A Transformer-Based Framework for Scene Text Recognition

PRABU SELVAM1, JOSEPH ABRAHAM SUNDAR KOILRAJ1, CARLOS ANDRÉS TAVERA ROMERO2, (Member, IEEE), MESHAL ALHARBI3, ABOLFAZL MEHBODNIYA4, (Senior Member, IEEE), JULIAN L. WEBBER4, (Senior Member, IEEE), AND SUDHAKAR SENGAN5, (Member, IEEE)

1School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu 613401, India
2COMBA R&D Laboratory, Faculty of Engineering, Universidad Santiago de Cali, Cali 76001, Colombia
3Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
4Department of Electronics and Communications Engineering, Kuwait College of Science and Technology, Safat 20185145, Kuwait
5Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, Tamil Nadu 627152, India

Corresponding author: Sudhakar Sengan (sudhasengan@gmail.com)

This work was supported by the Dirección General de Investigaciones at Universidad Santiago de Cali under Grant 01-2022.

ABSTRACT Scene Text Recognition (STR) has become a popular and long-standing research problem in computer vision communities. Almost all the existing approaches mainly adopt the connectionist temporal classification (CTC) technique. However, these existing approaches are not much effective for irregular STR. In this research article, we introduced a new encoder-decoder framework to identify both regular and irregular natural scene text, which is developed based on the transformer framework. The proposed framework is divided into four main modules: Image Transformation, Visual Feature Extraction (VFE), Encoder and Decoder. Firstly, we employ a Thin Plate Spline (TPS) transformation in the image transformation module to normalize the original input image to reduce the burden of subsequent feature extraction. Secondly, in the VFE module, we use ResNet as the Convolutional Neural Network (CNN) backbone to retrieve text image features maps from the rectified word image. However, the VFE module generates one-dimensional feature maps that are not suitable for locating a multi-oriented text on two-dimensional word images. We proposed 2D Positional Encoding (2DPE) to preserve the sequential information. Thirdly, the feature aggregation and feature transformation are carried out simultaneously in the encoder module. We replace the original scaled dot-product attention model as in the standard transformer framework with an Optimal Adaptive Threshold-based Self-Attention (OATSA) model to filter noisy information effectively and focus on the most contributive text regions. Finally, we introduce a new architectural level bi-directional decoding approach in the decoder module to generate a more accurate character sequence. Eventually, We evaluate the effectiveness and robustness of the proposed framework in both horizontal and arbitrary text recognition through extensive experiments on seven public benchmarks including IIIT5K-Words, SVT, ICDAR 2003, ICDAR 2013, ICDAR 2015, SVT-P and CUTE80 datasets. We also demonstrate that our proposed framework outperforms most of the existing approaches by a substantial margin.

INDEX TERMS Connectionist temporal classification, scene text recognition, self-attention, transformer, optical character recognition, deep learning.

I. INTRODUCTION Text information disclosed in natural scene images is vital for visual interpretation, conceptual understanding and reasoning [1]. Reading texts in natural scene images is useful in numerous deep learning systems [2] including, robot navigation, industrial automation and driver assistance. The areas of computer vision and pattern recognition have paid special attention to STR. Despite decades of research into Optical Character Recognition (OCR) [3], text recognition...
from natural scene images remains challenging due to a variety of factors, for example, huge variation in the text font and text colour, low contrast, complex backgrounds, perspective distortion, occlusion, uneven lighting and so on. A wide variety of STR approaches has been discussed in the literature with significant success in recent years, benefitting from the emergence of deep learning. Traditional approaches recognized text from scene images by first locating individual characters and then using a CNN to recognize each cropped character [4]. However, a great volume of inter-character and intra-character conflict will effectively decrease the recognition network’s performance. These methods rely largely on a robust character detector. An attentional Recurrent Neural Network (RNN) can effectively handle the sequence-to-sequence (seq2seq) problem of recognizing regular text. Whereas, recognizing arbitrary shape text is more challenging for an RNN model.

Conventional irregular STR methods can be divided into 4 main categories: text shape rectification-based approach [4], [5], Connectionist Temporal Classification (CTC) based approach [6], [7], multi-direction encoding-based approach [8], [9] and attention-based approach [10], [11], [12]. Shape rectification is used to normalize the word image, eliminate distortion and make irregular text recognition simpler. Methods based on shape rectification [5] attempt the transformation of irregular text to regular text before using regular text recognizers. A Spatial Transformer Network (STN) [13] is a first text normalization network, it was utilized to rectify individual character regions and complete word images. Later, Shi et al. [4] incorporated Thin-Plate Spline (TPS) transformation mechanism as a text normalization module, to handle more complex text distortions. Sophisticated rectification modules are necessary to manage a range of distortions, and they are becoming a new trend. However, these factors have an impact on the performance and memory use of recognition algorithms. CTC has made substantial improvements in a variety of areas, including voice recognition and web handwritten character identification. CTC is a widely used prediction algorithm in STR. Yu et al. [14] and Shi et al. [7] were the initial algorithms to apply CTC to STR, inspired by its achievement in speech recognition. Luo et al. [15] used a CTC-based prediction algorithm for model learning and have shown superior performance. Despite the remarkable performance in STR, However, CTC has certain shortcomings: CTC’s underlying approach is complex, CTC exhibits peaky distribution issues, and CTC is ineffective for two-dimensional prediction tasks like irregular STR.

Attention-based prediction methods have surpassed CTC in decoding in recent years, because of their capability to attend to the appropriate place. The RNN consistently use the attention mechanism in its prediction module for STR problem. The input text image pattern, the output text sequences pattern and its difference are primarily learned by the attention mechanism by examining the experience of the final character sequences and encoded feature vectors. A wide variety of attention-based approaches have evolved in the STR field, influenced by the growth of machine translation frameworks. There are various flaws in the attention mechanism: This technique needs additional storage space and computation power, it suffers from attention drift problem, and The latest attention mechanism research is primarily focused on languages with just a few character groups (e.g., English, French).

Transformer [16], a modern attention alternative, was extensively used to increase parallelization and minimize complexity for STR. Some efforts were made to replace recurrent neural networks with non-recurrent structures in the domain of regular text recognition, such as convolutional-based and attention-based approaches. Nevertheless, both approaches depend on seq2seq structures, which are inadequate for handling arbitrary shaped text. It’s worth mentioning that the Transformer is originally designed for language translation tasks such as English to French, French to English and so on. It takes one-dimensional sequences as its input. Since position information is not naturally encoded within the input set of sequences, the model is less sensitive to the positioning of input sequences than RNN and LSTM frameworks with associative bias. The transformer is permutative equivariance, because of its Self-Attention (SA) and Feed-Forward Network (FFN) layers, which calculate the result of each component in the input sequence separately. While the 1D Positional Encoding (PE) approach employed in Transformer may handle the permutation equivariance issue that may arise in 1D sequences associated with NLP, it cannot preserve the horizontal and vertical features produced by CNNs for a 2D input image.

In short, the primary contributions of our research work are as follows:

- **Transformer [16]** in Natural Language Processing (NLP), which takes only 1D sequences as its input. On the other hand, the scene text recognizers can only handle 2D images. To solve this permutation equivariance problem and to preserve the order of sequential information, we modify the conventional transformer architecture to recognize texts in the scene image. Here, we introduced a novel mechanism to convert the spatial encoder from 1D to 2D, by expanding the standard transformer’s 1D Positional Encoding (1DE) to 2D Positional Encoding (2DE).

