Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.029105
Article

Tech Science Press

Enhanced Attention-Based Encoder-Decoder Framework for Text Recognition

S. Prabu and K. Joseph Abraham Sundar*

School of Computing, SASTRA Deemed to be University, Thanjavur, 613401, India
*Corresponding Author: K. Joseph Abraham Sundar. Email: josephabrahamsundar@it.sastra.edu
Received: 25 February 2022; Accepted: 19 April 2022

Abstract: Recognizing irregular text in natural images is a challenging task in computer vision. The existing approaches still face difficulties in recognizing irregular text because of its diverse shapes. In this paper, we propose a simple yet powerful irregular text recognition framework based on an encoder-decoder architecture. The proposed framework is divided into four main modules. Firstly, in the image transformation module, a Thin Plate Spline (TPS) transformation is employed to transform the irregular text image into a readable text image. Secondly, we propose a novel Spatial Attention Module (SAM) to compel the model to concentrate on text regions and obtain enriched feature maps. Thirdly, a deep bi-directional long short-term memory (Bi-LSTM) network is used to make a contextual feature map out of a visual feature map generated from a Convolutional Neural Network (CNN). Finally, we propose a Dual Step Attention Mechanism (DSAM) integrated with the Connectionist Temporal Classification (CTC) - Attention decoder to re-weights visual features and focus on the intra-sequence relationships to generate a more accurate character sequence. The effectiveness of our proposed framework is verified through extensive experiments on various benchmarks datasets, such as SVT, ICDAR, CUTE80, and IIIT5k. The performance of the proposed text recognition framework is analyzed with the accuracy metric. Demonstrate that our proposed method outperforms the existing approaches on both regular and irregular text. Additionally, the robustness of our approach is evaluated using the grocery datasets, such as GroZi-120, WebMarket, SKU-110K, and Freiburg Groceries datasets that contain complex text images. Still, our framework produces superior performance on grocery datasets.

Keywords: Deep learning; text recognition; text normalization; attention mechanism; convolutional neural network (CNN)

1 Introduction

Text in natural scenes can provide rich and precise semantic information, which is beneficial for understanding the world around us. High-level semantics embodied in the text is vital in a wide range of vision-based application scenarios, such as Unmanned Aerial Vehicle (UAV) navigation, industrial automation, image search, robot navigation, intelligent navigation inspection, etc., [1]. As a result, the detection and recognition of scene text have become a popular and important research topic in computer

vision. Various backgrounds and arbitrary imaging conditions such as uneven illumination, low resolution, distortion, blurring, low contrast, and other artefacts discriminate text in natural scenes. Furthermore, the variety of font types and sizes available adds to the complexity of Scene Text Recognition (STR) algorithms.

There are two stages involved in recognizing the text, they are for text recognition: text detection and text recognition [2]. The goal of text detection is to find text areas in images, and the goal of text recognition is to transcribe the image into editable text format. Traditionally, Optical Character Recognition (OCR) techniques are used to recognize text from well-formatted scanned documents captured in a controlled environment. However, OCR techniques have exhibited excellent accuracy when recognizing characters in scanned document images. OCR has limited efficacy when used to recognize characters in images because of the distinctive properties of the text images [3]. There are two types of text images: Regular text with nearly horizontally aligned characters and Irregular text for the arbitrarily shaped text (e.g., curved text). Recognizing irregular text in an image is a challenging task.

CNNs such as VGGNet [4] and Residual Network (ResNet) [5] were utilized to extract features from the complex background images. The CTC and the encoder-decoder framework are two popular text recognition methods that use CNN as a feature extraction network to acquire high-level information. The CTC decoding module was designed for speech recognition [6]. Built-in Conditional Independence is one of CTC's flaws, and it's not ideal for many-to-many mapping. The combined decoding module has high recognition accuracy compared to a separate CTC or Attention method. On the other hand, Attention-based techniques failed to handle irregular text images because they cannot precisely match image feature regions and targets. In comparison, the encoder-decoder framework generates variable-length outputs, which satisfies the requirement of text recognition.

The summary of our work's contributions as follows:

- Designed a new framework by combining the CTC and Attention mechanism under encoder-decoder architecture to recognize the irregular texts from the images.
- Proposed a new Spatial Attention Module (SAM); it performs a feature filtering operation, enriching feature columns by eliminating redundancies and clutters. It can help to minimize the effects of inaccuracy in text detection.
- Proposed a Dual Step Attention Mechanism (DSAM); the first step in the attention process is to determine the visual and contextual elements that focus more on text regions. The second step concentrates more on the intra-sequence relationships. Finally, a Joint CTC-Attention decoder decodes both visual and contextual features simultaneously.
- The combined decoding module has high recognition accuracy compared to a separate CTC or Attention method. The proposed framework is optimized to be weakly supervised from top to bottom, requiring only ground truth text and input images. Extensive experiments on multiple benchmarks have been demonstrated that the proposed framework outperforms state-of-the-art text recognition methods on regular and irregular text datasets.

2 Related Works

Despite deep learning and CNN's dominance in visual recognition tasks [7–9], several new text recognition methods have been proposed, with significant improvements. Ghosh et al. [10] introduced a visual attention model using the Long-Short Term Memory (LSTM) network composed under the encoder-decoder. The model passes convolutional features from a typical CNN into an LSTM network, which selectively focuses attention to text regions of the image at each time step to recognize character sequence without using a pre-defined vocabulary. An Arabic text recognition system proposed by Zayene et al. [11] based on multi-dimensional LSTMs and CTC avoids complex OCR steps such as text line

segmentation and feature extraction. Liao et al. [12] use a common backbone architecture for detection and recognition tasks. The recognition module uses two different prediction branches. Semantic character segmentation was used to segment text images from the original image and a spatial attention decoder generates text from the segmented word image.

Liu et al. [13] proposed a method to detect and correct individual characters known as Character-Aware Neural Network (Char-Net). The attention network comprises of two levels: The character-level encoder uses a spatial transformer to reduce character distortions. The word-level encoder generates the feature map input into the recurrent RoIWarp layer. The decoder fed the feature maps into the LSTM. This method has significantly increased its ability to deal with various types of distortion. Li et al. [14] introduced an off-the-shelf deep neural network framework composed of ResNet, LSTM, and a 2D attention module called Show Attend and Read (SAR). The LSTM encoder was used to encode the feature map and considered as a holistic feature. The LSTM decoder was used to decode the feature maps into character sequences. 2D attention module was used to compute the sum of weights of features and efficiently handle text irregularity. To tackle the occurrence of undesired noise during text recognition, Huang et al. [15] proposed an Effective Parts Attention Network (EPAN). The Character Effective Parts Decoder (CEPD) in EPAN decodes important features from feature maps to generate character informations. The Effective Parts Decoder (EPD) ignores the noise in feature maps and enhances character information.

