International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Gated Dual Adaptive Attention Mechanism with Semantic Reasoning, Character Awareness, and Visual-Semantic Ensemble Fusion Decoder for Text Recognition in Natural Scene Images

A. S. Venkata Praneel[*1], Dr. T. Srnivasa Rao[2]

Submitted: 20/08/2023 **Revised**: 11/10/2023 **Accepted**: 23/10/2023

Abstract*:* Text recognition in natural scene images poses a significant challenge due to variations in font styles, sizes, orientations, complex backgrounds, and lighting conditions. In this paper, a Gated Dual Adaptive Attention Mechanism (GDAAM), a novel framework that combines Mask Scoring Region-based Convolutional Neural Networks (MS-RCNN), Pyramid-based Text Proposal Networks (PBTPN), and Transformation Component Networks (TCN) as encoder, along with semantic reasoning, character awareness, and a visual-semantic ensemble fusion decoder for accurate text recognition in natural scene images is proposed. The encoder component of GDAAM leverages two robust architectures: MS-RCNN and PBTPN+TCN. MS-RCNN is utilised for its strong object detection capabilities, allowing for accurate localisation of text regions within the scene images. PBTPN+TCN captures temporal dependencies and contextual information in images containing text sequences. GDAAM extracts comprehensive features from spatial and temporal dimensions by combining these encoders, enabling effective representation of text elements. To facilitate fine-grained attention modelling, it incorporates the GDAAM in its decoder. It allows the model to selectively focus on relevant visual and textual cues, dynamically adapting its attention weights based on the input. GDAAM efficiently integrates visual and textual information by incorporating gate mechanisms enhancing text recognition accuracy in challenging natural scene images. Semantic reasoning is another crucial aspect integrated into GDAAM. A reasoning module incorporates contextual information, enabling the model to reason and make informed decisions. GDAAM selectively attends to relevant visual and textual cues, leveraging attention mechanisms, enhancing its understanding, and promoting more accurate text recognition. GDAAM addresses character awareness to handle complex text layouts, irregularities, and occlusions commonly found in natural scene images. This awareness further improves the model's ability to accurately recognise text in challenging visual environments. The proposed visual-semantic ensemble fusion decoder in GDAAM combines the visual and semantic features to generate the final text recognition results. GDAAM achieves coherent and contextually consistent text recognition outputs by effectively fusing and integrating information from both modalities, improving overall performance. Extensive experiments on benchmark datasets like SVT, ICDAR 2013, ICDAR 2015, IIIT5K, SVTP and CUTE 80 for text recognition in natural scene images demonstrate the effectiveness of GDAAM. The results show that GDAAM outperforms state-of-the-art approaches in terms of accuracy and robustness. GDAAM demonstrates superior performance in challenging text recognition tasks. The proposed model surpasses existing approaches, opening new avenues for accurate and robust text recognition in complex visual environments.

Keywords: *Instance Segmentation, Text recognition, TCN, PBTPN, MS-RCNN, GDAAM, Semantic Reasoning, Character awareness, Visual cue, Semantic cue.*

1. Introduction

The field of computer vision and pattern recognition has seen active research in the domain of recognising deep text present in natural scene images. Over time, numerous approaches have been proposed to tackle the challenges associated with accurately extracting and recognising text from complex visual environments. In this section, we review and discuss the existing works related to text recognition in natural scene images, focusing on the fundamental methodologies, advancements, and limitations.

1Department of Computer Science and Engineering, GITAM (Deemed-to-be University), Visakhapatnam-530045, AP, India
ORCID ID: 0000-0002-9511-0645
2 Department of Computer Science and Engineering, GITAM (Deemed-to-be University), Visakhapatnam-530045, AP, India
ORCID ID: 0000-0002-6263-2666
** Corresponding Author Email: praneelsri@gmail.com*

1.1 Deep Learning-Based Approaches

The field of text recognition in natural scene images has been transformed by advancements in learning, enabling significant progress in terms of accuracy and robustness. Convolutional Neural Networks (CNNs) have been widely adopted for feature extraction from scene images. Methods such as the Fully Convolutional Network (FCN) and Region Convolutional Neural Network (RCNN) have been utilised to localise and segment text regions within the images. These CNN-based approaches have demonstrated impressive performance in terms of text localisation.

1.2 Attention Mechanisms

Attention mechanisms have been widely employed to address the challenges of recognising text within complex scenes. Attention mechanisms enable models to focus selectively on relevant regions or characters within the scene, improving text recognition accuracy. Methods such as Spatial Transformer Networks (STN) and Visual Attention Mechanisms (VAM) have been integrated

into deep learning architectures to enhance the localisation and recognition of text regions.

1.3 Character-Level Analysis

Character-level analysis plays a crucial role in accurate text recognition. Many approaches have focused on modelling the interactions between characters to improve recognition accuracy. Methods based on Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks, have been employed to grasp sequential patterns and contextual details between characters. By considering the relationships and connectivity between characters, these models have shown improved performance in handling irregularities and occlusions in text layouts.

1.4 Semantic Reasoning

Semantic reasoning has emerged as an important aspect of text recognition in natural scene images. Models can better understand and recognise text in complex scenes by incorporating contextual information and higher-level semantics. Reasoning mechanisms based on Graph Neural Networks (GNNs) and Graph Convolutional Networks (GCNs) have been proposed to model the relationships between text regions and other visual elements within the scene. These approaches have demonstrated improved performance in handling scene-level context and improving text recognition accuracy.

1.5 Ensemble Fusion Techniques

Ensemble fusion techniques have been widely explored to enhance text recognition performance further. Fusing information from multiple modalities, such as visual and textual cues, has shown promising results. Fusion mechanisms based on deep attention mechanisms, multimodal memory networks, and graph-based fusion have been proposed to effectively combine information from different modalities and improve the robustness and accuracy of text recognition.

1.6 Benchmark Datasets

Several benchmark datasets have been established to evaluate and compare text recognition methods. Datasets such as ICDAR (International Conference on Document Analysis and Recognition), Street View Text, and CUTE 80 provide diverse collections of scene images with annotated text regions. These datasets have been extensively used to evaluate the performance of text recognition models, fostering advancements in the field.

1.7 Transmissions

This section emphasizes the developments and improvements in recognising text within natural scene images. Approaches based on deep learning techniques, attention mechanisms, character-level analysis, semantic reasoning, and ensemble fusion techniques have improved text recognition accuracy. Benchmark datasets have enabled fair evaluations and comparisons between different methods. Despite significant progress, challenges such as handling variations in font styles, sizes, orientations, complex backgrounds, and lighting conditions remain open research problems. The proposed GDAAM framework addresses these challenges by combining multiple methodologies for accurate and robust text recognition in natural scene images.

Numerous computer vision-based applications, including autonomous driving, travel translations, product retrieval, etc., have taken advantage of the text's rich semantic information. A key component of the scene text reading system is STR. The large differences in scene text's colour, font, spatial layout, and sometimes even uncontrollable background make text recognition in the wild difficult, despite the tremendous advances made in sequence-to-sequence recognition over the years [1,2]. Recent research efforts have predominantly concentrated on improving backbone networks [3,4], incorporating rectification modules [4], and refining attention mechanisms [1,2]. These endeavours aim to bolster the performance of STR by extracting more resilient and effective visual features. However, it is true that a human's ability to recognise scene text depends on both comprehensive comprehensions of the semantic context of text at a higher level and knowledge about visual perception. When only visual cues are considered, it is highly challenging to differentiate each character in such images, as some instances in Fig.1 demonstrate, particularly the characters outlined with pink dotted boxes. Humans are more likely to deduce the correct answer using the entire word's content than by ignoring the semantic context information.

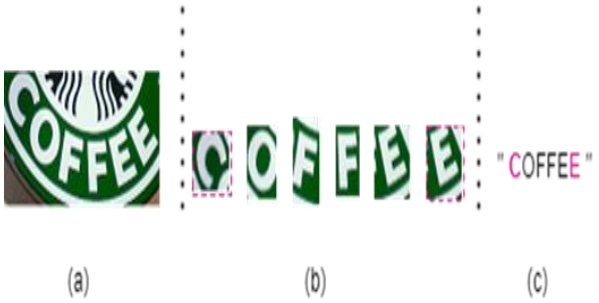

Fig .1: Example is an illustration demonstrating text in various scenarios: **(a)** a challenging scene text image, **(b)** individual characters extracted from the image **(a)**, and **(c)** the semantic content of the words. The characters enclosed in pink dotted boxes in **(b)** are prone to misclassification.

