IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 December 2023, accepted 30 December 2023, date of publication 11 January 2024,
date of current version 19 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3352748

RESEARCH ARTICLE

An Attention-Based Convolutional Recurrent Neural Networks for Scene Text Recognition

ADIL ABDULLAH ABDULHUSSEIN ALSHAWI, JAFAR TANHA, AND MOHAMMAD ALI BALAFAR
Electrical and Computer and Electrical Engineering Department, University of Tabriz, Tabriz 5166616471, Iran
Corresponding author: Jafar Tanha (tanha@tabrizu.ac.ir)

ABSTRACT Text recognition is critical in various domains, including driving assistance, handwriting recognition, and aiding the visually impaired. In recent years, deep learning-based methods have demonstrated outstanding performance in Scene Text Recognition (STR). However, STR poses significant challenges, and the scarcity of non-Latin language datasets further compounds these challenges. To address this, we collected a dataset of Persian digits, including 20000 images with different challenges, making the dataset appropriate for text recognition task. Furthermore, we propose a convolutional-based model that incorporates the squeeze and excitation gate, forcing the model to focus on latent features, and connectionist temporal classification, enabling end-to-end sequence learning, for Persian digit recognition. We conduct extensive comparisons with different architectures and models to evaluate the performance of our proposed model. As a result, our approach achieves an accuracy of 94.26 on our datasets. The results demonstrate that our model outperforms the other methods, highlighting its effectiveness in Persian digit recognition.

INDEX TERMS Convolutional neural network, scene text recognition dataset, recurrent neural networks, connectionist temporal classification.

I. INTRODUCTION

Text recognition has found applications in diverse fields, including driving assistance, handwritten recognition, and technologies aimed at aiding visually impaired individuals. Typically, text recognition can be categorized into two main areas: scanned document recognition (SDR) and scene text recognition (STR) [1]. While SDR has made significant progress, STR remains particularly challenging due to factors such as distorted text, linguistic variations, image quality, varying fonts, and irregular text shapes [2], [3]. Consequently, extensive research is still required in this area.

The success of deep learning methods has fueled the motivation of researchers in the field of STR to exploit these approaches [2]. In STR, recurrent neural networks (RNNs) are proper approaches to capture context and dependencies in sequential data, while convolutional neural networks (CNNs) excel at finding hidden patterns using local spatial information in the input [4]. RNNs, with their recurrent connections, are well-suited for handling sequential data, such as text, because they can retain information from previous time steps and utilize it to make predictions. In STR, where text is often represented as a sequence of characters or patches, RNNs can effectively capture the contextual relationships between these elements, enabling accurate recognition and understanding of the text. CNNs, on the other hand, are adept at extracting meaningful visual features from input images [2], [5]. By leveraging convolutional layers and pooling operations, CNNs can capture local spatial patterns and hierarchically learn complex representations [6]. These techniques are valuable for extracting distinctive features from text images and useful for classification or recognition tasks in the field of STR.

Previous research efforts aimed at enhancing STR performance using deep learning methods. These methods can be categorized into two main groups: segmentation-based methods and seq2seq-based approaches [7]. Seq2seq-based methods are further subdivided into three broad categories: connectionist temporal classification (CTC)-based methods, attention-based methods, and transformer-based methods [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Wei-Yen Hsu.

On the other hand, segmentation-based approaches typically involve two steps [7]: character segmentation and character recognition.

In STR tasks, the alignment of sequences, specifically aligning each character to its corresponding location, is essential [8]. Additionally, attention-based methods have proven to be advantageous in the STR tasks. These methods allow for the concentration on salient features by attending to important regions or features. As a result, attention mechanisms properly enhance the representation capabilities of CNNs [9].

Therefore, it is evident that CTC and squeeze-and-excitation (SE) attention [9] have significant relevance and can play a crucial role in the field of scene text recognition. CTC was specifically developed for temporal classification tasks, which involve sequence labeling problems where the alignment between the input data and the corresponding target labels is unknown [10]. SE block comprises a lightweight gating mechanism that focuses on enhancing the representational power of the network by modeling channel-wise relationships in a computationally efficient manner [9]. SE Net introduces additional processing between consecutive layers, enabling the exchange of information among channels. This approach prioritizes channels requiring information, thereby enhancing the efficiency of feature extraction [11]. It also suppresses the transmission of irrelevant information while improving the interaction efficiency of valuable information [12].

In response to the considerable availability of Latin datasets for STR and the scarcity of such datasets in other languages, especially those with variations in fonts and reading orders, a novel and valuable initiative has been provided in this work. An extensive dataset has been meticulously curated, comprising over 20,000 images of Persian digits sourced from electronic meters. This dataset not only introduces a substantial volume of data but also offers a diverse range of fonts and varying image qualities, rendering it a one-of-a-kind and exceptionally challenging resource in the realm of STR datasets (Figure 1). This innovative effort seeks to address the existing data gaps and further advance STR task in the context of Persian language and script.

In this study, drawing inspiration from the advantages of CTC and SE, we propose a novel attention-based convolutional recurrent neural network model for recognizing digits in our dataset. Our model comprises two individual models, each incorporating SE attention gates to enhance their representation capabilities. Additionally, bi-directional gated recurrent unit (GRU) and CTC loss are employed for sequence labeling.

The key contributions of our work can be summarized as follows:

- We have collected a substantial dataset of Persian digits designed explicitly for the task of scene text recognition (STR).
- We propose an attention-based convolutional recurrent neural network model that incorporates two CNN models, each of which is equipped with an embedded SE gate.
- Comprehensive experiments are conducted in which our approach is compared to different models and methods.

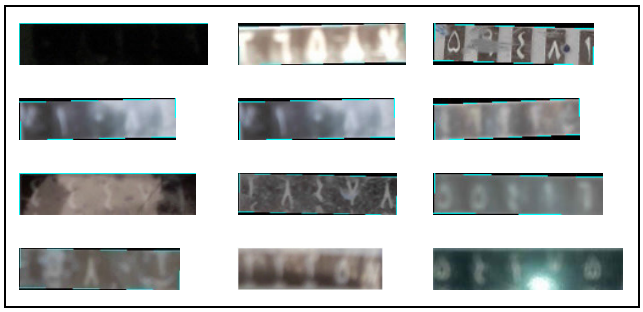

FIGURE 1. Examples of challenges in the dataset.

The experimental results obtained from the datasets used in this study demonstrate that the approach proposed in this paper outperforms state-of-the-art methods and yields an accurate recognition model.

The rest of the paper is structured as follow: Section II and III present the related works and our proposed model. Section IV provides experimental setup and experiments. Finally, in section V, conclusion and future work are reported.

II. RELATED WORKS

In this section, we provide a concise summary of the works in scene text recognition utilizing deep learning methods.

Cheng et al. [13] presented an arbitrary oriented network (AON), an end-to-end trainable network that can be trained using just pictures and word-level annotations. This model's design is based on CNN and long short-term memory (LSTM). The AON module is mostly used to extract deep features of irregular text in all four directions, as well as four placement hints. This network's output is sent to a filter gate module for the generation of an appropriate integrated sequence of features. Then, the attention-decoder module uses these attributes as input to generate anticipated character sequences. Despite the excellent performance of AON, it is hindered by a high number of parameters.

