Portmanteauing Features for Scene Text Recognition

Yew Lee Tan*, Ernest Yu Kai Chew* and Adams Wai-Kin Kong
School of Computer Science and Engineering
Nanyang Technological University (NTU)
Singapore

Jung-Jae Kim and Joo Hwee Lim
Institute for Infocomm Research
Agency for Science, Technology and Research
Singapore

Abstract—Scene text images have different shapes and are subjected to various distortions, e.g. perspective distortions. To handle these challenges, the state-of-the-art methods rely on a rectification network, which is connected to the text recognition network. They form a linear pipeline which uses text rectification on all input images, even for images that can be recognized without it. Undoubtedly, the rectification network improves the overall text recognition performance. However, in some cases, the rectification network generates unnecessary distortions on images, resulting in incorrect predictions in images that would have otherwise been correct without it. In order to alleviate the unnecessary distortions, the portmanteauing of features is proposed. The portmanteau feature, inspired by the portmanteau word, is a feature containing information from both the original text image and the rectified image. To generate the portmanteau feature, a non-linear input pipeline with a block matrix initialization is presented. In this work, the transformer is chosen as the recognition network due to its utilization of attention and inherent parallelism, which can effectively handle the portmanteau feature. The proposed method is examined on 6 benchmarks and compared with 13 state-of-the-art methods. The experimental results show that the proposed method outperforms the state-of-the-art methods on various of the benchmarks.

I. INTRODUCTION

Scene text recognition (STR) is the task of reading text from natural scenes. The semantic property of text in images embodies additional information that is useful for many practical applications such as intelligent document processing, street signs reading, and product recognition [1]. Because of these important applications, the field has drawn a great amount of attention from researchers and practitioners resulting in the emergence of various robust reading competitions [2]–[6]. STR is challenging because the images are acquired in a generally unstructured and uncontrolled setting. This, in turn, introduces more complexity and variability than traditional optical character recognition (OCR) tasks [1], [7], [8].

An approach in addressing such issues is to employ a rectification network to rectify perspective distorted and curved text before the text recognition process. Yang et al. [9] proposed the use of an auxiliary dense character detection scheme and an alignment loss to improve character localization in order to tackle the issues. In contrast, conceptually simpler methods which do not require explicit character detection are also introduced to address the problems [10], [11].

* these authors contributed equally to this work

Undoubtedly, the usage of rectification network has improved the overall performance of STR. However, the recognition network which comes after the rectification network may experience some form of penalty if, for example, suboptimal rectifications were being performed on regular text images. In other words, the recognition network will suffer from errors propagated from the rectification network, due to its linear input pipeline. An example is illustrated in Figure 1 where the rectification of “LASER” resulted in the loss of information and eventuated an incorrect prediction of “CASER” by model 2.

To mitigate the potential problems that may arise from using the rectification network, this study proposes the portmanteau feature which utilizes both the original and rectified images, together with the block matrix initialization, to form a non-linear pipeline. Like a portmanteau word whose meaning is derived from a blending of two or more morphemes (e.g. motel, brunch, smog), the portmanteau feature approach can better preserve the feature of the original image, which lightens the impact of unfavourable rectification, while incorporating the feature of the rectified image. This is shown in Figure 1 where the proposed approach (model 3) is able to generate the correct predictions in both cases as opposed to the model.

Fig. 1. The prediction results for multiple models from the experiments conducted. Model 1 uses the original image as input while model 2 utilizes a rectification module which rectifies the image before the recognition task. Model 3 uses both the rectified image as well as the original image as the inputs.
1 which failed to predict “OLDTOWN” and model 2 which failed to predict “LASER”.

Other than using a rectification network to solve the irregularities of text images, an attention model is also often employed [12], [13] due to its successes in both computer vision and natural language processing [14], [15]. The attention models can be used in tandem with a rectification network as it may be able to encode global context more effectively and hence, improve the overall performance of STR [11], [12], [16], [17].