- **Input word images in natural scenes take different forms, including curved and skewed texts. If such input word images are transmitted left unchanged, the feature extraction step must learn an invariant representation for such geometry. To eliminate distortion on the input word images and make text recognition easier, we employ a thin-plate spline (TPS) transformation [4]. The rectified or normalized images enhance text recognition accuracy, particularly for datasets with a majority of arbitrary texts and perspective distorted texts. TPS can be selected or deselected using our framework.**
We propose a new mechanism called the Optimal Adaptive Threshold-based Self Attention (OATSA) that explicitly ignores the least contributive components in the attention matrix to limit the extraction of irrelevant elements and improve attention focus. OATSA approach can efficiently filter noisy information and generate a more accurate word.

We introduce a unified bi-directional decoder architecture integrated into modified Transformers that works in both right-to-left (R2L) and left-to-right (L2R) directions for STR to obtain more robust character sequences. The bidirectional decoding method, which uses two distinct decoders outperforms the stand-alone decoders. We accomplished this by executing bidirectional decoding at the architectural-based systems, rather than at the input-based systems, which were used in earlier research works.

The proposed framework is a non-recurrent network, trained in an end-to-end manner. This framework can be trained concurrently without employing any RNN modules. Although, the proposed framework achieves comparable or superior performance compared with most existing methods on both horizontal and arbitrary shaped scene text benchmarks for example 97.1% on ICDAR03, 98.0% on ICDAR13, 87.2% on ICDAR15, and 89.3% on CUTE80 datasets.

II. RELATED WORK

In the past decade, there has been a growing interest in natural STR in the computer vision community, which differs from classical handwritten character and text recognition in terms of characteristics and difficulties. The extensive studies can be found in [1], [2], and [3].

Traditional STR systems used text detectors to extract various candidate character positions and then used a character classifier to recognize the characters. These traditional methods depend on low-level features for STR including stroke width transform [17], connected components [18], Histogram of Oriented Gradients (HOG) descriptors [19] and so on. Wang et al. [20] used the HOG descriptors to train a character recognizer, which then used a sliding window to recognize characters in cropped word images. Seok and Kim [21] represent the target character set as an Implicit Shape Model (ISM) to achieve robustness on the character set. Hough forest is trained to localize and group the character candidates; the semi-Markov conditional random field (semi-CRF) framework is constructed for the recognition of text candidates. However, the low capability of hand-crafted features limits the performance of traditional text recognition systems. Yao et al. [22] introduced a new way of reliably identifying individual characters called Strokelets, providing a histogram feature for recognizing character components in natural scenes image.

Several researchers have started using deep learning models for STR as a result of the fast growth of neural networks. For a 90k-word classification, Jaderberg et al. [23] designed a CNN classifier composed of four convolutional layers and two fully connected layers used to retrieve high-level feature maps from the text image. This technique, however, was confined to a pre-defined lexicon. To get around this constraint, several authors have recently considered STR as a sequence translation task. Shi et al. [7] introduced a new unified deep neural network architecture; named Convolutional Recurrent Neural Network (CRNN) combines the functions of both CNN and Recurrent Neural Networks (RNN). CRNN can handle input images of various dimensions and generate predictions of different lengths. CRNN is capable of handling random strings (e.g. phone numbers), sentences, and other scripts such as Chinese words, and is not confined to recognizing words from a known dictionary. Since the spatial dependencies between local image patches in CNN are not explored and utilized, Shi et al. [5] specially designed a new deep neural network that has both convolutional and recurrent layers to transform the irregular word images into a more readable regular word image via STN. This network generates a sequence of feature vectors for any arbitrary size input word image. Finally, an attention-based sequence recognizer is used to generate a character sequence. Lin et al. [24] designed a special network by combining a sequential transformation network, used to rectify the irregular text by dividing the complex transformation step into multiple basic transformation steps and an attention-based character recognizer adopted to classify and capture character sequences.

Recent deep networks can develop robust representations that are tolerant to imaging distortions and changes in text style, but they still have issues handling scene texts including viewpoint and curvature distortions. To deal with such issues, Zhan and Lu [25] established an end-to-end STR network called ESIR, which reduces viewpoint distortion and text line curvature iteratively that improves the performance of STR systems. The posture of text lines in scenes is estimated using an innovative rectification network that introduces a different line fitting transformation. In addition, an iterative rectification mechanism is being created, which corrects scene text distortions in a fronto-parallel perspective. Litman et al. [26] presented a new encoder-decoder architecture named Selective Context ATtentional Text Recognizer (SCATTER) for predicting character sequences against complicated image backgrounds. A deep Bi-LSTM encoder is designed for encoding contextual dependencies. A two-step 1D attention method is used to decode the text.

Instead of rectifying the complete text image, Liu et al. [27] suggested using Character-Level Encoder (CLE) to identify and rectify specific characters in the word image. The arbitrary orientation network (AON) was developed by Cheng et al. [28] to directly capture the deep feature representations of irregular texts in four directions along with character location clues. A filter gate mechanism was designed to integrate the four-direction character sequences of features, and an attention-based decoder was employed to generate
character sequences. To tackle the “attention drift” problem, Cheng et al. [29] proposed a Focusing Attention Network (FAN) method comprised of two main components: an attention network (AN) and a focusing network (FN). The AN is designed to recognize character targets and FN is designed to evaluate and alter the attention by paying attention to the character target locations in the word images properly. A Multi-Branch Guided Attention Network (MBGAN) was proposed by Wang and Liu [30] to acquire invariant semantic representations of decoded character sequences, which can handle numerous irregularity aspects for irregular text recognition. Furthermore, the MBGAN contribute to the prevention of attention drift. Similarly, Bai et al. [6] introduced a technique called edit probability (EP) to efficiently deal with the issue of missing or unnecessary characters causing misalignment between the training texts and the resultant probability distribution character sequence. Most deep learning models heavily rely on visual data such as well-annotated images, it does not efficiently employ linguistic data such as texts. Zhang et al. [31] introduced a novel network called Pre-trained Multi-Modal Network (PMMN) that uses both visual and linguistic data for accurate STR.

In recent years, some research has been done to eliminate the recurrent architecture from the seq2seq learning algorithms, allowing for entirely parallel computation and faster processing speeds. Vaswani et al. [16] introduced a novel standard network architecture solely based on attention mechanisms, with no recurrence or convolutions called “Transformer” for machine translation tasks. Position-pair computation by RNNs enables the core self-attention module to obtain interdependence between distinct places in a sequence, leading to greater parallelization and minimal network complexity. Dong et al. [32] incorporated a Transformer architecture for the voice recognition problem. Similarly, Yu et al. [33] developed a network for a reading comprehension problem by integrating internal convolution layers with a universal self-attention module. Both these models were inspired by the Transformer framework. Dehghani et al. [34] recently expanded the Transformer architecture by developing a new model called the “Universal Transformer” to handle string copying and other rational interpretation with string lengths that are longer than those seen during training. There have also been numerous attempts to interpret scene text without the use of recurrent networks. Based on the Transformer model, Chen et al. [35] developed a new non-recurrent seq2seq framework for STR, which includes a self-attention block functioning as a fundamental component in both the encoder and decoder architecture, to understand character dependencies. Yang et al. [36] proposed an STR network that is much more simple and powerful, based on holistic representation-guided attention. An attention-based sequence decoder is linked directly to two-dimensional CNN features. The holistic depiction may steer the attention-based decoder to more precisely concentrate on text regions. Because of their inherent model architecture, all of these existing approaches are mostly focused on regular STR and these are found hard to recognize irregular text.

In contrast with the convolution network and attention mechanism, we propose a simple but powerful STR model with an Optimal Adaptive Threshold-based Self-Attention (OATSA) mechanism in this paper. This method directly maps word images into character sequences and it also works well on both horizontal and arbitrary shaped scene text images.