Zhong et al. [14] introduced a novel approach called semantic gan and balanced attention network (SGBANet) for scene text recognition. It begins by generating simple semantic features using Semantic GAN and then uses the balanced attention module for text recognition. Unlike traditional image-to-image translation methods, Semantic GAN operates at the semantic level, aligning feature distributions between support and target domains. The semantic generator module generates compatible semantic features for target text images, and the semantic discriminator module distinguishes these features. Additionally, the balanced attention module corrects attention drift by learning and applying a balancing

parameter to achieve a balanced glimpse vector. However, they did not discuss the sequence length.

In [15], authors enhanced the Convolutional Recurrent Neural Network (CRNN) model for text recognition, addressing issues such as low accuracy, difficulty recognizing irregular text, and limited information acquisition. They improved accuracy by incorporating label smoothing for better generalization. Additionally, they introduced a speech recognition-inspired smoothing loss function and incorporated a language model to expand information channels, ultimately improving text recognition accuracy. Nevertheless, the method needed parameter optimization because of the high computational cost.

Liu et al. [16] developed a combination of a binary convolutional encoder-decoder network with a bidirectional recurrent neural network for accurate character recognition. Bartz et al. [30] presented an end-to-end model that represents sequence features by merging CNN and RNNs. CTC loss [31] was then applied to the neural network outputs to calculate the conditional probability between the anticipated and target sequences. The method has difficulty in recognition of irregular text and achieves low accuracy.

Gao et al. [17] employed a twofold supervision network in conjunction with an attention mechanism to recognize text in photos. This model's design is based on CNN and LSTM. Ghosh et al. [18] suggested a visual attention model based on LSTM for scene text recognition. In order to detect words without relying on a preset vocabulary, the model employs convolutional features from a typical CNN as input to an LSTM network that selectively responds to sections of the picture at each time step. The proposed method exhibited poor performance on both excessively small and large images due to the limitation of a one-layer feature extraction process.

Nanehkaran et al. [19] proposed an ensemble method called CBWME for Farsi digit recognition. The method was based on convolution bagging weighted majority ensemble learning that is the combination of three pre-trained models. However, the performance of the method was not evaluated on different sequence lengths. In [20] the machine- and deep-based methods were analyzed and compared for Farsi digit recognition.

To identify sequence-based scene text, Du et al. [21] presented a temporal convolutional encoder. The encoder's primary functions are a visual representation and feature sequence conversion. Su and Lu [22] suggested a technique for converting a word picture into a series of column vectors based on the histogram of oriented gradients (HOG). The RNN is then trained to categorize the consecutive feature vectors as words. Because preprocessing is distinct from the other pipeline components, existing RNN-based systems cannot undergo end-to-end training and optimization.

Bai et al. [23] devised a system that combines CNN and LSTM to classify scene images based on a variety of perspectives and levels. To account for image differences, the approach analyzes each image from many perspectives. Liu et al. [24] proposed a sequential transformation network based on attention for scene image text recognition. The network corrects erroneous text by dividing the task into a series of fundamental patch-wise adjustments. In addition, the model uses neighbor information to preserve the shape of characters in order to identify them. However, the method is less effective with the curved or shaped text that is common in motion photos. According to [25], rectifying features posed a significant challenge in achieving generalization for text with irregular shapes.

Shrivastava et al. [26] suggested a CNN-based approach. This model extracts low-level and high-level features in all four directions, as well as character placement indicators. Then, features and placement cues are combined and sent to the LSTM decoder for text prediction. The model needs enhancement when it comes to handling curved text. Wojna et al. [27] used an attention-equipped LSTM to detect and recognize text in street-view images. By integrating one-hot encoded spatial coordinates into the LSTM, they gave their model position awareness.

Zuo et al. [28] suggested an encoder-decoder process-based scene text recognition system. The feature sequence is first retrieved using the encoder code layer's convolutional neural network and then marked using the bidirectional LSTM (Bi-LSTM). Finally, a CTC and attention mechanism integrated module is built to decode and output the text sequence. Lei et al. [29] suggested a CNN-RNN hybrid model. The CNN component extracts image features and encodes them into feature sequences, while the Bi-LSTM component decodes the associated sequences into labeled sequences. Bartz et al. [30] proposed a method using a single deep neural network that learns to find and interpret text from natural images in a semi-supervised manner. SEE is a network that incorporates and jointly trains a spatial transformer network, which can learn to identify text areas in an image, and a text recognition network, which detects the textual content of the discovered text regions. The method performance is limited to sequence length; furthermore, detecting text in arbitrary locations is another weakness of the method.

Bai et al. [31] introduced a unique approach for scene text recognition dubbed edit probability (EP). EP attempts to estimate the likelihood of creating a string from the output sequence of a probability distribution conditioned on the input image while accounting for the possibility of missing/superfluous characters. Nevertheless, the misalignment is not solved in this study. A comprehensive evaluation and analysis of the various STR approaches was published by Chen et al. [32]. Shi et al. [33] presented an end-to-end model that represented sequence features by merging CNN and RNNs. CTC loss was then applied to the neural network outputs to calculate the conditional probability between the anticipated and target sequences. They did not deal with the challenge of irregular text recognition.

Zheng et al. [34] introduced TPS++, an attention-enhanced TPS transformation for text rectification. TPS++

incorporated the attention mechanism to improve the flexibility and accuracy of text correction. By jointly estimating foreground control points and attention scores, TPS++ generated natural and easier-to-read text corrections. It shared the feature backbone with the recognizer, minimizing parameter overhead and inference time.

Yan et al. [35] proposed a method for learning primitive representations of scene text images using graph-based modeling. They introduced pooling and weighted aggregators to learn these representations, which are then transformed into higher-level visual text representations using graph convolutional networks. The work also presented a framework called PREN2D that integrates visual text representations into an encoder-decoder model with a 2D attention mechanism to address misalignment issues in text recognition. This method identifies characters one by one, leading to a low processing speed.

Tran [4] proposed SAFL, a self-attention-based neural network model with focal loss for scene text recognition. SAFL utilized focal loss, which allows the model to focus more on training low-frequency samples. Additionally, to handle distortions and irregular texts, spatial transformer network (STN) is incorporated to rectify the text before passing it to the recognition network.

Wang et al. [36] proposed a decoupled attention network (DAN) for scene text recognition, addressing limitations in alignment and historical decoding. DAN consists of a feature encoder, convolutional alignment module, and decoupled text decoder. By decoupling the alignment operation, DAN achieves improved accuracy and flexibility in recognizing text. The experiments on text-like noises showed that the method struggled to align text.

Deelaka et al. [37] presented a new model architecture that includes a unique image feature encoding strategy and feature projection methods. The model was trained on images with a fixed number of characters, resulting in a fixed number of object labels without object location prediction. The aim was to reduce parameter spaces and computational complexity for real-time inference and efficient training in a federated learning setup. The model included a geometrical shape-based encoder and a feature localization unit for predicting ground-truth label sequences. The model assumed horizontally aligned input images with a single row of characters. The method is limited to digits; moreover, its performance is adversely affected by irregular texts.

III. PROPOSED MODEL

This section provides details of methodology, dataset, and the proposed model. In Figure 2, the main steps of the proposed model are presented in more details.

