IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 14, 2021, accepted July 17, 2021, date of publication July 27, 2021, date of current version August 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100717

Arabic Scene Text Recognition in the Deep Learning Era: Analysis on a Novel Dataset

HEBA HASSAN[1], **(Member, IEEE), AHMED EL-MAHDY**[1,2], **(Senior Member, IEEE), AND MOHAMED E. HUSSEIN**[2,3], **(Senior Member, IEEE)**
[1]Computer Science and Engineering Department, Egypt-Japan University for Science and Technology, Alexandria 21934, Egypt
[2]Department of Computer and Systems Engineering, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt
[3]Information Sciences Institute, University of Southern California, Los Angeles, CA 90007, USA

Corresponding author: Heba Hassan (heba.hassan@ejust.edu.eg)

ABSTRACT The problem of scene text recognition has recently gained extra attention, being an essential part of scene understanding systems. The broad scope of applications and the unresolved challenges has given this problem its popularity. However, the research focus has long been on languages with Latin characters while leaving behind other languages with different characteristics, such as the Arabic language. In this paper, we focus on Arabic scene text recognition and attempt to fill two main gaps regarding this research task. First, the Arabic language is lacking a publicly available benchmark dataset to compare different proposed methods on the same grounds. Therefore, we introduce a novel Arabic/English dataset: Everyday Arabic-English Scene Text dataset (EvArEST), to fill that need. Second, while deep learning methods have continuously evolved and pushed the state of the art in languages with Latin characters, their use for the Arabic language has been very limited. Therefore, we use our new dataset to evaluate the problem of Arabic scene text recognition from three perspectives: (1) using deep learning techniques and studying their suitability for Arabic scene text recognition, where we identify essential components required for the model to obtain good performance; (2) identifying Arabic text challenges that differ from Latin text and require special attention; (3) investigating a bilingual model that concurrently deals with Arabic and English words, since Arabic text is usually found along with other languages. We determine the best model to handle bidirectional text, its challenges, and possible ways to overcome them. We offer both Arabic and Bilingual text recognition results using EvArEST dataset for upcoming research to build upon and improve. We also point to directions for future research based on the analysis performed on the dataset. The dataset is publicly available at https://github.com/HGamal11/EvArEST-dataset.

INDEX TERMS Arabic scene text recognition, bilingual scene text recognition, deep learning, scene text recognition datasets.

I. INTRODUCTION

Text recognition in natural scenes is a vital component in image understanding systems since text is one of the most widely used means of communication and is around us everywhere. The problem of text recognition is a part of the text reading problem. Text reading starts with text detection, where text instances are located in the image, and then comes text recognition to convert those instances into readable words. Scene text reading has several applications in our daily lives, such as translation systems that could help overcome language boundaries and enable reading and translating text on the spot. Visual assistance could help the visually impaired with reading signs, ATM instructions, or books using text-to-voice systems. Other applications include intelligent inspection, multimedia retrieval, or product recognition.

Scene text recognition (STR) is a challenging problem in many aspects. In addition to the common challenges facing almost all computer vision tasks, such as image noise, scene complexity, viewpoint and brightness variations, text in natural scenes has its unique challenges. Text of any language typically has a large variety of font styles and shapes. On top of that, text in natural scenes exhibits other dimensions of variations due to artistic effects. For instance, it may appear in atypical orientations. It can also exhibit in-plane and

The associate editor coordinating the review of this manuscript and approving it for publication was Szidónia Lefkovits.

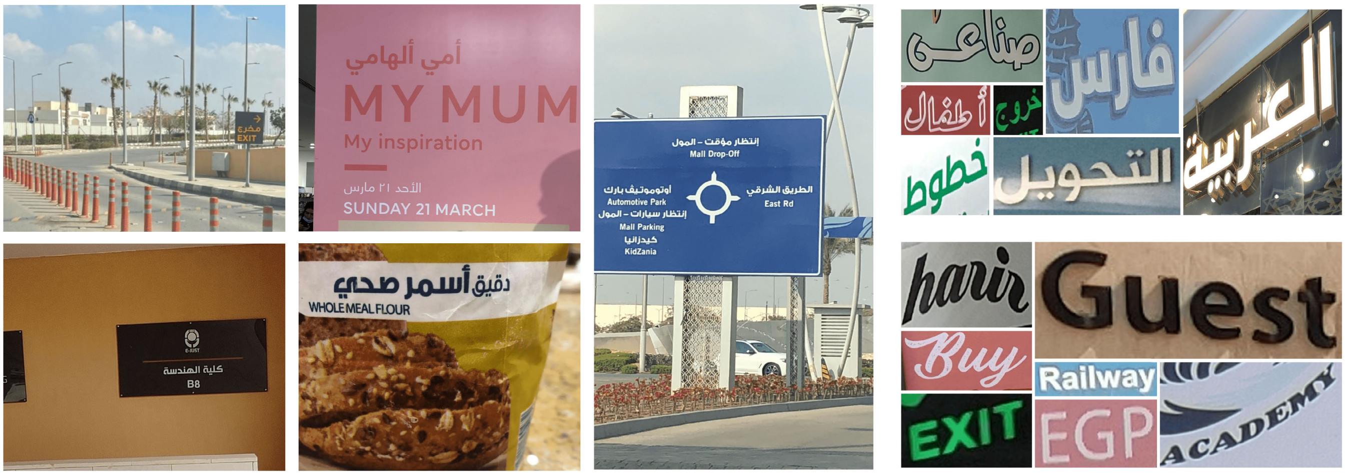


FIGURE 1. Samples from EvArEST dataset for Arabic–English scene text.

out-of-plane curvature and perspective transformation effects. All these factors mandate special attention to text recognition in natural scenes, which explains its popularity as a stand-alone problem in the research arena.

A typical deep-learning based STR framework has four main stages. It starts with preprocessing, where some models apply a transformation such as rectification [1], [2] to the image to ease the recognition. The second stage is feature extraction, where features from the text image are extracted typically using convolutional neural networks (CNN). Those features are then processed with a sequence processing network, and the word is predicted in the final stage. A comprehensive survey on recent STR techniques is offered elsewhere [3].

Much work has been recently proposed in STR [1], [2], [4]–[9], mainly focusing on languages written with Latin characters and, more precisely, English. However, other languages with different characteristics require more research effort. Moreover, with multilingual text being found in many cities nowadays, another problem that needs attention is multilingual text detection and recognition.

Arabic is the fifth most spoken language in the world and is the official language in 25 countries around the world. Arabic text is a cursive text. Therefore, in most cases, the characters are connected. Unlike English, where each character takes at most two shapes when its lower and upper case shapes differ from one another, Arabic characters can take up to four different shapes. Additionally, character shapes in Arabic can significantly vary from one another in size. Moreover, unlike English, in which character shapes depend on the type of the word and its location in the sentence, Arabic character shapes depend on their location in the word, which results in a higher difficulty predicting the character shape from the context. Furthermore, some Arabic characters are only distinguishable from one another by subtle differences, such as number and positions of dots. Another variation from English is the direction of the text, as Arabic is written from right to left. Fig. 4 presents examples of these variations, which have traditionally been handled in prior literature by adopting specialized techniques for Arabic text.

Reviewing the work done recently in Arabic STR [10]–[13], we identify two main issues. The first issue is the absence of publicly available datasets, such as the ones proposed for English. An important aspect to enrich the research in any language is to have benchmark datasets to serve as a references for all researchers. Most of the work done in Arabic text recognition uses different private datasets. Therefore, no fair comparison has been conducted to determine the relative quality of any of the proposed approaches. To deal with this issue, we propose the EvArEST (Everyday Arabic English Scene Text) dataset, which can serve as a benchmark for Arabic scene text recognition or as a bilingual dataset for Arabic-English scene text recognition. Some images from EvArEST are shown in Fig. 1.