Regrettably, most text recognition methods handle semantic information unidirectionally, akin to one-way serial transmission. This approach, depicted in Fig. 2 (a), can be observed in various works such as [1,2,4], where the character semantic information from the most recent decoding time step is recursively perceived. This approach has many clear disadvantages: It can only comprehend a very small amount of semantic context from each decoding time step, and the initial decoding time step has no meaningful semantic information. Second, if the incorrect decoding is highlighted earlier, it might transmit incorrect semantic information and accumulate errors. Nevertheless, due to the challenges in parallelizing Fig. 2 (b) the serial mode, it becomes inefficient and time-consuming.

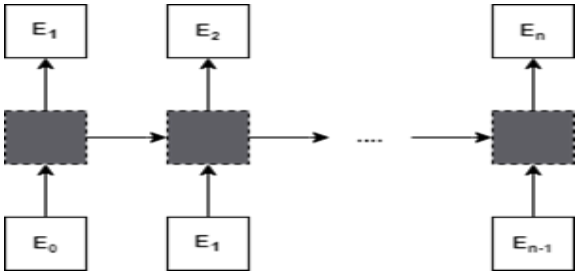

Fig .2(a): Semantic context delivery is a serial transmission.

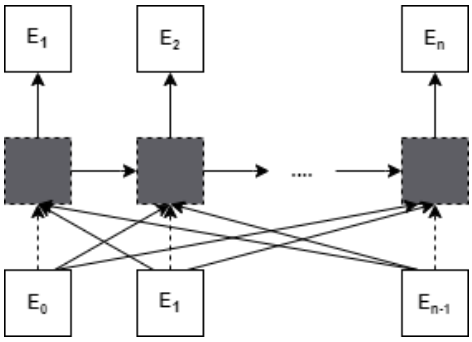

Fig .2(b): Semantic context delivery is a Parallel transmission.

Deciphering text from diverse real-world settings poses a significant and complex hurdle for the multimedia civilization. This endeavour seeks to convert visual depictions of text into a sequence of symbols that computers can comprehend [5]. This job has been widely applied in various real-world applications, including self-driving vehicles, the human-computer interface, and visual aids. Recent advancements in deep learning technology have led to substantial improvements in the field of text recognition within scenes. Recognition models typically adhere to a standard procedure, commencing with an encoding network [6] to gather contextual visual data and culminating in a decoding model [7] that transforms feature vectors into the desired sequence. This decoding module predominantly adopts a 1D sequence-to-sequence model [7], which inherently addresses irregular text instances, especially those involving perspective or curved text. This introduces a novel challenge in converting text images into textual symbols. Recently, various approaches [4,8,9,10] have surfaced, broadly categorised into two types—two-dimensional (2D) attention-based methods and rectification-based methods—aiming to tackle the issue of recognising irregular text. For the 2D attention processes [9], 2D attention maps are employed to align features and facilitate sequence decoding. The erroneous attention maps, however, have a negative impact on recognition performance. Rectification-based algorithms that address irregular text cases to produce canonical representations employ specific sampling procedures to establish control points. For example, ASTER [4] uses a spatial transform network (STN) [11] to predict control points along the boundaries of text instances directly, and ScRN [8] extends this concept by incorporating additional supervision within the rectification module. As illustrated in Fig. 1(a) and (b), the rectification pipelines of ASTER [4] and ScRN [8] are shown. The approach of text-level equidistant sampling generally overlooks character-level details, potentially leading to distorted characters despite the use of various local features to achieve accurate text lines. Accurate character translation is crucial for text recognition since characters are the fundamental building blocks of text instances. Recognizing perspective and curved texts depends heavily on how correctly each character is rectified into a canonical form. The rectification results involving control points and their immediate surroundings are further refined by the TPS transformation, which precisely aligns the selected control points to their predetermined positions. Given the insights discussed earlier, we propose that rectifying irregular text instances using a character-level sampling approach could yield improved outcomes, as it would incorporate more character-specific guidance.

Fig .3: Some illustrations of difficult scene text owing to character cropping, backdrop interference, occlusion, distortion, and blur.

STR has garnered much attention recently because of its use in visual question and answer, information retrieval, and understanding of the visual world. However, as seen in Fig. 3, it is still challenging to recognise irregular text in an unrestricted setting due to visual distortion, blur, and rotation. The final image in Fig. 3, in particular, can be identified as "guice" using the techniques from [9,12,13]. However, individuals are prone to correcting it as a "guide" through the use of contextual understanding. Methods centred around shape rectification [4], detection [14], and attention [15,16] can be categorised as earlier efforts to tackle the challenge of recognising irregular text. Rectification-based methods approximate regular text using traditional text recognition techniques after first transforming irregular material. However, fixing texts with a curved arrangement or significant deformation can be difficult. Character level annotation is followed by character identification using a localised character detection technique before processing all combined detected characters. Character- and word-level annotations are used in attention-based approaches to align each character with visual attributes. The majority of recognition techniques [4,14,15,17] still call for encoding an image's feature map, which is retrieved as one-dimensional (1D) sequences and supplied to an RNN decoder via a convolutional neural network (CNN). Although the positive results that RNN-based recognition methods have produced, some intrinsic drawbacks may hinder their widespread use. First, because of its sequential construction, the RNN requires much time and is difficult to employ to build a deep network. Second, a vanishing or exploding gradient makes it difficult for the RNN to converge. Additionally, prior techniques like [15] have demonstrated that paying attention to 2D features can enhance recognition performance. To this end, we construct a direct link between a relational attention module inspired by the method in [18] and 2D picture feature maps. Post-processing methods can be used for the model's generated sequence labels to improve performance.

There are two different post-processing methods: dictionaries and rectifications based on linguistic models. It would take a lot of time and be unpractical to retrieve label sequences for text recognition using a spelling check dictionary. Instead, a LM is used to choose the labels in the sequence based on the characters that were previously predicted. These post-processing methods, however, only take into account textual information and ignore visual cues. Our proposal for a unified framework that combines linguistic and visual representations to recognize scene text accurately is in response to the aforementioned observations.

2. Related Work

In this section, we delve into the recently introduced text recognition algorithms that employ CNNs and explore the latest advancements in instance segmentation problems within the same research domain.

2.1 Text Recognition

Reading the text in a natural setting is a difficult task that has drawn much attention from industry and academics. Top-down and bottom-up techniques are two categories under which various strategies have been proposed in recent years. A comprehensive exploration of the progress in scene text recognition (STR) has been presented in [19], and we will now offer a brief overview of some state-of-the-art models in this part. Bottom-up methods [20,21] first produce character-level predictions before connecting each character to the appropriate sequences. To forecast each character in a text instance, a classifier is used to collect text information using hand-crafted feature extraction modules like strokelet generation [21] and semi-markov conditional random field [22]. By substituting hand-crafted feature extraction techniques with neural networks, recent deep learning-based methods include dramatically increased performance, including those found [23] instance, LCSegNet [24] generates character-specific pixel-wise predictions using a segmentation model and smooths label assignments using a conditional random field, resulting in a promising performance in several open benchmarks. Another method of text recognition is used in a top-down approach, which reads full-text instances directly without making any predictions about individual letters. To recognise 90k words, Jaderberg [25] created a classification network with 90k categories inspired by the image classification problem. Due to the large number of classes in the classification network and the out-of-vocabulary words, this method cannot be widely applied. The sequence approaches, loosely categorised into CTC (Connectionist temporal classification)-based and Attention-based methods, have been proposed to read text instances of arbitrary duration. CTC-based approaches commonly employ deep networks to encode both visual context and sequence information, followed by utilising CTC to obtain conditional probabilities for texts of varying lengths, as seen in works like [26]. The integration of attention mechanisms into recognition models has gained prominence, as seen in [17] and [27], where focusing maps are constructed for each letter location within text regions to enhance recognition performance.