Chen et al. [16] proposed an Adaptive Embedding Gate (AEG) module that uses character language modelling to adaptively strengthen the impact of recent predictions in the decoding phase. Wang et al. [17] designed a Memory Augmented Attention Network (MAAN) to predict the character at the current time step by feeding the previously computed character sequence and entirely attended alignment history to the attention network. As a result, the glimpse vector computed by MAAN is more robust for character representation at each timestamp. Lee et al. [18] introduced a recursive CNN for image encoding; to extract contextual information, followed by feature selection and decoding using a 1D attention model. The fiducial transformation points on the slope and arbitrary texts are regressed in ASTER [19].

Lin et al. [20] applied a series of simple transformations such as translation, scaling and rotation to obtain flexible rectification. They also proposed a grid projection approach for smoother sequential transformation to make recognition tasks easier. However, this approach failed to recognize text under certain situations, such as severely curved text, low-resolution text images and noisy images. Yang et al. [21] suggested utilizing a coarse-to-refined method to decompose the classification process. Instead of decoding the character in a single iteration, a single attention module was utilized multiple times to improve the appropriate features until the outcome was satisfactory. This technique produces impressive results in character recognition. Bai et al. [22] developed a new method termed Edit Probability (EP) to solve the misalignment problem. The Arbitrary Orientation Network (AON) was proposed by Cheng et al. [23] to capture visual features of irregular texts. A filter gate mechanism and an attention-based decoder are employed to retain the sequence of characters.

Thin-Plate Spline transformation (TPS) [24] was used to handle the irregular shaped text and feed it into an attentional sequence-to-sequence model to recognize the sequence of characters. TPS proposed two networks to recognize perspective text and curved text: a rectification network to solve the alignment problem and a recognition network for scene text recognition. Distorted scene text is recognized via iterative rectification [25]. Cheng et al. [26] proposed Focusing Attention Network (FAN) to tackle the "attention drift" problem; it was made up of two major sections: Attention Network (AN) to recognize target characters and Focusing network (FN) to evaluate AN attention on target areas. An end-to-end model based on an attention-based RNN recognizes irregularly shaped text by presenting two important components auxiliary dense character detection and an alignment loss. These two components achieve fast convergence and high classification accuracy for perspectively distorted text or curved text.

3 Text Recognition Framework

In this paper, a new encoder-decoder architecture for text recognition is proposed. The input text image is fed into the text recognition framework to carry out an accurate text recognition task. The overall architecture of the proposed text recognition framework is shown in Fig. 1. The encoder block comprises of image transformation, feature extraction with SAM, a deep Bi-LSTM network to mark the most contributive text features and a proposed DSAM model. The decoder block contains a Joint CTC-Attention joint decoding mechanism that decodes the output and produces the final character sequence. Each module in the architecture is explained in the following sections.

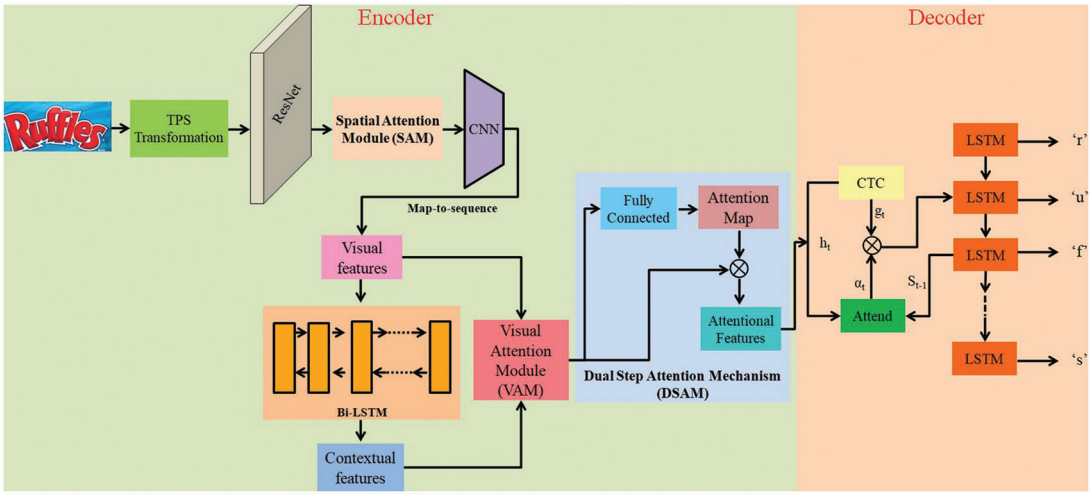

Figure 1: The overall architecture of the text recognition framework

3.1 Image Transformation

The main reason for employing an image transformation module is to normalize the input text image, eliminate distortion and simplify irregular text recognition tasks. In particular, the irregular text refers to text with a perspective distortion or an unusual curved shape, making recognition tasks more difficult. However, the text images come in various shapes, such as curved and slanted texts. If such diverse shaped input images are fed directly without altering, the feature extraction step must learn an invariant representation for such geometry. To reduce the burden of the additional learning process, it employs a TPS transformation to normalize the input text image (I).

The TPS transformation has been applied with its flexibility to various aspect ratios of text lines. TPS is an alternative to the Spatial Transformer Networks (STN) [27]. TPS interpolates between a collection of fiducial points using a smooth spline (see Fig. 2). TPS identifies a set of fiducial points at the top and bottom of the text area and reduces the projected region to a pre-defined size. Image Transformation is otherwise called as Image Normalization, in which the resulted normalized image is generally represented as (I').

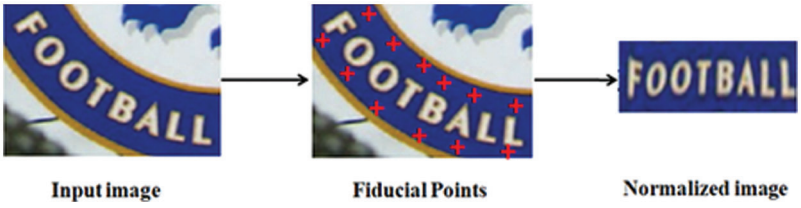

Figure 2: Illustration of the image transformation module

3.2 Spatial Attention Module (SAM)

Fig. 3 illustrates our proposed SAM. In this module, feature representation of the normalized image (I') is extracted using a fully convolutional network-based architecture that uses ResNet-34 as its backbone; the extracted features are named FeatureMaps. Original ResNet-34 has a stride of 32, which is large for cropped text images. As a result, significant adjustments are done to make it suitable for further processing. Also eliminated the 3 × 3 max-pooling and replace the 7 × 7 convolution with a 3 × 3 equivalent convolution. As an outcome, the height and width of FeatureMaps are remarkably reduced by 1/8 of the normalized image (I') and there are 512 channels.