Although the transformer model (which utilizes attention) [15] was conceptualized for natural language processing (NLP) tasks, it is becoming increasingly popular to be incorporated into vision related tasks due to its capabilities [12], [18], [19]. However, the span of the transformer attention is limited to one axis of direction as NLP tasks mainly involve a sequence of information (or words). As an image contains information in both directions, Dosovitskiy et al. [18] split up the image into patches and arranged them into one sequence as demonstrated in their vision transformer (ViT). However, such method will incur a large amount of computational overhead. Therefore, this work introduces the dual-axes vision transformer (DaViT) which is adapted from the Axial Transformer [20] in order to implicitly attend to features in two axes of direction.

The contribution of this work is as summarized: (1) portmanteau feature containing information from both original and rectified images, lightening the impact of unfavorable rectification, and (2) the proposed network surpresses current state-of-the-art methods.

The rest of the paper is organized as follows. Section II explores related works. Section III discusses the approach and methodology. Section IV reports the experimental results on six scene text benchmarks, and Section V concludes this study.

II. RELATED WORK

Irregular texts present major challenges to STR, and so, Shi et al. and various other works [10], [16], [21], [22] employed geometric transformation using thin plate splines (TPS) with spatial transformer network (STN). Subsequently, promising results from STN spurred several prominent works in text rectification. Luo et al. [11] developed a rectification method, called MORN, free from geometric constraints with the direct prediction of offset maps. Zhan and Lu [22] proposed a STN based rectification network by iteratively rectifying the images, with controls points derived from a polynomial curve and line segments. Yang et al. [23] proposed a rectification module that includes prediction of text center line as well as the various local attributes, all derived from the ground-truth bounding box labels.

Attention mechanism is often utilized in tandem with rectification network in order to enhance the overall performance in STR. Lee and Osindero [24] proposed the use of attention modeling which is conditioned upon the image feature from the CNN as well as the output of the RNN. Wang et al. [25] noticed that various STR works did not make full use of the alignment history and therefore, proposed the usage of a coverage vector to make adjustment for future attention.

Yu et al. [13] adopted the transformer architecture and introduced a parallel visual attention and a global semantic reasoning module for STR. The network employs a multi-way parallel transmission in order to make use of global semantic information more effectively. Bartz et al. [19] proposed the use of the vanilla transformer in their text recognition network which takes in regions of interest from their localization network.

III. APPROACH

The proposed network can be broken down into three stages as shown in Figure 2. The first stage is the portmanteau feature generation where an input image goes through two different reshaping schemes. One of the schemes reshapes the image into a fixed height with varying width which will be padded to produce a padded image, $I_p$. The other scheme reshapes the image to a fixed height and fixed width followed by a rectification network to produce a rectified image, $I_r$. $I_r$ and $I_p$ then form a concatenated image, $I_{con}$, which goes through a linear projection block and produces the portmanteau feature, $F_{port}$. The linear projection block utilizes the block matrix initialization which allows the portmanteau feature to retain more individual information from the features of the rectified and padded images.

The second stage is the feature extraction stage that learns representation of the portmanteau feature. A variant of the ViT and Axial Transformer, called the DaViT, is proposed as the basic building block for this stage. By employing the transformer, the network will be able to achieve parallelism in the processing of the portmanteau feature. Furthermore, the DaViT is computationally less intensive than the ViT and Axial Transformer. The feature that transverses through the second stage will also be denoted as $F_{port}$.

The final stage of the network is the sequence modelling stage which incorporates contextual cues from the sequence of embeddings, as well as the output from the encoder in order to predict the next character. The stage is made up of the vanilla transformer decoder and the softmax function is used to predict the next character.

A. Rectification

Two types of rectification networks are adopted in this study separately, STN and MORN [11]. The TPS of STN makes geometric adjustments to straighten text, while MORN specializes in correcting complex distortion without applying geometric constraints.