2.2 Semantic Free Context Approaches

Approaches based on context-free semantics consider STR solely as a visual classification task without explicitly incorporating semantic information. For instance, CRNN [26] combined CNN and RNN to extract sequential visual features from text images, while CTC [28] aimed to maximise the probability of all paths leading to the correct result based on position-specific visual classifications. Xie [29] introduced the aggregation cross entropy (ACE) loss to optimise character frequency along the time dimension, enhancing efficiency and reducing the computational burden of back-propagation in CTC loss. Liao [14] utilised FCN to predict character categories via pixel-level classification and grouped characters into text lines using heuristic criteria inspired by visual segmentation's success. However, this approach requires costly character-level annotations. Jaderberg [30] directly used CNNs to classify 90k text images representing individual words rather than optimising decoding accuracy at each step. These techniques often overlook the significance of semantic context.

2.3 Semantic Aware Context Approaches

Semantic context-aware techniques try recording semantic data to aid STR. The majority of these techniques use one-way semantic transmission. For instance, [27] guided visual features to attend the appropriate region using semantic information from the previous time step after horizontally encoding the input text image into 1D sequential visual features. As we previously said, some of the most recent efforts concentrate on extracting visual cues more successfully, particularly for irregular text. Before sequence recognition, [4] introduced a rectification module incorporating an STN [11], utilising numerous control point pairs to mitigate the adverse effects of perspective distortion and distribution curvature. End-to-end STR via iterative image rectification employed a line-fitting transformation with iterative refinement to correct asymmetric text images. Symmetry-constrained rectification network for STR further enhanced rectification outcomes by developing a symmetry-constrained rectification network based on rich local attributes. Boosting spatial visual information can alleviate irregular text identification challenges to some extent. [31] retrieved scene text features from four directions to control feature contribution from different directions and implemented a filter gate. [1] introduced spatial coordinate encoding to heighten sensitivity to sequential orders on feature maps. However, the semantic context information we want to concentrate on in this research must be properly utilised in these efforts.

2.4 Context Modelling Structure

Context modelling structures are designed to capture information across specific time or space. While RNNs excel at capturing sequence data dependencies, they require assistance with parallel computation during training and inference. To address these challenges, ByteNet [32] and ConvS2S [33] employed CNNs as encoders, enabling full parallelisation during training and inference for optimised hardware utilisation. However, due to limited receptive field size, they may need help to capture global relationships effectively. In contrast, the transformer architecture was developed to capture global dependencies and establish connections between two signals at any point with constant computational complexity. The transformer framework has succeeded in various computer vision and natural language processing [34] tasks. In this study, we leverage the same framework to reason about semantic content and enhance visual encoding using a transformer structure.

2.5 Regular Scene Text Recognition

Early text recognition methods [21,35] typically employ a two-stage recognition process involving character detection in the first step and character recognition in the second. Recent approaches have tried to use a model, like the RNN and CTC, to solve the problem since a scene text is primarily in sequence labels [27,36]. Using RNN modelling techniques, [26,37] handled CNN features as 1D sequences. These techniques eliminate the need for explicitly segmenting individual characters, and the CTC method primarily concentrates on efficient training for rapid predictions. Fang [38] introduced an approach to STR by combining character probabilities through a recognition module and linguistic rules.

2.6 Irregular Scene Text Recognition

Three main categories of methods for recognising irregular scene text are based on attention, as in Fig.4, shape-rectification, and pixel-wise character segmentation approaches. A framework was proposed by rectify-based processes [4] that uses an STN [11] to correct irregular text images. The altered text images are then recognised using a sequence-based recogniser. Rather than rectifying the entire input image, Liu [39] extended the rectification process to individual characters. However, badly distorted texts are difficult to repair, which makes recognition difficult. Methods like [15,18] focused on the feature maps of 2D images and used a customised RNN to classify irregular text images. This method [31] included encoding 4 feature sequences from an image running in various directions in order to construct a character sequence. Similarly, [14] proposed a character attention FCN for precise character localization in a 2D setting. The next step was to classify scene text using an SSN. This was followed by employing a semantic segmentation network for scene text classification. Our method shares similarities with the approach described in [18], particularly regarding visual and textual feature extraction. Our approach is fundamentally inspired by the dot product, which was used by [18] to create links between word representations and a CNN encoder. Although the prior techniques show promise, they frequently just pay attention to visual patterns, need time-consuming RNN models, or rely on separate post-processing procedures. This paper, in contrast, presents a unified framework that includes CNN for extracting image features, dot-product-based relational attention for connecting word representations and image features, and our proposed gated dual adaptive attention strategy for combining data from linguistic dependencies and visual cues. Our recognition model uses simple convolution and attention mechanisms for parallelisable training. To link word representations with image feature maps, this study offers a unified framework that uses mutual attention, dot-product-based relational attention, and CNN for image feature extraction. Simple convolution and attention techniques can be used to train the resulting recognition model in parallel.

2.7 Language Models (LM's)

The non-RNN models [34,40,41,42] that allow for data parallelisation and quicker training are summarised in this section. A completely CNN-based design that permits parallelization over sequential tokens was introduced [33,40]. A transformer machine translation model that relies just on attention was proposed [34]. The transformer concept was expanded [41] to address problems like string copying that the original transformer architecture was unable to manage. The bidirectional encoder representation for transformers (BERT) [42] allows the model to learn a word's dependencies depending on the entire context. In this study, we align character sequences and image characteristics using the dot product from the transformer [34].

3. Proposed Method

3.1 Framework

Fig. 5 depicts the whole structure of the supplied approach. We will discuss the architecture in this section, following the encoder–decoder architecture. Text recognition [15,17] and machine translation [41] both frequently employ this design. Our architecture consists of four components, which are as follows: 1) An image encoder (MS-RCNN +PBTPN+TCN) [43] responsible for extracting 2D features. 2) A relational attention mechanism based on dot products to compute the resemblance or likeness between visual and textual aspects. 3) a language module that extracts the context from words. To combine visual patterns and verbal representations, we devised the ground-breaking gated dual adaptive attention module.

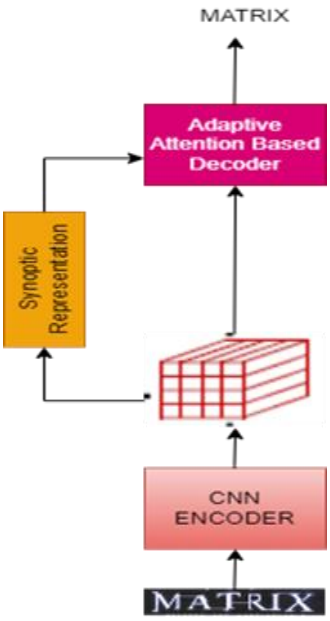

Fig .4: Convolutional Adaptive Attention-based Model

3.1.1 Notation

In the context of a textual image, we use the term "image feature map" represented by the matrix **K**; here, "**m**" means the no. of visual cues, and "**d**" refers to the dimensions of a feature vector at a specific point in space. The label sequence associated with "n" tokens in the text image is denoted as $\mathbf{Q = \{q_1, ..., q_n\}}$ in a matrix format of n rows and d columns. Each $\mathbf{q_n}$ represents the nth token's feature vector. The weights in the various strata are indicated by the letter **W**, together with any necessary subscripts or superscripts. We avoid overcomplicating the notation by omitting the term "**b**" from the discussion to maintain clarity.

3.1.2 Label embeddings

As seen in the lower right corner of Fig.5, during the training phase, we combine a start token, "BOS," with the decoded character sequence to form the decoder input. The real labels are combined with the associated labels, together with the end token "EOS." To accommodate varying sequence lengths within a batch to ensure sequence alignment, a padding token ⟨PAD⟩ is introduced. In this research, the tokens ⟨BOS⟩, ⟨EOS⟩, and ⟨PAD⟩ are assigned the values 1, 2, and 3, respectively. For simplicity in subsequent sections, we analyse a single sequence as input, omitting the ⟨PAD⟩ token. As a result, we can write $\mathbf{I_{input} = (1, i_1, i_2, ..., i_n)}$ and $\mathbf{I_{output} = (i_1, i_2, ..., i_n, 2)}$ to represent the input and output token sequences, respectively, where '**n**' denotes the token count and $\mathbf{i_n}$ denotes the index of the nth token in the character set.

Let's break down the process:1) Character Embedding: We employ a character embedding mechanism similar to that in other models that work with sequences. The i/p and o/p sequences are converted into a matrix, denoted as $\mathbf{I_{nxd}}$, where "**n**" is the sequence length and "**d**" denotes the dimension of the character embedding. We may

extract the character-specific information using this matrix. 2) Positional Encodings: Positional encodings are incorporated into the input embeddings at the beginning of the decoder to preserve the sequence order. These positional encodings offer a mechanism to identify each element's position within the sequence. This is by incorporating character embeddings and positional information, we create a richer representation of the input sequence that considers the characters themselves and their order within the sequence. This can enhance the model's ability to understand and generate meaningful outputs in tasks like text generation or translation.