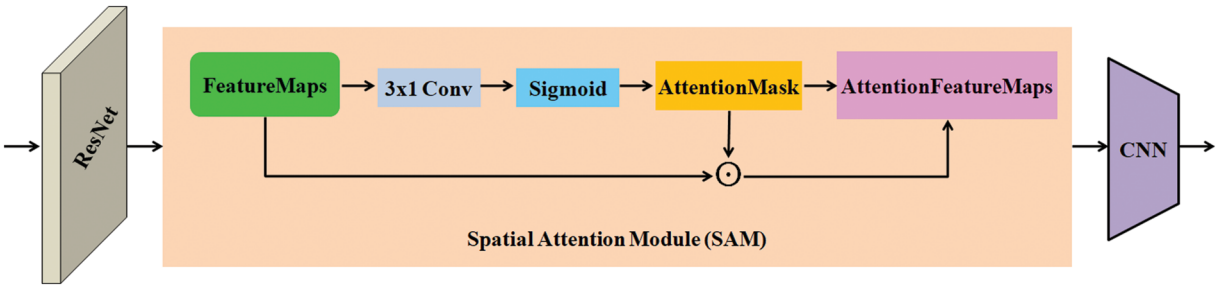

Figure 3: Illustration of spatial attention module (SAM)

The SAM is introduced to acquire high-level features. A 3 × 1 convolution with stride 1 × 1 is applied on the FeatureMaps, followed by a sigmoid activation function, which generates AttentionMask has the same resolution as FeatureMaps. The only difference between AttentionMask and FeatureMaps is the number of channels. AttentionMask contains only one channel, whereas FeatureMaps contains 512 channels. A high-level feature map (AttentionFeatureMaps) represented by $F = [f_1, f_2, \ldots, f_N]$ are generated by performing a broadcast element-wise product operation on FeatureMaps and AttentionMask. The SAM performs a feature filtering task, enriching feature columns by providing semantic information and eliminating clutters and redundancies in the feature maps. The proposed SAM guides the framework to focus on the most important parts of the text image. Meanwhile, it can also help to minimize the effects of false text localization; for example, the localized bounding box isn't near enough to the text region.

3.3 Visual Attention Features (VAF) Extraction

In this module, Recurrent layers are layered on top of convolutional layers to create a deep bi-directional Recurrent Neural Network (RNN). The recurrent layers have three advantages: RNN specializes in extracting contextual information within a sequence. For image-based sequence recognition tasks, these contextual cues are more helpful and accurate than handling each symbol independently. In addition, it allows training both the recurrent and convolutional layers in a fused network, because it has the ability to propagate error gradients back to its input. It can function on sequences of any length. LSTM is a class of RNN units. Hence, a bi-directional LSTM is designed by combining two LSTMs, one forward and one backward. A deep bi-directional LSTM is developed by stacking multiple bi-directional LSTMs. The deep structure provides a higher degree of abstraction than the shallow structure.

AttentionFeatureReps are transformed into a feature sequence. Assume that AttentionFeatureReps are H × W × D in size, where H, W and D represent the height, width and depth respectively. AttentionFeatureReps are converted into W feature vectors using a CNN. The propagated gradients are fused into maps at the bottom of the recurrent layers, reversing the translation of maps into sequences and giving back to the convolutional layers. This process is known as Map-to-Sequence, as used in [28].

Visual features are represented by $V = [v_1, v_2,\ldots,v_N]$, where N is the length of the feature sequence that contains valuable context information. Visual features are considered as the most crucial factor for the text recognition process. The CNN's receptive field is limited where the extracted features may suffer from a lack of contextual information. So, it employs a deep multi-layer Bi-LSTM to derive long-term dependencies in both directions, conduct bidirectional analysis of feature sequences and output additional contextual features of the same length to extend. The contextual feature can be represented as $C = [c_1, c_2, \ldots,c_N]$. The output of multi-layer Bi-LSTM is concatenated, i.e., contextual feature with the visual feature, which yields a new feature space J = (V, C) named as visual attention feature map represented by $J = [j_1, j_2,\ldots,j_N]$.

3.4 Dual Step Attention Mechanism (DSAM)

The schematic illustration of the proposed DSAM is shown in Fig. 1. Here, a 1D self-attention operation [29] is employed on a visual attention feature map (J) from these feature maps; an attention map (A) is computed using a fully connected layer. Next, the attentional features (H) are calculated by performing an element-wise product between a visual attention feature map (J) and an attention map (A). The newly generated attentional features (H) are decoded using a CTC-Attention-decoder [30], for each time the decoder generates an output (y_t) using the Eq. (1).

$$y_t = \text{Generate}(s_t, g_t) \tag{1}$$

$$s_t = \text{RNN}(y_{t-1}, g_t, S_{t-1}) \tag{2}$$

where s_t and g_t in Eqs. (2) and (3) are RNN hidden states at time t and a weighted sum of sequential feature vectors ($h_1, h_2, \ldots, h_k$). A feed-forward network and an LSTM recurrent network are represented by the Generate and RNN functions, respectively. The decoder produces a vector called glimpse by linearly combining a vector G with the columns of H.

$$g_t = \sum_{j=1}^{n} \alpha_{t,j} h_j \tag{3}$$

where $\alpha_t \in R^T$ is a vector of attention weights given in Eqs. (4) and (5), also known as alignment factors. Then, α_t is calculated by scoring each weighted sum of sequential feature vectors ($h_1, h_2,\ldots,h_k$) separately and normalizing the computed scores.

$$e_{t,i} = w_k \tanh\left(Ws_{t-1} + Vh_j, + b\right) \tag{4}$$

$$\alpha_{t,j} = \frac{\exp\left(e_{t,j}\right)}{\sum_{j=1}^{k} \exp\left(e_{t,j}\right)} \tag{5}$$

where V, W, b, and u are trainable parameters. At time t, S_{t-1} is the hidden state of the recurrent cell within the decoder, and h_j is a column of H. Following that, the decoder's recurrent cell is fed with Eq. (6).

$$(xt, st) = \text{RNN}(St - 1, (gt, f(yt - 1)) \tag{6}$$

The concatenation of gt and one-hot embedding y_{t-1}, represented by $(gt,f(y_{t-1}))$. The probability of a given character can be calculated using the following Eq. (7):

$$p(y_t) = \text{softmax}(W_0 x_t + b_0) \tag{7}$$

The following equation Eq. (8) is used to compute the attention model's loss function:

$$L_{attn} = \sum_{t} \ln P(\hat{y}_t | I', \theta) \tag{8}$$

where $\hat{y}_t$ and θ are the ground truth of t^{th} character and a vector containing all the network's parameters.