In this work, both STN and MORN used are pre-trained. The pre-trained STN follows ASTER STN [10], but instead of predicting the control points directly, it predicts a 4th order Legendre polynomial and 10 line segments to generate the control points, similar to ESIR [22]. Further details regarding the STN are available in the supplementary material. The

https://github.com/chewyukai/portmanteauing-features
SRNet framework used to train the STN, and the pre-trained MORN are available on GitHub.

B. Portmanteauing Features

A portmanteau word is a blending of two or more words such that the word expresses the combined meaning of its constituents. A portmanteau word, a concatenated connection is employed in the first stage of the framework in order to retain the information that exists in the padded image and the rectified image, as illustrated in Figure 2.

1) Concatenation: After the reshape and rectification, \( I_p \in \mathbb{R}^{H \times W \times 1} \) and \( I_r \in \mathbb{R}^{H \times W \times 1} \) are concatenated to form the image \( I_{con} = [I_p, I_r] \), where \( I_{con} \in \mathbb{R}^{H \times W \times 2} \) and \( H, W \) represent the height and width of the image. It is then split up into patches of \( P_h \times P_w \times 2 \) which is followed by a flattening, resulting in \( I_{con} \in \mathbb{R}^{N_x \times N_y \times 2P_wP_h} \), where \( N_x = \frac{H}{P_h} \) and \( N_y = \frac{W}{P_w} \). \( I_{con} \) will then enter a linear projection block (described in Section III-B3), which utilizes the block matrix initialization (described in Section III-B2), and be projected into a portmanteau feature, \( F_{port} \). In the experiments, \( P_w = 2 \) and \( P_h = 4 \).

2) Block Matrix Initialization: Through the utilization of the transformer’s intrinsic property (i.e. the multi-head attention module), the network is able to achieve parallelism in processing the portmanteau feature as if it is going through two separate transformers at the same time.

The block matrix initialization (BMI) scheme is proposed for the parameters in the linear layers in order to regulate the interaction between them. By initializing the parameters as such, it is assumed that the optimal solution exists within the premise of limited interaction and the network will be able to learn to facilitate the exchange of information with restraint.

The BMI enables \( I_{con} \) and \( F_{port} \) to start off as though \( I_{con} \) is going through two separate linear projection blocks followed by \( F_{port} = [F_p, F_r] \) going through two separate transformers. For an arbitrary input \( X = [X_1, X_2] \) where \( X_1, X_2 \in \mathbb{R}^{N \times D_{in}} \), and \( X \in \mathbb{R}^{N \times 2D_{in}} \). The transformation matrix in the linear layer, \( A \in \mathbb{R}^{2D_{in} \times 2D_{out}} \), can be broken down into a block matrix as such:

\[
A = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix},
A_{ij} \in \mathbb{R}^{D_{in} \times D_{out}}
\] (1)

By having \( A_{12} \) and \( A_{21} \) as zero matrices, the linear transformation on \( X \) is as though both \( X_1 \) and \( X_2 \) are going through their individual linear transformation with no interaction between them as follows:

\[
XA = [X_1A_{11} \quad X_2A_{22}]
\] (2)

The two off-diagonal block matrices in each of the linear layers of the encoder are initialized as zero matrices while the diagonal matrices are initialized with the Xavier initialization [26].

3) Linear Projection: The linear projection (LP) consists of two linear transformations as shown in Equation 3.

\[
LP(I_{con}) = (I_{con}A_1 + b_1)A_2 + b_2
\] (3)

The output of the LP is the portmanteau feature \( F_{port} = LP(I_{con}) \). For the experiments, the input dimension is \( 2P_wP_h = 16 \), the output dimension \( D_y \) which represents the dimensionality of the \( y \)-encoder layer is 96, and the inner dimension \( D_{LP} \) is 2048.