III. PROPOSED SYSTEM

We propose a modified Transformer-based architecture to recognize arbitrary shaped text from the natural scene image. Fig. 1 illustrates the overall pipeline of the proposed framework. The modified transformer can be categorized into four main modules: Image transformation, VFE, Encoder and Decoder. Both the encoder and decoder utilize a multi-layer stack of transformers. The encoder module is designed to obtain a high-level feature representation from a scene text image. The decoder block is designed to generate the sequence of characters from the feature maps while paying attention to the encoder output. Transformers are attention-based deep-learning architectures that use a self-attention module to scan through each constituent of a sequence and post updates by accumulating information from the entire sequence. An attention mechanism greatly helps the transformers to capture the global dependencies among the input and output sequences, where previous deep-learning approaches find it challenging to capture such relationships.

A. IMAGE TRANSFORMATION

Text recognizers are most effective when their input images include tightly confined regular text. This encourages us to do a spatial transformation before a recognition to convert input images into ones that recognizers can read easily. We employed a TPS transformation to transform an input text image (I) into a normalized image (I’) as shown in Fig. 2. Text images come in various shapes for example: tilted, perspective and curved texts. Such complex-shaped text images force the feature extraction steps to learn an invariant representation concerning such geometry. TPS rectification algorithm is an alternative to the STN, which has been used for various aspect ratios of text lines to reduce this complexity. TPS interpolates between a collection of fiducial points using a smooth spline. TPS identifies several fiducial points at the upper and lower enveloping points, and then normalizes the character region to a predefined rectangle.

B. VISUAL FEATURE EXTRACTION (VFE) AND 2D POSITIONAL ENCODING (2DPE)

According to recent research, ResNet [37] is gaining a lot of traction in the research community because of its unique features of dealing with the overfitting and vanishing gradient issue, parameter efficiency, capturing well-defined feature representations and introducing
FIGURE 1. Illustrate the overall pipeline of our proposed framework.

FIGURE 2. Visualization of image normalization step a) Input image (I) b) Set of fiducial points are predicted on the input text image (I) represented by red markers c) Normalized image (I').

“identity shortcut connection”. Therefore, we use ResNet as the CNN backbone for the VFE.

Table 1 illustrates the 50-layer residual network configuration as used in [10] we utilize for our feature extraction stage. CNN process the rectified image (I') to extract compact feature representation $F \in \mathbb{R}^{W \times H \times c}$, where $W$, $H$ and $C$ represent the rectified image’s width, height and the number of channels respectively. To reduce the encoding stage’s computational cost, we use $1 \times 1$ convolution to reduce the feature map channels represented by $F' \in \mathbb{R}^{W \times H \times d}$, where $d < c$.

Conventional Transformers lack recurrence and convolution layers, so it is important to provide absolute or relative location information of the characters in the word or sentence to the model which utilizes the sequence’s order for processing. To this end, we use 2D positional encoding maps $P \in \mathbb{R}^{W \times H \times d}$ as in [16] to capture the spatial information. The input is a tensor $F'$ of dimension $H \times W \times d$, and an attention score is used to correlate a query and a key at each position in the input $F'$, resulting in the attention score tensor $S$ with dimension $H \times W \times d'$. The multi-head self-attention layer can better capture 2D spatial information using the positional encoded map $F'$. By introducing a fixed 2D positional encoding $P(.)$ as given in Eq. (1) – Eq. (4), we generalize the original transformer’s 1D encoding to be suitable for the 2D image feature.

$$PE(\text{hor, ver}, 2i) = \sin(\text{pos(} \text{hor} \text{)} . \text{fre}_i)$$ (1)

$$PE(\text{hor, ver}, 2i + 1) = \cos(\text{pos(} \text{hor} \text{)} . \text{fre}_i)$$ (2)

$$PE(\text{hor, ver}, 2j + \text{ch}/2) = \sin(\text{pos(} \text{ver} \text{)} . \text{fre}_j)$$ (3)

$$PE(\text{hor, ver}, 2j + 1 + \text{ch}/2) = \cos(\text{pos(} \text{ver} \text{)} . \text{fre}_j)$$ (4)

where pos(hor) and pos(ver) represent the horizontal and vertical positions respectively, $\text{fre}_i, \text{fre}_j \in \mathbb{R}$ is the 2D positional encoding signal’s learnable frequencies, $c$ represents the number of channels in $F'$ and $i, j \in [0, d/4]$. Position code (P) and 2D PE ($F'$) are combined to enable to notice character’s information in each position before it. The 2D encoding maps P added to $F'$ represented by $F'' = F' + P$. The Transformer’s encoder only takes a set of vectors as input. Hence, it is necessary to vectorize the d channels of $F''$ and stacked together to create a single feature matrix $M$ as shown in Eq. (5). The function Mat2Vec is used to convert the feature map into a vector represented by $x_{i,j} = \text{Mat2Vec}(F''(:, :, j)) \in 1 \times HW$.

$$M = \begin{bmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,d} \end{bmatrix} \in \mathbb{R}^{d \times WH}$$ (5)

| Network Layer | Configuration | Output feature map dimension |
|---------------|---------------|-----------------------------|
| Input Layer   | Combined feature | $64 \times 16 \times 130$ |
| Block1        | Channel: 256  Kernel: $3 \times 3$ | $64 \times 16 \times 256$ |
| Conv1         | Channel: 256  Kernel: $3 \times 3$ | $64 \times 16 \times 256$ |
| MaxPool       | Kernel: $2 \times 2$ Padding: $1 \times 0$ Stride: $1 \times 2$ | $65 \times 8 \times 256$ |
| Block2        | Channel: 512  Kernel: $3 \times 3$ | $65 \times 8 \times 512$ |
| Conv2         | Channel: 512  Kernel: $3 \times 3$ | $65 \times 8 \times 512$ |
| Block3        | Channel: 512  Kernel: $3 \times 3$ | $65 \times 8 \times 512$ |
| Conv3         | Channel: 512 Stride: $1 \times 2$ Padding: $0 \times 0$ Kernel: $2 \times 2$ | $65 \times 4 \times 512$ |
| Conv4         | Channel: 512 Stride: $1 \times 1$ Padding: $0 \times 0$ Kernel: $2 \times 2$ | $65 \times 3 \times 512$ |
| AvgPool       | Kernel: $1 \times 3$ Padding: $1 \times 0$ Stride: $1 \times 2$ | $65 \times 1 \times 512$ |
C. ENCODER

The encoder is composed of two important layers: Multi-Head Self-Attention (MHSA) and Feed-Forward Network Layers (FFNL). An encoder captures a set of vectors ($M$) as its input and processes these vectors by giving input to the self-attention layer and FFN layer in order. Then the output from the FFN layer is sent to the next encoder. Around each of the two layers, we use a residual connection and proceed with layer normalization as shown in Fig. 3 i.e LayerNorm($x + \text{Sublayers}(x)$), where the function Sublayers($x$) is implemented by the sublayers themselves. All sublayers in the model, as well as the embedding layers, generate results with dimension 512 to facilitate these residual connections.

1) MULTIHEAD SELF ATTENTION (MHSA)

In a transformer-based STR network, the Multi-Head Self Attention (MHSA) mechanism is mainly used to manage size disparities in the text instances, the vector $M$ is linearly transformed into three matrices: Query ($Q$), Key ($K$), and Value ($V$). The dot product form is used for familiarity to assist the translation of vector calculation into matrix calculation. The weighted sum of the values matrix is output by the attention mechanism, which computes the weights of the values matrix $V \in \mathbb{R}^{W \times d}$ (see Eq. (9)) using the appropriate key matrix $K \in \mathbb{R}^{W \times d}$ (see Eq. (8)) and query matrix $Q \in \mathbb{R}^{W \times d}$ (see Eq. (7)). The queries, keys, and values for the self-attention modules are all generated in the same sequence. The attention output matrix is derived as follows (see Eq. (6)):

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d_k}} \right) V$$

$$Q = [q_1, q_2, \ldots, q_n]^T, \quad q_i = W_q x_i + b_q \quad (7)$$

$$K = [k_1, k_2, \ldots, k_n]^T, \quad k_i = W_k x_i + b_k \quad (8)$$

$$V = [v_1, v_2, \ldots, v_n]^T, \quad v_i = W_v x_i + b_v \quad (9)$$

where $b$, $W_q$, $W_k$, and $W_v$ are the bias, weight matrices of Query, Key and Value respectively. The scaling factor $\frac{1}{\sqrt{d_k}}$ represent the dimensions of queries and keys, the scaling factor is used to prevent a very small gradient of the softmax function.

