TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models

Minghao Li[1*], **Tengchao Lv**[2], **Jingye Chen**[2*], **Lei Cui**[2],
Yijuan Lu[2], **Dinei Florencio**[2], **Cha Zhang**[2], **Zhoujun Li**[1], **Furu Wei**[2]
[1]Beihang University
[2]Microsoft Corporation
{liminghao1630, lizj}@buaa.edu.cn
{tengchaolv, v-jingyechen, lecu, yijlu, dinei, chazhang, fuwei}@microsoft.com

Abstract

Text recognition is a long-standing research problem for document digitalization. Existing approaches are usually built based on CNN for image understanding and RNN for char-level text generation. In addition, another language model is usually needed to improve the overall accuracy as a post-processing step. In this paper, we propose an end-to-end text recognition approach with pre-trained image Transformer and text Transformer models, namely **TrOCR**, which leverages the Transformer architecture for both image understanding and wordpiece-level text generation. The TrOCR model is simple but effective, and can be pre-trained with large-scale synthetic data and fine-tuned with human-labeled datasets. Experiments show that the TrOCR model outperforms the current state-of-the-art models on the printed, handwritten and scene text recognition tasks. The TrOCR models and code are publicly available at https://aka.ms/trocr.

Introduction

Optical Character Recognition (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene photo or from subtitle text superimposed on an image. Typically, an OCR system includes two main modules: a text detection module and a text recognition module. Text detection aims to localize all text blocks within the text image, either at word-level or textline-level. The text detection task is usually considered as an object detection problem where conventional object detection models such as YoLOv5 and DBNet (Liao et al. 2019) can be applied. Meanwhile, text recognition aims to understand the text image content and transcribe the visual signals into natural language tokens. The text recognition task is usually framed as an encoder-decoder problem where existing methods leveraged CNN-based encoder for image understanding and RNN-based decoder for text generation. In this paper, we focus on the text recognition task for document images and leave text detection as the future work.

Recent progress in text recognition (Diaz et al. 2021) has witnessed the significant improvements by taking advantage of the Transformer (Vaswani et al. 2017) architectures. However, existing methods are still based on CNNs as the backbone, where the self-attention is built on top of CNN backbones as encoders to understand the text image. For decoders, Connectionist Temporal Classification (CTC) (Graves et al. 2006) is usually used compounded with an external language model on the character-level to improve the overall accuracy. Despite the great success achieved by the hybrid encoder/decoder method, there is still a lot of room to improve with pre-trained CV and NLP models: 1) the network parameters in existing methods are trained from scratch with synthetic/human-labeled datasets, leaving large-scale pre-trained models unexplored. 2) as image Transformers become more and more popular (Dosovitskiy et al. 2021; Touvron et al. 2021), especially the recent self-supervised image pre-training (Bao, Dong, and Wei 2021), it is straightforward to investigate whether pre-trained image Transformers can replace CNN backbones, meanwhile exploiting the pre-trained image Transformers to work together with the pre-trained text Transformers in a single framework on the text recognition task.

To this end, we propose **TrOCR**, an end-to-end Transformer-based OCR model for text recognition with pre-trained CV and NLP models, which is shown in Figure 1. Distinct from the existing text recognition models, TrOCR is a simple but effective model which does not use the CNN as the backbone. Instead, following (Dosovitskiy et al. 2021), it first resizes the input text image into 384×384 and then the image is split into a sequence of 16×16 patches which are used as the input to image Transformers. Standard Transformer architecture with the self-attention mechanism is leveraged on both encoder and decoder parts, where wordpiece units are generated as the recognized text from the input image. To effectively train the TrOCR model, the encoder can be initialized with pre-trained ViT-style models (Dosovitskiy et al. 2021; Touvron et al. 2021; Bao, Dong, and Wei 2021) while the decoder can be initialized with pre-trained BERT-style models (Devlin et al. 2019; Liu et al. 2019; Dong et al. 2019; Wang et al. 2020b), respectively. Therefore, the advantage of TrOCR is three-fold. First, TrOCR uses the pre-trained image Transformer and text Transformer models, which take advantages of large-scale unlabeled data for image understanding and language modeling, with no need for an external language model. Sec-

[*]Work done during internship at Microsoft Research Asia.