---

https://github.com/youdao-ai/SRNet
https://github.com/Canjie-Luo/MORAN_v2
C. Dual-Axes Vision Transformer Encoder

Scene text images can be densely-packed with characters. Hence, the size of image patches is required to be small enough such that the embedding of every character in the image can be learned. However, the reduction in image patch dimension increases the sequence length. As the complexity of the transformer self-attention layer is \(O(N^2 D)\) where \(D\) is the dimension of the representation, using such a method would be computationally intensive.

In order to solve the above-mentioned issue, an adaptation of the Axial Transformer [20], DaViT encoder is proposed. Instead of an explicit two-dimensional attention of the ViT, Axial Transformer alternates between vertical and horizontal attentions, thereby allowing the model to attend in two axes of direction. The range of patches that the attention can explicitly access will be more limited as compared to ViT for the lack of punctuation in scenes.

### Table I

| Attention Type | Complexity per Layer |
|----------------|-----------------------|
| ViT            | \(O(N_x^2 \cdot N_y \cdot D_x)\) |
| Axial Transformer | \(O(N_x^2 \cdot D_x \cdot N_y + N_y^2 \cdot D_x \cdot N_x)\) |
| DaViT          | \(O(N_x^2 \cdot D_x + N_y \cdot N_x \cdot D_y \cdot L_y)\) |

The transformer encoder layer that has an attention span across the x-axis is called the x-encoder layer while the one that has its attention span across the y-axis will be called the y-encoder layer as illustrated in the stage two of Figure 2. The y-encoder layer follows its respective architecture in the Axial Transformer where the \(N_x = \frac{W}{P_x}\) columns of features will go through the attention mechanism. The feature which transverses through the y-encoder layers will have the \(N_x \times N_y \times D_y\) will have the \(N_x\) technically treated as a “mini-batch” size with \(N_y = \frac{H}{P_y}\) representing the sequence length.

For the x-encoder layers, \(F_{port}\) will be reshaped into \(N_x \times D_x\) where the concatenation of \(D_y\) through \(N_y\) forms \(D_x\) \((D_x = D_y \times N_y)\) and \(N_x\) will represent the sequence length. This is in contrast with the Axial Transformer where the input to corresponding encoder layer would be of the dimension \(N_y \times N_x \times D_x\) and the features dimension is the same for both directions \((D_x = D_y)\). Specifically for each y-encoder layer, the complexity per layer for DaViT is \(N_x^2 \cdot D_y = N_y \cdot N_x \cdot D_x\). Conversely, the counterpart in Axial Transformer has a complexity per layer of \(N_y^2 \cdot D_x \cdot N_x\). For the x-encoder layer, the complexity per layer for DaViT is \(N_y^2 \cdot D_x \cdot N_x\) whereas the complexity for the Axial Transformer is \(N_y^2 \cdot D_x \cdot N_x\). In the experiments, the number of y-encoder layers \(L_y = 8\), the number of y-encoder layers \(L_y = 2\), \(N_y = 64\) and \(N_y = 8\).

1) Multi-Head Attention: Since DaViT performs attention sequentially across both the x and y directions, the model consists of both x and y multi-head attention with each having an attention spanning across the x-axis and y-axis respectively.

According to the transformer [15], the multi-head attention (MHA) is as follows:

\[
SDPA(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{D}}h\right)V
\]

\[
MHA(Q,K,V) = \text{Concat}(Att_0, ..., Att_{h-1})A_M
\]

where \(Att_i = SDPA(QA_i, KA_i, VA_i), i\) represents the index of the head, and the transformation matrices \(A_Q, A_K, A_V \in \mathbb{R}^{D \times \frac{D}{h}}\). For the y-encoder layers, \(h_y = 2\) and for the x-encoder layers, \(h_x = 16\).