The second issue with contemporary Arabic STR is that most of the work done uses special preprocessing or classical feature extraction methods with Arabic text, while all recent STR methods use deep learning. It is essential to use more generalized techniques such as deep learning to enhance the performance of Arabic STR and to be able to have a bilingual or multilingual model. In this paper, we apply multiple recently proposed methods for STR, which purely use deep learning models, to Arabic cursive text to observe how different techniques work with this kind of text. Another vital aspect to investigate is the possibility of having a bilingual model that could recognize Arabic and English words with no special preprocessing. This is essential due to the bilingual nature in many Arabic-speaking countries. While in many countries Arabic is the official language, other languages are often used in communication and public text. English is used in countries such as Egypt, Sudan, and some Arab Gulf states. French is often used in countries such as Morocco, Tunisia, Algeria, and Lebanon. While in Libya, Italian is the second language. Other languages such as Kurdish, Somali, and Farsi are also other examples of languages used in some countries whose main language is Arabic.

The contributions of this paper are as follows:

- We introduce a novel bilingual dataset that we call Everyday Arabic-English Scene Text Dataset (EvArEST). The dataset contains images with instances of Arabic and English words that are collected from many places under different conditions.
- We conduct comprehensive experiments and analysis on the problem of Arabic text recognition using a selected set of deep learning-based scene text recognition models to determine which works better with Arabic text. We also identify some key challenges associated with the Arabic language writing style.
- We investigate a bilingual text recognition model that could recognize two languages Arabic and English, using the same model without any special preprocessing. We also examine the challenges associated with this model and possible solutions for them.

The rest of the paper is organized a follows. In Section II, we review the recent work done in scene text recognition and its different stages, and also review the work done in Arabic scene text recognition. In Section III, we introduce the EvArEST dataset and its statistics. We then explain the STR framework and methodology used for our experiments in Section IV. Experiments and evaluations for Arabic STR are then presented in Section V. Next, in Section VI, we present the experiments and evaluation for the bilingual STR model. Finally, we provide the conclusion and suggested future work directions in Section VII.

II. RELATED WORK

In this section, we first review STR techniques, then we review the work done in Arabic STR and the existing Arabic datasets.

A. SCENE TEXT RECOGNITION

Before deep learning, scene text recognition was usually performed using hand-crafted features that utilize text characteristics to extract useful features from the image. Features such as Stroke Width Transform (SWT) [14] or Maximally Stable Extremal Region (MSER) [15] were used to detect characters. Alternatively, the characters were detected by using a sliding window equipped with a classifier to find the characters in the image [16], [17]. Those techniques start with the character to obtain the word in a bottom-up way.

The deep learning approach to the problem was different; it dealt with the word image as a whole in a top-down approach. One of the leading methods was the model proposed by Jaderberg *et al.* [18], where text recognition was treated as an image classification problem. A CNN was trained to classify 90k different words from a dictionary, where each word was considered a class. This approach achieved a breakthrough in the accuracy of text recognition systems, but the network could not recognize any out of dictionary words.

Sequential models, such as Recurrent Neural Network (RNN) [19], seemed more fitting than CNNs to the problem of text recognition to predict the word as a sequence. Shi *et al.* [4] introduced CRNN or Convolutional Recurrent Neural Network. In this network, CNN and RNN were combined. The image features were extracted by a CNN and then fed to an RNN as a sequence of features. Most of the research that followed this paper adopted the same strategy to deal with text recognition. This class of models has four main stages: preprocessing, feature extraction, sequence processing, and prediction, as shown in Fig. 2.

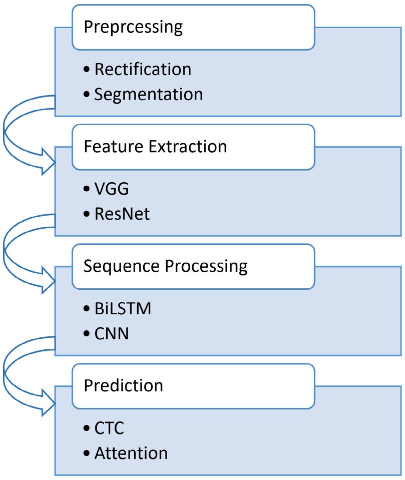


FIGURE 2. Stages of a typical scene text recognition system and examples for methods used in each stage.

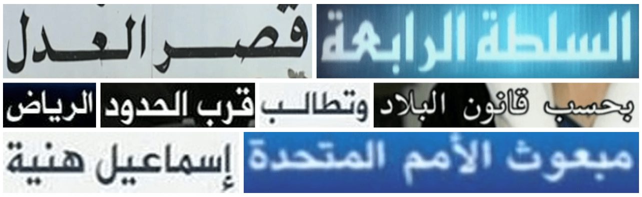
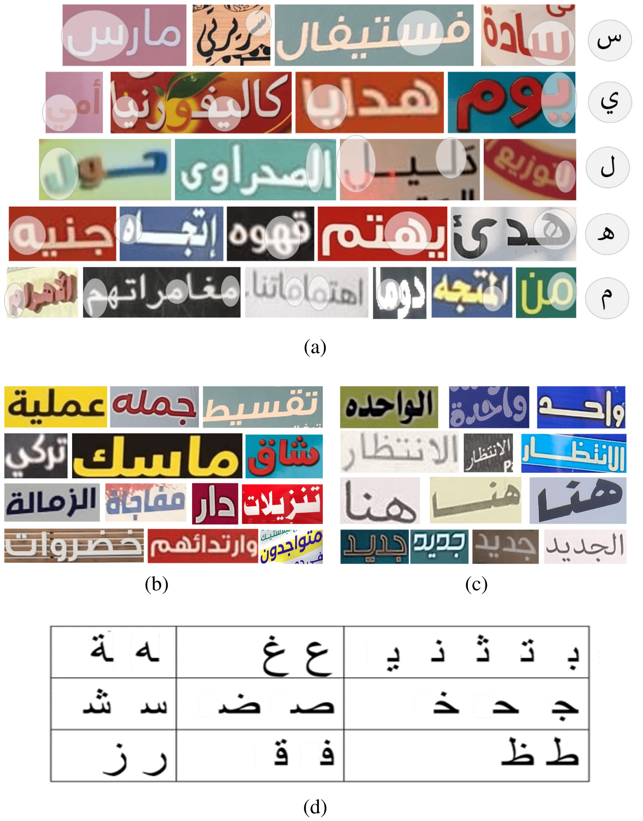


FIGURE 3. Examples from the ALIF dataset [20] for Arabic text in videos.

The most common form of preprocessing is the rectification process [2], [6], [7], [21], [22]. This is usually performed using a form of Spatial Transformation Network (STN) [23] to automatically learn a parametric transformation to the text image that leads to an improvement in the recognition accuracy. Thin-plate splines (TPS) [24] were also used [1], [2] as a form of nonlinear rectification to deal with different types of irregular text. Using rectification gave better results with irregular text. Another form of preprocessing is segmentation. Segmentation was used to detect the characters [9], and then a segmentation map with the characters' positions was used as the input to the next stage. Luo *et al.* [25] used a Generative Adversarial Network (GAN) to remove the background of the text before feature extraction.

Feature extraction is usually carried out using a CNN, such as a VGG network [26] or a ResNet network [27]. VGG was often used [1], [4], [5], [28], [29] as it offers a lightweight network for features extraction, while others [2], [6], [30], [31] used the more complex ResNet network for better feature representation. Recursive CNN [32], [33] was used to reduce



(a)

(b)

(c)

(d)

FIGURE 4. Variations in Arabic writing style, (a) Different character shape according to its place in the word, (b) Words with one ligature or multiple ligatures, (c) Word shape varies with different fonts and character stretches, (d) Examples of characters with the same shape but different dots.

memory consumption by having a deeper CNN with the same number of parameters through weight-sharing. The features are then fed to the next stage for sequence processing. This is usually done using a bidirectional long short term memory (BiLSTM) [34]. BiLSTM was used in many STR systems [1], [2], [4], [6], [9], [22], [33] to model the sequential features to enable the prediction of the word character by character. However, in some cases [30], no sequence model was used and only CNN was used to reduce inference time.