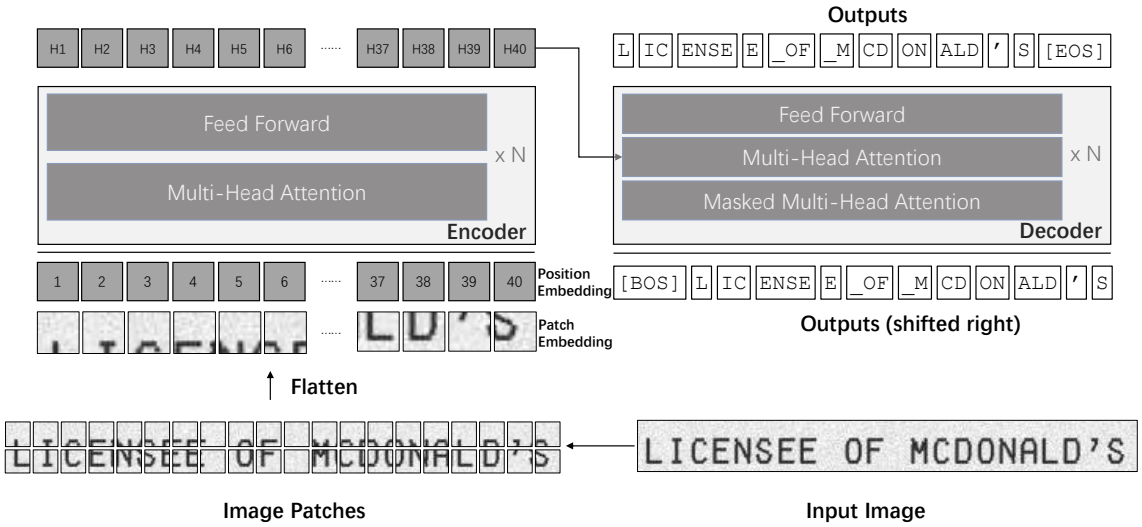

Figure 1: The architecture of TrOCR, where an encoder-decoder model is designed with a pre-trained image Transformer as the encoder and a pre-trained text Transformer as the decoder.

ond, TrOCR does not require any convolutional network for the backbone and does not introduce any image-specific inductive biases, which makes the model very easy to implement and maintain. Finally, experiment results on OCR benchmark datasets show that the TrOCR can achieve state-of-the-art results on printed, handwritten and scene text image datasets without any complex pre/post-processing steps. Furthermore, we can easily extend the TrOCR for multilingual text recognition with minimum efforts, where just leveraging multilingual pre-trained models in the decoder-side and expand the dictionary.

The contributions of this paper are summarized as follows:

1. We propose TrOCR, an end-to-end Transformer-based OCR model for text recognition with pre-trained CV and NLP models. To the best of our knowledge, this is the first work that jointly leverages pre-trained image and text Transformers for the text recognition task in OCR.
2. TrOCR achieves state-of-the-art results with a standard Transformer-based encoder-decoder model, which is convolution free and does not rely on any complex pre/post-processing steps.
3. The TrOCR models and code are publicly available at https://aka.ms/trocr.

TrOCR

Model Architecture

TrOCR is built up with the Transformer architecture, including an image Transformer for extracting the visual features and a text Transformer for language modeling. We adopt the vanilla Transformer encoder-decoder structure in TrOCR. The encoder is designed to obtain the representation of the image patches and the decoder is to generate the wordpiece sequence with the guidance of the visual features and previous predictions.

Encoder The encoder receives an input image $x_{\text{img}} \in \Re^{3\times H_0\times W_0}$, and resizes it to a fixed size (H, W). Since the Transformer encoder cannot process the raw images unless they are a sequence of input tokens, the encoder decomposes the input image into a batch of $N = HW/P^2$ foursquare patches with a fixed size of (P, P), while the width W and the height H of the resized image are guaranteed to be divisible by the patch size P. Subsequently, the patches are flattened into vectors and linearly projected to D-dimensional vectors, aka the patch embeddings. D is the hidden size of the Transformer through all of its layers.

Similar to ViT (Dosovitskiy et al. 2021) and DeiT (Touvron et al. 2021), we keep the special token "[CLS]" that is usually used for image classification tasks. The "[CLS]" token brings together all the information from all the patch embeddings and represents the whole image. Meanwhile, we also keep the distillation token in the input sequence when using the DeiT pre-trained models for encoder initialization, which allows the model to learn from the teacher model. The patch embeddings and two special tokens are given learnable 1D position embeddings according to their absolute positions.