2) Linear Block: The linear block (LB) in the network applies two linear layers to each position in the sequence identically, followed by a rectified linear unit (ReLU):

\[
LB(F_{port}) = \max(0, F_{port}A_1' + b_1')A_2' + b_2'
\]

where \(k \in \{x, y\}\), \(A_1' \in \mathbb{R}^{D_x \times D_{LH}}\), and \(A_2' \in \mathbb{R}^{D_{LH} \times D_x}\). X-linear blocks apply the linear transformations to the sequence along the x-direction whereas the y-linear blocks apply to the y-direction. For the x-linear blocks, the input and output are of the dimension \(D_x = 768\), and the inner-layer has the dimensionality of \(D_{LH} = 2048\). The input and output to the y-linear blocks have the dimensionality of \(D_y = 96\) with \(D_{LH} = 512\).

D. Transformer Decoder

All the building blocks in the decoder are similar to that of DaViT as can be seen on Figure 2. However the masked multi-head attention applies a mask which prevents the attention of subsequent positions (i.e. looking into the future). The decoder is accompanied by a softmax classifier which predicts the probability distribution over the character classes. For the experiments, the number of decoder layers, \(L_d = 4\).

IV. EXPERIMENTS

A. Implementation Details

The network was implemented using PyTorch and trained using the ADAM optimizer with a base learning rate of 0.02, betas of \((0.9, 0.98)\), and \(\epsilon = 10^{-9}\). The warmup duration was set to 6000 iterations with a decaying rate of \(\min(\text{steps}^{-0.5}, \text{steps} \times \text{warmup}^{-1.5})\). The model was trained on 8x NVIDIA RTX6000 24GB, with a batch size of 768 for 300,000 iterations.

The proposed network was trained with three synthetic datasets: MJSynth (MJ) [27] containing 9 million samples, SynthText (ST) [28] containing 8 million images, and SynthAdd (SA) [17] containing 1.6 million images used to compensate for the lack of punctuation in MJSynth and SynthText. 

1) Scene Text Benchmarks: The proposed network was evaluated on six scene text benchmarks. IIT 5K-Words (IITT) [29] contains 3000 test images. ICDAR 2013 (IC13) [4] contains 1015 images for testing where samples that contain non-alphanumeric characters, or have less than three characters were discarded. ICDAR 2015 (IC15) [3] contains 2077 test images. Street View Text (SVT) [30] consists of 647 testing images. Street View Text-Perspective (SVTP) [31] has 645 testing images. CUTE80 [32] (CT) contains 288 testing images.
TABLE II
TEXT RECOGNITION ACCURACY (IN PERCENTAGES) ON SIX SCENE TEXT BENCHMARKS. FOR EACH BENCHMARK, THE RESULT SHOWN IN **BOLD** REPRESENTS THE BEST PERFORMANCE, WHILE UNDERLINE REPRESENTS THE SECOND BEST. ST, MJ, AND SA DENOTE THE SYNTHETIC DATASETS OF SYNTHTEXT, MSYNTH, AND SYNTHADD RESPECTIVELY. R DENOTES THE USE OF REAL DATASETS IN TRAINING OR FINE-TUNING. **PORTMORN** AND **PORTSTN** REPRESENT PORTMANTEAU FEATURE GENERATION USING MORN AND STN, RESPECTIVELY.