In prediction, two main approaches are used. First, there is the Connectionist Temporal Classification (CTC) loss [35]. Many models used the CTC loss [2], [4], [30], [33] to predict text strings from the sequence of features without prior alignment. CTC maximizes the likelihood of the predicted output by calculating the probability for all possible input-output alignments. However, it increases the computation cost, especially for long sequences. The other approach is attention-based prediction [36], which is usually used with RNNs. Attention was used to learn the alignment between the input features and the text [1], [7], [9], [22], [32] without additional computation like CTC.

B. ARABIC SCENE TEXT RECOGNITION

Much work has been done on Arabic printed text in documents and on handwritten text [37]; however, the field of text detection and recognition in natural scenes has not received the same attention yet. Yousfi *et al.* [20] proposed the ALIF dataset for text recognition in videos of news broadcasts. They used a sliding window to detect the characters with two models for feature extraction: CNN and Deep Belief Networks (DBN). Then, a BiLSMT-CTC schema is used to transfer the features into a sequence of characters. This work was then followed up by adding a large-scale language model [11]. Zayene *et al.* [10] introduced AcTiV dataset, which is a larger dataset for text recognition in videos.

Text in video datasets offered a basis to take off, but this kind of text, as seen in Fig. 3, is different from the more general scene text. Text in news videos does not offer much variability in backgrounds and fonts. It also does not contain text in real-life situations with different illumination conditions, background noise, and perspective distortion. Ahmed *et al.* [13] introduced the EASTR dataset for text detection and recognition with text from natural scenes. They used MSER features with LSTM and performed their analysis on their dataset; which is not publicly available. ARASTI dataset was proposed by Tounsi *et al.* [38] for Arabic scene text recognition, yet again the word dataset was not found publicly available. Only the part of the dataset for Arabic characters is available.

Most of the work recently proposed in Arabic text recognition has either used text in video datasets or a private dataset that is not publicly available. In this paper, we propose a novel dataset that follows the same format as multiple publicly available English datasets. We also apply recent models covering different STR model component combinations to determine what fits Arabic text best. We also report the results obtained using those models to serve as a reference for future research.

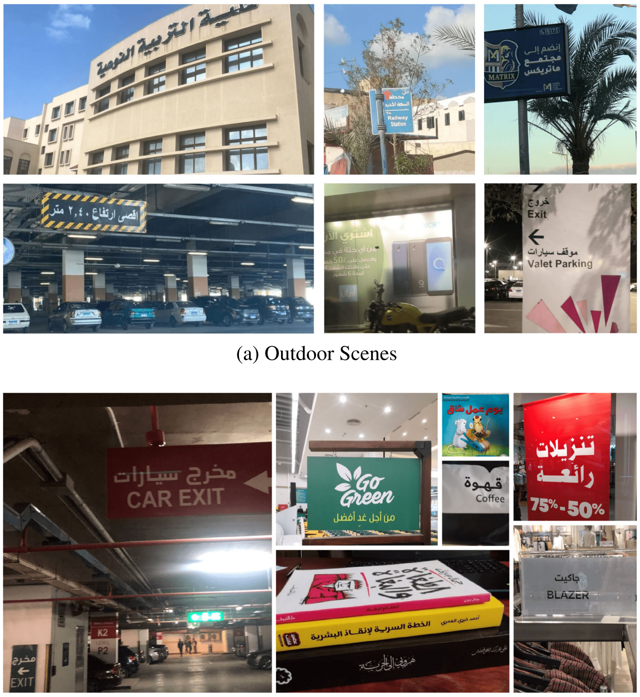
III. EvArEST: EVERYDAY ARABIC-ENGLISH SCENE TEXT DATASET

Arabic is spoken by around 422 million speakers around the world. Other languages such as Urdu and Farsi use a similar set of characters, and they are all written from right to left with a cursive nature, unlike Latin characters. We introduce EvArEST or Everyday Arabic-English scene text dataset to enrich the field of Arabic text reading and provide a publicly available dataset for future research. The dataset contains 7102 cropped word images extracted from 510 scene images with Arabic and English word instances. We perform experiments on the cropped words for Arabic STR and bilingual STR.

In this section, we will discuss the Arabic text writing style and the challenges it brings to the problem of scene text recognition. We also explain how the data is gathered and the diversity in the collected images. This is followed by the dataset statistics and the ground-truth format.

A. ARABIC WRITING STYLE

Arabic is a cursive language, meaning that most of the characters in one word are connected. Some characters cannot



(a) Outdoor Scenes

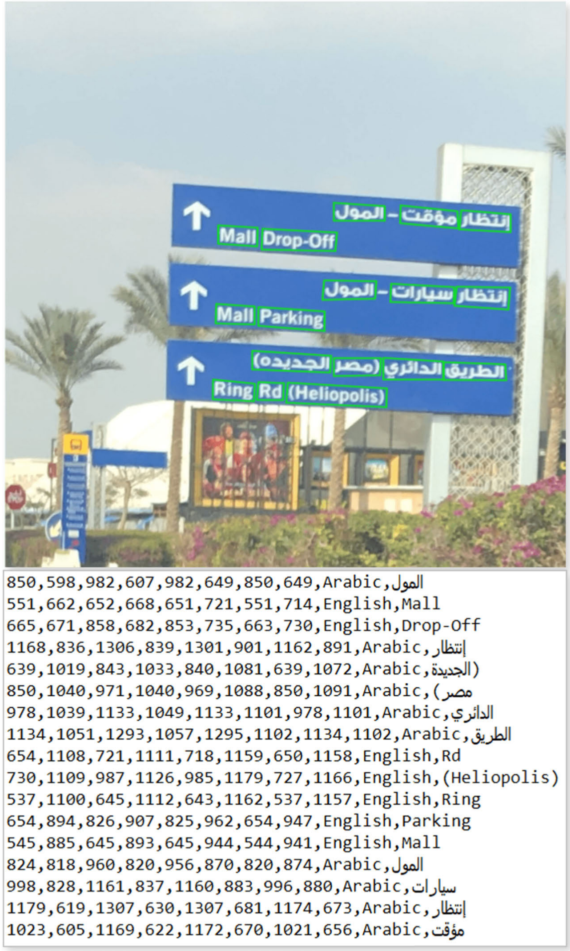
(b) Indoor Scenes

FIGURE 5. **Examples from the EvArEST dataset under different conditions.**

connect to others, which means that one word could consist of one ligature, two, or more as seen in Fig. 4b. The character's shape varies according to its place in the word and could vary from one font to the other. In Fig. 4a, we observe examples of some characters in different places in the word. Also, in some words, a character could be stretched, which changes its shape as seen in Fig. 4c. All the reasons mentioned above increase the intra-class variation, making deep learning a suitable choice for recognizing Arabic text. Another challenge with Arabic text is the presence of dots. Many characters could have the same shape but have different numbers or different places for dots as shown in Fig. 4d. That could lead to many confusions between characters, and maybe a background noise could be mistaken for a dot.