The multi-head attention method involves projecting the queries, keys, and values ‘n’ times with various learnable projection weights, allowing the model to collect useful data from several representation subsections at the same time. In Transformer, both dot-product attention and MHSA are effective in practice, with multi-head attention being a concatenation of dot-product attention as given in Eq. (10) and Eq. (11). Multiple heads of attention are employed in each transformer layer. Furthermore, a multi-head-attention mechanism has been introduced to the attention layer to improve the model’s expressive capability. Before calculating attention, all $Q$, $K$, and $V$ are partitioned into numerous heads and run through distinct, learnt linear projections. After the attention calculation, multiple heads will be joined together.

$$\text{MHAttention}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_n)W^O$$

$$\text{Where head}_i = \text{Attention}(QW^Q_i, KW^K_i, VW^V_i)$$

Using multiple attention heads has the advantage of allowing the model to learn to focus on various parts of the input image for each attention head at different phases of the encoding process and it provides several “representation subspaces” to the attention layer.

2) OPTIMAL ADAPTIVE THRESHOLD-BASED SELF-ATTENTION (OATSA)

The self-attention operation in the classic Transformer architecture, on the other hand, has an evident flaw in that it distributes credits to all context components. It’s inappropriate since a lot of credit could be given to information that’s not relevant and should be discarded. For example, the traditional self-attention method calculates attention weights by multiplying the specified query by the key from several modalities. The weighted sum is then calculated by applying the attention matrix to the value. However, many irrelevant words may have a minimal association with encoded image attributes, leading to a very modest amount after multiplying the provided query by key. When attention scores are relatively close, a SOTA approach namely constraint local attention cannot filter irrelevant information and will break the long-term dependency.

In our proposed work, we integrate a new threshold module namely Optimal Adaptive Threshold-based Self Attention (OATSA) in a standard self-attention calculation as shown
in Fig. 4(b) to discard irrelevant information and preserve the long-term dependencies by replacing Standard scaled dot-product attention as shown in Fig. 4(a). We integrate our OATSA module between scale and softmax function to converge attention. Since the softmax function is dominated by the elements with the higher numerical value, the OATSA module chooses the elements with higher numerical values and discards the elements with lower numerical values as shown in Fig. 5. In the first step, we perform the dot product operation between query and key to derive the attention matrix P. The elements of attention matrix P are represented by \{p_{i1}, p_{i2}, ..., p_{im}\}. The elements having higher values in attention matrix P are assumed to be the most contributive element. To aggregate focus, we choose the most contributive elements from each row in the attention matrix P. In the next step, we divide the attention matrix P into 'n' chunks and calculate the mean value for each chunk. The elements lower than the threshold multiplicative factor (mean(p_{iw}) \times t) are assigned to negative infinity (see Eq. (11)). Based on the hypotheses, the elements with negative infinity do not contain any relevant information, the elements having higher numerical values contain close relevance information. Finally, negative infinity values are replaced by zero by applying the softmax function (see Eq. (12)) on the attention matrix (OP). The working procedure of the OATSA module is given in the Table 2. The proposed Optimal Adaptive Threshold-based Self-Attention (OATSA) module remarkably eliminates noisy information.

3) FEED FORWARD NETWORK LAYERS (FFNL)

To make the original transformer more robust in preserving the features produced by the encoder's multi-head self-attention mechanism, we employed an improved version of an FFN layer. The main objective of FFN is to convert the attention vectors into a format that the following encoder or decoder layer can understand. The modified FFN is made up of two layers of 1 \times 1 convolution with a ReLU activation function, followed by a residual connection. Attention vectors are taken “one at a time” by the Feed Forward Network. The finest part is that, unlike with RNN, each of these attention vectors is independent of the others. As a result, parallelization may be used here, which makes a huge impact. We can now feed all of the words into the encoder block at the same time and obtain the set of encoded vectors for each word at the same time.

D. DECODER

The decoder is made up of n layers of transformer decoders. The embeddings of the decoded output sequence of characters are sent into the decoder. The decoder is made up of N identical layers, each of which contains three sub-layers. The MHSA is used in the first layer of the network; the masked mechanism in the first layer prevents the model from seeing future data. This masking approach ensures that the model only utilizes the previous words to generate the current word. An MHSA layer without the masked technique makes up the second layer. It applies a multi-head self-attention mechanism over the first layer’s result. This layer serves as the foundation for correlating text and image information with the self-attention layer. A position-wise fully connected FFN is incorporated in the third layer. Following layer normalization, the Transformer establishes a residual connection for all three layers. To convert the Transformer’s result into probabilities for each character sequence in a sentence, we attach a Fully Connected (FC) layer and a softmax layer at the top. All the characters in the phrase can be created simultaneously, unlike the LSTM. The top encoder’s output is then converted into a set of K and V attention vectors. These K and V attention vectors are utilized by each decoder in its “encoder-decoder attention” layer, which assists the decoder in focusing on the appropriate positions in the input sequence.

1) BI-DIRECTION EMBEDDING

The conventional Seq2seq model’s decoder preserves output only in one direction, leaving the other direction uncaptured.
TABLE 2. Optimal Adaptive Threshold-based Self-Attention (OATSA) algorithm.

| Algorithm 1: Optimal Adaptive Threshold-based Self-Attention (OATSA) |
|---------------------------------------------------------------|
| **Input:** Key vector $K = [k_1, k_2, \ldots, k_n]$ and Query vector $Q = [q_1, q_2, \ldots, q_m]$. |
| **Output:** Attention matrix $AM = [e_{11}, e_{21}, e_{31}, \ldots, e_{mn}]$. |

1. **Import** key vector $K = [k_1, k_2, \ldots, k_n]$.
2. **Import** Query vector $Q = [q_1, q_2, \ldots, q_m]$.
3. **Transpose** Query vector $(Q)$.
4. **Perform** Dot Product between $Q^T$ and $K$.

$$
\begin{bmatrix}
q_1 \\
q_2 \\
\vdots \\
q_m
\end{bmatrix} \cdot 
\begin{bmatrix}
k_1 & k_2 & \cdots & k_n
\end{bmatrix} = 
\begin{bmatrix}
P_{11} & P_{12} & \cdots & P_{1n} \\
P_{21} & P_{22} & \cdots & P_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
P_{m1} & P_{m2} & \cdots & P_{mn}
\end{bmatrix}
$$

5. **Divide** the matrix $P_{ij}$ into ‘n’ chunks.

$$
C_a = \begin{bmatrix}
P_{11} & \cdots & P_{1a} \\
\vdots & \ddots & \vdots \\
P_{ma} & \cdots & P_{mn}
\end{bmatrix},
C_b = \begin{bmatrix}
P_{1b} & \cdots & P_{1d} \\
\vdots & \ddots & \vdots \\
P_{mb} & \cdots & P_{mn}
\end{bmatrix}, \ldots, C_m = \begin{bmatrix}
P_{1f} & \cdots & P_{1z} \\
\vdots & \ddots & \vdots \\
P_{mf} & \cdots & P_{mn}
\end{bmatrix}
$$

6. **Compute** the mean value for each chunk ($C_a, C_b, \ldots, C_m$).