Unlike the features extracted by the CNN-like network, the Transformer models have no image-specific inductive biases and process the image as a sequence of patches, which makes the model easier to pay different attention to either the whole image or the independent patches.

Decoder We use the original Transformer decoder for TrOCR. The standard Transformer decoder also has a stack of identical layers, which have similar structures to the layers in the encoder, except that the decoder inserts the "encoder-decoder attention" between the multi-head self-attention and feed-forward network to distribute different attention on the output of the encoder. In the encoder-decoder attention module, the keys and values come from the en-

coder output, while the queries come from the decoder input. In addition, the decoder leverages the attention masking in the self-attention to prevent itself from getting more information during training than prediction. Based on the fact that the output of the decoder will right shift one place from the input of the decoder, the attention mask needs to ensure the output for the position i can only pay attention to the previous output, which is the input on the positions less than i:

$$h_i = Proj(Emb(Token_i))$$
$$\sigma(h_{ij}) = \frac{e^{h_{ij}}}{\sum_{k=1}^{V} e^{h_{ik}}} \quad for\ j = 1, 2, \ldots, V$$

The hidden states from the decoder are projected by a linear layer from the model dimension to the dimension of the vocabulary size V, while the probabilities over the vocabulary are calculated on that by the softmax function. We use beam search to get the final output.

Model Initialization

Both the encoder and the decoder are initialized by the public models pre-trained on large-scale labeled and unlabeled datasets.

Encoder Initialization The DeiT (Touvron et al. 2021) and BEiT (Bao, Dong, and Wei 2021) models are used for the encoder initialization in the TrOCR models. DeiT trains the image Transformer with ImageNet (Deng et al. 2009) as the sole training set. The authors try different hyperparameters and data augmentation to make the model data-efficient. Moreover, they distill the knowledge of a strong image classifier to a distilled token in the initial embedding, which leads to a competitive result compared to the CNN-based models.

Referring to the Masked Language Model pre-training task, BEiT proposes the Masked Image Modeling task to pre-train the image Transformer. Each image will be converted to two views: image patches and visual tokens. They tokenize the original image into visual tokens by the latent codes of discrete VAE (Ramesh et al. 2021), randomly mask some image patches, and make the model recover the original visual tokens. The structure of BEiT is the same as the image Transformer and lacks the distilled token when compared with DeiT.

Decoder Initialization We use the RoBERTa (Liu et al. 2019) models and the MiniLM (Wang et al. 2020b) models to initialize the decoder. Generally, RoBERTa is a replication study of (Devlin et al. 2019) that carefully measures the impact of many key hyperparameters and training data size. Based on BERT, they remove the next sentence prediction objective and dynamically change the masking pattern of the Masked Language Model.

The MiniLM are compressed models of the large pre-trained Transformer models while retaining 99% performance. Instead of using the soft target probabilities of masked language modeling predictions or intermediate representations of the teacher models to guide the training of the student models in the previous work. The MiniLM models are trained by distilling the self-attention module of the last Transformer layer of the teacher models and introducing a teacher assistant to assist with the distillation.

When loading the above models to the decoders, the structures do not precisely match since both of them are only the encoder of the Transformer architecture. For example, the encoder-decoder attention layers are absent in these models. To address this, we initialize the decoders with the RoBERTa and MiniLM models by manually setting the corresponding parameter mapping, and the absent parameters are randomly initialized.

Task Pipeline

In this work, the pipeline of the text recognition task is that given the textline images, the model extracts the visual features and predicts the wordpiece tokens relying on the image and the context generated before. The sequence of ground truth tokens is followed by an "[EOS]" token, which indicates the end of a sentence. During training, we shift the sequence backward by one place and add the "[BOS]" token to the beginning indicating the start of generation. The shifted ground truth sequence is fed into the decoder, and the output of that is supervised by the original ground truth sequence with the cross-entropy loss. For inference, the decoder starts from the "[BOS]" token to predict the output iteratively while continuously taking the newly generated output as the next input.

Pre-training

We use the text recognition task for the pre-training phase, since this task can make the models learn the knowledge of both the visual feature extraction and the language model. The pre-training process is divided into two stages that differ by the used dataset. In the first stage, we synthesize a large-scale dataset consisting of hundreds of millions of printed textline images and pre-train the TrOCR models on that. In the second stage, we build two relatively small datasets corresponding to printed and handwritten downstream tasks, containing millions of textline images each. We use the existed and widely adopted synthetic scene text datasets for the scene text recognition task. Subsequently, we pre-train separate models on these task-specific datasets in the second stage, all initialized by the first-stage model.