| Method | Train Datasets | Regular Text | Irregular Text |
|--------|----------------|--------------|----------------|
|        |                | III | IC13 | SVT | IC15 | SVTP | CT | III | IC13 | SVT | IC15 | SVTP | CT | III | IC13 | SVT | IC15 | SVTP | CT | III | IC13 | SVT | IC15 | SVTP | CT |
|        |                | 3000 | 1015 | 647 | 2077 | 645 | 288 |
| Luo et al. [11] | ST+MJ | 91.2 | 92.4 | 88.3 | 68.8 | 76.1 | 77.4 |
| Yang et al. [23] | ST+MJ | 94.4 | 93.9 | 88.9 | 78.7 | 80.8 | 87.5 |
| Zhan and Lu [22] | ST+MJ | 93.3 | 91.3 | 90.2 | 76.9 | 79.6 | 83.3 |
| Wang et al. [33] | ST+MJ | 94.3 | 93.9 | 89.2 | 74.5 | 80.0 | 84.4 |
| Yu et al. [13] | ST+MJ | 94.8 | 95.5 | 91.5 | - | 85.1 | 87.8 |
| Zhang et al. [34] | ST+MJ | 94.7 | 94.2 | 90.9 | - | 81.7 | - |
| Qiao et al. [35] | ST+MJ | 93.8 | 92.8 | 89.6 | 80.0 | 81.4 | 83.6 |
| Bartz et al. [19] | ST+MJ+SA | 94.6 | 93.1 | 89.2 | 74.2 | 83.1 | **89.6** |
| **PortSTN**+DaViT | ST+MJ | 93.8 | 95.0 | **94.3** | 80.5 | **89.9** | **88.5** |
| **PortMORN**+DaViT | ST+MJ+SA | **95.3** | **96.0** | 91.0 | 79.9 | 87.8 | 83.3 |
| **PortSTN**+DaViT | ST+MJ+SA | 95.1 | 94.3 | 92.3 | **80.5** | 87.6 | 86.8 |
| Yue et al. [36] | ST+MJ+R | 95.4 | 94.1 | 89.3 | 79.2 | 82.9 | **92.4** |
| Wang et al. [37] | ST+MJ+R | **95.7** | **94.9** | 92.7 | - | 84.8 | 91.6 |
| **PortSTN**+DaViT | ST+MJ+R | 95.0 | 94.1 | **94.1** | **82.5** | **90.0** | 87.8 |
| Li et al. [17] | ST+MJ+SA+R | 95.0 | 94.0 | 91.2 | 78.8 | 86.4 | 89.6 |
| Lu et al. [12] | ST+MJ+SA+R | **95.3** | 90.6 | 90.6 | 79.6 | 84.5 | 87.5 |
| Hu et al. [16] | ST+MJ+SA+R | **95.8** | 94.4 | 92.9 | 79.5 | 85.7 | **92.2** |
| **PortSTN**+DaViT | ST+MJ+SA+R | 95.0 | 93.6 | 94.6 | **82.7** | 87.0 | 89.2 |

B. Experimental Results

The network recognizes 100 token classes, including 10 digits, 52 case sensitive letters, 35 punctuation characters, a start token, an end token, and a pad token. During evaluation, only 37 classes which include 36 case-insensitive alphanumeric characters and an end token were taken into consideration. The maximum output sequence length is 30 and greedy decoding was used. No rotation strategy [17] was used in the experiments.

The network was evaluated on six benchmarks and the comparison results are shown in Table II. The results show that the proposed network with MORN (PortMORN+DaViT) outperforms the original MORAN [11] significantly for all the test datasets. Moreover, both PortMORN+DaViT and PortSTN+DaViT which utilize a full Transformer encoder-decoder architecture outperforms KISS [19] and MASTER [12], where both networks are ResNet-Transformer hybrids.

The result demonstrates the benefit of the transformer encoder over the ResNet.

Furthermore, the proposed models achieved state-of-the-art result for the majority of the datasets with PortSTN+DaViT leading IC15 (containing 2077 images) by 0.9% and SVTP (containing 645 images) by 1.2%. Comparing the results of other works which included real datasets in the training or fine-tuning process, PortSTN+DaViT achieved 82.7% accuracy on IC15 and 94.6% on SVT, outperforming the next best results from other works by 2.7% and 1.7% respectively. Although there is a drop in the accuracy for SVTP when real datasets are included, the result of 87.0% still outperforms the next best result from other works by 0.6%.

C. Ablation Studies

1) Discussion on Portmaneau Feature: Two additional networks were trained without the portmaneau feature in order to investigate its impact as follows: (1) the STN+DaViT which
contains a rectification network, and (2) the DaViT with only \( I_p \) as the input. The two networks were trained with the same hyperparameters as Port\(_{STN}+\text{DaViT} \).

