Textual Alchemy: CoFormer for Scene Text Understanding

Gayatri Deshmukh
Independent Researcher
dgayatri9850@gmail.com

Onkar Susladkar
Independent Researcher
onkarsus13@gmail.com

Dhruv Makwana
Independent Researcher
dmakwana503@gmail.com

Sparsh Mittal
IIT Roorkee
sparsh.mittal@mfs.iitr.ac.in

Sai Chandra Teja R
Green PMU Semi Pvt Ltd
saichandrateja@gmail.com

Abstract

The paper presents CoFormer (Convolutional Fourier Transformer), a robust and adaptable transformer architecture designed for a range of scene text tasks. CoFormer integrates convolution and Fourier operations into the transformer architecture. Thus, it leverages convolution properties such as shared weights, local receptive fields, and spatial subsampling, while the Fourier operation emphasizes composite characteristics from the frequency domain. The research further proposes two new pretraining datasets, named Textverse10M-E and Textverse10M-H. Using these datasets, we demonstrate the efficacy of pretraining for scene text understanding. CoFormer achieves state-of-the-art results with and without pretraining on two downstream tasks: scene text recognition (STR) and scene text editing (STE). The paper further proposes LISTNet (Language Invariant Style Transfer), a novel framework for bi-lingual STE. It also introduces three datasets, viz., TST500K for STE, CSTR2.5M and Akshara550 for STR. The source-code of CoFormer is available at https://github.com/CandleLabAI/CoFormer-WACV-2024.

1. Introduction

Scene text, also known as text in a visual context, is ubiquitous in our daily lives and is found on signs, labels, advertisements, and more. An accurate understanding of scene text is crucial for numerous applications, e.g., image retrieval and organization, accessibility for visually impaired individuals, preserving and communicating information and autonomous decision-making in human-designed environments. Scene text understanding (STU) refers to interpreting the text present in scenes, such as images, videos, and real-world sceneries.

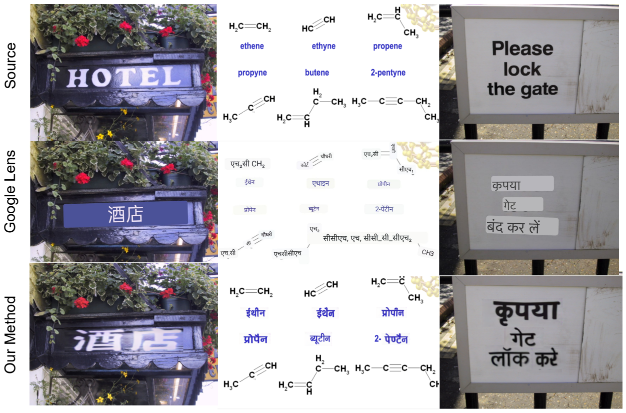


Figure 1. Sample results of LISTNet (our proposed network)

Scene text recognition (STR) and scene text editing (STE) are essential tasks in STU. STR involves recognizing and transcribing the text present in images and videos. With the increasing reliance on textual cues for interpretation in various domains, STR has become a crucial aspect of information retrieval and human-machine interaction. STR can help automate tasks such as document scanning, translation and text-to-speech conversion. STE complements text understanding by facilitating the generation of diverse text samples with different styles. It involves analyzing style features, training models on diverse style variations, and generating text in specific styles. By manipulating the appearance of text in a scene, such as changing its font, style, and background, STE provides valuable insights into how text styles impact information perception. STR and STE tasks are crucial for creating visually appealing designs and graphics in fields such as marketing and advertising.

Recent years have seen the introduction of novel deep-learning techniques for these tasks. Some works have proposed two-stage networks [24] for detecting and recognizing words. Some works have used "convolutional neural networks" (CNNs), and "recurrent neural networks" (RNNs) for scene text tasks [20, 39, 54]. A few works em-

Sparsh is the corresponding author. Dhruv, Gayatri and Onkar contributed to this paper while working as an intern at IIT Roorkee.

ploy attention mechanisms to selectively focus on relevant regions in an image [3, 16, 38, 39, 57], while others have used semantic segmentation [19]. Still, several challenges remain. Real-world scenes show noise, occlusions, and variability in style, font and orientation of the text. These factors make it difficult for deep learning models to generalize well to new data, leading to poor recognition accuracy. Transformer-based models [1, 33] address some of these challenges but incur huge memory and computation overheads. Further, the transformer lacks inductive bias and fails to focus on local features. In the field of pretraining, there has been only limited emphasis on generalizing the model to identify and process text in complex and cluttered environments. Also, the existing STU datasets are limited in terms of the number of images, complexities, and diversity of text styles, fonts and backgrounds.

Contributions: To address the aforementioned challenges, we propose (1) CoFormer, a backbone for diverse scene text tasks (2) LISTNet, a network for STE and (3) five novel datasets. Our major contributions are:

A. CoFormer brings together the best of convolution, self-attention and Fourier transformation (Sec. 3). Unlike the vision-transformer (ViT) [6], which takes image patches as input, CoFormer accepts the entire RGB image as input and passes it through C-block (convolution+GeLU+BatchNormalization). This simplifies network architecture and enables CoFormer to easily accommodate different resolutions of input images, which are crucial to many vision jobs. CoFormer includes a "multi-headed channel attention" (MCA) module that operates in the frequency domain and helps draw attention to critical regions. MCA identifies frequencies corresponding to background, boundaries and structures. On top of this, the convolution operation present in CoFormer specifically targets these frequencies, resulting in improved model performance. Overall, convolution processes spatial local characteristics, while Fourier analyzes global features. CoFormer has 48M parameters and performs 3.2G FLOPs.

B. We present a novel network named LISTNet (language invariant style transfer) for STE (Sec. 5.2). It consists of two stages: the first stage uses a CoFormer encoder to create the background and intermediate style-transferred text image. The second stage combines them to create the final image with style-transferred text and background. A novel decoder block architecture shows the benefits of applying the Fourier transform followed by channel attention.