Fine-tuning

Except for the experiments regarding scene text recognition, the pre-trained TrOCR models are fine-tuned on the downstream text recognition tasks. The outputs of the TrOCR models are based on Byte Pair Encoding (BPE) (Sennrich, Haddow, and Birch 2015) and SentencePiece (Kudo and Richardson 2018) and do not rely on any task-related vocabularies.

Data Augmentation

We leverage data augmentation to enhance the variety of the pre-training and fine-tuning data. Six kinds of image transformations plus keeping the original are taken for printed

Encoder	Decoder	Precision	Recall	F1
$\text{DeiT}_{\text{BASE}}$	$\text{RoBERTa}_{\text{BASE}}$	69.28	69.06	69.17
$\text{BEiT}_{\text{BASE}}$	$\text{RoBERTa}_{\text{BASE}}$	76.45	76.18	76.31
ResNet50	$\text{RoBERTa}_{\text{BASE}}$	66.74	67.29	67.02
$\text{DeiT}_{\text{BASE}}$	$\text{RoBERTa}_{\text{LARGE}}$	77.03	76.53	76.78
$\text{BEiT}_{\text{BASE}}$	$\text{RoBERTa}_{\text{LARGE}}$	**79.67**	**79.06**	**79.36**
ResNet50	$\text{RoBERTa}_{\text{LARGE}}$	72.54	71.13	71.83

Table 1: Ablation study on the SROIE dataset, where all the models are trained using the SROIE dataset only.

Model	Precision	Recall	F1
From Scratch	38.06	38.43	38.24
+ Pretrained Model	72.95	72.56	72.75
+ Data Augmentation	82.58	82.03	82.30
+ First-Stage Pretrain	95.31	95.65	95.48
+ Second-Stage Pretrain	**95.76**	**95.91**	**95.84**

Table 2: Ablation study of pretrained model initialization, data augmentation and two stages of pre-training on the SROIE dataset.

and handwritten datasets, which are random rotation (-10 to 10 degrees), Gaussian blurring, image dilation, image erosion, downscaling, and underlining. We randomly decide which image transformation to take with equal possibilities for each sample. For scene text datasets, RandAugment (Cubuk et al. 2020) is applied following (Atienza 2021), and the augmentation types include inversion, curving, blur, noise, distortion, rotation, etc.

Experiments

Data

Pre-training Dataset To build a large-scale high-quality dataset, we sample two million document pages from the publicly available PDF files on the Internet. Since the PDF files are digital-born, we can get pretty printed textline images by converting them into page images and extracting the textlines with their cropped images. In total, the first-stage pre-training dataset contains 684M textlines.

We use 5,427 handwritten fonts[1] to synthesize handwritten textline images by the TRDG[2], an open-source text recognition data generator. The text used for generation is crawled from random pages of Wikipedia. The handwritten dataset for the second-stage pre-training consists of 17.9M textlines, including IIIT-HWS dataset (Krishnan and Jawahar 2016). In addition, we collect around 53K receipt images in the real world and recognize the text on them by commercial OCR engines. According to the results, we crop the textlines by their coordinates and rectify them into normalized images. We also use TRDG to synthesize 1M printed textline images with two receipt fonts and the built-in printed fonts. In total, the printed dataset consists of 3.3M textlines. The second-stage pre-training data for the scene text recognition are MJSynth (MJ) (Jaderberg et al. 2014) and SynthText (ST) (Gupta, Vedaldi, and Zisserman 2016), totaling about 16M text images.

Benchmarks The SROIE (Scanned Receipts OCR and Information Extraction) dataset (Task 2) focuses on text recognition in receipt images. There are 626 receipt images and 361 receipt images in the training and test sets of SROIE. Since the text detection task is not included in this work, we use cropped images of the textlines for evaluation, which are obtained by cropping the whole receipt images according to the ground truth bounding boxes.

The IAM Handwriting Database is composed of handwritten English text, which is the most popular dataset for handwritten text recognition. We use the Aachen's partition of the dataset[3]: 6,161 lines from 747 forms in the train set, 966 lines from 115 forms in the validation set and 2,915 lines from 336 forms in the test set.