B. DATA COLLECTION

We asked several volunteers in different cities in Egypt to collect images of Arabic scene text from different places around them, either indoors or outdoors. The images were collected in an uncontrolled environment with mobile phone cameras of different resolutions. We had to ensure that the dataset covers a wide range of fonts, lighting conditions, background complexities, and perspective deformation when choosing the images. The dataset has images taken outdoors with different lighting conditions. Those images include images of billboards, road signs, places' names, etc. Indoor images include images taken in public places and images of everyday used items that contain text. Sample images from the dataset with different properties are shown in Fig. 5. The images in our dataset contain text instances of Arabic and English.



```
850,598,982,607,982,649,850,649,Arabic,المول
551,662,652,668,651,721,551,714,English,Mall
665,671,858,682,853,735,663,730,English,Drop-Off
1168,836,1306,839,1301,901,1162,891,Arabic,إنتظار
639,1019,843,1033,840,1081,639,1072,Arabic,الجديدة)
850,1040,971,1040,969,1088,850,1091,Arabic,(مصر
978,1039,1133,1049,1133,1101,978,1101,Arabic,الدائري
1134,1051,1293,1057,1295,1102,1134,1102,Arabic,الطريق
654,1108,721,1111,718,1159,650,1158,English,Rd
730,1109,987,1126,985,1179,727,1166,English,(Heliopolis)
537,1100,645,1112,643,1162,537,1157,English,Ring
654,894,826,907,825,962,654,947,English,Parking
545,885,645,893,645,944,544,941,English,Mall
824,818,960,820,956,870,820,874,Arabic,المول
998,828,1161,837,1160,883,996,880,Arabic,سيارات
1179,619,1307,630,1307,681,1174,673,Arabic,إنتظار
1023,605,1169,622,1172,670,1021,656,Arabic,مؤقت
```

FIGURE 6. **An example of an image and its ground-truth format from the dataset.**

Text content in many places would usually include the two languages or just one of them.

C. DATASET STATISTICS

We collected 510 images, all containing one or more instances of text. The images have outdoor and indoor text images as seen in Fig. 5 with English and Arabic words. The statistics of the word content of the dataset are shown in Table 1. Every word is annotated with a four-point polygon and not a rectangle to better represent irregular text. The given points start with the top left corner of the word and follow in the clockwise direction. The annotations for the images are provided in the same format as ICDAR datasets [39], [40]. Each image comes with a text file containing three elements: the four points polygon that contains the word, the language of the word, and the text, as seen in Fig. 6.

The polygons were used to extract the word instances from the images. We ended up with 7102 cropped word images of both Arabic and English languages. The distribution of the two languages and the split of the data is shown in Table 1. As seen in Fig. 7 the dataset covers a wide range of varieties in text. The dataset includes regular and irregular text,

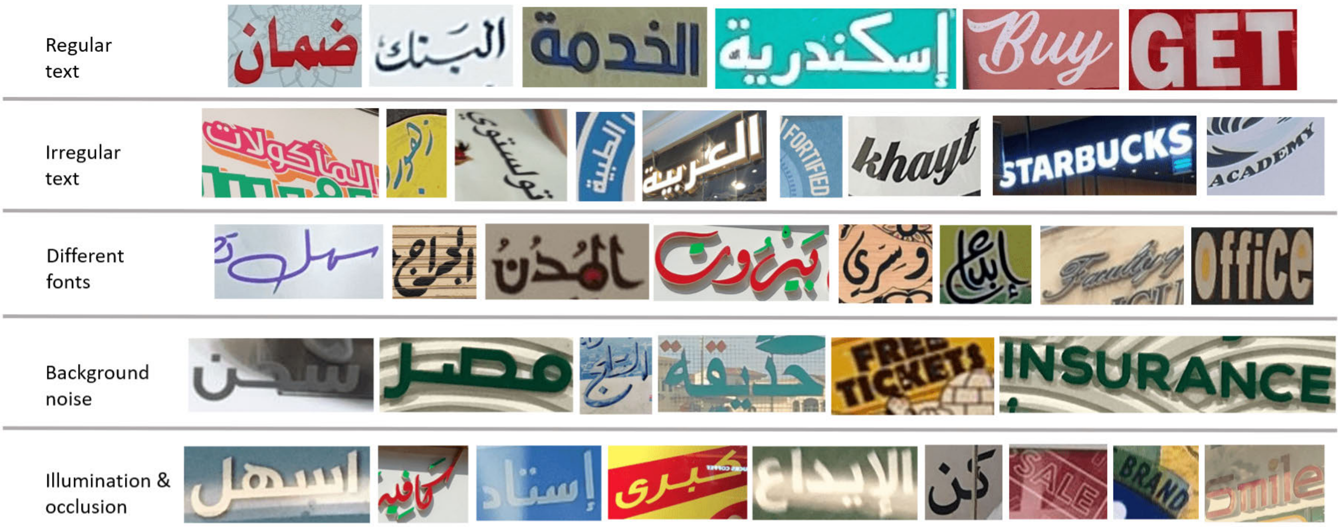


FIGURE 7. Images from EvArEST with different properties.

TABLE 1. Word count for each language and the test-train splits in the EvArEST dataset.

Language	No. Words	Test	Train
Arabic	5337	1150	4187
English	1765	498	1267
Bilingual	7102	1648	5454

different illumination conditions, different fonts, cluttered backgrounds, and occluded text. The ground-truth for the cropped words is given by a text file, in which each line refers to a word image name and the text in the image.

The dataset could be used for Arabic text recognition only and could be used for bilingual text recognition. We later provide the word accuracy results we obtained by applying a selected number of models of the recent work done in STR to this dataset. Furthermore, we also show our experiment with a bilingual text recognition model to study the possibility of having both English and Arabic languages combined in one model.

IV. METHODOLOGIES

As mentioned before, STR has four stages for text prediction. A number of representative recent papers were chosen to cover various settings for these different stages, as summarized in Table 2. Here we describe the different methods we used for each stage in the STR system.

A. PREPROCESSING

Two main techniques were used as a preprocessing step before feature extraction: rectification and segmentation. Rectification is usually done using Spatial Transformation Network (STN) [23] to transform the text image into a form better for word prediction. The network predicts a transformation matrix to apply to the image before the next step of the model. Thin Plate Splines (TPS) are used to apply nonlinear rectification to the image before feature extraction [1], [2]. Luo *et al.* [7] perform the rectification by predicting a position offset map that is applied to the text image before the next step. Another type for preprocessing is done by using segmentation [9], where a semantic segmentation network is used to predict a map with the characters' position and classes before the feature extraction stage.

TABLE 2. The different models used in our evaluation and the methods used in each stage for each model.

Method	Preprocessing	Feature Extraction	Sequence Modeling	Prediction
CRNN [4]	None	VGG	BiLSTM	CTC
RARE [1]	Rectification	VGG	BiLSTM	Attention
R2AM [32]	None	RCNN	None	Attention
STARNET [2]	Rectification	ResNet	BiLSTM	CTC
GRCNN [33]	None	RCNN	BiLSTM	CTC
Rosetta [30]	None	ResNet	None	CTC
WWSTR [22]	Rectification	ResNet	BiLSTM	Attention
Moran [7]	Rectification	VGG	BiLSTM	Attention
SCAN [9]	Segmentation	VGG	BiLSTM	Attention

B. FEATURE EXTRACTION

At this stage, features are extracted from the text image using a convolutional network. Different convolutional networks are used to extract the features from the text image, such as VGG [1], [4], [7] or ResNet [2], [30]. Lee and Osindero [32] used recursive CNN to obtain a deeper network with the same number of parameters, inspired by the recurrent convolutional neural network (RCNN) [41]. Gated RCNN (GRCNN) is introduced by Wang and Hu [33]. They introduced a gated recurrent convolutional layer to control the context information in RCNN.

C. SEQUENCE PROCESSING

Sequence processing captures the contextual information from the visual features obtained from the feature extraction stage to predict the presented words. BiLSTM is used in many STR models [1], [2], [4], [7], [9]. Others argued that BiLSTM is time consuming and used features from CNN only. Borisyuk *et al.* [30] only used ResNet without sequence modeling to obtain a faster network. Lee and Osindero [32] only used features from RCNN with attention.

D. PREDICTION

Two main methods are used for the prediction stage: attention mechanism or CTC loss. CTC was first introduced by Graves *et al.* [35] and has accomplished great results in fields such as voice recognition and handwritten text recognition. It enables alignment between the input and the output by calculating every possible input-output sequence alignment probability. Many STR models use CTC to obtain the character sequence [2], [4], [30], [33]. Attention is used to learn the alignment between the input and the output sequence. Typically, attention is used with BiLSTM to learn the alignment between the input features and the sequence of characters [1], [7], [9].