3.2 Visual Encoder

As our visual feature extractor, we use the MS-RCNN + PBTPN + TCN model's conventional ResNet-50 architecture. After the final average pooling and fully linked layers were eliminated, the relational attention module immediately got the 2D features. We pad the margins to maintain the original aspect ratios while resizing all input photos to 128 (width) x 400 (height) 3 (channel) dimensions. Using convolutional processes, a 2D feature map measuring 49x512 is created.

3.3 Reading Text with Relational Attention

A relational attention module is constructed based on the outlined approach in references [18,34] is adopted, which comprises three key components: **a)** Masked Multi-Head Attention Layer: This layer facilitates learning distant relationships in the data. **b)** 2D Attention Layer: This layer links the two parts as a connection between the encoder and the decoder. **c)** Feed-Forward Layer: This layer refines the information independently and uniformly to each position.

Let's concisely describe the fundamental technique for generating attention maps, which we later extend. In this method regularisation represented by d-dimensional vectors ($\boldsymbol{Q \in R_{n\,x\,d}}$) and m key-value pairs from the visual data ($\mathbf{K} \ \boldsymbol{\epsilon\, R_{m\,x\,d}}$), using regularization the similarity matrix between the character representations and image feature is calculated as follows:

$$\boldsymbol{S = softmax\left(\frac{QW_Q(KW_K)^T}{\sqrt{d}}\right) \in R^{n\,x\,m}} \qquad (1)$$

Where: $\boldsymbol{W_0 \epsilon R^{d\,x\,\frac{d}{H}}}$ and $\boldsymbol{W_k \epsilon R^{d\,x\,\frac{d}{H}}}$ are learned weight matrices. The parameter H is set to eight. The square root of d is used as a scaling factor to prevent excessively large dot products. This matrix S captures the weighted relationships between the image and character features, a crucial step in our method.

To capture information from different subspaces, the model simultaneously attends to multiple instances of attention, resulting in the combined output represented as **O** represented as:

$$\mathbf{O = [O^1, \dots, O^H]W_O \in R_{n\times d}} \qquad (2)$$

Here, **[·, ·]** signifies the concatenation of matrices horizontally, and $\boldsymbol{W_0 \ \epsilon \ R^{d\,x\,d}}$ is a matrix subject to learning.

The output of an instance of individual attention $\mathbf{O^i}$ is achieved through the values of weightes summation obtained as follows.

$$\boldsymbol{Q^i = S^i\, KW\, K_K^i \epsilon R^{d\,x\,\frac{d}{H}},\ i \epsilon \{1, \dots\dots, H\}} \qquad \mathbf{(3)}$$

where **'i'** takes values from **1** to **H**. In this context, $\boldsymbol{W_K^i \epsilon R^{d\,x\,\frac{d}{h}}}$ stands for a learned weight matrix, and **d/h** signifies the number of dimensions for the weights.

This mechanism of jointly attending to multiple attention instances and computing weighted summations contributes to the model's ability to assimilate information from diverse subspaces, enhancing its overall performance. crucial since the order of elements carries valuable context in sequential data. 3) Sin and Cos Functions: We use sine and cosine functions to construct the positional encodings, drawing on ideas covered in [34]. This choice of functions is designed to create distinct patterns for different positions, ensuring that the model can differentiate between various parts of the sequence based on their order. 4)Position Matrix: A position matrix is the outcome of using the sine and cosine functions, represented as $\mathbf{P_{n\times d}}$, where "n" again signifies the sequence length, and "**d**" denotes the dimensions of the positional embedding. 5) Label Embeddings: We combine everything by adding positional encodings ($\mathbf{P_{n\times d}}$) to character embeddings ($\mathbf{I_{n\times d}}$). This combination results in label embeddings, denoted as **Q**, which contain character-based and positional information. This can be expressed mathematically as $\mathbf{Q = I_{n\times d} + P_{n\times d}}$.

The attention module develops a function that transforms a query **Q** into a weighted sum of values, with the weights determined by the dot-product formula. Nevertheless, it is critical during training to keep the decoder from gaining access to places that still require decoding. This is accomplished at a certain decoding stage by masking out pointless places. The masked dot-product attention query's target is the 2D feature map that the 2D attention layer acquired from the CNN image encoder. The multi-head attention method presented in [34] enables us to build linkages between input and output sequences utilizing a fixed number of sequential operations (**O (1)**). With the help of this technique, we may convey the universal relationship between character encoding and visual cues irrespective of distance.

Following the 2D attention stage, a position-wise feed-forward network comes into play. This network operates on a per-position basis, involving two linear projections. We proceed with the computation $\mathbf{K_{vi}}$ using the formula:

$$\boldsymbol{K_{vi} = \mathbf{ReLU}\,(OW_{l1})W_{l2} \epsilon R^{n\,x\,d}} \qquad (4)$$

where $\boldsymbol{W_{l1} \epsilon R^{d\,x\,d_{ll}}}$ and $\boldsymbol{W_{l2} \epsilon R^{d_{ll}\,x\,d}}$ are matrices with learnable weights. $\boldsymbol{d_{ll}}$ signifies the dimensionality of the inner layer. The enhanced visual representation is represented by $\boldsymbol{K_{vi}}$. We create a preliminary estimation n of the results sequence based on the calculated relational properties by using:

$$\boldsymbol{P_r = Softmax\,(K_{vi}W_r) \epsilon R^{n\,x\,c}} \qquad (5)$$

Here: $\boldsymbol{W_r \epsilon R^{d\,x\,c}}$ is a matrix that is subject to learning. The letter "c" stands for all of the classes in the character alphabet.

As our baseline for recognition in this study, $\mathbf{OCR_{baseline}}$, we use the estimated value from Equation (5). In the sections that follow, we examine techniques to improve recognition accuracy.