Recognizing scene text images is more challenging than printed text images, as many images in the wild suffer from blur, occlusion, or low-resolution problems. Here we leverage some widely-used benchmarks, including IIIT5K-3000 (Mishra, Alahari, and Jawahar 2012), SVT-647 (Wang, Babenko, and Belongie 2011), IC13-857, IC13-1015 (Karatzas et al. 2013), IC15-1811, IC15-2077 (Karatzas et al. 2015), SVTP-645 (Phan et al. 2013), and CT80-288 (Risnumawan et al. 2014) to evaluate the capacity of the proposed TrOCR.

Model	Recall	Precision	F1
CRNN	28.71	48.58	36.09
Tesseract OCR	57.50	51.93	54.57
H&H Lab	96.35	96.52	96.43
MSOLab	94.77	94.88	94.82
CLOVA OCR	94.3	94.88	94.59
$\text{TrOCR}_{\text{SMALL}}$	95.89	95.74	95.82
$\text{TrOCR}_{\text{BASE}}$	96.37	96.31	96.34
$\text{TrOCR}_{\text{LARGE}}$	96.59	96.57	**96.58**

Table 3: Evaluation results (word-level Precision, Recall, F1) on the SROIE dataset, where the baselines come from the SROIE leaderboard (https://rrc.cvc.uab.es/?ch=13&com=evaluation&task=2).

Settings

The TrOCR models are built upon the Fairseq (Ott et al. 2019) which is a popular sequence modeling toolkit. For the model initialization, the DeiT models are implemented and initialized by the code and the pre-trained models from the timm library (Wightman 2019) while the BEiT models and the MiniLM models are from the UniLM's official repository[4]. The RoBERTa models come from the corresponding page in the Fairseq GitHub repository. We use 32 V100 GPUs with the memory of 32GBs for pre-training and 8 V100 GPUs for fine-tuning. For all the models, the batch size is set to 2,048 and the learning rate is 5e-5. We use the

[1] The fonts are obtained from https://fonts.google.com/?category=Handwriting and https://www.1001fonts.com/handwritten-fonts.html.

[2] https://github.com/Belval/TextRecognitionDataGenerator

[3] https://github.com/jpuigcerver/Laia/tree/master/egs/iam

[4] https://github.com/microsoft/unilm

BPE and sentencepiece tokenizer from Fairseq to tokenize the textlines to wordpieces.

We employ the 384×384 resolution and 16×16 patch size for DeiT and BEiT encoders. The $\text{DeiT}_{\text{SMALL}}$ has 12 layers with 384 hidden sizes and 6 heads. Both the $\text{DeiT}_{\text{BASE}}$ and the $\text{BEiT}_{\text{BASE}}$ have 12 layers with 768 hidden sizes and 12 heads while the $\text{BEiT}_{\text{LARGE}}$ has 24 layers with 1024 hidden sizes and 16 heads. We use 6 layers, 256 hidden sizes and 8 attention heads for the small decoders, 512 hidden sizes for the base decoders and 12 layers, 1,024 hidden sizes and 16 heads for the large decoders. For this task, we only use the last half of all layers from the corresponding RoBERTa model, which are the last 6 layers for the $\text{RoBERTa}_{\text{BASE}}$ and the last 12 layers for the $\text{RoBERTa}_{\text{LARGE}}$. The beam size is set to 10 for TrOCR models.

We take the CRNN model (Shi, Bai, and Yao 2016) as the baseline model. The CRNN model is composed of convolutional layers for image feature extraction, recurrent layers for sequence modeling and the final frame label prediction, and a transcription layer to translate the frame predictions to the final label sequence. To address the character alignment issue, they use the CTC loss to train the CRNN model. For a long time, the CRNN model is the dominant paradigm for text recognition. We use the PyTorch implementation[5] and initialized the parameters by the provided pre-trained model.

Evaluation Metrics

The SROIE dataset is evaluated using the word-level precision, recall and f1 score. If repeated words appear in the ground truth, they are also supposed to appear in the prediction. The precision, recall and f1 score are described as:

$$Precision = \frac{\text{Correct matches}}{\text{The number of the detected words}}$$
$$Recall = \frac{\text{Correct matches}}{\text{The number of the ground truth words}}$$
$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}.$$

The IAM dataset is evaluated by the case-sensitive Character Error Rate (CER). The scene text datasets are evaluated by the Word Accuracy. For fair comparison, we filter the final output string to suit the popular 36-character charset (lowercase alphanumeric) in this task.