C. We introduce five new datasets (refer to supplementary). (1) Textverse10M-E (English) and (2) Textverse10M-H (Hindi) are utilized for pre-training on STU tasks (Sec. 4). (3) Text Style Transfer-500K (TST500K), which has 100K images for STE between each of the following five language pairs: English-Hindi, Hindi-English, English-Tamil, English-Chinese and English-Bengali. (4) Complex Scene Text Recognition-2.5M (CSTR2.5M) for text recognition. (5) Akshara550 is a real-life dataset of 557 images for STR.

D. We evaluate CoFormer for two downstream tasks: STR and STE (Sec. 5). We show that (1) our proposed pretraining datasets improve the performance of both our technique and the previous techniques on the downstream tasks. (2) CoFormer is the best on nearly all STR datasets (Sec. 6-Sec. 7). For example, on the COCO-Text dataset, without and with pretraining, CoFormer achieves a word accuracy of 64.09% and 82.96%, respectively. For STE, LISTNet is the best on all language pairs (e.g., English-Tamil) of the TST500K dataset. For instance, on English-Hindi STE, LISTNet achieves a PSNR of 34.98 and an SSIM of 0.9099.

E. While CoFormer design is inspired by scene text characteristics, the architecture itself is not limited to this domain. For image classification on ImageNet-22K [36] dataset, CoFormer achieves a top-1 accuracy of 85.72%, surpassing CNN models, e.g., ConvNeXt-T [22] (82.9%) and ConvNeXt-S (84.6%), and transformer models, e.g., Swin-B [21] (85.2%). Although ConvNeXt-L (86.6%) and Swin-L (86.3%) yield higher accuracy than CoFormer, they require more complexity and twice the FLOPs. Detailed results are presented in the supplementary.

Application of our proposed model: Generating engaging and readable content is essential to content marketing strategies, particularly in social media, advertising, and blogging. However, creating content that resonates with diverse audiences with varying preferences, languages, and cultural backgrounds can be challenging. For example, a tagline written in the Chinese language may be inaccessible to those who do not understand Chinese. Visual translation methods such as Google Lens only translate the text content without copying the style, background, and font (Fig. 1). Due to this, they leave a patchy effect. As a result, the impact and appeal that the marketer intends to deliver through the tagline's style and appearance may be lost.

This is where STE comes into play, which enables mimicry of the original image's style and appearance and replacing the text with the target text. As shown in Fig. 1, LISTNet, our proposed STE model, can effectively replace scene text on images with target text. It adapts the original text style without leaving any patchy artifacts on the image. LISTNet is a powerful tool to get stylistically appropriate and highly appealing visually translated results for scene text. By using it in conjunction with a visual translation method like Google Lens, its application scope can be increased further. It can be used to translate PowerPoint slides, social media memes and taglines of brands, where the appearance and style of text are crucially important.

2. Related Work

Text recognition: Recent advancements [42, 58, 60] have significantly improved word-level STR. Some meth-

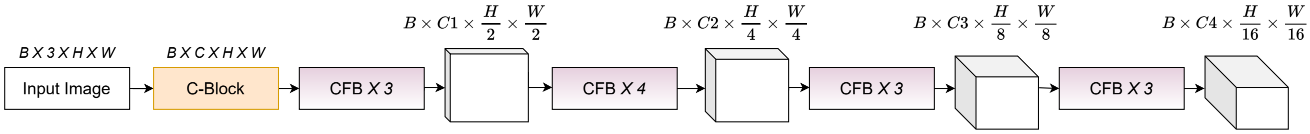

B X 3 X H X W
B X C X H X W
$B \times C1 \times \frac{H}{2} \times \frac{W}{2}$
$B \times C2 \times \frac{H}{4} \times \frac{W}{4}$
$B \times C3 \times \frac{H}{8} \times \frac{W}{8}$
$B \times C4 \times \frac{H}{16} \times \frac{W}{16}$
Input Image
C-Block
CFB X 3
CFB X 4
CFB X 3
CFB X 3

Figure 2. (a) CoFormer architecture. It has four stages. The number of CoFormer blocks (CFB) in each stage is shown inside each box.

ods [30, 56] use attention maps to focus on target text regions and improve recognition results. However, these models face challenges in preserving content and generating visually appealing results. Transformer-based approaches such as ViTSTR [8], and MaskOCR [25] seek to improve vision extraction capabilities. ABINet [7] uses a transformer-based language modality to improve vision model prediction. However, these models are computationally intensive and struggle with complex scene text images. They often produce unnatural results when applied to stylized text or mixed styles.

Scene Text Editing: Both SRNet [54] and SwapText [57] divide STE into three tasks: copying input text style to the target text, background inpainting, and fusing these two to produce the final text image. SRNet is a GAN-based model, and SwapText enhances SRNet by considering text geometry using spatial points. RewriteNet [17] uses an encoder to separate content and style features, followed by a generator that produces the image with desired content and style. AGGAN [47] uses attention mechanisms to guide the transfer process. Self-supervised methods such as TextStyleBrush [14] teach the networks to preserve both source style and target content using text perceptual loss functions. MOSTEL [31] proposes semi-supervised hybrid learning to train the model using both: labeled synthetic images and unpaired real-world images. STEFANN [35] is a font adaptive neural network that replaces individual characters in the source image with a target alphabet. This approach assumes per-character segmentation, which is impractical in many real-world images.

Transformers in computer vision: Recently introduced transformer-based models [21, 51, 52, 55] solely focus on patch relationships and drastically increase computational complexity. CvT [53] uses convolution-based projection within the transformer to capture the spatial structure and low-level details in image patches. CoFormer provides superior results than these networks (Sec. 7).