V. ARABIC SCENE TEXT RECOGNITION

Here, we apply the selected methods from the literature to the problem of Arabic scene text recognition. We present the obtained results and discuss the suitability of these methods with a cursive language, such as Arabic. For the implementation, we used the code from Baek *et al.* [22]. They provided an implementation for different methods in each stage of STR. For the other studied methods not included in this framework, the publicly available author implementations were used.

A. ARABIC DATASETS

Here, we list the data used for training and testing the models only for the Arabic language:

- **SynthText Arabic**: The dataset consists of about 50k images for text detection with about 245k cropped word images. The images are generated with a modified code from SynthText [42]. The data was released with the ICDAR MLT dataset [43].
- **Generated Synthetic Data**: We generated 200k images with segmentation maps to be used for training the model that requires segmentation. The code generates an Arabic word and its segmentation map, applies geometric transformations to the text, and then embeds it in a randomly chosen background. Each image has a segmentation mask ground-truth along with the word ground-truth. Samples of these images are shown in Fig. 8. This data is publicly available with EvArEST dataset.
- **MLT**: The ICDAR19 Multi-lingual scene text detection and recognition dataset [43] has 10k images of 10 different languages. The training dataset has 1000 images that contain instances of Arabic text. We extracted cropped word images with Arabic text, obtaining 4334 real images for training.

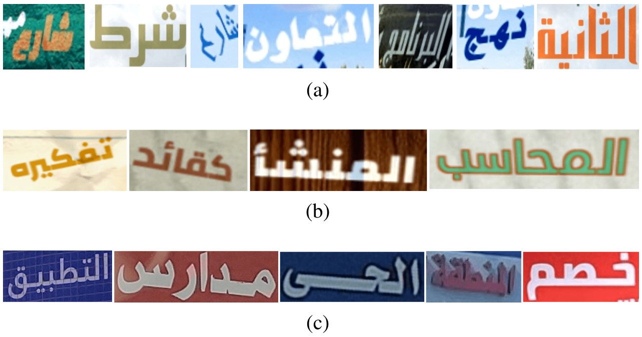


FIGURE 8. (a) Arabic SynthText, (b) Generated synthetic data, (c) Real data from (EvArEST).

- **EvArEST-Ar**: The part of EvArEST dataset with Arabic text images, it has 4187 real images for training and 1150 for testing.

B. TRAINING

All images were resized to 64 × 256 with a maximum sequence length of 32 characters. We used both synthetic and real data for training the models. All models were trained for the same number of iterations, starting with the synthetic data, the model is trained for 150k iterations and then using the real data from EvArEST and MLT datasets for 50k iterations with a batch size of 32. We used 40 classes: 29 classes for the Arabic characters, 10 classes for the numbers from 0 to 9, and one class for special characters.

C. EXPERIMENTS AND ANALYSIS

1) EXPERIMENTAL RESULTS

We obtain the results of applying nine recent methods from the literature to the proposed Arabic text recognition dataset from the EvArEST dataset. The word accuracy results, the techniques used in each of the nine methods, and the number of parameters for each method can be seen in Table 3. From the table, we can observe that the method WWSTR [22] achieves the best accuracy of 91.2%. This method has rectification along with BiLSTM and attention and uses ResNet for feature extraction. In the second place, comes the RARE method [1] with 89.8% accuracy, the method also has rectification, BiLSTM, and attention but uses VGG for feature extraction. Therefore, we can conclude that ResNet delivers a better representation for the image but at the expense of model complexity, as noticed from the number of parameters.

The results are close in most models, but we can notice that a crucial component with Arabic text is the BiLSTM for sequence processing. The two methods without BiLSTM obtain the lowest accuracies of 85.4 and 84.0. We can also observe that the method SCAN [9] got good results, even though it uses character segmentation, which is more difficult with cursive text. That indicates that deep learning models can probably handle cursive text with the same efficiency as non-cursive text.

Observing the model's size, we notice that the model with the highest accuracy is also the one with the highest number of

TABLE 3. The accuracy results for Arabic text recognition from the EvArEST-Ar dataset.

Method	Rectification	Segmentation	BiLSTM	CTC	Attetnion	Accuracy %	Param. $\times 10^6$
CRNN [4]	✗	✗	✓	✓	✗	86.5	8.4
RARE [1]	✓	✗	✓	✗	✓	89.8	10.8
R2AM [32]	✗	✗	✗	✗	✓	84.0	2.8
Star-Net [2]	✓	✗	✓	✓	✗	89.6	48.8
GRCNN [33]	✗	✗	✓	✓	✗	87.4	4.7
Rosseta [30]	✗	✗	✗	✓	✗	85.4	44.2
WWSTR [22]	✓	✗	✓	✗	✓	**91.2**	49.5
Moran [7]	✓	✗	✓	✗	✓	89.4	20.3
SCAN [9]	✗	✓	✓	✗	✓	88.4	17.7

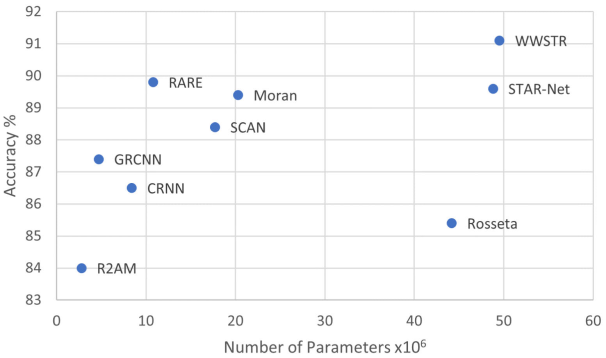


FIGURE 9. Accuracy vs. model size for Arabic scene text recognition on EvArEST dataset.

parameters, as seen in Fig. 9. The best choice for accuracy and memory consumption together is the RARE model [1]. It has rectification, BiLSTM, and attention, just as WWSTR. The only difference is that WWSTR uses ResNet, while RARE uses VGG.

2) TRAINING DATA ANALYSIS

We started training the model using synthetic data, and then we used the real data to fine tune the model. We reported the results after training with synthetic data and then again after fine tuning with real data. As seen in Table 4, the accuracy significantly improves after training with real data. In other STR work, e.g. [22], we can notice the effect of training with real data and how it improves the accuracy. However, in the Arabic case, we can see that training with synthetic data gave very low accuracy, and we think it is due to the quality of the synthetically rendered text. As we can see in Fig. 8, our synthetically generated Arabic text, in many words, looks disconnected and different from real text. The cursive nature of the Arabic text makes it hard to render text character by character in the precise place.

3) FAILURE CASE ANALYSIS

We view some of the failure cases from the model WWSTR [22] with the highest accuracy. Some failure cases are known to be common challenging shapes for all STR systems, and some are due to properties of Arabic text. We categorized the main reasons for wrongly recognizing words as follows:

TABLE 4. Arabic STR accuracy when training on synthetic data only, and accuracy after training on real data.

Method	Accuracy % Synthetic	Accuracy % Real
CRNN	67.2	86.5
RARE	69.5	89.8
R2AM	64.0	84.0
Star-Net	71.5	89.6
GRCNN	68.1	87.4
Rosseta	66.3	85.4
WWSTR	71.8	91.2
Moran	68.5	89.4
SCAN	68.1	88.4

Low Resolution: Images with low resolution are challenging for any STR system and could require special preprocessing.

Rotated Text: The dataset has many examples for rotated text, and by using preprocessing, many of these examples are correctly recognized. However, some of the images with more significant rotations are still hard to recognize.

Occlusion: Some occluded text could be recognized using context and the language model that the network learns. While in other cases, the model is unable to recognize the word correctly.

Difficult Fonts: Some fonts make it difficult to correctly recognize the characters, and hence cause the model to confuse one character with another. Some calligraphic fonts could also be hard even for humans to read.