Results

Architecture Comparison We compare different combinations of the encoder and decoder to find the best settings. For encoders, we compare DeiT, BEiT and the ResNet-50 network. Both the DeiT and BEiT are the base models in their original papers. For decoders, we compare the base decoders initialized by $\text{RoBERTa}_{\text{BASE}}$ and the large decoders initialized by $\text{RoBERTa}_{\text{LARGE}}$. For further comparison, we also evaluate the CRNN baseline model and the Tesseract OCR in this section, while the latter is an open-source OCR Engine using the LSTM network.

[5] https://github.com/meijieru/crnn.pytorch

Table 1 shows the results of combined models. From the results, we observe that the BEiT encoders show the best performance among the three types of encoders while the best decoders are the $\text{RoBERTa}_{\text{LARGE}}$ decoders. Apparently, the pre-trained models on the vision task improve the performance of text recognition models, and the pure Transformer models are better than the CRNN models and the Tesseract on this task. According to the results, we mainly use three settings on the subsequent experiments: $\textbf{TrOCR}_{\textbf{SMALL}}$ (total parameters=62M) consists of the encoder of $\text{DeiT}_{\text{SMALL}}$ and the decoder of MiniLM, $\textbf{TrOCR}_{\textbf{BASE}}$ (total parameters=334M) consists of the encoder of $\text{BEiT}_{\text{BASE}}$ and the decoder of $\text{RoBERTa}_{\text{LARGE}}$, $\textbf{TrOCR}_{\textbf{LARGE}}$ (total parameters=558M) consists of the encoder of $\text{BEiT}_{\text{LARGE}}$ and the decoder of $\text{RoBERTa}_{\text{LARGE}}$. In Table 2, we have also done some ablation experiments to verify the effect of pre-trained model initialization, data augmentation, and two stages of pre-training. All of them have great improvements to the TrOCR models.

SROIE Task 2 Table 3 shows the results of the TrOCR models and the current SOTA methods on the leaderboard of the SROIE dataset. To capture the visual information, all of these baselines leverage CNN-based networks as the feature extractors while the TrOCR models use the image Transformer to embed the information from the image patches. For language modeling, MSO Lab (Sang and Cuong 2019) and CLOVA OCR (Sang and Cuong 2019) use LSTM layers and H&H Lab (Shi, Bai, and Yao 2016) use GRU layers while the TrOCR models use the Transformer decoder with a pure attention mechanism. According to the results, the TrOCR models outperform the existing SOTA models with pure Transformer structures. It is also confirmed that Transformer-based text recognition models get competitive performance compared to CNN-based networks in visual feature extraction and RNN-based networks in language modeling on this task without any complex pre/post-process steps.

IAM Handwriting Database Table 4 shows the results of the TrOCR models and the existing methods on the IAM Handwriting Database. According to the results, the methods with CTC decoders show good performance on this task and the external LM will result in a significant reduction in CER. By comparing the methods (Bluche and Messina 2017) with the TrOCR models, the $\text{TrOCR}_{\text{LARGE}}$ achieves a better result, which indicates that the Transformer decoder is more competitive than the CTC decoder in text recognition and has enough ability for language modeling instead of relying on an external LM. Most of the methods use sequence models in their encoders after the CNN-based backbone except the FCN encoders in (Wang et al. 2020a), which leads to a significant improvement on CER. Instead of relying on the features from the CNN-based backbone, the TrOCR models using the information from the image patches get similar and even better results, illustrating that the Transformer structures are competent to extract visual features well after pre-training. From the experiment results, the TrOCR models exceed all the methods which only use synthetic/IAM as the sole training set with pure Transformer structures and

Model	Architecture	Training Data	External LM	CER
(Bluche and Messina 2017)	GCRNN / CTC	Synthetic + IAM	Yes	3.2
(Michael et al. 2019)	LSTM/LSTM w/Attn	IAM	No	4.87
(Wang et al. 2020a)	FCN / GRU	IAM	No	6.4
(Kang et al. 2020)	Transformer w/ CNN	Synthetic + IAM	No	4.67
(Diaz et al. 2021)	S-Attn / CTC	Internal + IAM	No	3.53
(Diaz et al. 2021)	S-Attn / CTC	Internal + IAM	Yes	2.75
(Diaz et al. 2021)	Transformer w/ CNN	Internal + IAM	No	2.96
$\text{TrOCR}_{\text{SMALL}}$	Transformer	Synthetic + IAM	No	4.22
$\text{TrOCR}_{\text{BASE}}$	Transformer	Synthetic + IAM	No	3.42
$\text{TrOCR}_{\text{LARGE}}$	Transformer	Synthetic + IAM	No	2.89

Table 4: Evaluation results (CER) on the IAM Handwriting dataset.