Fourier transform in computer vision: Tanick et al. [46] use Fourier features for capturing high-frequency characteristics with low dimensions, whereas LaMa [45] propose a Fourier-based image restoration network. GAFNet [44] uses "Fourier Feed Forward" block to learn the relationship between text and image modalities. FNet [18] replaces the self-attention module in the transformer encoder with a 2D FFT operation, which mixes tokens and makes them available to the feed-forward module. FrFNet [37] extends FNet by introducing fractional-order Fourier transform, which allows accessing any intermediate domain between time and frequency. CoFormer benefits from the properties of convolution, such as shared weights and focusing on local features. With the help of the Fourier transform, it learns the presence of specific frequencies, such as edges and textures, to detect objects or features in images. We hypothesize that a model with an inductive bias and rotational equivariance feature extractor towards edge and texture information will perform better on images with these features. For scene text images, these features are the edges and structure of the text. As shown in Fig. 4, CoFormer effectively interprets scene text images by focusing on the edges and structure.

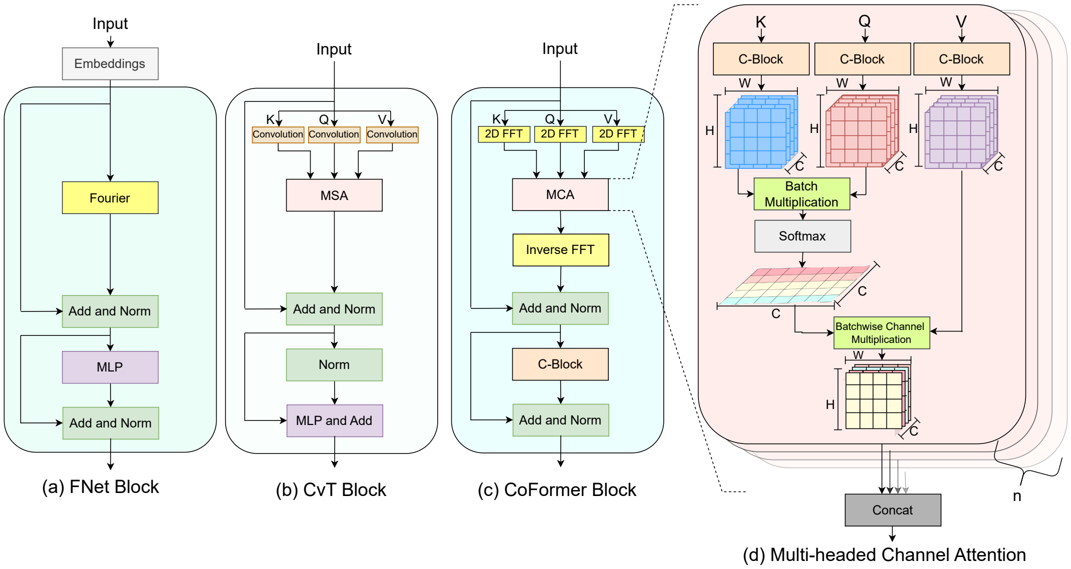
3. CoFormer: Our proposed architecture

This section covers CoFormer architecture (Sec. 3.1), its CoFormer block (Sec. 3.2), and the MCA module (Sec. 3.3).

3.1. Overall Architecture

Fig. 2 presents the CoFormer architecture, which extends the transformer design with convolution-based operations and uses Fourier transform as a foundation. We eliminate positional embedding and use C-Block at the input. The C-Block consists of a 3×3 convolution layer, GeLU activation [11], and a batch normalization (BN). Unlike ViT [6], which takes image patches as input, CoFormer accepts the entire RGB image and passes it through C-block. This not only simplifies the architectural design but also allows CoFormer to easily accommodate different resolutions of input images, which are crucial to many vision jobs.

The CoFormer architecture (Fig. 2), consists of four stages that generate feature maps of different sizes, with each stage comprising a fixed number of CoFormer blocks. The design of CoFormer block is shown in Fig. 3(c). The first stage involves three CoFormer blocks, resulting in a feature map of dimension $B \times C1 \times \frac{H}{2} \times \frac{W}{2}$. Here, B, C1, H, and W stand for batch size, number of channels, and height and width of the feature map, respectively. The output of



(a) FNet Block

(b) CvT Block

(c) CoFormer Block

(d) Multi-headed Channel Attention

Figure 3. (a) FNet [18] (b) CvT [53] (c) Proposed CoFormer block (CFB) (d) Proposed multi-headed channel attention (MCA).

each stage is passed to the next stage, and corresponding feature maps are produced. The feature map from the last stage can be utilized for downstream tasks by modifying the tail part of the model. For example, for STR, CoFormer can be used as an encoder/backbone to extract the image representations, and these representations can be given to the BLSTM (bi-directional LSTM) to recognize the text.

3.2. CoFormer block

Figures 3(a), (b), and (c) show the design of FNet, CvT and CoFormer, respectively. As for the input, FNet takes embeddings; CvT takes convolution projections, and CoFormer takes 2D FFT projections. FNet does not use attention at all; CvT uses an MSA [48] block, whereas CoFormer uses a multi-headed channel attention module. In the latter part of the block, both FNet and CvT use MLP, whereas CoFormer uses a C-block (convolution+GeLU+BN).

CoFormer block combines the advantage of convolution (viz., shared weights, local receptive fields, and spatial subsampling) and the Fourier transform. Given an input feature map, we first generate three copies of the input for the key, query, and value. Then, we perform the Fourier transform to transfer the values from the spatial domain to the frequency domain to obtain the constituent frequency components. Then, we project each of these components n times, where n shows the number of attention heads. These projections are then fed into the MCA. Here, Fourier transform determines the frequency present in the feature map and, thus, automatically differentiates between high-frequency and low-frequency features. This enables the model to draw attention to crucial aspects with the help of the MCA block.