A. DATASET AND PREPROCESSING

Considering the limited availability of datasets in non-Latin languages for text recognition tasks, we have created a dataset, designed to fill this gap. This dataset focuses on Persian digits found in images of electricity meter devices and

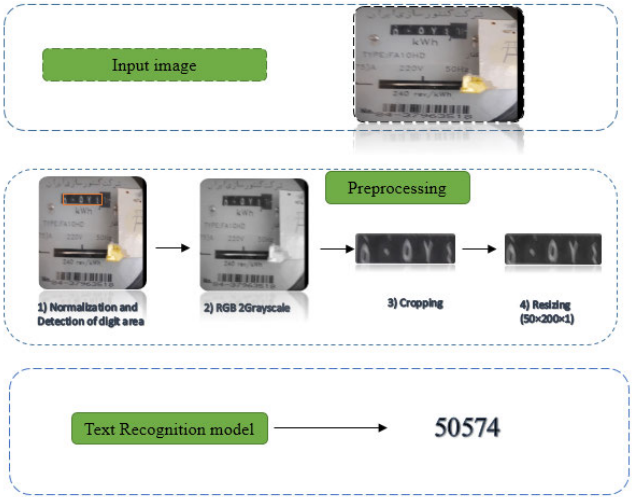

FIGURE 2. The general overview of the proposed model.

contains 20,000 images sourced from these devices. Notably, this dataset presents significant challenges due to the presence of low-quality, tilted, blurry images, and various fonts. Additionally, some images feature additional difficulties like dirtiness, obstructions from objects, or distant perspectives. Figure 3 displays samples illustrating these image-related issues.

To prepare the images for the text recognition task, we follow a specific procedure. Firstly, we normalize the images by dividing each pixel value by 255. This normalization step ensures that the pixel values are within a standardized range. Additionally, the images are annotated to provide labels for the objects of interest. In our research, we employed the bounding box technique for annotation. This involves drawing boxes around the items in the images to precisely identify their placement. Creating accurate bounding boxes and avoiding overlap among them are crucial for developing a robust and accurate model. To perform the annotation task, we utilized the vgg image annotator (VIA) [38], [39], an open-source software designed for annotating images. VIA allows us to define regions of interest in the images and generate textual descriptions for those regions. Once the annotation process is complete, the images are converted from RGB color space to grayscale. This conversion reduces the dimensionality of the images and simplifies the subsequent processing steps. Subsequently, the system proceeds to detect the digit area within the images, which is then isolated through cropping and resized to a standardized dimension of 50×200 pixels. These cropped and resized images are subsequently input into the text recognition module for further in-depth analysis. To provide a visual representation of the dataset, we include some sample images in Figure 4. These samples showcase the variety of images in our dataset and highlight the challenges posed by factors such as image quality, tilting, blurring, and different fonts. Figure 5 displays the distribution of digits across different positions. Indeed,

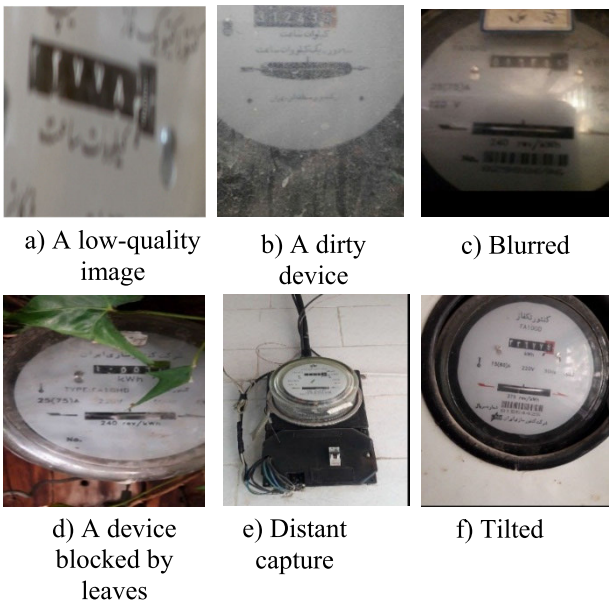

FIGURE 3. Examples of the hard situations depicted in the collected images presented.

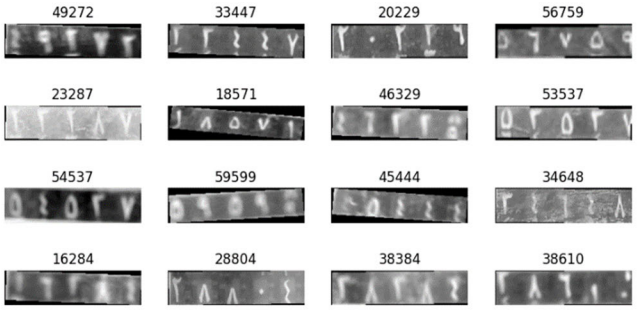

FIGURE 4. Samples of pre-processed images.

as illustrated in Figure 5, it becomes apparent that the distribution of all digits exhibits a nearly uniform consistency. This indicates that the occurrence of digits is relatively balanced across the dataset. Except for position one, numbers two, three, and four have more occurrences.

B. PROPOSED MODEL

To facilitate the recognition of digits, we have proposed a two-head CNN architecture that utilizes different network architectures to extract distinct features from the input data. The input image is represented as $I \in R^{H\times W\times C}$ which H, W, and C correspond to the height, width, and number of channels, respectively. The image is input into the two distinct networks within our model. The schematic diagram of our proposed model can be found in Figure 6.

Each block in the first model consists of two convolutional layers, and the rectified linear unit (ReLU) activation function, defined as $f(x) = max(0, x)$, is applied. To enhance the convergence of the model and improve the initialization performance, the extracted features are passed through a batch normalization (BN) layer. Subsequently, a max pooling (MP) layer is employed to reduce the output map size. On the other hand, the second model follows a slightly different architecture. Each block in this model includes a single CNN block, and the remaining layers are iterated similarly to the first model.

The outputs of the final block in both models are followed by a SE block. Finally, the output of the two models is concatenated.

The purpose of the SE block is to encourage the models to focus more on salient features. SE blocks are introduced by Hu et al. [9] to enhance the representation power of the model by modeling the relationships among channels. This enables the network to recalibrate features by suppressing unnecessary or less relevant features.

The SE gate consists of two main modules: the squeeze and excitation modules. In the squeeze module, global average pooling is employed to generate channel-wise statistics. This is achieved by reducing the spatial dimensions of the feature map, $F \in R^{H\times W\times C}$. Formally, a statistic $z \in R^{C}$ is generated by squeezing U through its spatial dimensions H × W, such that the c^{th} element of z is calculated by:

$$Z_c = F_{sq}(U) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} U(i,j) \tag{1}$$

where $F_{sq}$is the squeeze function. The information obtained from the squeeze module is then transferred to the excitation module to capture channel-wise dependencies. To achieve this goal, a gating mechanism with the sigmoid function is employed as follows:

$$\begin{aligned} S = F_{ex}(Z, W) &= \sigma\left(g(Z, W)\right) \\ &= \sigma(W_2\left(\delta\left(W_1 Z\right)\right)) \end{aligned} \tag{2}$$

where σ and δ are sigmoid and ReLU activation functions, $W_1 \in R^{C/r\times C}$, $W_2 \in R^{C\times C/r}$, and r is the reduction ratio. In order to manage model complexity and enhance generalization, the gating mechanism is parameterized by introducing a bottleneck structure. This involves incorporating two fully connected (FC) layers around a non-linearity. The final output of the block is obtained by rescaling U with the activations s.

$$\tilde{X}_C = F_{scale}(U_C, s_c) = u_c s_c \tag{3}$$

F_{scale} depicts the channel-wise multiplication between $s_c$and U_c, and $\tilde{X}_C = [x_1, x_2, \ldots, x_c]$. In other words, the output of the excitation module is multiplied element-wise with the input feature map, resulting in enhanced features that are more attentive to important channels. Figure 7 provides a visual depiction of the structure of the SE gate.