Model	Parameters	Total Sentences	Total Tokens	Time	Speed #Sentences	Speed #Tokens
$\text{TrOCR}_{\text{SMALL}}$	62M	2,915	31,081	348.4s	8.37 sentences/s	89.22 tokens/s
$\text{TrOCR}_{\text{BASE}}$	334M	2,915	31,959	633.7s	4.60 sentences/s	50.43 tokens/s
$\text{TrOCR}_{\text{LARGE}}$	558M	2,915	31,966	666.8s	4.37 sentences/s	47.94 tokens/s

Table 5: Inference time on the IAM Handwriting dataset.

achieve a new state-of-the-art CER of 2.89. Without leveraging any extra human-labeled data, TrOCR even gets comparable results with the methods in (Diaz et al. 2021) using the additional internal human-labeled dataset.

Scene Text Datasets In Table 6, we compare the $\text{TrOCR}_{\text{BASE}}$ and $\text{TrOCR}_{\text{LARGE}}$ models of fine-tuning with synthetic data only and fine-tuning with synthetic data and benchmark datasets (the training sets of IC13, IC15, IIIT5K, SVT) to the popular and recent SOTA methods. Compared to all, the TrOCR models establish five new SOTA results of eight experiments while getting comparable results on the rest. Our model underperforms on the IIIT5K dataset, and we find some scene text sample images contain symbols, but the ground truth does not. It is inconsistent with the behavior in our pre-training data (retaining symbols in ground truth), causing the model to tend still to process symbols. There are two kinds of mistakes: outputting symbols but truncating the output in advance to ensure that the number of wordpieces is consistent with the ground truth, or identifying symbols as similar characters.

Inference Speed Table 5 shows the inference speed of different settings TrOCR models on the IAM Handwriting Database. We can conclude that there is no significant margin in inference speed between the base models and the large models. In contrast, the small model shows comparable results for printed and handwriting text recognition even though the number of parameters is an order of magnitude smaller and the inference speed is as twice as fast. The low number of parameters and high inference speed means fewer computational resources and user waiting time, making it more suitable for deployment in industrial applications.

Related Work

Scene Text Recognition

For text recognition, the most popular approaches are usually based on the CTC-based models. (Shi, Bai, and Yao 2016) proposed the standard CRNN, an end-to-end architecture combined by CNN and RNN. The convolutional layers are used to extract the visual features and convert them to sequence by concatenating the columns, while the recurrent layers predict the per-frame labels. They use a CTC decoding strategy to remove the repeated symbols and all the blanks from the labels to achieve the final prediction. (Su and Lu 2014) used the Histogram of Oriented Gradient (HOG) features extracted from the image patches in the same column of the input image, instead of the features from the CNN network. A BiLSTM is then trained for labeling the sequential data with the CTC technique to find the best match. (Gao et al. 2019) extracted the feature by the densely connected network incorporating the residual attention block and capture the contextual information and sequential dependency by the CNN network. They compute the probability distribution on the output of the CNN network instead of using an RNN network to model them. After that, CTC translates the probability distributions into the final label sequence.

The Sequence-to-Sequence models (Zhang et al. 2020b; Wang et al. 2019; Sheng, Chen, and Xu 2019; Bleeker and de Rijke 2019; Lee et al. 2020; Atienza 2021) are gradually attracting more attention, especially after the advent of the Transformer architecture (Vaswani et al. 2017). SaHAN (Zhang et al. 2020b), standing for the scale-aware hierarchical attention network, are proposed to address the character scale-variation issue. The authors use the FPN network and the CRNN models as the encoder as well as a hierarchical attention decoder to retain the multi-scale features. (Wang et al. 2019) extracted a sequence of visual features from the input images by the CNN with attention module and BiL-