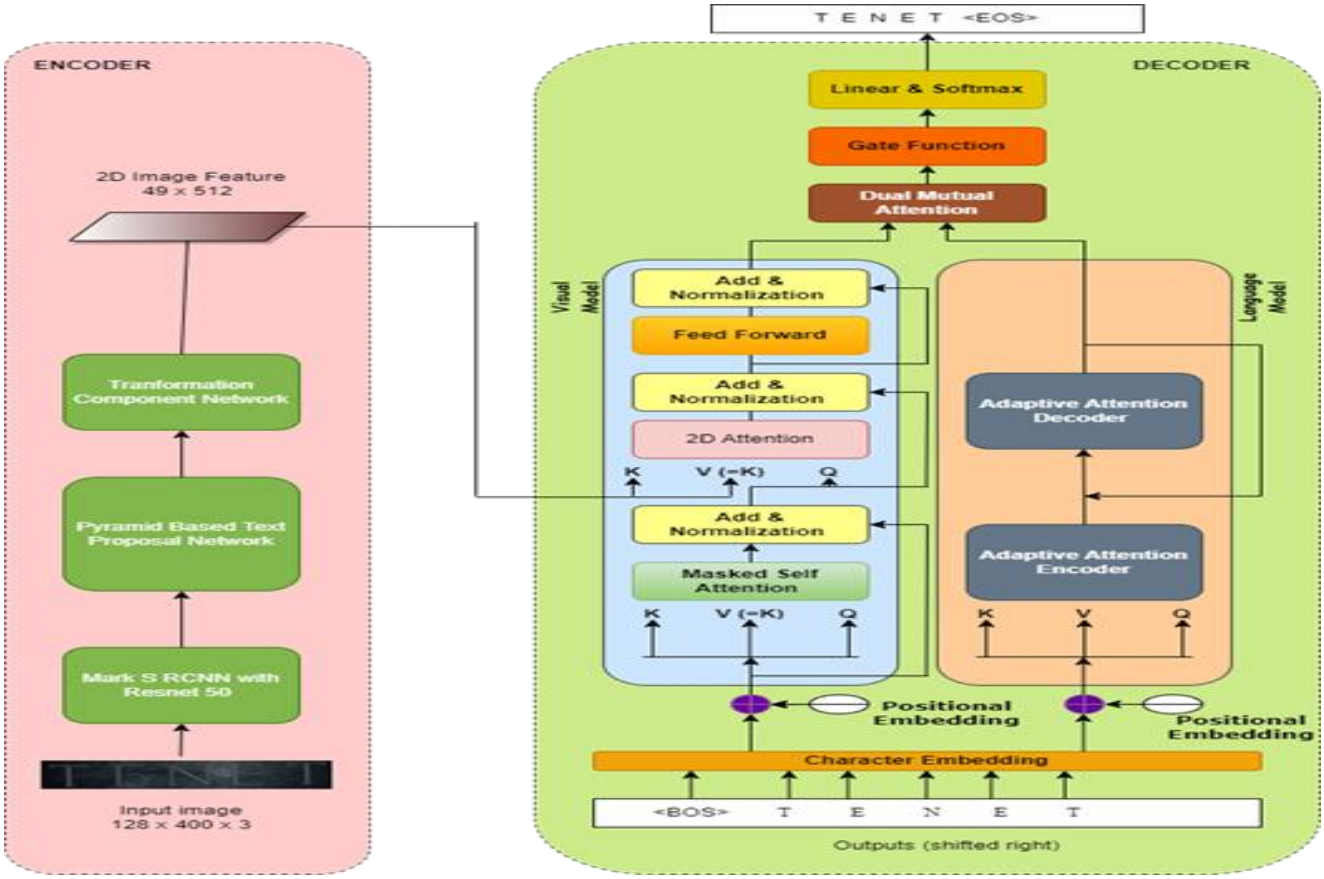

Fig .5: The proposed model overview. We use Resnet 50 as the backbone in the MS RCNN with PBTPN and TCN as Encoder, and the Decoder uses a relational attention module and an LM.

3.4 Linguistic Encoder

The standard transformer architecture described in reference [34] as a LM to comprehend character relationships is used. This LM takes in the identical character sequence $\boldsymbol{Q \epsilon R^{n\,x\,d}}$, like the information provided to the relational attention module outlined in Section 3.3. The output of the LM yields attentive linguistic dependencies $\boldsymbol{Q_{li} \epsilon R^{n\,x\,d}}$, as defined by Equation (2).

The LM proceeds to predict the next most probable token using the equation:

$$\boldsymbol{P_{li} = Softmax\,(Q_{li} W_{li}) \epsilon R^{n\,x\,c}} \qquad (6)$$

Where $\boldsymbol{W_{li} \epsilon R^{d\,x\,c}}$ represents a matrix with learnable weights.

3.5 Using Gated Dual Adaptive attention to generate visual and linguistic representations

The ability to mix features from many models is one approach to conceptualize the attention process [44]. In this part, we propose the "gated dual adaptive attention" method, a straightforward yet powerful method for capturing the interaction of cross-modal stimuli originating from visual and linguistic representations.

With "n" standing for the decoded text length and "d" standing for model's dimensionality, the inputs, indicated as $\boldsymbol{Q_{li} \epsilon R^{n\,x\,d}}$ and $\boldsymbol{K_{vi} \epsilon R^{n\,x\,d}}$, are drawn from the relational attention and LM's outputs. Following the principles outlined in Equation (1), the adaptive attention maps for linguistics conditioned on visual data termed adaptive attention I, can be computed as:

$$\boldsymbol{S_{vi} = Softmax\left(\frac{Q_{li} W_{Q_{li}} (K_{vi} W_{K_{vi}})^T}{\sqrt{d}}\right) \epsilon R^{n\,x\,n}} \qquad (7)$$

The following is how the attention maps for linguistically conditioned visual data are created.:

$$\boldsymbol{S_{li} = S_{vi}^T \epsilon R^{n\,x\,n}} \qquad (8)$$

By using adaptive attention, the attended visual and language characteristics can be succinctly expressed as:

$$\boldsymbol{O_{Q_{li}} = [O_{Q_{li}}^1, \ldots\ldots, O_{Q_{li}}^H] W_{O_{li}} \epsilon R^{n\,x\,d}} \qquad (9a)$$

$$\boldsymbol{O_{K_{vi}} = [O_{K_{vi}}^1, \ldots\ldots, O_{K_{vi}}^H] W_{O_{vi}} \epsilon R^{n\,x\,d}} \qquad (9b)$$

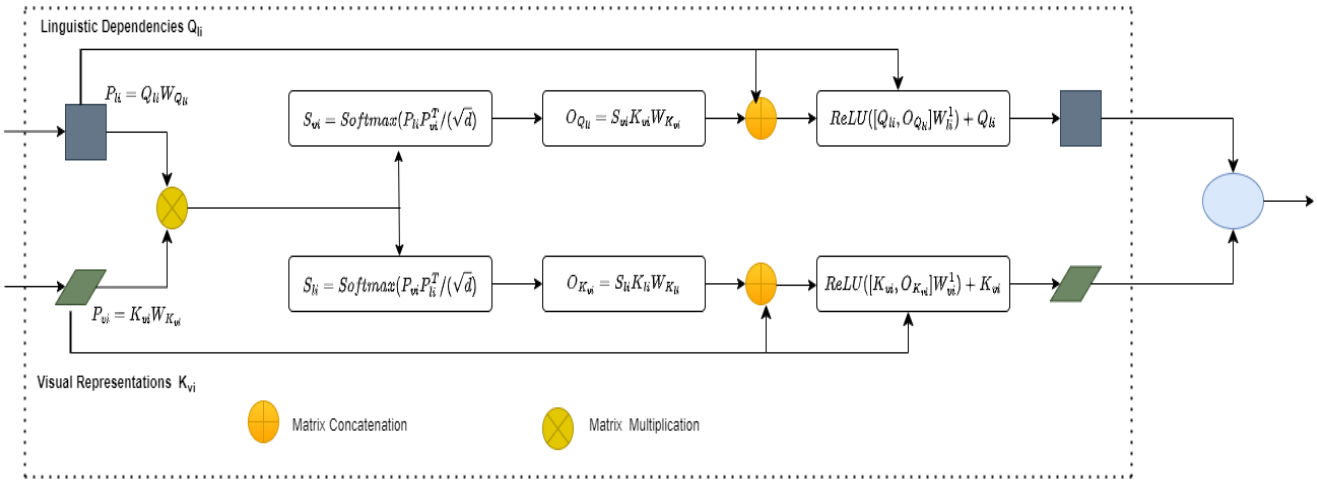

Fig 6: The internal organization of a dual adaptive attention layer..

Using a residual link and a feed-forward network, we integrate the visual representations once the attention-based features have been collected as indicated in Equations (9a) and (9b):

$$\boldsymbol{K'_{vi} = ReLU\big(\,[\,K_{vi}, O_{K_{vi}}]W'_{vi}\,\big) + K_{vi} \in R^{n\,x\,d}} \qquad (10)$$

In this equation, [·, ·] represents horizontal matrix concatenation, and $\boldsymbol{W'_{vi} \in R^{2d\,x\,d}}$ is a weight matrix subject to learning.

Likewise, linguistic information can be combined as follows:

$$\boldsymbol{Q'_{li} = ReLU\big(\,[\,Q_{li}, O_{Q_{li}}]W'_{li}\,\big) + Q_{li} \in R^{n\,x\,d}} \qquad (11)$$

This approach effectively fuses cross-modal features using attention mechanisms and feed-forward networks, capturing intricate relationships between visual and linguistic representations.

On the other hand, the matrix $\boldsymbol{W'_{li} \in R^{2d\,x\,d}}$ represents a weight matrix with learnable parameters.

This approach refines multimodal features by iteratively extracting and propagating intricate relationships within each modality, enhancing the overall understanding and representation of the cross-modal information.

3.6 Fusion of the Visual and Linguistic representation using a gated network

To simplify the example given in Fig. 6, we add a layer to the linguistic and visual representations of single attention. The combined representations generated by Equations (10) and (11) are used to compute the final result using a gated technique that integrates a portion of visual attributes with verbal context:

$$\boldsymbol{P_{fu} = Softmax\,\big(F_{gate}[\,K'_{vi}\,, Q'_{li}\,]\big)W'_{gate} \in R^{n\,x\,c}} \qquad (12)$$

In this equation $\boldsymbol{W'_{gate} \in R^{2d\,x\,c}}$ is a matrix with learnable weights.