7. **Perform** the following condition (see Eq. (11)): Compare the elements of each chunk with an optimal threshold value ($t$), window size ($w$), if $p_{ij}$ is smaller than the multiplicative factor then assign elements by $-\infty$.

$$
P_i = \begin{cases}
    p_{ij}, & \text{if } p_{ij} \geq \frac{\sum_{j=1}^{w} \sum_{i=1}^{w} p_{ij}}{w} + t \\
    -\infty, & \text{if } p_{ij} < \frac{\sum_{j=1}^{w} \sum_{i=1}^{w} p_{ij}}{w} + t
\end{cases}
$$

8. **Apply** softmax function on updated attention matrix ($P_i$), replace $-\infty$ to 0.

$$
a_{ij} = \frac{e^{p_{ij}}}{\sum e^{p_{ij}}} \quad \text{(12)}
$$

Attention Matrix = Softmax($P_i$)

For instance, in certain fonts, a decoder that recognizes character sequence from L2R may have trouble choosing the initial letter between upper-case ‘I’ and lower-case ‘l’. These initial characters are difficult to differentiate perceptible and the decoder has no memory of the previous deciphered characters. Such challenging characters can be recognized using an R2L decoder easily since the succeeding characters suggest the initial character based on the language.
datasets and three irregular datasets. The datasets descriptions challenge the STR benchmark datasets, including four regular scenes. Numerous experiments were performed on the output sequence processing order by adding the direction embedding. Similar to position embedding, this direction embedding also gives additional context information to the framework.

The directional embedding enables the network to decipher the text information not only from the L2R direction but also from the R2L direction. If the decoding is performed in one direction (L2R) then the character loss for L2R deciphered images cannot be decreased. We considered the output sequence decoding direction as two subtasks. Each character sequence decoding (L2R direction and R2L direction) are sub-tasks of the conditional output sequence algorithm. To channel the result in a correct decoding direction, we create two separate 512-d vectors at the beginning of training. Each scene text image in the set is decoded two times during each training iteration step, first from the L2R direction and next from the R2L direction. The reserved ground truth of the original description is the ground truth description for the R2L deciphered character sequence. The decoder achieves strong performance by combining the outputs of the two directions, which also helps the classifier to predict the right character. The total loss is the sum of the losses suffered by both the L2R and R2L directions given in Eq. (13).

\[
\text{Loss}_{\text{total}} = -\frac{1}{2} \sum_k (\log P_{\text{L2R}}(y_k | I) + \log P_{\text{R2L}}(y_k | I)) \quad (13)
\]

where \(y_k\), \(P_{\text{L2R}}\), \(P_{\text{R2L}}\) and I represent the ground truth of the \(k\)th character, predicted result on left to the right direction, predicted result on R2L direction and the input image, respectively. We also include a supervisory branch that projects each visual component from the ‘c’ dimension to the number of alphabet classes in order to predict the character it belongs to, as well as calculate the cross-entropy loss between ground truth and prediction.

IV. EXPERIMENT

In this section, we exhaustively evaluate the performance of our model. Numerous experiments were performed on challenging STR benchmark datasets, including four regular datasets and three irregular datasets. The datasets descriptions are as follows: The results from the experiment show that our proposed framework comprehensively outperforms current SOTA methods.

A. DATASETS

In this paper, we train our framework with only two synthetic datasets: Synth90k [45] and SynthText [46]. We evaluated the significance and robustness of our proposed STR framework on seven standard benchmark datasets, four regular scene text datasets and three irregular scene text datasets.

Synth90k is the synthetic text dataset proposed by Gupta et al. [45]. A total of 9 million word pictures were acquired from a collection of 9k frequent English language. The entire dataset images were only used for training. Each image in Synth90k has a word-level ground-truth annotation. These images were generated with the help of a synthetic text engine and are quite realistic.

Synthtext is another synthetic text dataset that was only used for training, which is proposed by Jaderberg et al. [46]. The process of image generation was similar to that of [45]. Originally, the Synthext dataset was created for text detection, unlike [45]. Characters are rendered as a full-size images.

IIIT5K-Words [38] (IIIT5K) there are 3k cropped word test images in the IIIT5K dataset collected from the internet. Each image has a vocabulary of 50 short words and 1,000 long words. A few words were created randomly and the rest were created from the dictionary.

Street View Text [39] (SVT) dataset comprises of 647 cropped text pictures acquired from Google Street View (GSV). Each image contains a 50-word lexicon. Most of the images in the SVT dataset are severely distorted, noisy, blurred and low resolution.

ICDAR 2003 [40] (IC03) consists of 251 scene text images with text-labelled bounding boxes. For a fair comparison, we excluded word images containing non-alphanumeric characters or images with less than 3 characters as suggested by Wang et al. [20]. The updated dataset comprises of 867 cropped word pictures. Images in the IC03 dataset include both 50-word lexicon and “full-lexicon”.

ICDAR 2013 [41] Most of the (IC13) dataset image samples are inherited from its successor IC03. For a fair comparison, words with non-alphanumeric characters were removed from the dataset. The filtered test dataset contains 1015 cropped word images with no lexicon associated with them.

ICDAR 2015 [42] (IC15) dataset consists of 6545 cropped text images, 4468 images used for training and 2077 images used for testing. No lexicon is associated with it. Most of the text in the word images in this dataset are irregular shapes such as horizontal, oriented and curved. IC15 dataset images are captured by Google Glasses without proper positioning and focusing.

SVT-Perspective [43] (SVT-P) dataset consists of 645 cropped word pictures. Images in the SVT-P dataset include both a 50-word lexicon and a “full-lexicon”. SVT-P dataset images are collected from GSV, most of the
images captured at a side-view angle. Therefore, the images in the SVT dataset are heavily distorted, noisy, blurred and low resolution.

CUTE80 [44] (CUTE) dataset contains a collection of 80 high-resolution images taken in naturalistic environments. It contains 288 cropped word pictures for testing. CUTE is the most challenging dataset since most of the word images consist of arbitrarily shaped letters. No lexicon is associated with this dataset. It was collected with the intent of evaluating the performance of irregular STR.

B. IMPLEMENTATION DETAILS
The specifications of ResNet-based CNN network architecture for robust text feature extraction are shown in Table 1. We implemented our method using the Pytorch framework. A single NVIDIA GTX-1080Ti GPU with 12 GB memory was used to conduct all the experiments. We constructed our model completely from scratch using synthetic images of SynthText and Synth90k. The training data consists of 14 synthetic images, 6 million from Gupta et al. [45] and 8 million from Jaderberg et al. [23] and. There is no additional data utilized. In our tests, we don’t employ any geometric-level or pixel-level labelling. The model is trained using just synthetic text, with no fine-tuning for each dataset. The AdaDelta optimization strategy is used to train our model with a batch size of 64. The learning rate is initially set to 1.0, as suggested by Shi et al. [4], learning rate decays to 0.1 at step 0.6 M and 0.01 at step 0.8 M. The optimizer AdaDelta’s learning rate is adjustable, we discover that slower learning rates lead to greater convergence. The proposed model can classify 37 classes, including 26 alphabets (a-z), 10 digits (0-9) and a special symbol representing “<EOS>”.

C. ABLATION STUDIES
For the image encoding and the feature extraction, we experimented with various CNN models (see Table 3) such as VGG16, ResNet18, ResNet34, ResNet50 and ResNet164. In that, ResNet50 provides an excellent balance between model size and accuracy and it captures well-defined feature representations. Hence, we choose ResNet50 as our CNN’s backbone.

After numerous experiments, the number of attention heads in the encoder and decoder is kept at 16. Similarly, the number of decoder blocks is dynamically changed during each iteration and finally set to 3. The results suggest that \( N = 3 \) yields the best outcomes for our model as shown in Table 4. This effect contradicts the Transformer’s experimental results, which suggest that utilizing additional blocks improves language translation and irregular text recognition performance. Increasing the number of heads leads to an overfitting problem.