3.5 CTC-Attention Decoder

In this paper, global attention replaces with local attention for decoding. Here, a directed one-way Long Short Term Memory (LSTM) network is used for decoding. To enhance the performance of the recognition network, the CTC module and the Attention module get integrated and made as CTC-Attention mechanism [31]. The CTC-Attention integrated approach can take all the necessary feature information and utilize this feature information by computing global probability in the CTC. The decoding network decodes the attention-based feature sequence (H) produces $Y = (y_1, y_2, \ldots, y_k)$ under the joint CTC-Attention mechanism. This integration process improves recognition accuracy and speeds up network convergence.

The decoder input attention feature sequence $H = [h_1, h_2, \ldots, h_n]$, hidden variable $Z = \{z_t \in D \cup \text{blank} \mid t = 1, \ldots, W\}$, Here, t and W are the total number of characters and encoding sequence respectively, produces a character sequence of length (l), Attention feature sequence $H = \{h_l \in D \mid l = 1, \ldots, W\}$, The character dictionary (D) contains all the available characters. The posterior probability of the predicted text sequence is computed using Bayes' theorem given in Eq. (9); the CTC strategy assumes that generated labels are independent.

$$P_{CTC} = (Y|X) \approx \sum_{Z} \prod_{t} p(z_t | z_{t-1}, Y) p(z_t | X) p(Y) \tag{9}$$

where $p(Y)$ is letter-based language model, $p(z_t | z_{t-1}, Y)$ is the hidden variable's conditional probability derived from the hidden variable at the previous timestamp, and the probability of a hidden variable is represented by $p(z_t | X)$ derived from the input feature map. The probability of the joint attention predicted text sequence is given in Eq. (10).

$$P_{atten} = (Y|X) \approx \prod_{l} p(y_l | y_{1:l-1}, X) \tag{10}$$

where $p(y_l | y_{1:l-1}, X)$ is the probability of the predicted text sequence, X is the input characteristic, and l is the output obtained. The attention mechanism can predict long text sequences [32]; it failed to impose any constraint that leads to misalignment during the decoding process and is sensitive to noise. The two crucial difficulties, such as training long sequence input and extracting information from extended characters, are effectively addressed by the CTC-Attention algorithm. The linear combination of the joint CTC-Attention prediction is given by:

$$\hat{Y} = \arg\max_{Y \in D} \{\lambda \log p_{ctc}(Y|X) + (1 - \lambda) \log p_{atten}(Y|X)\} \tag{11}$$

The decoder calculates a score for each partial hypothesis during the word beam search process using Eq. (11). Combining the CTC-Attention score in the beam search is unfair because CTC performs frame synchronously while attention performs out label synchronously. Rescoring [33] and the forward algorithms [34] incorporate CTC probabilities into the hypothesis score. Rescoring generates and rescores complete hypotheses. The forward algorithm computes CTC probabilities (see Eq. (11)). CTC and the attention model to calculate the probability of each partial hypothesis in One-Pass decoding based on Eq. (12). The one-pass CTC-Attention decoding is given in Tab. 1.

Table 1: Joint CTC-attention decoding algorithm

Function *one_pass_beam_search(X, S_{max})*
{
S_{max} = *the maximum number of hypotheses to be examined*
Ψ_0 = {<sos>}
$\hat{\Psi}$ = ø
for *l = 1 to* S_{max} **do**
Ψ_l = ø
while Ψ_{l-1} != ø **do**
b = head (Ψ_{l-1})
dequeue(Ψ_{l-1})
for each *t* ϵ Υ ∪ {<eos>} **do**
h=b.t
$\alpha(Y)$ = {λ *log* p_{ctc} *(Y|X)+ (1-*λ*) log* p_{atten} *(Y|X)}*
if *t =* <eos> **then**
enqueue($\hat{\Psi}$,*Y)*
else
enqueue(Ψ_l,*Y)*
if $|\Psi_l| > beamwidth$ **then**
removeworst(Ψ_l)
end if
end if
end for
end while
if *stopdetect*($\hat{\Psi}$,*l) = true* **then**
break
end if
end for
return *argmax*$\hat{Y}\epsilon$ $\hat{\Psi}$ $\alpha(\hat{Y})$
}

Function *log* p_{ctc} *(Y|X), Y, X*
{
g,c ← *Y*
if *c =* <eos> **then**
return $log\{\gamma_t^{(n)}(g) + \gamma_t^{(b)}(g)\}$
else
$\gamma_t^{(n)}(Y) \leftarrow \begin{cases} p(z_1 = c|X) & if\ g = <sos> \\ 0 & otherwise \end{cases}$
$\gamma_t^{(b)}(Y) \leftarrow 0$
$\varphi \leftarrow \gamma_t^{(n)}(Y)$
for *t=2,...., T* **do**
$\beta \leftarrow \gamma_{t-1}^{(b)}(g) + \begin{cases} 0 & if\ last(g) = 0 \\ \gamma_{t-1}^{(n)}(g) & otherwise \end{cases}$
$\gamma_t^{(n)}(Y) \leftarrow (\gamma_{t-1}^{(n)}(Y) + \beta)\ p(z_t = c|X)$
$\gamma_t^{(b)}(Y) \leftarrow (\gamma_{t-1}^{(b)}(Y) + \gamma_{t-1}^{(n)}(Y))$ $p(z_t = <b>|X))$
$\varphi \leftarrow \varphi + \beta\ .p(z_t = c|X)$
end for
return *log*(φ)
end if
}

$$p_{ctc}(h, \ldots|X) = \sum_{v \in (u \cup \{<eos>\})^+} p_{ctc}(h.v|X) \tag{12}$$

CTC score is computed based on Eq. (13),

$$\alpha_{ctc}(h, X) = \log p_{ctc}(h, \ldots.|X) \tag{13}$$

where v contains all possible label sequences does not contain empty sequence characters.

4 Experiment

In this section, the performance of the proposed text recognition framework is evaluated against state-of-the-art algorithms on several public benchmark datasets, including both regular and irregular datasets.

4.1 Datasets

In this paper, two synthetic datasets (SynthText, SynthAdd) proposed by Gupta et al. [35] and Jaderberg et al. [36] respectively, are used to train the proposed framework. In addition, seven standard benchmarks that contain four 'regular' datasets (IC03, IC13, SVT, IIIT5K) and three 'irregular' datasets (IC15, SVTP, CUTE) are used to evaluate the proposed framework.