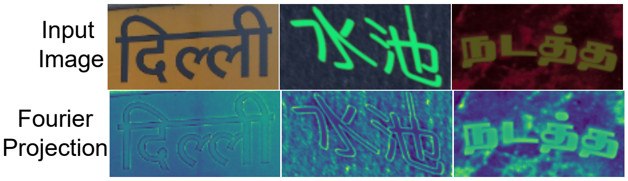


Figure 4. Fourier projection

Fig. 4 shows sample input images along with Fourier projections obtained by taking the output after four CoFormer blocks and projecting it along the input image. Here, the Fourier transform converts feature maps of input images to the frequency domain. Since the background, text boundaries and structure present in the image correspond to different frequencies in the frequency domain, CoFormer can clearly distinguish between them. This explains why CoFormer, which uses the Fourier transform, performs well in the scene text understanding tasks. On top of this, the convolution operation present in CoFormer specifically targets these frequencies, resulting in improved model performance. Here, convolution processes spatial local characteristics, while Fourier analyses the global features.

Our MCA block operates in the frequency domain. MCA helps in identifying and improving the essential characteristics. The feature vector acquired by MCA is transformed back into the spatial domain using the inverse Fourier transform. The hidden representation acquired after the inverse Fourier transform is fed into layer-normalization (LN), and the resulting feature map is added to the input of the CoFormer block. Finally, the feature vector is sent via C-block, LN, and residual connection.

3.3. Multi-headed Channel Attention (MCA)

While traditional MSA operates in the spatial domain, our proposed MCA operates in the frequency domain. MCA differentiates between distinct frequencies by encapsulating several intricate interactions that exist among them. The MCA has n self-attention heads (Fig. 3(d)). Each head i has its own weight matrices, which are learnable and denoted by K_i, Q_i, and V_i.

While traditional MSA uses linear layers, MCA uses C-blocks to make K, Q, and V matrices learnable and capture correlations across different frequencies. This provides two key advantages (1) The linear layer function is too simplistic to learn all frequency characteristics. In convolution, a sliding window convolves over the feature map and computes the weighted sum of the feature map's value within the window. This outputs the new feature map that reflects the local information within the window. In our case, the input of the MCA is the feature map in the frequency domain containing the frequency distribution. So, convolution helps to approximate these distributed frequencies and create efficient attention matrices allowing CoFormer to understand which frequencies to preserve or eliminate. (2) Convolution layers require fewer parameters than linear layers. We use multi-headed channel attention to capture various frequency relationships and improve performance. The outputs of all the heads in the MCA module are concatenated and sent to the next CoFormer block layer. More details about MCA are presented in the supplementary.

4. Pre-training

To understand the scene-text image, we adopt two pre-training tasks, STR and masked image modeling (MIM). We pre-train CoFormer for both Hindi and English to learn a universal representation of the two text scripts.

Fig. 5 depicts the pre-training architecture. We first randomly mask around 40% of the image regions using irregular patches and feed this masked image to the CoFormer. The concealed representation obtained from CoFormer is passed to both STR and MIM branches. In the STR branch, the feature map generated by the CoFormer block is fed into a BLSTM for STR. This branch employs the "connectionist temporal classification" (CTC) loss [9] to penalize text recognition errors. The MIM branch upsamples the CoFormer feature map to reconstitute the original image with missing patches. This branch uses L1 loss. Patch completion and text prediction assist the models in comprehending visual and textual representations.

5. Down-stream tasks

We showcase the use of CoFormer for scene text recognition and scene text editing.

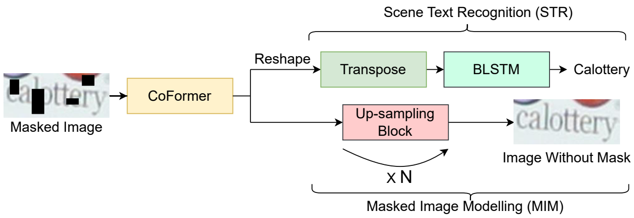


Figure 5. Pretrainng approach consisting of STR and MIM

5.1. Scene text recognition

As shown in Fig. 6, CoFormer serves as the backbone of the architecture, providing robust 2D feature representations for the input image. The output is derived from each stage of CoFormer and fed into an ASPP (atrous spatial pyramid pooling) block [4]. We use the ASPP block to get different levels of spatial information from CoFormer. ASPP output is provided to the transformer decoder (GPT) [32], which predicts the text present in the image in an auto-regressive manner. The auto-regressive version is shown as CoFormer-A. We use six blocks of GPT decoder to make the training more robust and accurate. We also trained our model with the normal CTC-based loss; this version is termed CoFormer-N. Although this version takes lower latency for training and inference than CoFormer-A, it provides lower accuracy (Table 1) due to its inability to properly align the text in the image.

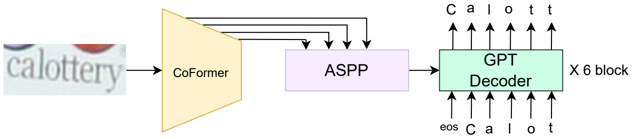


Figure 6. (Autoregressive) Network for scene text recognition task

5.2. LISTNet: Our proposed architecture for STE

We propose a novel STE architecture, named LISTNet, which is shown in Fig. 7(a). We explain it for the example of English-to-Hindi STE. It works in two stages. The first stage has a dual-stream setup, which utilizes CoFormer encoders to separately process the target text image (I_T) and input style image (I_{IS}). The intermediate style image (I_S) and background image (I_B) are produced by separate decoders in the first stage. The final style transferred image (I_{ST}) is then generated in the second stage by concatenating the outputs from the first stage decoders and feeding them into a single decoder.