Finally, the output of each model is concatenated, merging the information from both models. This concatenated output is then sent to Bi-GRU, which is responsible for decoding the feature sequence and capturing sequential dependencies in both forward and backward directions. The output from the Bi-GRU is subsequently fed into a CTC layer. It is designed

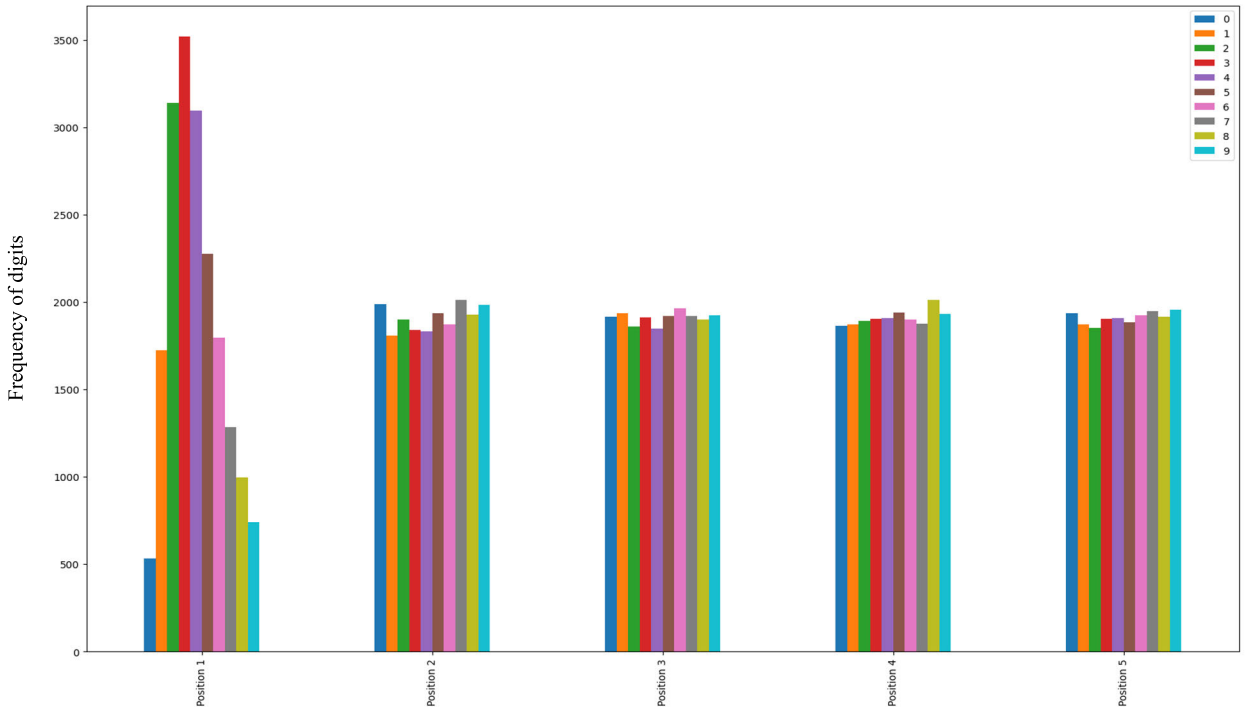

FIGURE 5. The distribution of digits according to the position of digits.

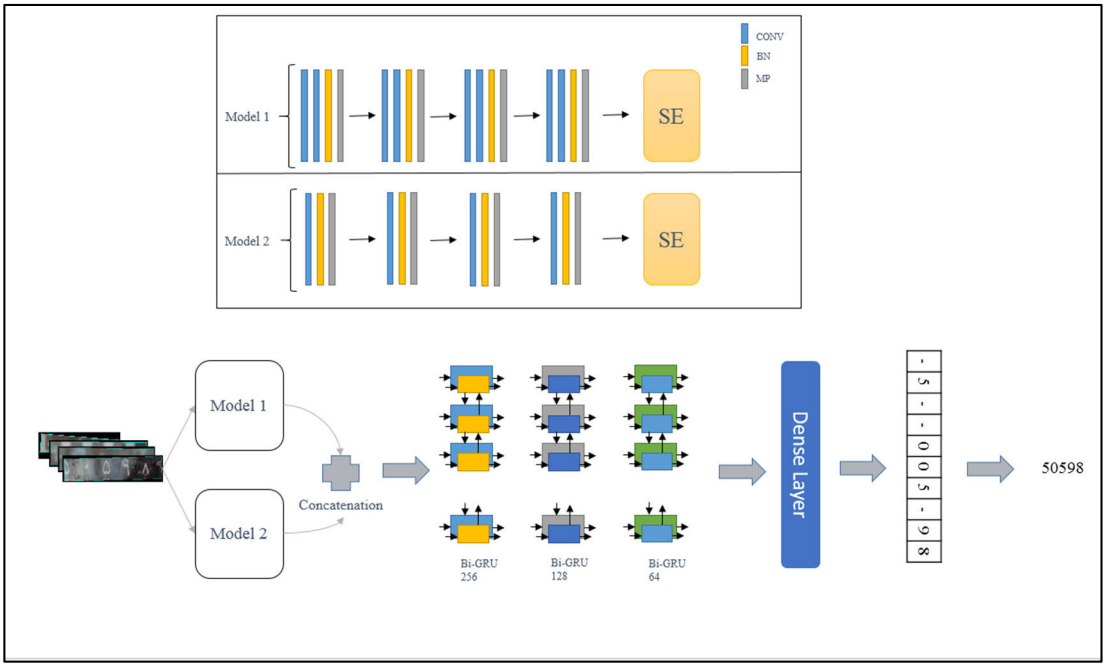

FIGURE 6. The architecture of the proposed model.

to handle sequences of variable length, making it suitable for recognizing text in images where the number of characters can vary.

The CTC layer is responsible for decoding and mapping the sequential output to the corresponding text labels or digits, thus enabling the recognition of the digits present in the input image. In other words, it transfers the Bi-GRU output sequence to the detected text. However, the Bi-GRU input sequence may be translated to many output labels.

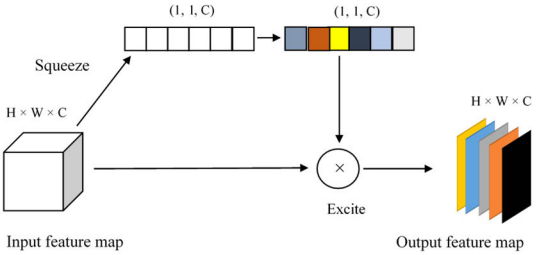

FIGURE 7. Schematic view of SE gate [9].