Table III shows the result of the experiments. To a certain extent, it demonstrates that the introduction of the STN causes an adverse impact to the text recognition capability of the model. Such effect is explicitly evident on the IC13 dataset as the accuracy dropped by 1.8\% when STN is added into DaViT although the average accuracy on the datasets increases.

Figure 4 shows some examples from the test datasets on the adverse effect of the STN on the predictions and the alleviation of the effect by portmanteauing of features. More examples are provided in the supplementary material.

Comparing Port\(_{STN}+\text{DaViT} \) with STN+DaViT and DaViT, where only one image was used, shows that the portmanteauing of feature generally improves the overall accuracy of the test datasets. The network, appears to be able to mitigate the adverse effect of STN as it outperforms STN+DaViT on four benchmarks including IC13 dataset. The incorporation of the features from \( I_p \) and \( I_r \) also allows the network to attain higher accuracy on SVT and IC15 than what could have been achieved individually. Some examples are shown in Figure 5 where the Port\(_{STN}+\text{DaViT} \) is able to attain correct predictions for the text images where both DaViT and STN+DaViT result in wrong predictions. More examples are provided in the supplementary material.

From Table III, it appears that BMI slightly improves the average accuracy of the test datasets. It is interesting to note that the parameters of the linear layers in Port\(_{STN}+\text{DaViT} \) with BMI is able to retain the characteristic of the block matrix initialization after the training as seen on Figure 3. Such patterns can be observed in all linear layers where the block matrix initialization was used. This suggests that a more optimal solution exists where the interaction between the features of the two images are regulated by the BMI.

Furthermore, results from Table III show that using the same reshaping scheme of unpadding for both images (i.e. method names denoted by “U”) results in worst overall performance. Although some datasets benefited from this, there is a drop of about 1\% in average accuracy across the 6 benchmarks as compared with their counterparts (without “U”).

V. CONCLUSION

In this paper, the adverse impact of rectification network was discussed. Although the rectification network improves overall text recognition accuracy, this study found that in some instances, rectification network can produce undesirable results. In order to address this issue, the portmanteauing of feature was proposed together with DaViT which can effectively handle the portmanteau feature. Experimental results show that the approach, to a certain extent, was able to lighten the adverse impact of rectification and also achieved the state-of-the-art results for most scene text recognition benchmarks.

VI. ACKNOWLEDGEMENT

This work is partially supported by NTU Internal Funding – Accelerating Creativity and Excellence (NTU–ACE2020-03).
REFERENCES

[1] S. Long, X. He, and C. Yao, “Scene text detection and recognition: The deep learning era,” International Journal of Computer Vision, vol. 129, no. 1, pp. 161–184, 2021.

[2] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto et al., “Icdar 2003 robust reading competitions: entries, results, and future directions,” International Journal of Document Analysis and Recognition, vol. 7, no. 2-3, pp. 105–122, 2005.

[3] D. Karatzas, L. Gomez-Bigorda, A. Nicoloau, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu et al., “Icdar 2015 competition on robust reading,” in Proceedings of the 13th International Conference on Document Analysis and Recognition, 2015, pp. 1156–1160.

[4] D. Karatzas, F. Sfaifat, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, “Icdar 2013 robust reading competition,” in Proceedings of the International Conference on Document Analysis and Recognition, 2013, pp. 208–212.

[5] R. Gomez, B. Shi, L. Gomez, L. Numann, A. Veit, J. Matas, S. Belongie, and D. Karatzas, “Icdar 2017 robust reading challenge on coco-text,” in Proceedings of the IAPR International Conference on Document Analysis and Recognition, 2017, pp. 98–102.

[6] C. Rigaud, D. Doucet, A. Antoine, M. Cousot, and J.-P. Moreux, “Icdar 2019 competition on post-ocr text correction,” in Proceedings of the International Conference on Document Analysis and Recognition, 2019, pp. 1588–1593.