Regular Datasets the performance of the proposed framework was evaluated using standard benchmark datasets such as IIIT5K-Words [37], Street View Text [38], ICDAR 2003 [39] and ICDAR 2013 [40]. The majority of the text images in these datasets are almost horizontal text images.

Irregular Datasets ICDAR 2015 Incidental Text [41], Street View Text Perspective [42] and CUTE80 [43] are the benchmark datasets used to evaluate the performance of our framework. Most text images in this dataset are curved, rotated and low-quality.

Grocery Datasets Four publicly available datasets such as GroZi-120 [44], WebMarket [45] and SKU-110K [46], Freiburg Groceries Dataset [47] are used to train and test the proposed text recognition framework.

4.2 Implementation Details

Our framework experiments are performed using PyTorch on an NVIDIA GTX 1080Ti GPU with 12 GB memory. The training datasets SynthText and SynthAdd consists of 6-million and 8-million synthetic images, respectively. Additionally, text images from grocery datasets that contain complex text shapes and styles are cropped to train our text recognition framework. The framework is trained using the AdaDelta optimizer. Different batch sizes were rehearsed, and it was observed that a batch size of 128 was found to be the most efficient. Similarly, various learning rates were examined, and a learning rate of 0.01 was the most effective. The number of training epochs was set to 80 and 36 different symbol classes were employed including 10 digits and 26 letters. Three special punctuation letters are added to the decoder: "<sos>", "<eos>", and "<unk>", which indicate the start, end, and unknown characters, respectively. The average decoder probabilities are calculated until the "<eos>" symbol is encountered to compute a prediction confidence score. The final prediction is then decided as the highest confidence score. The beam-search algorithm was used for decoding. However, the authors of [14] claim it increases accuracy by around 0.5%. The world beam algorithm-based text recognition decoding enhances performance by eliminating grammatical mistakes, allowing arbitrary numbers and punctuation marks, and employing a word-level language model.

4.3 Ablation Studies

In this section, experimental trials were carried out to interpret the comparison of performance improvements and assess the effect of our key contributions. Tab. 2 shows the impact of the key contributions in the text recognition framework. Experiments (b), (d), (f) and (i) show the effect of the image transformation module in the text recognition task. The image transformation module reduces the burden of feature extraction steps and normalizes the distorted or irregular text images. The transformation module significantly improves the performance of the proposed framework on all datasets, notably on IC15 (1.1%), SVTP (0.8%) and CUTE (0.6%).

The impact of the proposed Spatial Attention Module (SAM) is shown in rows (a), (d), (g), (h) and (i) of Tab. 2. SAM is primarily designed to suppress redundant features and clutters. Compared to baseline (without SAM), accuracy was improved around 3.1% on IC15, 4.3% on SVTP and 3.7% on CUTE80 datasets. The performance of the SAM can be improved, particularly for the irregular datasets (IC15, SVT, and CUTE), which comprise low-quality text images and complex text shape images. Similarly, The importance of obtaining the Visual Attention Feature (VAF) is given in rows (b), (c), (d),

(g) and (i) of Tab. 2. The CNN's receptive field is limited. The features it extracts may suffer from a lack of contextual information. A deep bi-directional LSTM network was used to generate a contextual feature map over CNN's outputting feature map to address this shortcoming. Then, the contextual feature map from the deep Bi-LSTM network is combined with the visual feature map to generate the contributive VAF. The VAF greatly supports the proposed framework to improve its recognition accuracy marginally on both regular (3.4% on IC13 and 3.9% on IIIT5K) and irregular datasets (5.4% on IC15, 2.8% on SVTP and 3.9% on CUTE80) by comparing with baseline. The effect of our final key contribution, DSAM, is given in rows (b), (f), (g), (h) and (i) of Tab. 2. The sixth row shows the improvement over the baseline results on regular text datasets (3.5% on IC03, 2.5% on IC13, 5.7% on SVT and 2.7% on CUTE80) and irregular text datasets (4.4% on IC15, 2.2% on SVTP and 1.5% on CUTE80).

Table 2: Performance comparison among four variations. In the rectification column, "Yes" indicates image normalization is performed. "No" indicates image normalization is not performed. Our key contributions: SAM, VAF, and DSAM are achieved superior performance on the text recognition task. ✗ represent the particular proposed module is not carried out for the particular iteration. ✓ represent the particular proposed module is performed for the specific iteration

Exp	Rectification	Key contributions			Regular text dataset				Irregular text dataset		
		SAM	VAF	DSAM	IIIT 5K	SVT	IC 03	IC 13	IC 15	SVT-P	CUTE80
(a)	No	✓	✗	✗	83.3	82.8	79.2	81.6	77.5	80.2	72.5
(b)	Yes	✗	✓	✓	92.6	91.6	89.4	90.2	84.8	84.6	82.1
(c)	No	✗	✓	✗	84.2	85.1	82.8	84.2	82.9	84.1	79.5
(d)	Yes	✓	✓	✗	90.8	93.5	92.4	93.6	84.9	88.0	82.6
(e)	No	✗	✗	✗	82.4	81.7	74.6	80.3	77.5	81.3	75.6
(f)	Yes	✗	✗	✓	85.9	84.2	80.3	83.0	81.9	83.5	77.1
(g)	No	✓	✓	✓	95.9	95.7	94.0	94.0	86.0	87.9	85.2
(h)	No	✓	✗	✓	93.0	92.3	91.2	91.8	85.1	86.3	84.3
(i)	Yes	✓	✓	✓	**96.3**	**96.1**	**94.7**	**94.7**	**87.1**	**89.7**	**87.8**

Our proposed framework used a powerful feature extractor and advanced attention-based sequence network, which deals with complex texts. Compared to the separate CTC or Attention mechanism, the combined CTC-Attention mechanism substantially improved text recognition accuracy and the training speed of the model is much faster than existing methods. Unlike traditional attention decoders that perform recognition alone, our decoupled text decoder takes encoded features and attention maps as input and performs alignment and recognition simultaneously. In our encoder-decoder framework, the total LSTM layers in the model significantly impact recognition accuracy. As the number of LSTM layers increases, it is observed that the encoder has a higher advantage over the decoder.

The two-layer LSTM decoder and the three-layer LSTM encoder is used in our experiment shown in Tab. 3; the accuracy value of the model was better. Despite this, an attempt to expand the layers count in both the encoder and decoder is performed, but the model faces an overfitting problem. The CTC-Attention-based text recognition model can recognize the curved shaped text, image with non-uniform spacing between the words and multiple texts in an image. Our architecture is built with four Bi-LSTM

layers based on intermediate supervision levels. The relative improvement in accuracy on regular and irregular text is +0.4% and +1.8%, respectively.