Therefore, maximum likelihood is used to choose the final output label sequence. The following formula is used to determine the probability of the output label sequence:

$$p(l \mid x) = \sum_{\pi \in F^{-1}(l)} p(\pi \mid x)$$
$$p(\pi \mid x) = \prod_{t=1}^{T} y_{\pi_t}^{t} \tag{4}$$

where x is the input sequence of CTC, and T is sequence length. π indicates the output sequence with blank labels. l is output label sequence. The relationship between sequence π and label l is specified by the mapping function F. $y_{\pi_t}^{t}$ represents the probability of possessing label π_t at timestamp t.

IV. EXPERIMENTS

This section includes two subsections; firstly, the details of experimental setups are provided, and secondly, to assess the performance of our model, we conduct different experiments and compare its performance with other methods.

A. IMPLEMENTATION DETAILS

1) SETUP

In this study, an implementation of the proposed model was developed using Python and TensorFlow, a popular deep learning framework. The model was trained and evaluated in an end-to-end fashion, optimizing various training parameters to achieve optimal performance.

During the training phase, specific parameters were chosen to guide the learning process, and they were adjusted by trial and error. A learning rate of 0.001 was employed, and a learning rate decay strategy was implemented, reducing the learning rate by a factor of 0.8. An epsilon value of 0.001 was utilized, and a patience value of five was set to monitor the training process and determine when to stop if no improvement was observed. Additionally, early stopping with a patience of ten was employed to prevent overfitting and ensure the best model performance.

To optimize the hyperparameters of the model, the Adam optimizer was employed, as it generally demonstrated superior performance compared to other optimizers [40], [41]. A clip value of 0.2 was used to limit the gradients during the optimization process, and the training was performed for a total of 200 epochs. Throughout the model architecture, a dropout rate of 0.5 was applied to all layers, which aids in regularization and reduces the risk of overfitting. Furthermore, the input shape of models is 50 × 200×1, and the output shape is 12 × 12. The summary of used parameters is presented in Table 1.

TABLE 1. The hyper-parameters of the proposed model.

Item	Value
Epochs	400
Batch size	16
Learning rate	0.001
Epsilon	0.001
Input size	(50, 200)

The Google Colab platform, which was utilized for training and testing the model, adhered to specific environment requirements for model development. These requirements included using Python version 3.9, Tensorflow version 2.x, an Nvidia K80 GPU, and a RAM capacity of 12GB.

Moreover, the accuracy is assessed at the word level; it measures how frequently the model predicts right words. The accuracy measure is calculated as follows:

$$accuracy = \frac{number\ of\ correctly\ classified\ words}{all\ words} \tag{5}$$

2) NETWORKS

The configurations of two CNN models, mentioned in section III, are presented in Tables 2.

TABLE 2. Configuration of CNN models (k, s, p, and c stand for kernel size, stride, padding, and channel, respectively).

First			Second		
Layer		Configuration	Layer		Configuration
Block1	Convolutional 1	k=5, s=1, p=same, c=16	Block1	Convolutional 1	k=5, s=1, p=same, c=16
	Convolutional 2				
	Maxpooling	s=2		Maxpooling	s=2
	Batchnormalization	-		Batchnormalization	-
	Dropout	rate = 0.5		Dropout	rate = 0.5
Block2	Convolutional 1	k=3, s=1, p=same, c=32	Block2	Convolutional 1	k=3, s=1, p=same, c=32
	Convolutional 2				
	Maxpooling	s=2		Maxpooling	s=2
	Batchnormalization	-		Batchnormalization	-
	Dropout	rate=0.5		Dropout	rate = 0.5
Block3	Convolutional 1	k=3, s=1, p=same, c=64	Block3	Convolutional 1	k=3, s=1, p=same, c=64
	Convolutional 2				
	Maxpooling	s=2		Maxpooling	s=2
	Batchnormalization	-		Batchnormalization	-
	Dropout	rate=0.5		Dropout	rate = 0.5
Block4	Convolutional 1	k=3, s=1, p=same, c=128	Block4	Convolutional 1	k=3, s=1, p=same, c=128
	Convolutional 2				
	Maxpooling	s=2		Maxpooling	s=2
	Batchnormalization	-		Batchnormalization	-
	Dropout	rate=0.5		Dropout	rate = 0.5

B. RESULTS

In this section, we present the results of various approaches employed in our study. Firstly, we assess the performance of individual pre-trained models when used in conjunction with an SE block. Secondly, we evaluate the performance of pre-trained models instead of our CNN blocks. Finally, we compare the performance of our proposed method to the state-of-the-art models in this field.

1) PRE-TRAINED MODELS

Three pre-trained deep learning models, including Resent50 [42], MobileNet [43], and Inception [44], are utilized in the experiments. The performance of each model is presented in Table 3. The training dataset and setup are identical for the models. According to Table 3, MobileNet performed better in terms of accuracy in the text recognition task, and Inception acquired the worst performance. It is pertinent to highlight that, within the aforementioned models, the top layer—specifically, the classification layer—is excluded, and instead, an SE gate is employed before transmitting the model outputs to the RNN and CTC components.

TABLE 3. The performance of pre-trained models.

Model	Accuracy
Resnet50 [42]	90.93
MobileNet [43]	**91.44**
Inception [44]	45.13

2) COMBINATION OF PRE-TRAINED MODELS

In this section, we delve into the performance evaluation of the mentioned pre-trained models and compare them against lightweight CNN models, as presented in Section III. Based on the individual performance of the pre-trained models, a multi-head model with a SE gate is designed. This multi-head model is constructed by concatenating the deep learning architectures of ResNet50, MobileNet, and Inception. To simplify the model and mitigate overfitting, some convolution layers are removed; moreover, the top layer of each architecture, which represents the classification layer, is eliminated. Consequently, the ResNet50 component of the multi-head model outputs (25, 7, 512), while the MobileNet and Inception components output (25, 6, 728), (25, 7, 192) respectively. To ensure compatibility among the outputs of the models, a max pooling layer is added.

In order to conduct a comprehensive evaluation of the different combinations of pre-trained models, we consider four specific combinations: ResNet with Inception, ResNet with MobileNet, Inception with MobileNet, and all models together. The results of these combinations in terms of accuracy are presented in Table 4. Based on the findings, it is evident that the combination of MobileNet and Inception yielded the highest performance among the different combinations. This particular combination achieved the highest accuracy rate compared to the other combinations, indicating its effectiveness in the text recognition task. Furthermore, the fusing of ResNet50 and Inception acquires the second-best accuracy, showing the importance of diversity.

TABLE 4. The performance of combined pre-trained models on the Persian digit dataset.

Model	Accuracy	F1	Precision	Recall
Resnet50-MobileNet	89.93	97.83	97.86	97.82
Resnet50-Inception	93.39	98.81	98.83	98.79
MobileNet-Inception	93.64	98.87	98.93	98.81
Resnet50-MobileNet-Inception	92.0	98.60	98.60	98.61

On the other hand, the combination of all three models together exhibited the lowest performance among the four combinations. This suggests that the integration of ResNet50, Inception, and MobileNet in a unified model did not yield the desired improvements in accuracy. It is important to note that while the combination of all three models may offer potential benefits in terms of overall model diversity, it may also introduce challenges related to model complexity and potential conflicts in the learned representations.