The letter "**c**" stands for how many classes there are in the alphabet. For each modality, $\boldsymbol{F_{gate}}$, a non-linear function, calculates attentiveness.

Essentially, the gated network gives the model the ability to dynamically learn and ascertain the contributions made by each modality to the representations across modalities.

Additionally, in the ablation studies described in Section 4.3. This exploration helps to comprehend the effectiveness of different fusion techniques in the context of the model's performance.

3.7 GDAAM algorithm

1. Begin
2. Function:GDAAM_Model(input_feature_map, target_text)
3. num_classes # Number of output classes , input_channels # Number of input image channels (e.g., RGB) , kernel_size # Size of the convolutional kernel , stride # Stride of the convolution operation, hidden_units # Number of hidden units in the fully connected layers , dropout_prob # Dropout probability
4. model = Model(num_classes, input_channels, kernel_size, stride, hidden_units, dropout_prob)

 # Encoder: Feature Map

 # Input feature map is generated using MS-RCNN, PBTPN, and TCN

5.ms_rcnn= models.detection.maskrcnn_resnet50_fpn(pretrained=True)

6.features_ms_rcnn = ms_rcnn(input_image)

7. pbtpn = module.PBTPNModel()
8. features_pbtpn = pbtpn(features_ms_rcnn)
9. tcn = module.TCNModel()
10. feature_map = tcn(features_pbtpn)

11. return feature_map

 # The feature map represents high-level features extracted from the input data

 # Text Preprocessing

 # Tokenize and encode the target_text into character-level embeddings

12. encode_text_to_char_embeddings(target_text, char_to_index, max_sequence_length)
13. target_text: The input target text as a string.
14. char_to_index: A dictionary that maps characters to their corresponding numerical indices.
15. max_sequence_length: The maximum sequence length for padding.
16. char_embeddings= encode_text_to_char_embeddings(target_text, char_to_index, max_sequence_length)
17. return char_embeddings

 # Decoder: Adaptive Attention

 # Initialize the Adaptive Attention Decoder

18. input_size # Number of unique characters in the input , output_size # Number of unique words in the output vocabulary,char_embedding_dim # Dimension of character embeddings, pos_embedding_dim # Dimension of positional embeddings ,hidden_dim # Dimension of hidden state in the, num_layers # Number of layers, dropout_p # Dropout probability, attention_type # Choose the appropriate attention mechanism, max_sequence_length # Maximum sequence length for positional embeddings
19. encoder = CharacterEmbeddingEncoder (input_size, char_embedding_dim, hidden_dim, num_layers, dropout_p)
20. decoder = AdaptiveAttentionDecoder (output_size, char_embedding_dim, pos_embedding_dim, hidden_dim, num_layers, dropout_p, attention_type)

 # Dual Adaptive Attention Mechanism

 # Calculate bidirectional attention scores between feature map and character embeddings

21. bidirectional_attention= BidirectionalAttention(feature_map_dim, char_embedding_dim)
22. attention_scores_fm_to_char, attention_scores_char_to_fm = bidirectional_attention(feature_map, char_embeddings)

 # Apply gating mechanisms to control information flow between modalities

23. gated_fusion = GatedFusionModule(text_dim, visual_dim)
24. fused_representation = gated_fusion(text_input, visual_input)

 # Text Generation

 # Initialize the decoder with the combined context

#Generate text description character by character using the decoder

25. generated_text = generate_text_description(decoder, hidden, cell, max_length=100, char_to_index)

 # Predict the next character based on the previous character and context information

26. model = CharLanguageModel(input_size, hidden_size, output_size, num_layers)

 # Define an initial hidden state

27. hidden = (torch.zeros(num_layers, 1, hidden_size), torch.zeros(num_layers, 1, hidden_size))

 # Sample the next character based on the previous character and context

28. previous_char = torch.tensor([1]) # Replace with the index of the actual previous character
29. next_output, hidden = model(previous_char, hidden)
30. next_char = sample_next_char(next_output)

 # Use a character embedding vocabulary to sample the next character

31. Return target_text
32. End Function

3.8 Optimization

The model underwent optimisation through a multitask loss approach, incorporating three distinct cross-entropy loss functions as follows:

$$L\_total = L\_rel * W_{L\_rel} + L\_li * W_{L\_li} + L\_gate * W_{L\_gate} \quad (13)$$

Where: $L\_rel = -\log(P_{rel})$, $L\_li = -\log(P_{li})$, and $L\_gate = -\log(P_g)$ (5), (6), and (12), as specified, respectively.

$W_{L\_rel} \in R^{c \times c}$, $W_{L\_li} \in R^{c \times c}$, and $W_{L\_gate} \in R^{c \times c}$ represents matrices of learnable weights.

The model can learn and improve based on the relational attention, linguistic attention, and fused attention outputs using this configuration of the multitask loss function, with the corresponding weight matrices contributing to the final learning process.

3.9 Bidirectional Training

In our method, we make use of a bidirectional transformer network, drawing inspiration from recent developments. During the training phase, we input the decoder for a specific s sequence with both left-to-right **(Fig .7 (a))** and right-to-left **(Fig .7 (b))** sequences. This method enhances the model's comprehension by enabling it to view the image and related text from two alternative perspectives.

Table 1. Scene Text Recognition Accuracy on different Datasets. "None" represents that no lexicons are used.

Method	*Regular Text*			*Irregular Text*		
	IIIT5K	***IC13***	***SVT***	***IC15***	***SVTP***	***CUTE 80***
	None	*None*	*None*	*None*	*None*	***None***
Shi et al. 2016 [6]	81.9	88.6	81.9	-	71.8	59.2
Cheng et al. 2017 [17]	87.4	93.3	85.9	-	71.5	63.9
Shi et al. 2018 [4]	93.4	91.8	93.6	-	73.0	79.5
Li et al. 2019 [15]	91.5	91.0	84.5	69.2	76.4	83.3
Wan et al. 2020 [12]	93.9	92.9	90.1	79.4	84.3	83.3
Zhiguang et al. 2021[44]	96.6	96.4	94.1	81.6	85.6	91.4
Ours et al. 2023	**97.8**	**97.5**	**96.2**	**83.7**	**86.9**	**93.1**

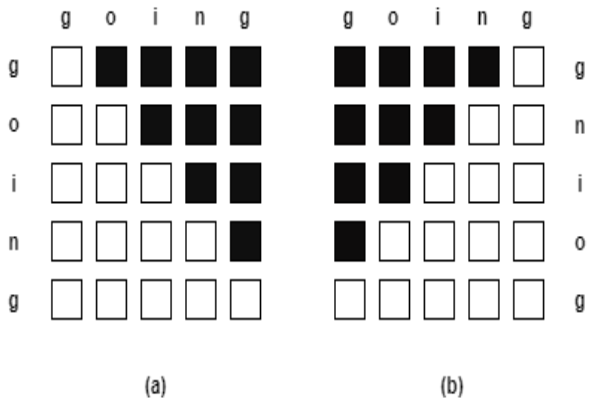

Fig .7: (a) Left to Right Decoding and **(b)** Right to Left Decoding for the bidirectional training.

The bidirectional training process, as shown in **Fig 7**, resembles unidirectional learning with one important exception—we mask out future tokens. In the Fig.7 the black blocs will represent the allow to attend characters and the hallow blocks are the preventing from attending. The model's ability to successfully capture contextual information coming from different directions is greatly improved by this bidirectional technique.

4. Experiments

The only synthetic datasets used for the proposed model's training were the Synth90K dataset, SynthText dataset which contains nine million items [30], and eight million items [45] respectively. On a number of publicly accessible benchmark datasets, our trained model was assessed, and its performance was measured against that of cutting-edge techniques.