Self-attention is critical in many seq2seq activities such as chatbots, language translation and so on because it is capable of capturing long-term relationships. We explored the effect of improved self-attention block in our proposed framework for regular and irregular STR. To improve the depiction of relationships between distant image patches, initially, we built a self-attention layer on top of the convolutional layers. On the other hand, we drop the self-attention layer from the decoder to analyze the influence of the self-attention layer on the decoder side. The recognition accuracy of the generalized model is marginally lower than the standard system (91.6% v.s. 97.7% on the IIIT5K dataset and 83.3% v.s. 90.6% on the SVT-P dataset) as shown in Table 5, but it is still comparable to earlier approaches.

In contrast to language translation approaches, we identified that applying the self-attention mechanism in STR has a significant impact on performance. We believe there are three alternative reasons: Firstly, the length of character sequences represented in standard STR tasks is often less than those required for machine translation. Secondly, the CNN-based encoder effectively represents the long-range relationships between the words. For example, the receptive field produced by ResNet50’s final feature layer has a great influence on long-term dependencies. Finally, self-attention is often employed in machine translation to represent the relationships between words in a phrase or even a paragraph.

### TABLE 3. The performance comparison of different modified CNN backbones. Modified ResNet50 captures well-defined feature representations and it provides an excellent balance between model size and accuracy.

| Modified CNN Backbone | Accuracy | IIIT5K | IC13 | IC15 | CUTE |
|-----------------------|----------|--------|------|------|------|
| VGG16                 |          | 92.1   | 90.4 | 76.3 | 75.6 |
| ResNet18              |          | 93.5   | 94.6 | 82.3 | 84.5 |
| ResNet34              |          | 94.0   | 96.7 | 85.8 | 86.3 |
| ResNet50              |          | 97.3   | 98.0 | 88.2 | 91.3 |
| ResNet164             |          | 94.1   | 97.3 | 87.1 | 89.9 |

### TABLE 4. The performance comparison of adding or removing decoder blocks and attention head counts from the proposed framework. It can be seen that increasing the number of heads decreases the performance marginally due to overfitting (with \( N=4, H=16 \)).

| Number of Decoder Block | Number of Head | Accuracy | IIIT5K | IC13 | IC15 | CUTE |
|-------------------------|---------------|----------|--------|------|------|------|
| 1                       | 8             |          | 93.7   | 96.3 | 85.9 | 86.2 |
| 1                       | 16            |          | 93.9   | 96.8 | 86.0 | 87.1 |
| 2                       | 16            |          | 94.0   | 97.2 | 86.5 | 87.4 |
| 3                       | 16            |          | 96.2   | 97.9 | 87.2 | 88.6 |
| 3                       | 8             |          | 97.3   | 98.0 | 88.2 | 91.3 |
| 4                       | 16            |          | 96.4   | 97.3 | 87.0 | 88.8 |
TABLE 5. The encoder and decoder performance comparison with and without self-attention block. When comparing row 1 and row 2, we find that dropping the self-attention block from the decoder side of our framework produces a significant performance drop. Rows 2 and 3 illustrate that the self-attention block to the encoder side has shown slight improvement.

| Encoder Self-attention | Decoder Self-attention | Accuracy |
|------------------------|------------------------|----------|
| ✓                      | ×                      | IIT5K: 91.6  | SVT: 89.2  | IC03: 94.3  | IC13: 95.1  | IC15: 82.6  | SVT-P: 83.3  | CUTE: 81.8 |
| ×                      | ✓                      | IIT5K: 95.4  | SVT: 93.8  | IC03: 96.7  | IC13: 97.4  | IC15: 86.6  | SVT-P: 85.9  | CUTE: 88.4 |
| ✓                      | ✓                      | IIT5K: 97.7  | SVT: 96.6  | IC03: 97.1  | IC13: 98.0  | IC15: 88.2  | SVT-P: 90.6  | CUTE: 91.3 |

TABLE 6. Experiment with various decoders. "Normal" denotes an L2R direction, "Reversed" signifies an R2L direction and "Bidirectional" denotes a combination of them.

| Method       | Accuracy |
|--------------|----------|
|              | IIT5K    | SVT      | IC03     | IC13     | IC15     | SVT-P    | CUTE     |
| Normal       | 93.7     | 91.8     | 96.6     | 97.8     | **87.2** | 85.1     | 88.6     |
| Reversed     | 95.1     | 91.5     | 96.8     | 97.5     | 87.0     | 85.8     | 88.3     |
| Bidirectional| **97.7** | 96.6     | 97.1     | **98.0** | 88.2     | **90.6** | 91.3     |

TABLE 7. The performance of the proposed method with and without the image transformation technique.

| Method                      | Accuracy |
|-----------------------------|----------|
|                            | IIT5K    | SVT      | IC03     | IC13     | IC15     | SVT-P    | CUTE     |
| Without Image Transformation| 96.6     | 94.0     | 96.9     | 97.0     | 86.5     | 86.1     | 88.8     |
| With Image Transformation   | **97.7** | 96.6     | 97.1     | **98.0** | **88.2** | **90.6** | **91.3** |

Between words that are far apart, there are nevertheless wealthy syntactic and semantic relationships. In contrast, each input image in STR generally comprises a single word, a self-attention module is mainly employed to represent the character relationships of an input text. The ties that bind the relationship between the letters in a word are usually weaker than those that bind the relationship between the words of a sentence. This might clarify why self-attention does not assist to enhance irregular text recognition performance. We analyze the recognition accuracies of several decoders to assess the efficacy of the bidirectional decoder: The normal decoder only understands text that is read from the L2R direction. Reversed recognizes text only in the R2L direction; the Bidirectional decoder works in both L2R and R2L directions and chooses the one with the greatest recognition accuracy. In many cases, Normal and Reversed decoders produce equivalent accuracies as shown in Table 6. Normal surpasses reversed on SVT, IC15 and CUTE while reversed excels IIT5K, IC03 and SVT-P. In the worst case, the difference in recognition accuracy between Normal and Reversed decoders is minimal, when they are combined, they provide a significant performance improvement. We carried out experiments to see how text rectification impacted our framework. As a text rectification approach, we employed the image normalization technique proposed in the STR method. Without an image normalization block, the proposed Transformer based 2D-attention mechanism can locate a single character scattered in 2D space. In this context, the image normalization block has minimal effect on our framework (see Table 7). We provide a novel technique called Optimal Adaptive Threshold-based Self-Attention (OATSA) that effectively explicit sparse Transformer. The performance and importance of the OATSA technique are shown in Table 8. The OATSA technique preserves the long-term dependencies, which are defined by the distribution of neighbour nodes. It can focus the attention of the standard Transformer on the most contributive components. Optimal Adaptive Threshold is integrated into self-attention and performs as an attention mechanism in the decoder, allowing the model to produce more accurate words.
TABLE 8. Performance comparison among four variations. ResNet50 is used as the proposed model’s feature extraction module. Rectification and no rectification indicate the image normalization step was performed and the image normalization step not performed. 2D positional encoding represents the model that performs recognition tasks; it keeps tracking the character position in each iteration. The OATSA represent the Optimal Adaptive Threshold-based Self-Attention Algorithm. Different decoders are "Normal" denotes an L2R direction, "Reversed" signifies an R2L direction and "Bidirectional" denotes a combination of them.