Moreover, methods are compared based on F1-score, Precision, and Recall. Unlike the accuracy measure, which assesses the overall correctness, these metrics are used to evaluate the number of correctly predicted single digits. According to the results, our model with MobileNet-Inception achieves higher scores in F1-Score, Precision, and Recall.

Figure 8 compares the convergence rate of pre-trained models. The training loss for all combinations exhibits a similar convergence rate. Upon closer Inception, the combination of three pre-trained models shows better performance, and the results of Table 4 confirm it.

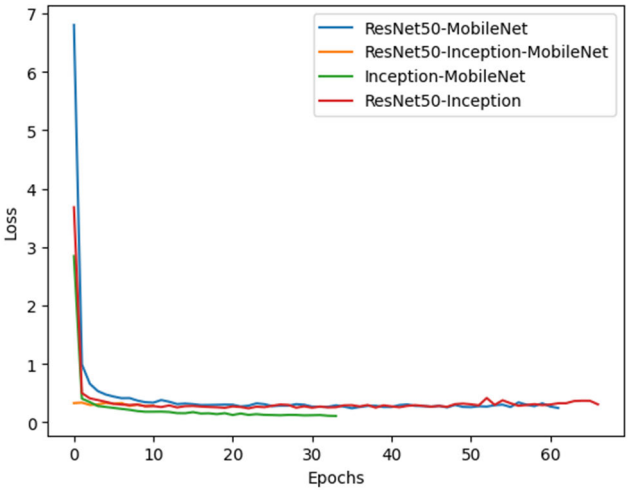

FIGURE 8. The loss by iteration of the combination of pre-trained models.

3) COMPARISON TO THE STATE-OF-THE-ART APPROACHES

We conduct a performance evaluation of our proposed approach by comparing it to other methods using our dataset. Moreover, we compare the effect of different recurrent neural networks (RNNs), including Bi-LSTM, GRU, and LSTM, on our method. Table 5 provides the results of different methods and models. We utilize a simple CNN model as the baseline, which achieves an accuracy of 91.71%. The architecture of the CNN model is presented in Figure 9. The next model we evaluate is the CRNN [19]. In the base paper, Bi-GRU was utilized in the CRNN model. However, for the purpose of further comparison, we also embed Bi-LSTM in the CRNN model. The results show that the CRNN model achieved an accuracy of 89.13% with Bi-GRU and an accuracy of 88.11% with Bi-LSTM.

TABLE 5. The result comparison of our model with others.

Model	Accuracy	F1	Precision	Recall
Our model (Bi-GRU)	94.26	98.92	98.92	98.93
Our model (Bi-LSTM)	93.03	98.23	98.24	98.21
Our model (simple GRU)	93.64	98.85	98.83	98.88
Our model (simple LSTM)	92.57	97.87	98.04	97.73
Our model with MobileNet and Inception	93.39	98.81	98.83	98.79
CRNN [33] (Bi-GRU)	89.13	97.97	97.97	97.98
CRNN [33] (Bi-LSTM)	88.11	97.87	98.04	97.73
GeoTRNet[37]	45.30	68.63	68.63	68.64
CNN	91.71	96.98	96.98	96.99

Next, we evaluate the GeoTRNet [23], a state-of-the-art model that has achieved remarkable results. However, in our evaluation, the GeoTRNet achieved an accuracy of 45.30%. This suggests that the GeoTRNet may not be well-suited for the specific characteristics or requirements of our dataset or task. To further explore the performance of different RNN architectures in our proposed method, we replace the lightweight CNNs in our model with pre-trained models, specifically MobileNet and Inception. Our model with MobileNet and Inception achieved an accuracy of 93.39%.

In comparison to these methods, our proposed model, utilizing Bi-GRU in the CRNN, achieved the highest accuracy of 94.26%. The second-best accuracy was achieved by our model with simple GRU, which obtained an accuracy of 93.64%. We also observed slightly lower accuracies with our model variants using Bi-LSTM (93.03%) and simple LSTM (92.57%). Overall, the results demonstrate that our proposed approach outperforms the other methods, and the choice of RNN architecture can have a significant impact on the performance of the model in the given task. Moreover, methods are compared based on F1-score (F1), Precision, and Recall. Unlike the accuracy measure, which assesses the overall correctness, these metrics are used to evaluate the number of correctly predicted single digits. According to the results, our model with Bi-GRU achieves higher scores in F1-Score, Precision, and Recall. Nevertheless, in our work, the accuracy measure, which calculates the number of correctly classified words, holds importance.

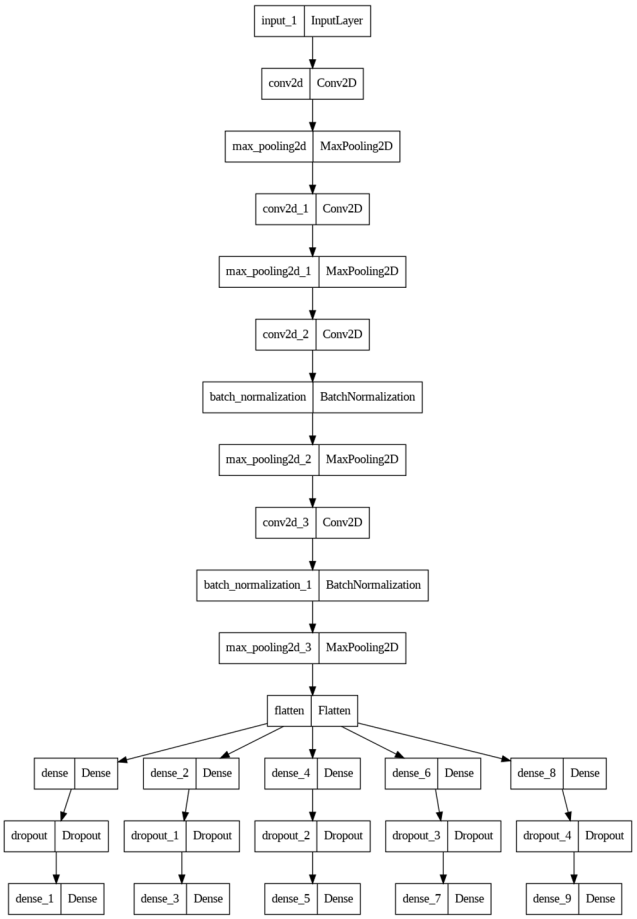

FIGURE 9. The architecture of CNN.

In Figure 10, the confusion matrix for our model with Bi-GRU is plotted. For each single digit, true negative, true positive, false negative, and false positive are reported. Present and not present indicate the presence and absence of a specific digit.

The convergence curves of our approach, CRNN, and GeoTRNet are presented in Figure 11. Analyzing the training loss curves depicted in Figure 11, it is evident that our model is faster compared to the others in terms of text recognition.

C. ABLATION STUDY

We conducted an ablation study, as detailed in Table 6, with the primary goal of evaluating the efficacy of the SE gate. The study consists of two experiments presented in Table 6, with the first experiment incorporating the SE module and the second experiment omitting it.