Stage-1: The Stage-1 generator has two streams: Stream-ST for the target text image (I_T) and Stream-Bg for the style image (I_{IS}). These streams use CoFormer to exchange information and improve their understanding of the images. Stream-ST uses a CoFormer to replicate the style of I_{IS} to I_T, while Stream-Bg uses a CoFormer to predict the background of I_{IS}. Our model uses the query trans-

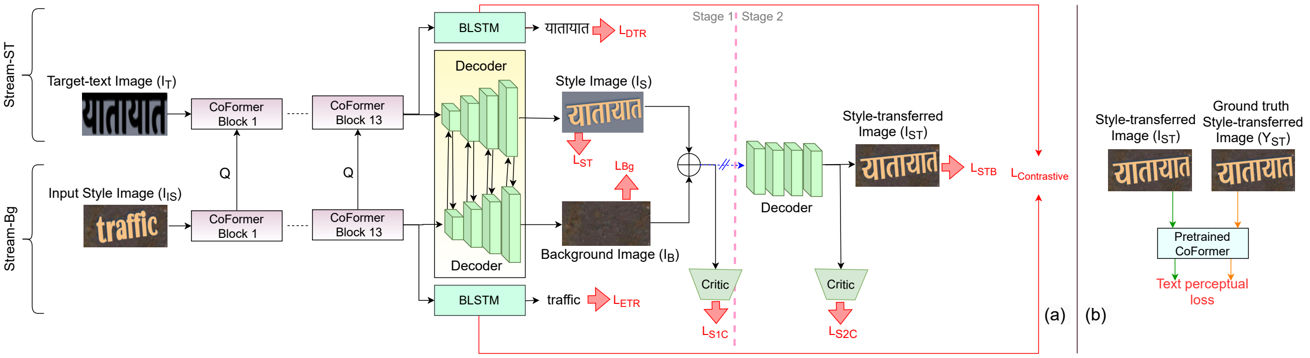


Figure 7. (a) LISTNet architecture for English-Hindi text editing (b) perceptual loss computation

mission method from ViLBERT [23] to incorporate cross-visual attention between I_{IS} and I_T. The CoFormer blocks in Stream-ST obtain hidden representations from Stream-Bg's CoFormer blocks and use these representations as a query in the MCA. This query transfer mechanism enables style-text-conditioned attention.

The outputs of CoFormers from both streams are fed into BLSTM and the decoder. Stream-Bg's BLSTM predicts English text on I_{IS}, and Stream-ST's BLSTM predicts Hindi text on I_T. Both BLSTM modules use CTC loss, denoted by L_{DTR} and L_{ETR} in Fig. 7(a). This helps the model recognize text on the image. Further, a contrastive loss ($L_{Contrastive}$) is calculated between their outputs to grasp the context of Hindi and English text.

Our novel decoder (refer to supplementary) for both streams uses cross Fourier attention, enabling information flow between streams through lateral connections. Fig. 7(a) shows decoder losses (L_{ST} and L_{BG}) for style and background images (I_S and I_B). These losses allow models to evaluate diverse but complementary aspects using L1 loss and SSIM loss. The L1 loss quantifies pixel-level differences, while SSIM loss accounts for structural information like edges, color contrasts, and brightness variations. The Stage-1 critic module receives a stacked output from both decoders. The critic uses a WGAN loss with a gradient penalty (L_{S1C}) to keep training constant. At last, Stage-2 receives the stacked I_S and I_B as input. The gradients are obstructed during the weight update of Stage-2. This is shown by the blue arrow in Fig. 7(a).

Stage-2: In Stage-2, the background texture (I_B) and style image (I_S) are fused to generate the final style transferred image. The decoder of Stage-2 (refer to supplementary) uses perceptual loss, L1 loss, and SSIM loss. As shown in Fig. 7(b), the perceptual loss is the L1 distance between the hidden representation of the ground truth style-transferred image (Y_{ST}) and the predicted style-transferred image (I_{ST}). Both these images are derived from a pre-trained CoFormer model trained on Hindi scene text data. Similar to Stage-1, the Critic module present in Stage-2 performs GAN-based training. The whole network is end-to-end trained. The critic modules of both stages have the same structure. They are made of five C-blocks.

6. Experimental results

The details on the dataset and implementation are provided in the supplementary material.

6.1. Results on Scene Text Recognition

Tab. 1 presents STR results on ten English datasets. The first half of the table reports the outcomes without any pretraining, while the second half reports results after pre-training on the Textverse10M-E dataset. In the former case, we utilize the results of previous methods as reported in [3]. Our proposed CoFormer model outperforms all previous techniques for both synthetic and real training datasets, achieving state-of-the-art (SOTA) results. Pretraining the models on the Textverse10M-E dataset substantially increases the word accuracy of every model, demonstrating the effectiveness of our proposed pretraining dataset and the pretraining method in improving the understanding of scene text. Moreover, CoFormer consistently achieves competitive results on all the benchmark datasets, with and without pre-training and for both synthetic and real training data. This indicates the effectiveness of our convolution-based Fourier projection model in identifying scene text.

Tab. 2 displays STR results on Hindi datasets. The IIIT-ILST [26] and MLT-Devnagari [28] datasets contain synthetic images, while our proposed Akshara550 dataset contains real images. CoFormer outperforms all previous techniques on all three datasets. The accuracy is lower on the real-scene datasets since they are inherently more challenging. Clearly, CoFormer is effective for a range of applications involving STR in complex environments.

Table 1. Word accuracy results on benchmark datasets. Train datasets are divided into two categories: synthetic (S) and real (R). The synthetic datasets (S) include MJ and ST, while the real datasets (R) include Uber [61], ArT [5], COCO-Text [49], RCTW17 [40], LSVT [43], ReCTS [59], MLT19 [28], OpenVINO [15] and TextOCR [41]. Normal and autoregressive models are denoted by N and A, respectively.