Table 3: Experimental results are comparing the number of LSTM layers in decoding *vs*. Recognition accuracy

Decoding layer	Regular text dataset				Irregular text dataset		
	IIIT5K	SVT	IC03	IC13	IC15	SVT-P	CUTE80
1	85.4	82.8	82.3	86.5	81.7	80.0	81.0
2	91.2	88.7	88.8	92.6	86.6	86.4	84.5
3	**96.3**	**96.1**	**96.6**	**94.7**	**87.1**	**89.3**	**87.8**
4	89.7	87.3	86.1	88.4	86.2	86.7	82.6
5	84.7	84.0	82.3	84.5	85.1	83.9	84.1

By adjusting the learning hyperparameter value (λ) between 0 and 1, the CTC puts various weight restrictions on attention. A learning hyperparameter (λ) is introduced in the joint decoding to balance the weight of the CTC and Attention mechanism. When λ = 0, attention alone is used for decoding; similarly, When λ = 1, CTC alone is used for decoding. A possible value between 0 and 1 is taken as a hyperparameter value shown in Tab. 4. When λ = 0.4, the most significant level of recognition accuracy; when the hyperparameter increases gradually, the accuracy decreases. Compared to the separate CTC or Attention method, the combined CTC-Attention mechanism achieves high recognition accuracy.

Table 4: Experimental results varying Hyperparameter *vs*. Recognition accuracy

Hyperparameter	Regular text dataset				Irregular text dataset		
	IIIT5K	SVT	IC03	IC13	IC15	SVT-P	CUTE80
CTC (λ = 0)	88.1	85.9	89.5	90.5	84.2	83.6	81.7
Attention (λ = 1)	88.6	86.3	89.7	90.6	83.1	82.4	83.5
CTC – Attention (λ = 0.1)	91.0	89.4	90.3	92.1	84.7	84.6	83.8
CTC – Attention (λ = 0.2)	93.4	93.9	92.0	93.8	87.5	85.4	85.2
CTC – Attention (λ = 0.3)	93.7	94.6	92.6	94.5	89.4	88.9	87.4
CTC – Attention (λ = 0.4)	**96.3**	**96.1**	**96.6**	**94.7**	**87.1**	**89.3**	**87.8**
CTC – Attention (λ = 0.5)	93.7	93.2	93.4	95.0	92.8	91.6	88.3
CTC – Attention (λ = 0.6)	92.5	92.4	92.7	94.6	91.4	90.7	87.5

4.4 Comparison with Existing Algorithms

In this section, the recognition accuracy of the proposed method is compared with state-of-the-art methods on both regular and irregular datasets are shown in Tab. 5. Some precise and false recognition results on the public benchmark text recognition and retail product datasets are shown in Fig. 4, green

characters represent correctly predicted characters, and red characters represent wrongly predicted characters. Liao et al. [12] and Liu et al. [13] used a self-designed feature extractor to generate a high-level feature extractor. Unlike Liao et al. [48], word-level annotated text images are used to train our framework. A powerful feature extractor, ResNet is utilized as CNN's Backbone to obtain relevant and high-level features. Our proposed text recognition framework obtained the highest accuracy on the IIIT5K, SVT, IC03, IC15 and SVT-P datasets and the second-highest on the CUTE80 dataset in lexicon-free mode.

Table 5: Comparisons with existing methods over seven public benchmark datasets. All the values from columns 3 to 9 are expressed in percentage (%). In column 2, "90 K", "ST", "SA" and "Wiki" stand for Synth90K, SynthText, SynthAd and WikiText-103 respectively. Also, "Self" denotes the use of a self-designed convolution network or self-made synthetic datasets. The outcomes are all in the no lexicon indicated by "None"

Method	ConvNet, Data	Regular text				Irregular text		
		IIIT5K	SVT	IC03	IC13	IC15	SVT-P	CUTE80
		None	None	None	None	None	None	None
Liao et al. [12]	Self, 90K+ST	95.3	91.8	95.0	95.3	78.2	83.6	88.5
Liu et al. [13]	Self, 90K+ST	83.6	84.4	91.5	90.8	–	73.5	–
Li et al. [14]	ResNet, 90K+ST	95.0	91.2	–	94.0	78.8	86.4	89.6
Huang et al. [15]	ResNet, 90K+ST	94.0	88.9	95.0	94.5	73.9	79.4	82.6
Shi et al. [19]	ResNet, 90K+ST	93.4	93.6	94.5	91.8	78.5	76.1	79.5
Lin et al. [20]	ResNet, 90K+ST	94.1	90.6	95.1	92.8	76.7	82.2	83.3
Yang et al. [21]	ResNet, 90K+ST	88.3	87.5	94.6	94.4	–	73.9	–
Cheng et al. [23]	BCNN, 90K+ST	87.0	82.8	91.5	–	68.2	73.0	76.8
Shi et al. [24]	CNN, 90K+ST	81.9	81.9	90.1	88.6	71.8	–	59.2
Zhan et al. [25]	ResNet, 90K+ST	93.3	90.2	95.0	92.4	79.6	76.9	83.3
Cheng et al. [26]	ResNet, 90K+ST	87.4	85.9	94.2	93.3	71.5	–	63.9
Shi et al. [28]	CNN, ST	78.2	80.8	89.4	86.7	66.8	–	54.9
Gao et al. [29]	Self, ST	81.8	82.7	89.2	88.0	–	–	–
Yu et al. [30]	ResNet, 90K+ST	94.8	91.5	–	95.5	82.7	85.1	87.8
Wang et al. [31]	ResNet, 90K+ST	93.2	89.2	94.3	92.8	76.6	82.6	82.6
Lu et al. [32]	ResNet, 90K+ST+SA	95.0	90.6	96.4	95.3	79.4	84.5	87.5
Wu et al. [33]	ResNet, 90K+ST	89.5	86.5	94.1	90.1	76.6	–	78.0
Zhang et al. [34]	ResNet, 90K+ST+Wiki	96.2	93.9	–	**97.7**	85.5	89.0	**91.9**
Liao et al. [48]	ResNet, ST	91.9	86.4	–	91.5	–	–	79.9
Litman et al. [49]	ResNet, 90K+ST+SA	93.7	92.7	96.3	93.9	82.2	86.9	87.5
Luo et al. [50]	Self, 90K+ST	91.2	88.3	95.0	92.4	76.1	68.8	77.4
Liu et al. [51]	CNN, ST	83.3	83.6	89.9	89.1	73.5	–	–
Liu et al. [52]	Self, 90K	89.4	87.1	94.7	94.0	73.9	–	62.5
Yang et al. [53]	ResNet, 90K+ST	94.4	88.9	95.0	93.9	78.7	80.8	87.5
Proposed Framework	ResNet, 90K+ST	**96.3**	**96.1**	**96.6**	94.7	**87.1**	**89.3**	87.8

Figure 4: Visualization of precise and false results by our proposed framework on both regular and irregular text datasets. Green indicates true predicted characters whereas red indicates false predicted characters

Our proposed framework outperforms the state-of-the-art techniques such as Liao et al. [12] and Li et al. [14] approach by an absolute margin of 3.8% and 2.3% on average, respectively, on irregular text datasets (IC15, SVTP, CUTE). Our framework improves the recognition rate of irregular texts, such as those with a curved shape, perspective distortion, or arbitrarily orientated, common yet difficult to recognize. Litman et al. [49] used the additional dataset (SynthAdd represent as SA) in their model. Even though, our framework surpasses Litman et al. [49] on both regular and irregular text datasets.