| Exp | Backbone (ResNet) | Image Transformation | 2D Positional Encoding | OATSA Model | Decoding Direction | HIT5K | SVT | IC03 | IC13 | IC15 | SVT-P | CUTE |
|-----|------------------|----------------------|-----------------------|-------------|--------------------|-------|-----|------|------|------|-------|------|
| (a) | ✓                | ✓                    | ✓                     | ×           | Normal             | 80.6  | 75.7| 82.3 | 72.8 | 70.4 | 79.3  | 71.0 |
| (b) | ✓                | ×                    | ✓                     | ×           | Normal             | 79.4  | 74.6| 80.2 | 69.4 | 68.3 | 77.4  | 74.9 |
| (c) | ✓                | ✓                    | ✓                     | ×           | Reversed           | 82.5  | 81.2| 84.8 | 83.6 | 75.7 | 83.8  | 80.2 |
| (d) | ✓                | ×                    | ✓                     | ×           | Reversed           | 81.6  | 80.4| 84.3 | 85.3 | 75.1 | 74.8  | 76.2 |
| (e) | ✓                | ✓                    | ✓                     | ✓           | Normal             | 93.7  | 91.8| 96.6 | 97.8 | 87.2 | 85.1  | 86.6 |
| (f) | ✓                | ✓                    | ✓                     | ✓           | Reversed           | 94.1  | 91.5| 96.8 | 97.5 | 87.0 | 85.8  | 85.3 |
| (g) | ✓                | ✓                    | ✓                     | ✓           | Bi-direction        | 97.7  | 96.6| 97.1 | 98.0 | 88.2 | 90.6  | 91.3 |

FIGURE 6. The attention heat maps provide a visual representation of the 2D attention weights obtained from all of the decoding stages on a standard benchmark.

After extensive comparative experiments, the optimal value of w and t is 4 and 0.67, respectively. Fig. 7 shows the visual representation of the 2D attention weights obtained from all of the decoding stages on a standard benchmark.

D. COMPARISONS WITH EXISTING METHODS

In this section, we compare the effectiveness and robustness of the proposed method by setting the number of decoder blocks (N) to 3, the number of heads (H) to 16 and d = 1024 in comparison with the current SOTA methods on a variety of regular and irregular text benchmarks datasets. To be fair, we merely list all the performance results in lexicon-free mode. Most of these existing approaches are trained on the same datasets. The proposed Transformer based scene recognition model was compared with 21 existing methods on both regular and irregular datasets and the results are shown in Table 9. The proposed framework comfortably outperforms the current SOTA approach of Yang et al. [36] and Lu et al. [48] on standard datasets such as SVT, IC13, SVT-P, CUTE and IC15. The proposed method produces better recognition accuracy on regular text datasets such as IC03, IC13 and IC15. We do not compare the results of Liao et al. [49] since they employed an extra word image with character-level annotations for model training. Note that Litman et al. [26] and Lu et al. [48] use an additional image dataset SynthAdd [47] for training and produced a better recognition accuracy of results of 86.9% and 84.5% on SVT-P and 87.5% on CUTE80. Zhang et al. [31] use the Wiki dataset for additional training to achieve the top performance on CUTE80. Still, the proposed method outperforms Litman et al. [26], Lu et al. [48] and Zhang et al. [31] on IC03, IC13, and IC15. The recognition accuracy of the proposed method on all datasets substantially surpasses that of linguistic-based approaches, notably on irregular texts (leading by +3.4%...
TABLE 9. On several benchmarks, the overall performance of our STR model is compared with that of the previous state of art approaches. All values are expressed as a percentage (%). The outcomes are all in the no lexicon represented by “None”, “90K”, “ST”, “SA” and “Wiki” stand for Synth90K, SynthText, SynthAdd and Wikitext-103, respectively; “word” and “char” denote the use of word-level or character-level annotations; and “self” denotes the use of a self-designed convolution network or self-made synthetic datasets.

| Method         | ConvNet | Annos | Training dataset | IHT5K | SVT | IC03 | IC13 | IC15 | SVT-P | CUTE |
|----------------|---------|-------|------------------|-------|-----|------|------|------|-------|-----|
|                |         |       |                  | None  | None| None | None | None | None  | None |    |
| Shi et al. [4] | ResNet  | Word  | 90K+ST           | 93.4  | 93.6| 94.5 | 91.8 | 76.1 | 78.5  | 79.5 |
| Shi et al. [5] | CNN     | Word  | 90K+ST           | 81.9  | 81.9| 90.1 | 88.6 | -    | 71.8  | 59.2 |
| Bai et al. [6] | ResNet  | Word  | 90K+ST           | 88.3  | 87.5| 94.6 | 94.4 | 73.9 | -     | -    |
| Gao et al. [8] | Self    | Word  | ST               | 81.8  | 82.7| 89.2 | 88.0 | -    | -     | -    |
| Liao et al. [12]| ResNet | Word  | 90K+ST           | 93.9  | 90.6| -    | 95.3 | 77.3 | 90.6  | 87.8 |
| Yu et al. [14] | ResNet  | Word  | 90K+ST           | 94.8  | 91.5| -    | 95.5 | 82.7 | 85.1  | 87.8 |
| Luo et al. [15] | Self   | Word  | 90K+ST           | 91.2  | 88.3| 97.8 | 92.4 | 68.8 | 76.1  | 77.4 |
| Lin et al. [24]| ResNet  | Word  | 90K+ST           | 94.1  | 90.6| 95.1 | 92.8 | 76.7 | 82.2  | 83.3 |
| Zhan et al. [25]| ResNet | Word  | 90K+ST           | 93.3  | 90.2| -    | 91.3 | 76.9 | 79.6  | 83.3 |
| Litman et al. [26] | ResNet | Word  | 90K+ST+SA       | 93.7  | 92.7| 96.3 | 93.9 | 82.2 | 86.9  | 87.5 |
| Liu et al. [27] | Self    | Word  | 90K+ST           | 83.6  | 84.4| 91.5 | 90.8 | -    | 73.5  | -    |
| Cheng et al. [28]| BCNN   | Word  | 90K+ST           | 87.0  | 82.8| 91.5 | 68.2 | 73.0 | 76.8  | -    |
| Cheng et al. [29]| ResNet | Word  | 90K+ST           | 87.4  | 85.9| 94.2 | 93.3 | 85.3 | -     | -    |
| Wang et al. [30]| ResNet | Word  | 90K+ST           | 93.2  | 89.2| 94.3 | 92.8 | 76.6 | 82.6  | 82.6 |
| Zhang et al. [31]| ResNet | Word  | 90K+ST−Wiki     | 96.2  | 93.9| -    | 97.7 | 85.5 | 89.0  | 91.9 |
| Chen et al. [35]| ResNet | Word  | 90K+ST           | 93.6  | 89.6| -    | 92.8 | 80.0 | 81.4  | 83.6 |
| Yang et al. [36]| ResNet | Word  | 90K+ST           | 94.7  | 88.9| -    | 93.2 | 74.0 | 80.9  | 85.4 |
| Lu et al. [48] | ResNet  | Word  | 90K+ST+SA       | 95.0  | 90.6| 96.4 | 95.3 | 79.4 | 84.5  | 87.5 |
| Liao et al. [49]| ResNet | Word  | ST               | 91.9  | 86.4| -    | 91.5 | -    | -     | 79.9 |
| Huang et al. [50]| ResNet| Word  | 90K+ST           | 94.0  | 88.9| 95.0 | 94.5 | 79.4 | 82.6  | 73.9 |
| Wu et al. [51] | ResNet  | Word  | 90K+ST           | 89.5  | 86.5| 94.1 | 90.1 | 76.6 | -     | 78.0 |
| Proposed Method| ResNet  | Word  | 90K+ST           | 97.7  | 96.6| 97.1 | 98.0 | 88.2 | 90.6  | 91.3 |