Method	Data	IIIT5K [27]	SVT [50]	ICDAR 2013 [13]	ICDAR 2015 [12]	SVTP [29]	CUTE80 [34]	COCO-Text [49]	ArT [5]	UBER [61]	CSTR-2.5M
Without any pre-training											
CRNN [2]	S	91.20	85.70	90.90	70.80	73.50	78.70	49.30	57.30	33.10	57.65
VITSTR-S [1]	S	94.00	91.70	95.10	78.70	83.90	88.20	56.40	66.10	37.60	74.76
TRBA [2]	S	96.30	92.80	95.00	80.60	86.90	91.30	61.40	68.20	38.00	78.82
ABINET [8]	S	95.30	93.40	95.00	79.10	87.10	89.70	57.10	65.40	34.90	80.00
PARSeq-N [3]	S	95.70	92.60	95.50	81.40	87.90	91.40	60.20	69.10	39.90	83.92
PARSeq-A [3]	S	97.00	93.60	96.20	82.90	88.90	92.20	64.00	**70.70**	42.00	84.32
CoFormer-N	S	96.01	93.01	95.76	82.90	88.78	92.99	63.98	69.92	41.87	84.02
CoFormer-A	S	**97.20**	**93.66**	**96.32**	**82.91**	**89.05**	**93.33**	**64.09**	70.12	**42.12**	**85.61**
CRNN [2]	R	94.60	90.70	94.50	78.50	80.60	89.10	62.20	66.80	51.00	64.64
VITSTR-S [1]	R	98.10	95.80	97.70	87.10	91.40	96.10	74.10	81.10	78.20	86.62
TRBA [2]	R	98.60	97.00	97.60	88.70	93.70	97.70	77.50	82.50	81.20	88.02
ABINET [8]	R	98.60	97.80	97.80	88.50	93.90	97.70	76.40	81.20	71.50	87.76
PARSeq-N [3]	R	98.30	97.50	98.10	88.40	94.60	97.70	77.00	83.00	82.40	89.89
PARSeq-A [3]	R	99.10	97.90	98.40	89.60	95.70	98.30	79.80	84.50	84.50	90.09
CoFormer-N	R	98.03	97.09	98.12	89.82	96.00	98.21	79.87	84.40	83.92	89.92
CoFormer-A	R	**99.18**	**98.03**	**98.49**	**90.23**	**96.08**	**98.33**	**80.01**	**85.12**	**87.00**	**91.01**
With pre-training on Textverse10M-E											
CRNN [2]	S	93.29	87.88	92.36	73.47	75.57	80.92	53.21	60.01	36.78	60.12
VITSTR-S [1]	S	95.92	93.21	96.94	79.88	86.76	90.01	59.02	67.97	40.00	75.61
TRBA [2]	S	**98.32**	94.03	96.55	83.09	88.04	92.07	63.76	70.09	40.21	80.11
ABINET [8]	S	96.72	94.93	**97.92**	81.29	88.91	90.92	57.17	66.99	37.90	83.31
PARSeq-A [3]	S	98.21	94.40	97.71	84.93	89.91	95.31	66.29	73.31	44.78	87.72
CoFormer-N	S	96.67	94.98	96.60	84.98	90.41	94.75	65.93	72.99	44.09	86.92
CoFormer-A	S	98.04	**95.00**	97.88	**85.01**	**91.37**	**96.00**	**66.98**	**74.07**	**44.98**	**88.09**
CRNN [2]	R	95.21	92.87	95.06	80.88	82.34	91.19	64.75	69.19	53.08	68.19
VITSTR-S [1]	R	98.41	96.71	98.00	89.32	93.21	97.92	76.18	84.09	79.99	78.98
TRBA [2]	R	98.79	97.67	98.34	91.99	95.02	98.76	78.82	86.54	81.07	84.41
ABINET [8]	R	99.00	97.99	98.21	92.35	94.99	98.65	78.81	86.76	81.01	87.88
PARSeq-A [3]	R	**99.97**	98.78	**99.02**	94.04	98.02	**98.78**	81.21	**86.99**	87.32	93.32
CoFormer-N	R	98.39	98.79	98.98	93.91	98.02	98.31	80.98	85.48	87.13	93.00
CoFormer-A	R	99.89	**98.92**	99.00	**94.49**	**98.49**	98.49	**82.96**	86.57	**88.97**	**94.75**

Table 2. Word accuracy results on Hindi datasets (Training was done on Textverse10M-H dataset)

Method	IIIT-ILST [26]	MLT-Devnagari [28]	Akshara550
CRNN [2]	66.71	65.90	52.23
VITSTR-S [1]	72.30	67.82	58.98
TRBA [2]	74.88	68.29	60.03
ABINET [8]	76.42	69.98	62.39
PARSeq-A [3]	**77.98**	71.21	64.32
CoFormer-N	76.91	70.67	63.77
CoFormer-A	**77.98**	**72.09**	**64.99**

6.2. Results on Scene Text Editing

We now evaluate several STE models on the TST500k dataset and for English-English STE we have utilised dataset from SRNet. To ensure a fair comparison, we do not pre-train any of the networks since SRNet, RewriteNet, and SwapText networks do not utilize any backbone that can be pre-trained using pretraining tasks mentioned in Sec. 4. Tab. 3 shows the results. SRNet, which uses a modular GAN-based architecture, does not perform well, possibly due to the known limitations of GANs, such as instability during training and the issue of mode collapse. SwapText outperforms SRNet, as it accounts for the spatial transformations of text. However, Rewritenet has limitations, as it does not share information between the encoders responsible for generating style features and content features. Due to this, it gains only a limited contextual understanding of how the style was transferred to the text.

MOSTEL and TextStyleBrush outperform SRNet and SwapText by using stroke guidance maps to find explicit text regions to alter. Yet, they are unable to match the higher generational quality of our LISTNet. We observe that the image encoders from MOSTEL and TextstyleBrush cannot catch the scene text stroke-level features. Hence, both perform poorly on the Indian scene text datasets, where the font stroke is particularly important. The supplementary section provides empirical support for our claim, offering a visual breakup of how our LISTNet excels in capturing intricate stroke-level features compared to the other models. The LISTNet encoder learns the boundaries and edges quickly due to the use of Fourier transform and channel-wise paired self-attention. It performs the best on all the metrics. It uses CoFormer as a backbone, which enables the model to understand different frequency patterns corresponding to the style, background, or structure of the image. Also, the use of WGAN-GP [10] training method ensures stable training of the network and avoids mode collapse.