The results outlined in Table 6 highlight the substantial positive impact of integrating the SE module on our model's performance when working with the dataset. This finding shows the pivotal role played by SE in enhancing our model's performance by enabling it to focus more effectively on salient features, ultimately leading to a notable performance boost.

FIGURE 10. The confusion matrix of our approach with Bi-GRU.

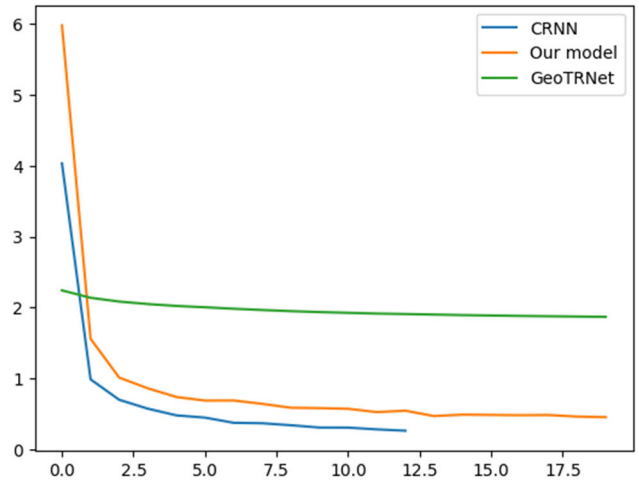

FIGURE 11. Comparison of Convergence rate of models. The vertical axis represents the training loss, while the horizontal axis indicates the iteration number.

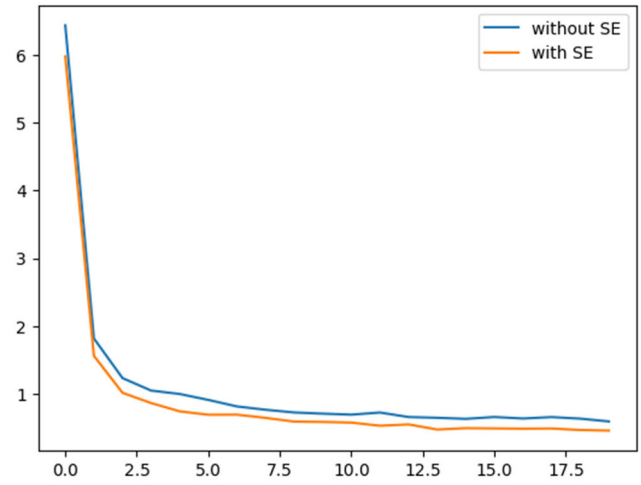

FIGURE 12. A comparison of the convergence rates with and without SE, highlighting the impact of SE on model performance.

TABLE 6. Text recognition accuracy with and without se.

Model	Accuracy
Our model with SE	**94.26**
Our model without SE	93.13

Furthermore, a visual assessment through a comparison of convergence curves, as depicted in Figure 12, clearly underscores the advantages of incorporating the SE module. The superior performance of our model is prominently evident when SE is integrated, reinforcing its pivotal role in the overall success of our approach.

D. TIME COMPLEXITY

In the comparative analysis of various models, Table 7 explores the time complexities of different architectural configurations. Our own models, incorporating Bi-GRU and Bi-LSTM, exhibit higher time complexities, indicative of their computational demands.

TABLE 7. Comparison of models' time complexity.

Model	Time complexity
Our model (Bi-GRU)	9343.14
Our model (Bi-LSTM)	8542.12
Our model (simple GRU)	7775.45
Our model (simple LSTM)	5659.55
Our model with MobileNet and Inception	5098.88
CRNN [33] (Bi-GRU)	1181.60
CRNN [33] (Bi-LSTM)	996.64
GeoTRNet[37]	11176.10

Simpler variants, such as models utilizing only GRU or LSTM, demonstrate reduced time complexities. Intriguingly, the integration of MobileNet and Inception layers within our model architecture results in a substantial decrease in overall

time complexity, suggesting enhanced computational efficiency. Comparative benchmarks with other models reveal that models based on CRNN, particularly those employing Bi-LSTM, exhibit lower time complexities, signifying superior computational efficiency compared to our models. However, the GeoTRNet model stands out with a higher time complexity, underscoring potential trade-offs between model complexity and computational efficiency in the quest for optimal performance across diverse applications.

V. DISCUSSION AND CONCLUSION

Given the limited availability of non-Latin datasets for text recognition tasks, this paper focuses on addressing this gap by collecting a dataset of Persian digits sourced from meter devices. The dataset comprises more than 20,000 images. Additionally, we introduce a novel model specifically designed for Persian digit recognition. Our proposed model consists of two CNN models with SE gates, followed by Bi-GRU and CTC loss. To evaluate the efficacy of our model, we conducted extensive experiments. Firstly, we investigated the impact of CNN models by replacing them with pre-trained models. Furthermore, we compared our model with existing approaches in the field. The experimental results demonstrate that our proposed model achieves state-of-the-art performance on our dataset and surpasses other models in terms of accuracy and effectiveness.

Despite the advantages of our approach, it is not suitable for irregular text and noisy text. Furthermore, our dataset is limited to five digits; as a result, its performance may not be suitable for digits with fewer or more than five digits.

In the future, one can improve the proposed model and apply it to our dataset. Moreover, data augmentation can affect the performance of the proposed model. Meanwhile, different architectures can be used. Furthermore, the proposed model can be used to predict images whose numbers are in Persian and English. Additionally, the model can be improved on irregular text.

REFERENCES

[1] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," 2018, *arXiv:1801.01671*.

[2] C. Xue, J. Huang, W. Zhang, S. Lu, C. Wang, and S. Bai, "Image-to-character-to-word transformers for accurate scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12908–12921, Nov. 2023, doi: 10.1109/TPAMI.2022.3230962.

[3] X. Xie, L. Fu, Z. Zhang, Z. Wang, and X. Bai, "Toward understanding wordart: Corner-guided transformer for scene text recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 303–321.

[4] B. H. Tran, T. Le-Cong, H. M. Nguyen, D. A. Le, T. H. Nguyen, and P. Le Nguyen, "SAFL: A self-attention scene text recognizer with focal loss," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 1440–1445.

[5] Y. Liu, Y. Wang, and H. Shi, "A convolutional recurrent neural-network-based machine learning for scene text recognition application," *Symmetry*, vol. 15, no. 4, p. 849, Apr. 2023.

[6] A. Raza, K. Munir, M. Almutairi, F. Younas, M. M. S. Fareed, and G. Ahmed, "A novel approach to classify telescopic sensors data using bidirectional-gated recurrent neural networks," *Appl. Sci.*, vol. 12, no. 20, p. 10268, Oct. 2022.

[7] H. Cai, J. Sun, and Y. Xiong, "Revisiting classification perspective on scene text recognition," Feb. 2021, *arXiv:2102.10884*.

[8] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, and X. Bai, "Scene text recognition from two-dimensional perspective," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 8714–8721.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[10] H. Liu, S. Jin, and C. Zhang, "Connectionist temporal classification with maximum entropy regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 839–849.

[11] A. Rehman, A. Raza, F. S. Alamri, B. Alghofaily, and T. Saba, "Transfer learning-based smart features engineering for osteoarthritis diagnosis from knee X-ray images," *IEEE Access*, vol. 11, pp. 71326–71338, 2023.