Similarly, Zhang et al. [34] utilized an additional training dataset (WikiText-103 represented as Wiki). However, proposed framework outperforms Zhang et al. [34] on IC15 (87.1% *vs*. 85.5%) and SVT-P (89.3% *vs*. 89.0%) datasets, whereas it surpasses proposed framework on CUTE (91.9% *vs*. 87.8%) dataset. The proposed text recognition framework accuracy is considerably outperformed the linguistic-based approaches (such as Luo et al. [50]) on all datasets, notably irregular texts (2.5% on IC15 and 3.6% on CUTE).

Our framework performs substantially better on regular and irregular text datasets than the baseline approach (see Tab. 2). It boosts accuracy by 13.9 % (from 82.4% to 96.3%) on IC03, 14.4% (from 80.43% to 94.7%) on IIIT5K, 14.4% (from 81.7% to 96.1%) on IC13, 20.1 % (from 74.6% to 94.7%) on SVT, 9.6% (from 77.5% to 87.1%) on IC15, 8.4% (from 81.3% to 89.7%) on SVT-P and 12.2% (from 75.6% to 87.8%) on CUTE80 datasets. This indicates that our key contributions, such as SAM, VAF and DSAM, effectively recognize text. To conclude, our framework obtains the top recognition score on five (IIIT5K, SVT, IC03, IC15 and SVT-P) out of seven benchmarks. Unlike existing approaches, the proposed framework is a top performer as it works on both regular and irregular text datasets. However, the additional training with cropped text images from the grocery dataset improved the recognition performance significantly by around 0.5%.

5 Conclusion

In this paper, a simple but effective text recognition model based on an encoder-decoder framework is presented. The main aim of the proposed work is to recognize both regular and irregular text. Different modules supported our framework to accomplish the task. Thin Plate Spline (TPS) is incorporated to convert the irregular text image into more readable text. The proposed Spatial Attention Module (SAM) extracts the enriched feature maps. It avoids the problem of learning an invariant representation by directing the model to focus more on the text regions. The proposed Dual Step Attention Mechanism (DSAM) integrated with the CTC-Attention decoder to generate a more accurate character sequence. The joint one-pass CTC-Attention decoding helps to boost the convergence of the training and enhances sequence recognition. The research trials infer that the CTC-Attention joint mechanism increases the model's text recognition performance and gives it an advantage in recognizing text images. The robustness of our approach is evaluated using public benchmarks, including the grocery datasets, such as GroZi-120, WebMarket, SKU-110K, and Freiburg Groceries datasets, which contain complex text shapes. Our proposed framework outperforms the state-of-the-art approaches on both regular and irregular text recognition methods.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Chen, L. Jin, Y. Zhu, C. Luo and T. Wang, "Text recognition in the wild: A survey," *Journal of the Association for Computing Machinery*, vol. 54, no. 2, pp. 42:1–42:35, 2021.

[2] J. Seytre, J. Wu and A. Achille, "Texttubes for detecting curved text in the wild," *Computing Research Repository (CoRR)*, vol. 1912.08990, pp. 1–10, 2019.

[3] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 38–62, 2000.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of the Int. Conf. on Learning Representations*, San Diego, USA, pp. 2326–2335, 2015.

[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770–778, 2016.

[6] T. Hori, Y. Kubo and A. Nakamura, "Real-time one-pass decoding with recurrent neural network language model for speech recognition," in *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing*, Florence, Italy, pp. 6364–6368, 2014.

[7] S. N. Kumar, A. F. Lenin, P. Padmanabhan, B. Gulyas, H. A. Kumar *et al.,* "Deep learning algorithms in medical image processing for cancer diagnosis: Overview, challenges and future," *Deep Learning for Cancer Diagnosis*, vol. 908, pp. 37–66, 2021.

[8] S. N. Kumar, J. Sebastin and H. A. Kumar, "Segmentation of magnetic resonance brain images using 3D convolution neural network," in *Deep Learning for Biomedical Applications*, 1st ed., vol. 1, Boca Raton, CRC Press, pp. 63–82, 2021.

[9] S. N. Kumar, A. L. Fred, H. A. Kumar, P. S. Varghese and A. S. Jacob, "Segmentation of anomalies in abdomen CT images by convolution neural network and classification by fuzzy support vector machine," *Hybrid Machine Intelligence for Medical Image Analysis*, vol. 841, pp. 157–196, 2020.

[10] S. K. Ghosh, E. Valveny and A. D. Bagdanov, "Visual attention models for scene text recognition," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, Kyoto, Japan, pp. 943–948, 2017.

[11] O. Zayene, S. E. Amamou and N. E. BenAmara, "Arabic video text recognition based on multi-dimensional recurrent neural networks," in *Proc. of the Int. Conf. on Computer Systems and Applications*, Hammamet, Tunisia, pp. 725–729, 2017.

[12] M. Liao, P. Lyu, M. He, C. Yao, W. Wu *et al.,* "MaskTextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Lecture Notes in Computer Science*, Cham: Springer, pp. 71–88, 2018.

[13] W. Liu, C. Chen and K. Y. K. Wong, "Char-Net: A character-aware neural network for distorted scene text recognition," in *Proc. of the AAAI Conf. on Artificial Intelligence*, New Orleans, USA, pp. 7154–7161, 2018.

[14] H. Li, C. Shen, P. Wang and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. of the Artificial Intelligence*, Honolulu, USA, pp. 8610–8617, 2018.

[15] Y. Huang, Z. Sun, L. Jin and C. Luo, "EPAN: Effective parts attention network for scene text recognition," *Neurocomputing*, vol. 376, no. 7, pp. 202–213, 2020.

[16] X. Chen, T. Wang, Y. Zhu, L. Jin and C. Luoa, "Adaptive embedding gate for attention-based scene text recognition," *Neurocomputing*, vol. 381, no. 11, pp. 261–271, 2020.