Unusual Character's Shape: We earlier discussed the large intra-class variability among the Arabic characters. This could lead to difficulty in recognizing some characters, especially when they take unusual shapes due to calligraphic effects.

Misrecognized Dots: As mentioned earlier, a major problem when dealing with Arabic text is that many characters could have the same shape but different locations and/or numbers of dots. With low resolution, occlusion, background noise, or uncommon fonts, such characters can be confused, and hence the word is wrongly recognized. In Fig. 11, we display some of these words to show how dots could affect the recognition.

Background Noise: Some cluttered backgrounds could lead to misrecognizing a word. Also, when combined with the dots problem, any noise in the background could be misclassified as a dot.

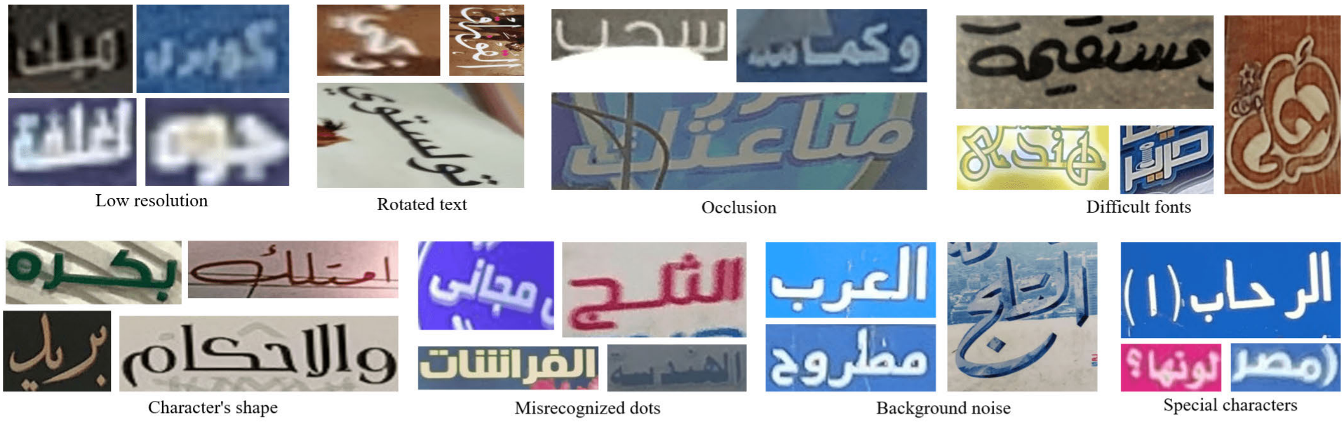


FIGURE 10. Failure cases for Arabic STR using WWSTR model.

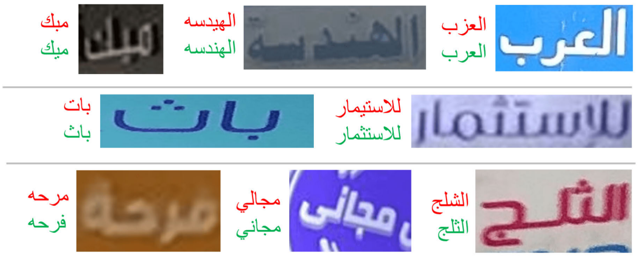


FIGURE 11. Characters misrecognized because of misrecognized dots. Misrecognized word in red and ground truth in green.

Special Characters: Some special characters are sometimes recognized as alphanumeric characters.

In Fig. 10, we present examples for each of the problems mentioned above.

VI. BILINGUAL SCENE TEXT RECOGNITION

In many modern cities, text from more than one language can often be found within the same scene. Having a model that can simultaneously recognize multiple languages would facilitate text recognition in these situations. In this section, we analyze the possibility of having a bilingual model that can recognize both Arabic and English words. Combining Arabic and English is particularly interesting because of the significant difference between these two languages, the most important of which is having completely different character sets. Considering that each of the two languages share its character set with many other languages, the success of such bilingual model opens the door for developing multi-lingual models that leverage the union of the two character sets.

A. DATASETS

We talked earlier about the Arabic datasets used; here we talk about the English datasets used in the bilingual and English models. The bilingual dataset used for evaluation is combined Arabic and English words from EvArEST with 1648 images for testing. The English datasets used for training and testing are:

- **Synthtext**: The dataset [42] is a large synthetic dataset. It contains about 8M cropped word images. We used only 1M images from this dataset when training the bilingual model.
- **COCO-Text**: This is one of the largest real datasets for text recognition [44]. It contains real images with text annotations. The dataset has about 42k images that we used for training.
- **EvArEST-En**: The part of EvArEST dataset with English text images, which has 1267 English words for training and 498 English words for testing.
- **IIIT5k-Words**: This dataset [45] has regular, curved, and perspective text. It has 3000 images for testing and 2000 images for training.

B. TRAINING

The idea here is to train the model to recognize Arabic and English words without a special preprocessing. This could be done by using the characters from both languages as the classes and training with samples from each language at each iteration.

For each of the evaluated models, the network is trained using Arabic and English words with 50% for each language in a batch. Following a similar training strategy to the Arabic model, we started with training for 150k iterations with synthetic data, and then real data were used for 100k iterations. All the images were resized to 256 × 64 and the maximum sequence length is 32 characters. In these models, we have 66 classes, 29 classes for Arabic characters, 26 classes for English characters, 10 classes for numbers 0 to 9, and one class for special characters.

C. EXPERIMENTS AND ANALYSIS

1) EXPERIMENTAL RESULTS

We test the model with EvArEST's bilingual dataset and the IIIT5K dataset. To understand the effect of training with the two languages together, we need to obtain each test dataset's results in bilingual and monolingual models. We already trained each of these models using only Arabic data, so we

TABLE 5. The accuracy result for Arabic, English, and Bilingual datasets. For each model, the first row shows the accuracy of the monolingual model, trained separately for each language, the second row (B1) is the accuracy using the bilingual model with both language's ground truth made to follow the same direction (left-to-right), and the third row (B2) is the accuracy results using the bilingual model with each language's ground truth left in its original direction (left-to-right or right-to-left).

Method	Preprocessing	BiLSTM	Attention	English		Arabic	Bilingual
				IIIT5K	EvArEST-En	EvArEST-Ar	EvArEST-B
CRNN CRNN-B2 CRNN-B1	✗	✓	✗	87.9 87.1 87.5	90.9 89.9 90.3	86.5 54.3 79.7	- 72.1 85.0
RARE RARE-B2 RARE-B1	✓	✓	✓	91.3 91.3 91.5	91.9 89.9 92.1	89.8 83.1 84.7	- 86.5 88.4
R2AM R2AM-B2 R2AM-B1	✗	✗	✓	84.2 81.8 83.4	88.6 86.2 84.1	84.0 67.8 71.2	- 77.0 77.7
Star-Net Star-Net-B2 Star-Net-B1	✓	✓	✗	91.5 91.3 91.8	92.5 91.3 91.7	89.6 69.0 81.3	- 80.2 86.5
WWSTR WWSTR-B2 WWSTR-B1	✓	✓	✓	93.7 93.1 93.4	94.2 92.0 94.5	91.2 87.2 87.2	- 89.6 **90.8**
SCAN SCAN-B2 SCAN-B1	✓	✓	✓	92.6 88.0 89.5	92.0 89.3 90.2	88.4 85.6 84.7	- 87.5 87.5

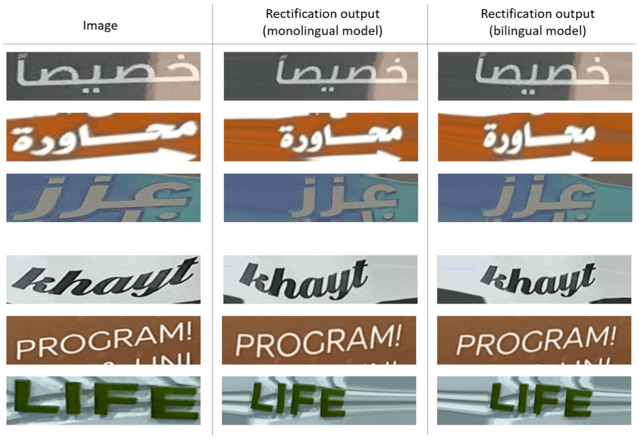


Image
Rectification output (monolingual model)
Rectification output (bilingual model)

FIGURE 12. Rectification output from monolingual and bilingual WWSTR model.

have the results for the Arabic models. What is missing is to train a monolingual model for the English language using the same English data used for the bilingual model. We calculated the accuracy for one Arabic dataset, which is EvArEST-Ar, and two English datasets, which are EvArEST-En and IIIT5k. Again, here, the best model was WWSTR as it obtained 90.8% accuracy in the bilingual dataset, followed by RARE, whose accuracy reached 88.4%.

