelerator-Optimized Neural Networks with AutoML*; Google AI Blog: San Francisco, CA, USA, 2019; Volume 2.
84. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
85. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
86. Le, Q.V.; Jaitly, N.; Hinton, G.E. A simple way to initialize recurrent networks of rectified linear units. *arXiv* **2015**, arXiv:1504.00941.
87. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent advances in recurrent neural networks. *arXiv* **2017**, arXiv:1801.01078.
88. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
89. Sun, S.; Sun, J.; Wang, Z.; Zhou, Z.; Cai, W. Prediction of battery SOH by CNN-BiLSTM network fused with attention mechanism. *Energies* **2022**, *15*, 4428. [CrossRef]
90. Adil, M.; Wu, J.Z.; Chakrabortty, R.K.; Alahmadi, A.; Ansari, M.F.; Ryan, M.J. Attention-based STL-BiLSTM network to forecast tourist arrival. *Processes* **2021**, *9*, 1759. [CrossRef]
91. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.

92. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014; Volume 27.
93. Antoun, W.; Baly, F.; Hajj, H. AraELECTRA: Pre-training text discriminators for Arabic language understanding. *arXiv* **2020**, arXiv:2012.15516.
94. Antoun, W.; Baly, F.; Hajj, H. Arabert: Transformer-based model for rabic language understanding. *arXiv* **2020**, arXiv:2003.00104.
95. Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; I Bigorda, L.G.; Mestre, S.R.; Mas, J.; Mota, D.F.; Almazan, J.A.; De Las Heras, L.P. ICDAR 2013 robust reading competition. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, IEEE, Washington, DC, USA, 25–28 August 2013; pp. 1484–1493.
96. Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V.R.; Lu, S.; et al. ICDAR 2015 competition on robust reading. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE, Tunis, Tunisia, 23–26 August 2015; pp. 1156–1160.
97. Veit, A.; Matera, T.; Neumann, L.; Matas, J.; Belongie, S. Coco-text: Ddtaset and benchmark for text detection and recognition in natural images. *arXiv* **2016**, arXiv:1601.07140.
98. Ch'ng, C.K.; Chan, C.S. Total-text: A comprehensive dataset for scene text detection and recognition. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), IEEE, Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 935–942.
99. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
100. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.