on IC15 and +2.8% on CUTE datasets). Our method outperforms the prior SOTA approach by Yang et al. [36] by a margin of +7.7% on SVT, +4.8% on IC13, +14.2% on IC15, +9.7% on SVT-P and +5.9% on CUTE. The significant improvement validates the effectiveness of our method. We are only 0.6% behind Zhang et al. [31] on CUTE.
In this paper, we presented a new simple yet powerful framework for both regular and irregular STR based on the transformer framework. The proposed framework breaks down into four modules: image transformation, feature extraction, encoder and decoder. First, the transformation module utilizes a Thin Plate Spline (TPS) transformation to normalize the irregular or arbitrary word image into a more readable word image that greatly helps to reduce the complexity of extracting text features. Second, the Visual Feature Extraction (VFE) module uses ResNet as the CNN backbone to extract well-defined feature representations and expand the standard transformer’s 1D Positional Encoding (1DPE) to 2D Positional Encoding (2DPE) information to capture the order of sequential information from the 2D rectified word image. Third, Multi-Head Self-Attention (MHSA) and Feed-Forward Network Layers (FFNL) in the encoder module perform feature aggregation and feature transformation concurrently. Finally, we proposed a new Optimal Adaptive Threshold-based Self-Attention (OATSA) model and an architectural level bi-directional decoding approach comprised in the decoder module greatly supports the framework to generate a more accurate character sequence. The OATSA model replaces the standard Scaled Dot-Product Attention. Moreover, it can be used in both encoder and decoder modules to filter noisy information effectively and choose the most contributive components to focus on image text regions. The proposed framework is trained with world-level annotations; it can handle words of any length in the lexicon-free mode. Comprehensive experimental results on challenging standard benchmarks (91.3% v.s. 91.9%). In addition, it is highlighted that the IIIT5k dataset has plenty of images with background noise. Still, the proposed approach achieves the highest recognition accuracy of 97.7%. Our model employs only word-level annotation, it outperforms the character-level model Liao et al. [49] on IIIT5K (97.7% v.s. 91.9%). Many samples in the IC15 dataset were not horizontally positioned, which is beyond the scope of the present study. As a result, we normalized the sample based on the image ratio. Most of the image samples in the SVT-perspective dataset were perspective and low-resolution. Although the proposed method performs as same as the model presented by Liao et al. [12], it did obtain SOTA performance in the majority of SVT-perspective scenarios. As shown in Fig. 8, our method outperforms Yang et al. [36] in the STR problem, the proposed framework is capable of recognizing blurred word images and irregular text. Our model learns both self-attention and input-output attention, where the encoder and decoder both preserve feature-feature and target-target interactions. This makes the intermediate representations more resistant to spatial distortion. Furthermore, our approach considerably reduces the issue of attention drifting.

V. CONCLUSION

In this paper, we presented a new simple yet powerful framework for both regular and irregular STR based on the transformer framework. The proposed framework breaks down

![FIGURE 7. Illustration of success and failure cases of the proposed method. “GT” stands for “Ground Truth,” “Pred” stands for “Prediction”. Blurry, low resolution (LR) and illumination are some of the reasons for failure.](image-url)
including IIIT5K-Words, Street View Text, CUTE80 and ICDAR datasets show that our methodology outperforms SOTA approaches.

REFERENCES

[1] Y. Zhu, C. Yao, and X. Bai, “Scene text detection and recognition: Recent advances and future trends,” Frontiers Comput. Sci., vol. 10, no. 1, pp. 19–36, 2016.

[2] Y. Ye and D. Doermann, “Text detection and recognition in imagery: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 7, pp. 1480–1500, Jul. 2015.

[3] S. Long, X. He, and C. Yao, “Scene text detection and recognition: The deep learning era,” Int. J. Comput. Vis., vol. 129, no. 1, pp. 1–26, 2018.

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

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

[6] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, “Edit probability for scene text recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 1508–1516.

[7] 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. 2016.

[8] Y. Gao, Y. Chen, J. Wang, M. Tang, and H. Lu, “Reading scene text with fully convolutional sequence modeling,” Neurocomputing, vol. 339, pp. 161–170, Apr. 2019.

[9] P. Selvam and J. A. S. Koilraj, “A deep learning framework for grocery product detection and recognition,” Food Anal. Methods, to be published, doi: 10.1007/s12261-022-02384-2.

[10] Y. Baek, S. Shin, J. Baek, S. Park, J. Lee, D. Nam, and H. Lee, “Character region attention for text spotting,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 504–521.

[11] S. Prabu and K. J. A. Sundar, “Enhanced attention-based encoder-decoder framework for text recognition,” Intell. Automat. Soft Comput., vol. 35, no. 2, pp. 2071–2086, 2023.

[12] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, “Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 2, pp. 532–548, Feb. 2021.

[13] M. Jaderberg, “Spatial transformer networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015, pp. 2017–2025.

[14] D. Yu, X. Li, C. Zhang, J. Han, J. Liu, and E. Ding, “Towards accurate scene text recognition with semantic reasoning networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 12110–12119.

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

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017, arXiv:1706.03762.

[17] C. Yao, X. Bai, and W. Liu, “A unified framework for multitiered text detection and recognition,” IEEE Trans. Image Process., vol. 23, no. 11, pp. 4737–4749, Nov. 2014.

[18] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “Object segmentation based on the integration of adaptive K-means and GrabCut algorithm,” in Proc. Int. Conf. Wireless Commun. Signal Process. (WISPNET), Mar. 2022, pp. 213–216, doi: 10.1109/WISPNET54241.2022.9767089.

[19] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, “End-to-end text recognition with convolutional neural networks,” in Proc. Int. Conf. Pattern Recognit. (ICPR), Nov. 2012, pp. 3304–3308.

[20] K. Wang, B. Babenko, and S. Belongie, “End-to-end scene text recognition,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Nov. 2011, pp. 1457–1464.

[21] J.-H. Seok and J. H. Kim, “Scene text recognition using a Hough forest implicit shape model and semi-Markov conditional random fields,” Pattern Recognit., vol. 48, no. 11, pp. 3584–3599, 2015.
ABOLFZAL MEHBOODIYAN (Senior Member, IEEE) received the Ph.D. degree from the INRS-EMT University of Quebec, Montreal, Canada, in 2010. He is currently an Associate Professor and the Head of the Department of Electronics and Communication Engineering (ECE), Kuwait College of Science and Technology. Before coming to KCST, he worked as a Marie Curie Senior Research Fellow at University College Dublin, Ireland. Prior to that, he worked as an Assistant Professor at Tohoku University, Japan, and as a Research Scientist at the Advanced Telecommunication Research (ATR) International, Kyoto, Japan. His research interests include communications engineering, the IoT and artificial intelligence in wireless networks, and real-world applications. He received numerous awards, including the JSPS Young Faculty Startup Grant, the KDDI Foundation Grant, the Japan Radio Communications Society (RCS) Active Researcher Award, the European Commission Marie Skodowska-Curie Fellowship, and the NSERC Visiting Fellowships in Canadian Government Laboratories. He is a Senior Member of IEICE.

JULIAN L. WEBBER (Senior Member, IEEE) received the Ph.D. degree from Bristol University, in 2004. Following postdoctoral research on wireless communications at Hokkaido University, in 2007. In 2012, he joined the Advanced Telecommunications Research Institute International, Kyoto. Since 2018, he has been a Visiting Researcher and a Research Assistant Professor at Osaka University. He is currently an Associate Professor and the Head of the Department of Electronics and Communication Engineering (ECE), Kuwait College of Science and Technology. His research interests include communications engineering, machine learning, signal and image processing, and emphasizing real-time implementation. He is a member of IEICE.

P. Selvam et al.: Transformer-Based Framework for Scene Text Recognition

MESHAL ALHARBI received the M.Sc. degree in computer science from Wayne State University, USA, in 2014, and the Ph.D. degree in computer science from Durham University, U.K., in 2020. He has ten years of experience in teaching/research/industry. He is currently an Assistant Professor of artificial intelligence with the Department of Computer Science, Prince Sattam Bin Abdulaziz University, Saudi Arabia. His research interests include artificial intelligence applications and algorithms, agent-based modeling and simulation applications, disaster/emergency management and resilience, optimization applications, and machine learning.