[17] C. Wang, F. Yin and C. Liu, "Memory-augmented attention model for scene text recognition," in *Proc. of the ICFHR*, Niagara Falls, USA, pp. 62–67, 2018.

[18] C. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. of the Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 2231–2239, 2016.

[19] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao *et al.,* "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2035–2048, 2019.

[20] Q. Lin, C. Luo, L. Jin and S. Lai, "STAN: A sequential transformation attention-based network for scene text recognition," *Pattern Recognition*, vol. 111, no. 9, pp. 1–9, 2021.

[21] X. Yang, D. He, Z. Zhou, D. Kifer and C. L. Giles, "Improving offline handwritten chinese character recognition by iterative refinement," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, Kyoto, Japan, pp. 5–10, 2017.

[22] F. Bai, Z. Cheng, Y. Niu, S. Pu and S. Zhou, "Edit probability for scene text recognition," in *Proc. of the Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 1508–1516, 2018.

[23] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu *et al.,* "AON: Towards arbitrarily-oriented text recognition," in *Proc. of the Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 5571–5579, 2018.

[24] B. Shi, X. Wang, P. Lyu, C. Yao and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. of the Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 4168–4176, 2016.

[25] F. Zhan and S. Lu, "ESIR: End-to-end scene text recognition via iterative image rectification," in *Proc. of the Computer Vision and Pattern Recognition*, Long Beach, USA, pp. 2054–2063, 2019.

[26] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu *et al.,* "Focusing attention: Towards accurate text recognition in natural images," in *Proc. of the Int. Conf. on Computer Vision*, Venice, Italy, pp. 5086–5094, 2017.

[27] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. of the Neural Information Processing Systems*, Cambridge, USA, pp. 2017–2025, 2015.

[28] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[29] Y. Gao, Y. Chen, J. Wang, M. Tang and H. Lu, "Reading scene text with fully convolutional sequence modeling," *Neurocomputing*, vol. 339, no. 1, pp. 161–170, 2019.

[30] D. Yu, X. Li, C. Zhang, J. Han, J. Liu *et al.,* "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. of the Computer Vision and Pattern Recognition*, Seattle, USA, pp. 12110–12119, 2020.

[31] C. Wang and C. L. Liu, "Multi-branch guided attention network for irregular text recognition," *Neurocomputing*, vol. 425, no. 1, pp. 278–289, 2021.

[32] N. Lu, W. Yu, X. Qi, Y. Chen, P. Gong *et al.,* "MASTER: Multi-aspect non-local network for scene text recognition," *Pattern Recognition*, vol. 117, no. 6, pp. 1–10, 2021.

[33] Y. Wu, J. Fan, R. Tao, J. Wang, H. Qin *et al.,* "Sequential alignment attention model for scene text recognition," *Journal of Visual Communication and Image Representation*, vol. 80, pp. 1–8, 2021.

[34] Y. Zhang, Z. Fu, F. Huang and Y. Liu, "PMMN: Pre-trained multi-modal network for scene text recognition," *Pattern Recognition Letters*, vol. 151, no. 1, pp. 103–111, 2021.

[35] Gupta, A. Vedaldi and A. Zisserman, "Synthetic data for text localization in natural images," in *Proc. of the Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 2315–2324, 2016.

[36] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *Computing Research Repository (CoRR)*, vol. 1406.2227, pp. 1–12, 2014.

[37] K. Alahari Mishra and C. V. Jawahar, "Scene text recognition using higher order language priors," in *Proc. of the British Machine Vision Conf.*, Surrey, UK, pp. 127.1–127.11, 2012.

[38] K. Wang, B. Babenko and S. Belongie, "End-to-end scene text recognition," in *Proc. of the Computer Vision*, Barcelona, Spain, pp. 1457–1464, 2011.

[39] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong *et al.,* "ICDAR 2003 robust reading competitions: Entries, results, and future directions," *in Proc. of the Int. Journal of Document Analysis and Recognition*, vol. 7, no. 2–3, pp. 105–122, 2003.

[40] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda *et al.,* "ICDAR 2013 robust reading competition," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, Washington, USA, pp. 1484–1493, 2013.

[41] D. Karatzas, L. G. Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov *et al.,* "ICDAR 2015 competition on robust reading," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, Tunis, Tunisia, pp. 1156–1160, 2015.

[42] T. Q. Phan, P. Shivakumara, S. Tian and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. of the Computer Vision*, Sydney, Australia, pp. 569–576, 2013.

[43] A. Risnumawan, P. Shivakumara, C. S. Chan and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8027–8048, 2014.

[44] M. Merler, C. Galleguillos and S. Belongie, "Recognizing groceries in situ using in vitro training data," in *Proc. of the Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, pp. 1–8, 2007.

[45] Y. Zhang, L. Wang, R. Hartley and H. Li, "Where's the weet-bix?," *Lecture Notes in Computer Science*, Berlin, Heidelberg: Springer, pp. 800–810, 2007.

[46] E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger and T. Hassner, "Precise detection in densely packed scenes," in *Proc. of the Computer Vision and Pattern Recognition*, Long Beach, USA, pp. 5222–5231, 2019.

[47] P. Jund, N. Abdo, A. Eitel and W. Burgard, "The freiburg groceries dataset," *Computing Research Repository (CoRR)*, vol. 1611.05799, pp. 1–7, 2016.

[48] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang *et al.,* "Scene text recognition from two-dimensional perspective," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8714–8721, 2019.

[49] R. Litman, O. Anschel, S. Tsiper, S. Mazor and R. Manmatha, "SCATTER: Selective context attentional scene text recognizer," in *Proc. of the Computer Vision and Pattern Recognition*, Seattle, USA, pp. 11959–11969, 2020.

[50] C. Luo, L. Jin and Z. Sun, "MORAN: A multi-object rectified attention network for scene text recognition," *Pattern Recognition*, vol. 90, no. 9, pp. 109–118, 2019.

[51] W. Liu, C. Chen, K. Y. K. Wong, Z. Su and J. Han, "STAR-Net: A spatial attention residue network for scene text recognition," in *Proc. of the BMVC*, York, UK, pp. 1–13, 2016.

[52] Y. Liu, Z. Wang, H. Jin and I. Wassell, "Synthetically supervised feature learning for scene text recognition," in *Proc. of the ECCV*, Munich, Germany, pp. 435–451, 2018.

[53] M. Yang, Y. Guan, M. Liao, X. He, K. Bian *et al.,* "Symmetry-constrained rectification network for scene text recognition," in *Proc. of the ICCV*, Seoul, Korea (South), pp. 9146–9155, 2019.