2) EFFECT OF TEXT DIRECTION

An important point to consider when training a model with these two languages is that each of them has its writing direction; Arabic is written from right to left, while English is written from left to right. A simple way to deal with this problem is to flip the ground-truth for the Arabic words so that they are predicted from left to right. In Table 5, we can see the results for some of the models we used earlier that cover the methodologies variations to we want to test using the two ways to handle the writing direction's difference. Our experiment tested the models using the word's ground-truth as it is, having two different directions, either left to right or right to left according to the language. And, we also tested them when trained after flipping the Arabic word's ground-truth, to predict the words from one direction for the two languages.

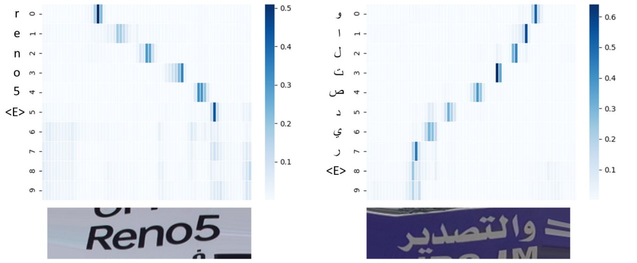
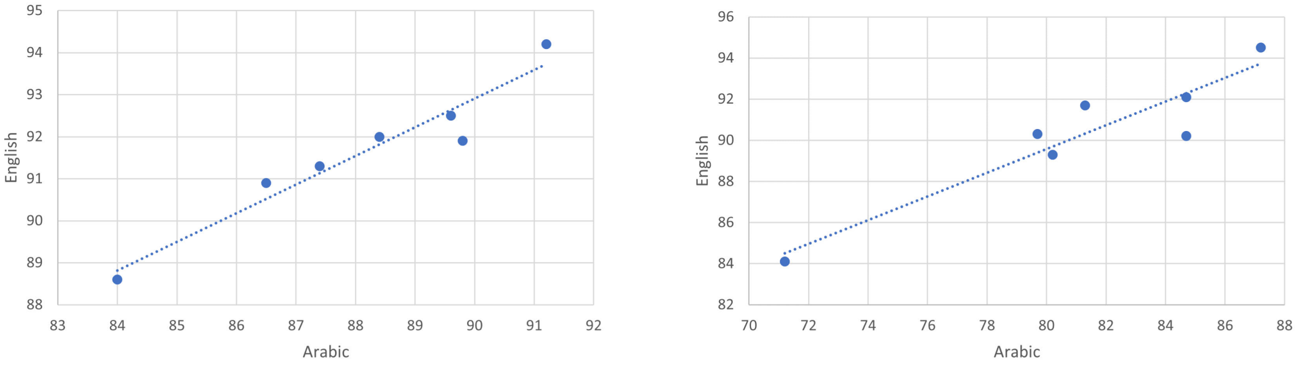


FIGURE 13. Attention weights for Arabic and English examples using WWSTR Bilingual model with two directions.

As noticed from Table 5, some models were able to adapt to the two languages, each having its directions, others could not. While BiLSTM is needed to handle the two languages using the same model, attention is required to process the two languages, each from a different direction. And in general, the results were better when unifying the direction of the word prediction. For example, the model R2AM, which does not include BiLSTM, achieved poor results in one direction and two directions. On the other hand, the models with BiLSTM and attention obtained good performance using one and two directions. Models such as CRNN, StarNet, and GRCNN obtained fair results when the two languages were trained in the same direction, while they failed to retain the same results when using two directions. The common characteristic



(a) Monolingual Models

(b) Bi-Lingual Models

FIGURE 14. **Correlation between Arabic and English accuracy results. (a) Arabic and English results in monolingual models, (b) Arabic and English results in bilingual models.**

among these three models is missing the attention module, which appears to help in adapting the model's output to the language direction.

To understand the role of the attention part in the bilingual model, we visualized the attention weight when predicting both Arabic and English words in the WWSTR model. As seen in Fig. 13, the attention weights for the feature vector of the image adapt to the direction of the text, which makes the model able to recognize both English and Arabic words even when each has a different direction. Also, when viewing the output from the rectification stage of the same model, we can notice that the model aligns the text in the image according to the language direction. As seen in Fig. 12, while it takes a left or a right direction in the monolingual model, it aligns the text in the center in the bilingual model.

3) CORRELATION BETWEEN ARABIC AND ENGLISH RESULTS

When we observe the accuracy of English and Arabic, either using monolingual training or bilingual training, we can notice that the two results are correlated in most of the models. From Fig. 14 we can observe that the accuracy for the two languages is positively correlated. If a model obtains a higher accuracy in English, it will most likely obtain higher accuracy in Arabic. This indicates that the advantage of one model compare to another is more related to the way each model handles the many common challenges in STR in general rather than the way it handles the specifics of the target language.

4) CONFUSION BETWEEN THE TWO LANGUAGES

We examined the prediction results from the best-performing bilingual model to determine if any confusion happened between the two languages, meaning that a word from one language is predicted as a word from the other language. For Arabic, we found that confusion only happened in 19 words out of 1150 words (1.6%), some of these words can be seen in Fig. 15. As for English, no confusion happened. An interesting point is that most of the misrecognized words had characters from only one language. That shows that the model learned to separate the two languages and predict a word that has either all English or all Arabic characters. In the first word in Fig. 15, one Arabic character was recognized as the English character (a), this is an example of a word recognized with mixed characters. The image has low resolution, and that character occurs in a single-character ligature; which could explain this error. Other examples from the figure show that some Hindi numerals are confused with English letters with similar shapes. This type of error is not expected to happen if the context is considered, i.e. when the numerals appear in the context of other Arabic words.

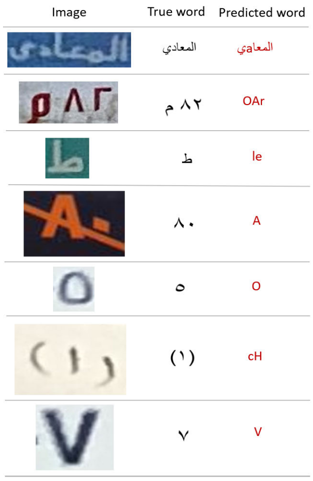


Image	True word	Predicted word
	المعادي	المعaي
	٨٢ م	OAr
	ط	le
	٨٠	A
	٥	O
	(١)	cH
	٧	V

FIGURE 15. **Arabic characters misrecognized as English characters.**

VII. CONCLUSION

In this paper, we introduce a novel dataset, EvArEST, for bilingual Arabic-English text in the hope of advancing research in the field of Arabic and bilingual scene text recognition. We also assess the performance of different deep learning-based models in Arabic and bilingual text recognition using the proposed dataset. Our evaluation establishes a benchmark for future research in this area.