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

[8] Y. Zhai, Y. Cong, and X. Bai, “Scene text detection and recognition: Recent advances and future trends,” Frontiers of Computer Science, vol. 10, no. 1, pp. 19–36, 2016.

[9] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, “Learning to read irregular text with attention mechanisms,” in Proceedings of the International Joint Conference on Artificial Intelligence, 2017, p. 3.

[10] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, “Aster: An attentional scene text recognizer with flexible rectification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 9, pp. 2035–2048, 2018.

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

[12] N. Lu, W. Yu, X. Qi, Y. Chen, P. Gong, and R. Xiao, “Master: Multi-aspect non-local network for scene text recognition,” arXiv preprint arXiv:1910.02256, 2019.

[13] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding, “Towards accurate scene text recognition with semantic reasoning networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2020.

[14] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6077–6086.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, vol. 30, 2017.

[16] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, “Gtc: Guided training of ctc towards efficient and accurate scene text recognition,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 2, pp. 11005–11012, 2020.

[17] H. Li, P. Wang, C. Shen, and G. Zhang, “Show, attend and read: A simple and strong baseline for irregular text recognition,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, July 2019, pp. 8610–8617.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Mostafa, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proceedings of the 9th International Conference on Learning Representations, 2021.

[19] C. Bartz, J. Bethge, H. Yang, and C. Meinel, “Kiss: Keeping it simple for scene text recognition,” arXiv preprint arXiv:1911.08400, 2019.

[20] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, “Axial attention in multidimensional transformers,” arXiv preprint arXiv:1912.12180, 2019.

[21] B. Shi, X. Bai, and C. Yao, “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298–2304, 2016.

[22] F. Zhan and S. Lu, “Eisr: End-to-end scene text recognition via iterative image rectification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2019.

[23] M. Yang, Y. Guan, M. Liao, X. He, K. Bian, S. Bai, C. Yao, and X. Bai, “Symmetry-constrained rectification network for scene text recognition,” in Proceedings of the IEEE International Conference on Computer Vision, October 2019.

[24] C.-Y. Lee and S. Ozerker, “Recursive recurrent nets with attention modeling for ocr in the wild,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2016.

[25] C. Wang, F. Yin, and C.-L. Liu, “Memory-augmented model for scene text recognition,” in Proceedings of the 16th International Conference on Frontiers in Handwriting Recognition, 2018, pp. 62–67.

[26] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, vol. 9, May 2010, pp. 249–256.

[27] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Synthetic data and artificial neural networks for natural scene text recognition,” arXiv preprint arXiv:1406.2227, 2014.

[28] A. Gupta, A. Vedaldi, and A. Zisserman, “Synthetic data for text localisation in natural images,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2315–2324.

[29] A. Mishra, K. Alahari, and C. Jawahar, “Scene text recognition using higher order language priors,” in Proceedings of the British Machine Vision Conference, 2012.

[30] K. Wang, B. Bahenen, and S. Belongie, “End-to-end scene text recognition,” in Proceedings of the International Conference on Computer Vision, 2011, pp. 1457–1464.

[31] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, “Recognizing text with perspective distortion in natural scenes,” in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 569–576.

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

[33] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai, “Decoupled attention network for text recognition,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, April 2020, pp. 12216–12224.

[34] H. Zhang, Q. Yao, M. Yang, Y. Xu, and X. Bai, “Autostr: Efficient backbone search for scene text recognition,” in Proceedings of the European Conference on Computer Vision, 2020, pp. 751–767.

[35] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, “Seed: Semantics enhanced encoder-decoder framework for scene text recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2020.

[36] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, “Robustscanner: Dynamically enhancing positional clues for robust text recognition,” in Proceedings of the European Conference on Computer Vision, Cham, 2020, pp. 135–151.

[37] Z. Wan, M. He, H. Chen, X. Bai, and C. Yao, “Textscanner: Reading characters in order for robust scene text recognition,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, April 2020, pp. 12 120–12 127.