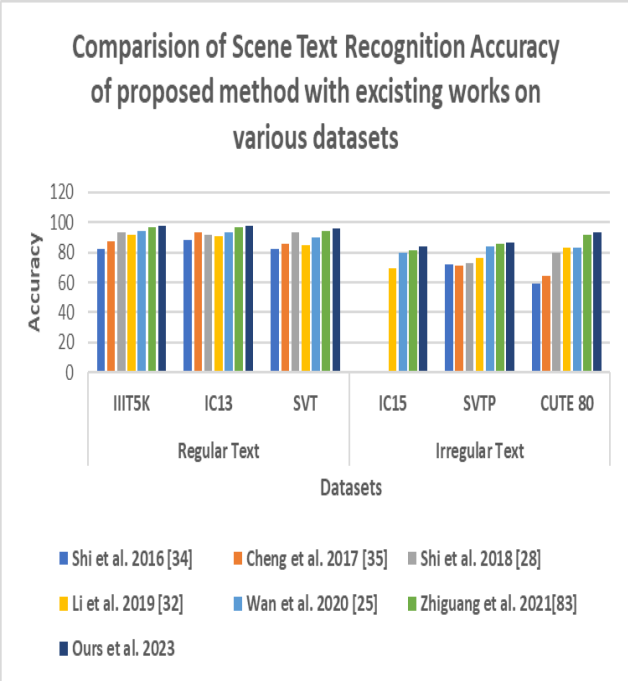

Fig .8: Comparison of STR accuracy of proposed work with existing works on various Datasets.

4.1 Datasets

The evaluation took place on the following benchmark datasets:

I.IIIT5K: Comprising 3000 test images, predominantly featuring regular text instances.

II. Street View Text (SVT): Consisting of 647 test samples extracted from Google Street View images.

III. SVT-Perspective (SVTP): Derived from Google Street View images, this dataset contains text instances distorted due to perspective.

IV. ICDAR 2013 (IC13): Involving 1,095 regular word patches for testing. Non-alphanumeric characters were removed, leaving 1,015 test images.

V. ICDAR 2015 (IC15): Comprising 2077 images captured incidentally at varying angles. We removed non-alphanumeric characters to create a subset known as IC15-1811 that allows for a fair comparison with earlier techniques.

VI. CUTE80: Encompassing 288 cropped patches.

These benchmark datasets served as a basis for assessing the model's performance and comparing it with the achievements of previous approaches.

4.2 Experimental Results

The state-of-the-art approaches are compared with our proposed solution in this section utilizing a variety of datasets that contain both regular and irregular text instances. Table 1 displays the recognition's outcomes. The Table 1 results of the STR accuracy of different methods are compared with our proposed method on different datasets is shown in Fig 8.

Table 1's findings show that, even when using the lightweight image encoder ResNet 50, our model consistently outperformed competing methods on test datasets comprising both regular and irregular texts. By using a baseline model that is more reliable than the OCR $_{\text{baseline}}$, the recognition performance could be further enhanced.

Fig. 9: A few examples of our model's success. The terms "**Gt**" and "**Pred**" stand for the ground truth, "**Ours**" for multimodal fusion utilizing our gated dual adaptive attention technique, and "**Pred**" for model prediction **OCR**$_{\text{baseline}}$ based purely on visual information.

A few successful applications of our technology are shown in Fig. 9. For instance, our **OCR**$_{\text{baseline}}$, which relies purely on visual features, struggled to deal with noisy images (Fig. 9(a) and 9(d)) caused by continuous scanning and printing methods. Through the GDAAM, our system made use of both language and visual representations to make more precise predictions. As seen in the examples in Fig. 9(b),9(c) the model was adept at resolving images with characters who had similar appearances. In Fig. 9(f), where the background noise obscured the letter "I", it also demonstrated the capability of recovering overlooked characters that the **OCR** $_{\text{baseline}}$ missed.

While our method showed triumphs, it's crucial to highlight that there were also failures and certain restrictions, which we go over in Section 5.

4.3 Ablation Study

Within this section, we delve into an examination of the factors that impact our model's performance.

4.3.1 Encoder backbone

In order to evaluate the effectiveness of several CNN architectures as image encoders, we looked into the VGG-16, VGG-19, ResNet-34, ResNet-50, and ResNet-101 models. The performance improvements were negligible as we used deeper convolutional architectures to increase the parameter count. Because the ResNet -50 produces noticeably superior results, we chose it.

4.3.2Multi-Task Loss Function

Three loss functions were used to optimize the suggested model, and the following three terms were found to be effective. We discovered that employing just the $\mathbf{L}_{\text{rel}}$ as our baseline, represented by OCR$_{\text{baseline}}$, already produces competitive performance in comparison to the other approaches. However, when employing simply $\mathbf{L}_{\text{gate}}$, which comprised a pre-trained LM with a significant amount of previous information about the labels, the recognition performance fell short of **OCR**$_{\text{baseline}}$. During training, this caused convergence to happen quickly.

Consequently, the visual models were not properly learned, such We tested the efficacy of these terms by optimizing our suggested model using three different loss functions. Using merely the $\mathbf{L}_{\text{rel}}$ term as our baseline, abbreviated as **OCR**$_{\text{baseline}}$, already produced competitive performance when compared to other approaches. Rapid convergence during training, however, resulted in a drop in recognition performance when only using the L_{gate} term, which comprised a pre-trained LM equipped with significant preliminary label information. This rapid convergence was helpful for the LM, but it interfered with the correct acquisition of visual representations, which eventually affected prediction accuracy.

We merged the two loss terms, L_{gate} and L_{rel}, to strike a balance between acquiring useful visual representations and avoiding biases from the linguistic model. In comparison to our baseline, **OCR**$_{\text{baseline}}$, this combination of linguistic rules and visual relational characteristics produced a significant 2.1 % improvement (from 81.6% to 83.7%) on the IC15 dataset.

The pre-trained LM's introduction considerably improves model convergence while lowering loss. In contrast to RNN-based techniques, which frequently encounter problems like vanishing/exploding gradients, our method makes use of the benefits of sequence training to improve both training and model performance. Using labels from training examples of text images to fine-tune the LM may be an option to further improve recognition performance.

4.3.3 Ensemble Strategies

First, we looked at how deep mutual attention affected system performance. The proposed GDAAM, straightforward feature concatenation, and bilinear interpolation were the next strategies we experimented with.

Although the suggested dual adaptive attention may theoretically be stacked with several layers, we found that, unlike machine translation, adding too many deep joint attention layers could have a negative impact on identification performance. As a result, we assessed the first, third, and eighth layers of adaptive attention. On the IC13 dataset, the resulting recognition accuracy values were 97.5%, 96.7%, and 95.3%, respectively. This phenomenon is similar to those found in [7], where it was discovered that long-range dependency modeling was not as important for brief sequences. As a result, we used a single-layer dual adaptive attention technique.

The independent examination of linguistic and visual representations in the logit summation strategy may have enhanced performance. On the other hand, combining multimodal features by means of straightforward concatenation improves the starting point. Bilinear interactions [46] between multimodal representations [47] produced a noticeable improvement in performance. The best results were obtained using the gated adaptive attention approach we devised, which allowed the gated network to find more cross-modality correlations that improved model discrimination. We also evaluated the model size and training time of our suggested approach. Using a single, dual adaptive attention layer as stated in Section 3.4, we calculated the average training speed for five hundred batches with a total of 256 samples.

4.3.4 Implementation

The tests were run on a server that had sixteen gigabytes of memory and eight NVIDIA Tesla V100 GPU cards. All 8 GPU cards were used to parallelize the model's training, and an ADAM optimization batch size of 2,048 was used. We started the image encoder using pre-trained weights for picture classification from ImageNet. We used a learning rate schedule where the rate was linearly raised over the initial 10 training steps and subsequently decreased, according to [34]. After 15 epochs, the model has attained convergence. To fully represent all categories in the open benchmark datasets used for evaluation, we selected 172 label classes containing 52 case-sensitive letters, ten digits, 106 non-alphanumeric characters, and four unique tokens. It was decided to use feature dimensions (d) of 512.

The model used to generate the results in Table 1 was trained using the synthesised datasets Synth90K [30] and SynthText [46]. To preserve a constant aspect ratio, input photos were scaled and padded to dimensions of 128x400.

We used the benchmark corpus of One Billion Words [48] to train the LM. Then, we either applied the pre-trained LM or improved it by using all of the labels from the two synthesis datasets plus training data from publicly available datasets. The synthesised training images were so distinct from the accessible benchmark datasets that overfitting was prevented because our model took both text and images as inputs.

Fig .10: Examples of our model's failure cases. We use "**Gt**" to stand for "ground truth," "**Pred**" for "model prediction," and we provide "**Reason**" potential explanations for failure scenarios.

5. Limitation

Our method was unable to handle vertical text images since they weren't present in the training data (Fig. 10(f)). We illustrate a few examples in Fig. 10 when our technique fell short. Our method attempted to infer the text from noisy images, as shown in Fig. 10(a-b), however it provided unfavourable predictions due of occlusions or blurring. Low R (resolution) and unique lighting (Fig. 10(c-d)). Additional sources of erroneous recognition included the placement of some character special positions and many text lines (Fig. 10(e) and Fig. 10(f)).