Table 3. Results on TST-500K. Values are shown as PSNR/SSIM/LPIPS (For PSNR/SSIM: higher is better. For LPIPS: lower is better)

	SRNet [54]	Swaptext [57]	RewriteNet [17]	MOSTEL [31]	TextStyleBrush [14]	LISTNet
English-Hindi	26.11/ 0.7943/ 0.57	31.09/ 0.8376/ 0.49	32.04/ 0.8667/ 0.36	32.98/ 0.8821/ 0.21	33.01/ 0.8800/ 0.20	**34.98/ 0.9099/ 0.19**
English-Bengali	25.67/ 0.7532/ 0.51	27.31/ 0.7787/ 0.42	31.02/ 0.8789/ 0.29	33.01/ 0.8901/ 0.20	33.98/ 0.9001/ 0.16	**34.56/ 0.9127/ 0.15**
English-Tamil	24.31/ 0.7631/ 0.52	27.01/ 0.7831/ 0.47	29.98/ 0.8567/ 0.33	32.01/ 0.8892/ 0.19	32.96/ 0.8900/ 0.19	**33.98/ 0.9012/ 0.18**
English-Chinese	23.09/ 0.7402/ 0.49	27.07/ 0.7978/ 0.44	28.98/ 0.8645/ 0.30	30.12/ 0.8909/ 0.18	31.88/ 0.8992/ 0.18	**32.99/ 0.9147/ 0.17**
Hindi-English	25.05/ 0.7500/ 0.58	27.96/ 0.7977/ 0.49	30.09/ 0.8812/ 0.35	31.87/ 0.9001/ **0.17**	32.09/ 0.9119/ 0.20	**33.97/ 0.9232/** 0.21
English-English	27.88/ 0.7881/ 0.52	24.32/ 0.7772/ 0.44	24.99/ 0.7808/ 0.36	31.82/ 0.8871/ 0.18	33.00/ 0.9001/ 0.19	**34.00/ 0.9219/ 0.16**

Qualitative results: The visual results in Figure 8 support above findings. LISTNet excels at manipulating text under extreme geometric distortion. In contrast, MOSTEL and TextStyleBrush struggle to effectively transfer styles between different languages due to their inability to capture scene text stroke-level features, particularly in Chinese and Tamil. LISTNet stands out in this aspect by leveraging Fourier transform and attention mechanisms to accurately identify text structures, resulting in impressive style and background generation. This is especially noticeable for complex text symbols found in languages like Chinese, Tamil, and Bengali. In English-to-English style transfer, while MOSTEL and TextStyleBrush successfully transfer style, LISTNet outperforms both by generating high-quality results. The images produced by LISTNet also exhibit greater clarity and sharpness.

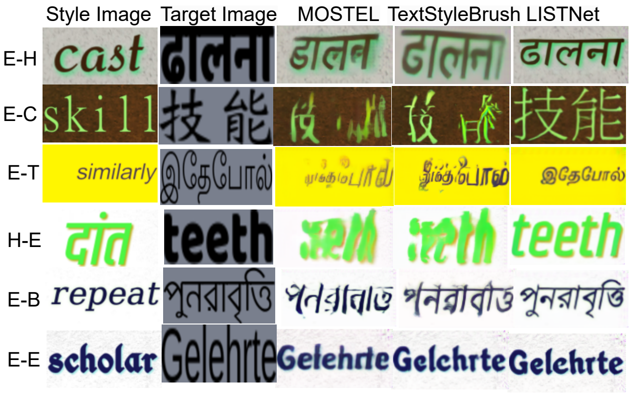


Figure 8. Sample STE results. E, H, C, T, and B denote English, Hindi, Chinese, Tamil, and Bengali, respectively. “E-H” signifies English to Hindi scene text editing, and so on.

7. Ablation Study

Tab. 4 shows the ablation results of LISTNet on the English-Hindi set of the TST500K dataset. The results of a similar ablation study for the STR task have been presented in the supplementary section.

Table 4. STE ablation results.

Experiments	PSNR	SSIM	LPIPS
0. Default LISTNet (without pretraining)	34.98	0.9099	0.19
1. LISTNet with pretrainning	38.91	0.9300	0.16
2. Without Fourier	32.91	0.8862	0.21
3. Without Fourier and without pretraining	31.98	0.8991	0.23
4. Without contrastive loss	33.19	0.8901	0.20
5. Replacing MCA with MSA	31.98	0.8692	0.24
6. Different training methodologies (Default uses WGAN-GP)			
6a. Pix2Pix	33.79	0.9001	0.18
6b. WGAN	34.02	0.9037	0.19
7. Different model size (Default uses 13 CoFormer blocks)			
7a. LISTNet (With 4 CoFormer Blocks)	32.98	0.8901	0.21
7b. LISTNet (With 8 CoFormer Blocks)	33.46	0.8957	0.20
8. Different backbones (Default uses CoFormer)			
8a. ViT	31.34	0.8992	0.20
8b. Swin Transformer	32.97	0.9011	0.19
8c. PVT-L	32.75	0.9005	0.19
8d. CvT-13	33.42	0.9087	0.18

1. We examine the effect of pre-training by incorporating pre-trained CoFormers into LISTNet. Evidently, pre-training improves model performance on both metrics. **2.** We assess the contribution of Fourier projections in CoFormer by removing 2D FFT and inverse FFT layers from the CFB block. The results indicate that model performance degrades without Fourier projections. **3.** Removal of both FFT and pretraining from the model degrades the quality metrics. Our results demonstrate that FFT improves model performance by highlighting frequency domain composite features, while pre-training enhances the model’s ability to understand scene text images.