For Arabic STR, different deep learning-based models are evaluated to determine the best that fits the problem. The models that obtained the best performance were the models with rectification, BiLSTM, and attention, while the models with no BiLSTM obtained the lowest accuracy. One crucial factor that affects the performance is the data used for training. We used synthetic and real data for training, and we found that real data could noticeably enhance recognition performance. However, comparing the number of available real data for Arabic with English data, we can predict that the performance on Arabic can approach the performance on English if comparable amounts of training data become available for Arabic. Also, the quality of the synthetic data is another factor that potentially contributes to the gap between the performance on Arabic vs. English. More realistic synthetic data generation for cursive languages, such as Arabic, is expected to add a significant boost to the performance of STR on such languages.

Even though Arabic is a challenging language with many variations in writing style, deep learning techniques with the right data can handle many of those challenges well. In addition, the generalization property that deep learning techniques offer would facilitate using the same model to recognize other languages with Arabic. However, other challenges such as dots misrecognition and unusual character shapes might require special attention when dealing with Arabic text.

Combining English and Arabic training in one model showed the possibility of having a bilingual model with no special preprocessing. Two components were found essential to handle bidirectional text. First, BiLSTM is needed to recognize the two languages using the same model. Second, attention enabled the model to handle bidirectional and unidirectional text with the same efficiency. However, the difference in accuracy between the bilingual and monolingual models, especially for Arabic, leaves the door open for further research to obtain matching performance for the bilingual and monolingual models.

Another critical problem is text detection, the task that precedes text recognition in the scene text recognition framework. In EvArEST, we provide the ground-truth for the text instances polygons to enable the usage of the dataset for bilingual text detection or even end-to-end STR with the detection and recognition tasks.

ACKNOWLEDGMENT

The authors would like to express their great appreciation to Dr. Marwan Torki (Alexandria University) for his valuable ideas and suggestions during the planning of this work. His willingness to give his time so generously is very much appreciated.

REFERENCES

[1] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4168–4176.

[2] W. Liu, C. Chen, K.-Y. Wong, Z. Su, and J. Han, "STAR-Net: A SpaTial attention residue network for scene text recognition," in *Proc. Brit. Mach. Vis. Conf.*, vol. 2, 2016, p. 7.

[3] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild: A survey," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–35, 2021.

[4] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[5] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5076–5084.

[6] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[7] C. Luo, L. Jin, and Z. Sun, "MORAN: A multi-object rectified attention network for scene text recognition," *Pattern Recognit.*, vol. 90, pp. 109–118, Jun. 2019.

[8] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "SEED: Semantics enhanced encoder-decoder framework for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13528–13537.

[9] H. Hassan, M. Torki, and M. Hussein, "SCAN: Sequence-character aware network for text recognition," in *Proc. 16th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2021, pp. 602–609.

[10] O. Zayene, J. Hennebert, S. M. Touj, R. Ingold, and N. E. Ben Amara, "A dataset for Arabic text detection, tracking and recognition in news videos-AcTiV," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 996–1000.

[11] S. Yousfi, S.-A. Berrani, and C. Garcia, "Contribution of recurrent connectionist language models in improving LSTM-based Arabic text recognition in videos," *Pattern Recognit.*, vol. 64, pp. 245–254, Apr. 2017.

[12] O. Zayene, S. M. Touj, J. Hennebert, R. Ingold, and N. E. Ben Amara, "Multi-dimensional long short-term memory networks for artificial Arabic text recognition in news video," *IET Comput. Vis.*, vol. 12, no. 5, pp. 710–719, Aug. 2018.

[13] S. B. Ahmed, S. Naz, M. I. Razzak, and R. B. Yusof, "A novel dataset for English-Arabic scene text recognition (EASTR)-42K and its evaluation using invariant feature extraction on detected extremal regions," *IEEE Access*, vol. 7, pp. 19801–19820, 2019.

[14] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.

[15] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 770–783.

[16] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[17] K. Wang and S. Belongie, "Word spotting in the wild," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 591–604.

[18] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] S. Yousfi, S.-A. Berrani, and C. Garcia, "ALIF: A dataset for Arabic embedded text recognition in TV broadcast," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1221–1225.

[21] M. Yang, Y. Guan, M. Liao, X. He, K. Bian, S. Bai, C. Yao, and X. Bai, "Symmetry-constrained rectification network for scene text recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9147–9156.

[22] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4715–4723.

[23] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2015, *arXiv:1506.02025*. [Online]. Available: http://arxiv.org/abs/1506.02025

[24] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, Jun. 1989.

[25] C. Luo, Q. Lin, Y. Liu, L. Jin, and C. Shen, "Separating content from style using adversarial learning for recognizing text in the wild," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 960–976, Apr. 2021.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[28] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, vol. 1, no. 2, p. 3.

[29] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai, "Decoupled attention network for text recognition," in *Proc. AAAI*, 2020, pp. 12216–12224.

[30] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 71–79.

[31] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8610–8617.

[32] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2231–2239.

[33] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 334–343.

[34] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[35] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.

[36] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[37] M. Al-Ayyoub, A. Nuseir, K. Alsmearat, Y. Jararweh, and B. Gupta, "Deep learning for Arabic NLP: A survey," *J. Comput. Sci.*, vol. 26, pp. 522–531, May 2018.

[38] M. Tounsi, I. Moalla, and A. M. Alimi, "ARASTI: A database for Arabic scene text recognition," in *Proc. 1st Int. Workshop Arabic Script Anal. Recognit. (ASAR)*, Apr. 2017, pp. 140–144.

[39] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

[40] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[41] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3367–3375.

[42] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.

[43] S. Saha, N. Chakraborty, S. Kundu, S. Paul, A. F. Mollah, S. Basu, and R. Sarkar, "Multi-lingual scene text detection and language identification," *Pattern Recognit. Lett.*, vol. 138, pp. 16–22, Oct. 2020.

[44] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*. [Online]. Available: http://arxiv.org/abs/1601.07140

[45] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–12.

HEBA HASSAN (Member, IEEE) received the M.Sc. degree from the Computer and Automatic Control Department, Faculty of Engineering, Tanta University. She is currently pursuing the Ph.D. degree with Egypt-Japan University of Science and Technology (E-JUST). In 2016, she was awarded the Ph.D. Scholarship by Ministry of Higher Education. She is also currently a Teaching Assistant at Kafrelshiek University. Her research interests include computer vision and deep learning.

AHMED EL-MAHDY (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Alexandria University, and the Ph.D. degree from the School of Computer Science, The University of Manchester, U.K., where he contributed to one of the early multicore processors (JAMAICA). He is currently a Full Professor at the Computer Science and Engineering Department, Egypt-Japan University of Science and Technology (E-JUST). He is also on leave from the Computer and Systems Engineering Department, Alexandria University. He has visited the Group of Advanced Processor Technologies contributing to porting the IBM Jikes dynamic compiler for JAMAICA. He has also been a Visiting Scientist at the IBM Centre for Advanced Studies, Cairo, where he was the first inventor of many issued patents in high-performance computing and image processing. He is currently the Founding Director of the Parallel Computing Laboratory at E-JUST with many funded research grants/support from IBM, Amazon, ITIDA, STDF, Academy of Science and Technology in embedded compilers, high-performance GPU acceleration, and high-performance computation on the cloud. He is a Senior Member of ACM. He is also a TPC Member of ICCD and ARCS conferences.

MOHAMED E. HUSSEIN (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Maryland, College Park, MD, USA, in 2009. He then spent close to two years as an Adjunct Member Research Staff at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, before moving to Alexandria University as a Faculty Member. Prior to joining ISI, he spent three years at Egypt-Japan University of Science and Technology (E-JUST), Alexandria, Egypt. He is currently a Computer Scientist and a Research Lead at the USC Information Sciences Institute and an Associate Professor (on leave) at Alexandria University, Egypt. He has many published research articles, and three issued patents. He is a member of the AAAI and a Senior Member of ACM.