6. Conclusion

In this study, we suggested a unified, neural component-based STR system that does not require a vocabulary. The decoder comprises an LM that computes contextual information and a relational dual adaptive attention module that connects character and visual embeddings. An MS-RCNN and PBTPN+TCN encoder that extracts 2D visual patterns uses ResNet 50 as its backbone. A dual adaptive attention module was developed to incorporate language dependencies with visual cues. This greatly enhanced identification efficiency. The suggested design is flexible because it offers encoder and decoder alternatives. The model was simultaneously trained using the teacher-forcing method, and it converged much more quickly when a pre-trained LM was used.

The STR architecture is extensible since the recommended GDAAM module can be modified to provide more text recognition outputs, such as [15]. In the future, we'll run an experiment to confirm this. The provided framework can be enhanced in a variety of ways as a meta-algorithm. In order to improve recognition performance, it is first plausible to investigate a wider range of visual representation strategies, such as layered multi-scale picture characteristics with 2D attention. Using a bidirectional LM, like the BERT, can increase language dependence.

References

[1] Wojna, Z., Gorban, A. N., Lee, D. S., Murphy, K., Yu, Q., Li, Y., & Ibarz, J. (2017, November). Attention-based extraction of structured information from street view imagery. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 1, pp. 844-850). IEEE.

[2] Yang, X., He, D., Zhou, Z., Kifer, D., & Giles, C. L. (2017, August). Learning to read irregular text with attention mechanisms. In IJCAI (Vol. 1, No. 2, p. 3).

[3] Lyu, P., Liao, M., Yao, C., Wu, W., & Bai, X. (2018). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In Proceedings of the European conference on computer vision (ECCV) (pp. 67-83).

[4] Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., & Bai, X. (2018). Aster: An attentional scene text recognizer with flexible rectification. IEEE transactions on pattern analysis and machine intelligence, 41(9), 2035-2048.

[5] Yao, C., Bai, X., & Liu, W. (2014). A unified framework for multioriented text detection and recognition. IEEE Transactions on Image Processing, 23(11), 4737-4749.

[6] Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust scene text recognition with automatic rectification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4168-4176).

[7] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[8] Yang, M., Guan, Y., Liao, M., He, X., Bian, K., Bai, S., ... & Bai, X. (2019). Symmetry-constrained rectification network for scene text recognition. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9147-9156).

[9] Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., ... & Cai, M. (2020, April). Decoupled attention network for text recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 12216-12224).

[10] Dai, P., Zhang, H., & Cao, X. (2019). Deep multi-scale context aware feature aggregation for curved scene text detection. IEEE Transactions on Multimedia, 22(8), 1969-1984.

[11] Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015, December). Spatial transformer networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2 (pp. 2017-2025).

[12] Wan, Z., He, M., Chen, H., Bai, X., & Yao, C. (2020, April). Textscanner: Reading characters in order for robust scene text recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 12120-12127).

[13] Yue, X., Kuang, Z., Lin, C., Sun, H., & Zhang, W. (2020, August). Robustscanner: Dynamically enhancing positional clues for robust text recognition. In European Conference on Computer Vision (pp. 135-151). Cham: Springer International Publishing.

[14] Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., ... & Bai, X. (2019, July). Scene text recognition from two-dimensional perspective. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 8714-8721).

[15] Li, H., Wang, P., Shen, C., & Zhang, G. (2019, July). Show, attend and read: A simple and strong baseline for irregular text recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 8610-8617).

[16] Luo, C., Jin, L., & Sun, Z. (2019). Moran: A multi-object rectified attention network for scene text recognition. Pattern Recognition, 90, 109-118.

[17] Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., & Zhou, S. (2017). Focusing attention: Towards accurate text recognition in natural images. In Proceedings of the IEEE international conference on computer vision (pp. 5076-5084).

[18] Yang, L., Wang, P., Li, H., Li, Z., & Zhang, Y. (2020). A holistic representation guided attention network for scene text recognition. Neurocomputing, 414, 67-75.

[19] Long, S., He, X., & Yao, C. (2021). Scene text detection and recognition: The deep learning era. International Journal of Computer Vision, 129, 161-184.

[20] Mishra, A., Alahari, K., & Jawahar, C. V. (2012, June). Top-down and bottom-up cues for scene text recognition. In 2012 IEEE conference on computer vision and pattern recognition (pp. 2687-2694). IEEE.

[21] Yao, C., Bai, X., Shi, B., & Liu, W. (2014). Strokelets: A learned multi-scale representation for scene text recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4042-4049).

[22] Seok, J. H., & Kim, J. H. (2015). Scene text recognition using a Hough forest implicit shape model and semi-Markov conditional random fields. Pattern Recognition, 48(11), 3584-3599.

[23] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012, November). End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st international conference on pattern recognition (ICPR2012) (pp. 3304-3308). IEEE.

[24] Wu, X., Chen, Q., Xiao, Y., Li, W., Liu, X., & Hu, B. (2020). LCSegNet: An efficient semantic segmentation network for large-scale complex Chinese character recognition. IEEE Transactions on Multimedia, 23, 3427-3440.

[25] Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. International journal of computer vision, 116, 1-20.

[26] Shi, B., Bai, X., & Yao, C. (2016). 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, 39(11), 2298-2304.

[27] Lee, C. Y., & Osindero, S. (2016). Recursive recurrent nets with attention modeling for ocr in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2231-2239).

[28] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning (pp. 369-376).

[29] Xie, Z., Huang, Y., Zhu, Y., Jin, L., Liu, Y., & Xie, L. (2019). Aggregation cross-entropy for sequence recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6538-6547).

[30] Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227.

[31] Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., & Zhou, S. (2018). Aon: Towards arbitrarily-oriented text recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5571-5579).

[32] Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. V. D., Graves, A., & Kavukcuoglu, K. (2016). Neural machine translation in linear time. arXiv preprint arXiv:1610.10099.

[33] Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017, July). Convolutional sequence to sequence learning. In International conference on machine learning (pp. 1243-1252). PMLR.

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[35] Wang, K., Babenko, B., & Belongie, S. (2011, November). End-to-end scene text recognition. In 2011 International conference on computer vision (pp. 1457-1464). IEEE.

[36] Bai, F., Cheng, Z., Niu, Y., Pu, S., & Zhou, S. (2018). Edit probability for scene text recognition. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1508-1516).

[37] He, P., Huang, W., Qiao, Y., Loy, C., & Tang, X. (2016, March). Reading scene text in deep convolutional sequences. In Proceedings of the AAAI conference on artificial intelligence (Vol. 30, No. 1).

[38] Fang, S., Xie, H., Zha, Z. J., Sun, N., Tan, J., & Zhang, Y. (2018, October). Attention and language ensemble for scene text recognition with convolutional sequence modeling. In Proceedings of the 26th ACM international conference on Multimedia (pp. 248-256).

[39] Liu, W., Chen, C., & Wong, K. Y. (2018, April). Char-net: A character-aware neural network for distorted scene text recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[40] Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017, July). Language modeling with gated convolutional networks. In International conference on machine learning (pp. 933-941). PMLR.

[41] Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., & Kaiser, Ł. (2018). Universal transformers. arXiv preprint arXiv:1807.03819.

[42] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[43] Praneel, A. V., & Rao, T. S. (2023). Scene Text Detection Using Pyramid-Based Text Proposal Network and Transformation Component Network. (pp.21-32). IJCSE.

[44] Liu, Z., Wang, L., & Qiao, J. (2022). Visual and semantic ensemble for scene text recognition with gated dual mutual attention. International Journal of Multimedia Information Retrieval, 11(4), 669-680.

[45] Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2315-2324).

[46] Ben-Younes, H., Cadene, R., Thome, N., & Cord, M. (2019, July). Block: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 8102-8109).

[47] Venkata Praneel, A. S., Srinivasa Rao, T., & Ramakrishna Murty, M. (2020). A survey on accelerating the classifier training using various boosting schemes within cascades of boosted ensembles. In Intelligent Manufacturing and Energy Sustainability: Proceedings of ICIMES 2019 (pp. 809-825). Springer Singapore.

[48] Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., & Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. arXiv preprint arXiv:1312.3005.