4. The model performance degrades on training the LISTNet model without using contrastive loss between the output of two BLSTM modules. This shows that the contrastive loss helps the model understand the context of both languages. **5.** We replace MCA with MSA while maintaining the overall design of CoFormer. This reduces both the metrics. **6.** WGAN-GP-based training methodology provides stable training and superior performance than other GAN-based methodologies. **7.** Our default LISTNet utilizes 13 CoFormer blocks. We further test with LISTNet versions having 4 and 8 CoFormer blocks. Expectedly, these versions provide inferior results than the default LISTNet. Interestingly, they do provide better results than the previous techniques (refer Tab. 3). This shows the efficacy of our technique. **8.** We evaluate LISTNet with a different transformer backbone than CoFormer. Clearly, CoFormer is superior to previous transformer backbones.

Conclusion: We introduce CoFormer, an innovative vision transformer architecture that leverages the strengths of Fourier and convolution operations. Experimental results confirm superiority of CoFormer. Our future work will establish CoFormer’s efficacy on downstream tasks, e.g., text super-resolution, text erasure, and scene text segmentation.

References

[1] Rowel Atienza. Vision transformer for fast and efficient scene text recognition. In *International Conference on Document Analysis and Recognition*, pages 319–334. Springer, 2021. 2, 7

[2] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3113–3122, 2021. 7

[3] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. *arXiv preprint arXiv:2207.06966*, 2022. 2, 6, 7

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 5

[5] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1571–1576. IEEE, 2019. 7

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[7] Shancheng Fang, Zhendong Mao, Hongtao Xie, Yuxin Wang, Chenggang Yan, and Yongdong Zhang. Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[8] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7098–7107, 2021. 3, 7

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. 5

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 7

[11] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[12] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 7

[13] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013. 7

[14] Praveen Krishnan, Rama Kovvuri, Guan Pang, Boris Vassilev, and Tal Hassner. Textstylebrush: transfer of text aesthetics from a single example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3, 8

[15] Ilya Krylov, Sergei Nosov, and Vladislav Sovrasov. Open images v5 text annotation and yet another mask text spotter. In *Asian Conference on Machine Learning*, pages 379–389. PMLR, 2021. 7

[16] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2231–2239, 2016. 2

[17] Junyeop Lee, Yoonsik Kim, Seonghyeon Kim, Moonbin Yim, Seung Shin, Gayoung Lee, and Sungrae Park. RewriteNet: Realistic Scene Text Image Generation via Editing Text in Real-world Image. *arXiv preprint arXiv:2107.11041*, 2021. 3, 8

[18] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021. 3, 4

[19] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020. 2

[20] Yang Liu, Zhaowen Wang, Hailin Jin, and Ian Wassell. Synthetically supervised feature learning for scene text recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451, 2018. 1

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 3

[22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 2

[23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 6

[24] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. 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)*, pages 67–83, 2018. 1

[25] Pengyuan Lyu, Chengquan Zhang, Shanshan Liu, Meina Qiao, Yangliu Xu, Liang Wu, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Maskocr: Text recognition with masked encoder-decoder pretraining. *arXiv preprint arXiv:2206.00311*, 2022. 3

[26] M. Mathew, M. Jain, and C. V. Jawahar. Benchmarking scene text recognition in devanagari, telugu and malayalam. In *ICDAR MOCR Workshop*, 2017. 6, 7

[27] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC-British machine vision conference*. BMVA, 2012. 7

[28] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 1582–1587. IEEE, 2019. 6, 7

[29] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 569–576, 2013. 7

[30] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13528–13537, 2020. 3

[31] Yadong Qu, Qingfeng Tan, Hongtao Xie, Jianjun Xu, Yuxin Wang, and Yongdong Zhang. Exploring stroke-level modifications for scene text editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2119–2127, 2023. 3, 8

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 5

[33] Zobeir Raisi, Mohamed A Naiel, Paul Fieguth, Steven Wardell, and John Zelek. 2d positional embedding-based transformer for scene text recognition. *Journal of Computational Vision and Imaging Systems*, 6(1):1–4, 2020. 2

[34] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014. 7

[35] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Umapada Pal. Stefann: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13228–13237, 2020. 3

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 2

[37] Furkan Şahinuç and Aykut Koç. Fractional fourier transform meets transformer encoder. *IEEE Signal Processing Letters*, 29:2258–2262, 2022. 3

[38] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176, 2016. 2

[39] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018. 1, 2

[40] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *2017 14th iapr international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1429–1434. IEEE, 2017. 7

[41] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8802–8812, 2021. 7

[42] Ron Slossberg, Oron Anschel, Amir Markovitz, Ron Litman, Aviad Aberdam, Shahar Tsiper, Shai Mazor, Jon Wu, and R Manmatha. On calibration of scene-text recognition models. *arXiv preprint arXiv:2012.12643*, 2020. 2

[43] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1557–1562. IEEE, 2019. 7

[44] Onkar Susladkar, Gayatri Deshmukh, Dhruv Makwana, Sparsh Mittal, R Teja, and Rekha Singhal. Gafnet: A global fourier self attention based novel network for multi-modal downstream tasks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5242–5251, 2023. 3

[45] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 3

[46] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 3

[47] Hao Tang, Dan Xu, Nicu Sebe, and Yan Yan. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 3

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[49] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 7

[50] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International conference on computer vision*, pages 1457–1464. IEEE, 2011. 7

[51] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 3

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 3

[53] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021. 3, 4

[54] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1500–1508, 2019. 1, 3, 8

[55] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 3

[56] Ruijie Yan, Liangrui Peng, Shanyu Xiao, and Gang Yao. Primitive representation learning for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2021. 3

[57] Qiangpeng Yang, Jun Huang, and Wei Lin. Swaptext: Image based texts transfer in scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14700–14709, 2020. 2, 3, 8

[58] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020. 2

[59] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 1577–1581. IEEE, 2019. 7

[60] Xinyun Zhang, Binwu Zhu, Xufeng Yao, Qi Sun, Ruiyu Li, and Bei Yu. Context-based contrastive learning for scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3353–3361, 2022. 2

[61] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *SUNw: Scene Understanding Workshop-CVPR*, volume 2017, page 5, 2017. 7