[12] A. Raza, K. Munir, and M. Almutairi, "A novel deep learning approach for deepfake image detection," *Appl. Sci.*, vol. 12, no. 19, p. 9820, Sep. 2022.

[13] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," 2017, *arXiv:1711.04226*.

[14] D. Zhong, S. Lyu, P. Shivakumara, B. Yin, J. Wu, U. Pal, and Y. Lu, "SGBANet: Semantic GAN and balanced attention network for arbitrarily oriented scene text recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 464–480.

[15] W. Yu, M. Ibrayim, and A. Hamdulla, "Scene text recognition based on improved CRNN," *Information*, vol. 14, no. 7, p. 369, Jun. 2023.

[16] Z. Liu, Y. Li, F. Ren, W. L. Goh, and H. Yu, "SqueezedText: A real-time scene text recognition by binary convolutional encoder–decoder network," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 7194–7201, doi: 10.1609/aaai.v32i1.12252.

[17] Y. Gao, Z. Huang, Y. Dai, C. Xu, K. Chen, and J. Guo, "DSAN: Double supervised network with attention mechanism for scene text recognition," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4, doi: 10.1109/VCIP47243.2019.8965779.

[18] S. K. Ghosh, E. Valveny, and A. D. Bagdanov, "Visual attention models for scene text recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 943–948, doi: 10.1109/ICDAR.2017.158.

[19] Y. A. Nanehkaran, J. Chen, S. Salimi, and D. Zhang, "A pragmatic convolutional bagging ensemble learning for recognition of Farsi handwritten digits," *J. Supercomput.*, vol. 77, no. 11, pp. 13474–13493, Nov. 2021.

[20] Y. A. Nanehkaran, D. Zhang, S. Salimi, J. Chen, Y. Tian, and N. Al-Nabhan, "Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits," *J. Supercomput.*, vol. 77, no. 4, pp. 3193–3222, Apr. 2021.

[21] X. Du, T. Ma, Y. Zheng, H. Ye, X. Wu, and L. He, "Scene text recognition with temporal convolutional encoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2383–2387, doi: 10.1109/ICASSP40776.2020.9054269.

[22] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Proc. Asian Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9003, 2015, pp. 35–48, doi: 10.1007/978-3-319-16865-4_3.

[23] S. Bai, H. Tang, and S. An, "Coordinate CNNs and LSTMs to categorize scene images with multi-views and multi-levels of abstraction," *Expert Syst. Appl.*, vol. 120, pp. 298–309, Apr. 2019, doi: 10.1016/j.eswa.2018.08.056.

[24] Y. Liu, H. Chen, C. Shen, T. He, L. Jin, and L. Wang, "ABCNet: Real-time scene text spotting with adaptive bezier-curve network," 2020, *arXiv:2002.10200*.

[25] S. Qin, A. Bissaco, M. Raptis, Y. Fujii, and Y. Xiao, "Towards unconstrained end-to-end text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4703–4713.

[26] A. Shrivastava, J. Amudha, D. Gupta, and K. Sharma, "Deep learning model for text recognition in images," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–6, doi: 10.1109/ICCCNT45670.2019.8944593.

[27] Z. Wojna, A. N. Gorban, D.-S. Lee, K. Murphy, Q. Yu, Y. Li, and J. Ibarz, "Attention-based extraction of structured information from street view imagery," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 844–850, doi: 10.1109/ICDAR.2017.143.

[28] L.-Q. Zuo, H.-M. Sun, Q.-C. Mao, R. Qi, and R.-S. Jia, "Natural scene text recognition based on encoder–decoder framework," *IEEE Access*, vol. 7, pp. 62616–62623, 2019, doi: 10.1109/ACCESS.2019.2916616.

[29] Z. Lei, S. Zhao, H. Song, and J. Shen, "Scene text recognition using residual convolutional recurrent neural network," *Mach. Vis. Appl.*, vol. 29, no. 5, pp. 861–871, Jul. 2018, doi: 10.1007/s00138-018-0942-y.
[30] C. Bartz, H. Yang, and C. Meinel, "See: Towards semi-supervised end-to-end scene text recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 6674–6681, doi: 10.1609/aaai.v32i1.12242.
[31] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," 2018, *arXiv:1805.03384*.
[32] 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, Mar. 2022, doi: 10.1145/3440756.
[33] 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, doi: 10.1109/TPAMI.2016.2646371.
[34] T. Zheng, Z. Chen, J. Bai, H. Xie, and Y.-G. Jiang, "TPS++: Attention-enhanced thin-plate spline for scene text recognition," 2023, *arXiv:2305.05322*.
[35] R. Yan, L. Peng, S. Xiao, and G. Yao, "Primitive representation learning for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 284–293.
[36] 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 Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12216–12224.
[37] P. N. Deelaka, D. R. Jayakodi, and D. Y. Silva, "Geometric perception based efficient text recognition," 2023, *arXiv:2302.03873*.
[38] A. Dutta, A. Gupta, and A. Zissermann, "VGG image annotator (VIA)," 2016. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/software/via
[39] A. Dutta and A. Zisserman, "The VIA annotation software for images, audio and video," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2276–2279, doi: 10.1145/3343031.3350535.
[40] C. C. Ukwuoma, Z. Qin, M. B. B. Heyat, F. Akhtar, O. Bamisile, A. Y. Muaad, D. Addo, and M. A. Al-antari, "A hybrid explainable ensemble transformer encoder for pneumonia identification from chest X-ray images," *J. Adv. Res.*, vol. 48, pp. 191–211, Jun. 2023.
[41] P. Poudel, S. H. Bae, and B. Jang, "Comparison of different deep learning optimizers for modeling photovoltaic power," *J. Chosun Natural Sci.*, vol. 11, no. 4, pp. 204–208, 2018.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[43] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[44] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

ADIL ABDULLAH ABDULHUSSEIN ALSHAWI received the B.Sc. degree in computer science from Basra University, Iraq, in 2008, and the M.Sc. degree in computer science from the University of Utara Malaysia (UUM), Malaysia, in 2016. He is currently pursuing the Ph.D. degree in artificial intelligence with Tabriz University. His research interests include machine learning and pattern recognition.

JAFAR TANHA was born in Bonab, Iran. He received the B.Sc. and M.Sc. degrees in computer science from the Amirkabir University of Technology (Polytechnic), Tehran, Iran, in 1999 and 2001, respectively, and the Ph.D. degree in computer science-artificial intelligence from the University of Amsterdam (UvA), Amsterdam, The Netherlands, in 2013. He joined with the INL Institute, Leiden, The Netherlands, as a Researcher, from 2013 to 2015. Since 2015, he has been with the Department of Computer Engineering, Payame Noor University, Tehran, where he was an Assistance Professor. He has held lecturing positions with the Iran University of Science & Technology, Tehran, in 2016. He is currently an Associate Professor with the University of Tabriz, Tabriz, Iran. His research interests include machine learning, pattern recognition, and document analysis.

MOHAMMAD ALI BALAFAR received the B.S. degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 1995, and the M.S. and Ph.D. degrees in computer engineering in Malaysia, in 2000 and 2007, respectively. He is currently a Professor with the Department of Computer Engineering, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran. His research interests include AI, big data, and visualization.