Model	Test datasets and # of samples							
	IIIT5k	**SVT**	**IC13**		**IC15**		**SVTP**	**CUTE**
	3,000	647	857	1,015	1,811	2,077	645	288
PlugNet (Mou et al. 2020)	94.4	92.3	–	95.0	–	82.2	84.3	85.0
SRN (Yu et al. 2020)	94.8	91.5	95.5	–	82.7	–	85.1	87.8
RobustScanner (Yue et al. 2020)	95.4	89.3	–	94.1	–	79.2	82.9	92.4
TextScanner (Wan et al. 2020)	95.7	92.7	–	94.9	–	83.5	84.8	91.6
AutoSTR (Zhang et al. 2020a)	94.7	90.9	–	94.2	81.8	–	81.7	–
RCEED (Cui et al. 2021)	94.9	91.8	–	–	–	82.2	83.6	91.7
PREN2D (Yan et al. 2021)	95.6	94.0	96.4	–	83.0	–	87.6	91.7
VisionLAN (Wang et al. 2021)	95.8	91.7	95.7	–	83.7	–	86.0	88.5
Bhunia (Bhunia et al. 2021b)	95.2	92.2	–	95.5	–	84.0	85.7	89.7
CVAE-Feed.[1] (Bhunia et al. 2021a)	95.2	–	–	95.7	–	**84.6**	88.9	89.7
STN-CSTR (Cai, Sun, and Xiong 2021)	94.2	92.3	96.3	94.1	86.1	82.0	86.2	–
ViTSTR-B (Atienza 2021)	88.4	87.7	93.2	92.4	78.5	72.6	81.8	81.3
CRNN (Shi, Bai, and Yao 2016)	84.3	78.9	–	88.8	–	61.5	64.8	61.3
TRBA (Baek, Matsui, and Aizawa 2021)	92.1	88.9	–	93.1	–	74.7	79.5	78.2
ABINet (Fang et al. 2021)	96.2	93.5	97.4	–	86.0	–	89.3	89.2
Diaz (Diaz et al. 2021)	96.8	94.6	96.0	–	80.4	–	–	–
PARSeq$_A$ (Bautista and Atienza 2022)	**97.0**	93.6	97.0	96.2	86.5	82.9	88.9	92.2
MaskOCR (ViT-B) (Lyu et al. 2022)	95.8	94.7	98.1	-	87.3	-	89.9	89.2
MaskOCR (ViT-L) (Lyu et al. 2022)	96.5	94.1	97.8	-	**88.7**	-	90.2	92.7
TrOCR$_{\mathrm{BASE}}$ (Syn)	90.1	91.0	97.3	96.3	81.1	75.0	90.7	86.8
TrOCR$_{\mathrm{LARGE}}$ (Syn)	91.0	93.2	98.3	97.0	84.0	78.0	91.0	89.6
TrOCR$_{\mathrm{BASE}}$ (Syn+Benchmark)	93.4	95.2	98.4	**97.4**	86.9	81.2	92.1	90.6
TrOCR$_{\mathrm{LARGE}}$ (Syn+Benchmark)	94.1	**96.1**	**98.4**	97.3	88.1	84.1	**93.0**	**95.1**

Table 6: Word accuracy on the six benchmark datasets (36-char), where "Syn" indicates the model using synthetic data only and "Syn+Benchmark" indicates the model using synthetic data and benchmark datasets.

STM. The decoder is composed of the proposed Gated Cascade Attention Module (GCAM) and generates the target characters from the feature sequence extracted by the encoder. For the Transformer models, (Sheng, Chen, and Xu 2019) first applied the Transformer to Scene Text Recognition. Since the input of the Transformer architecture is required to be a sequence, a CNN-based modality-transform block is employed to transform 2D input images to 1D sequences. (Bleeker and de Rijke 2019) added a direction embedding to the input of the decoder for the bidirectional text decoding with a single decoder, while (Lee et al. 2020) utilized the two-dimensional dynamic positional embedding to keep the spatial structures of the intermediate feature maps for recognizing texts with arbitrary arrangements and large inter-character spacing. (Yu et al. 2020) proposed semantic reasoning networks to replace RNN-like structures for more accurate text recognition. (Atienza 2021) only used the image Transformer without text Transformer for the text recognition in a non-autoregressive way.

The texts in natural images may appear in irregular shapes caused by perspective distortion. (Shi et al. 2016; Baek et al. 2019; Litman et al. 2020; Shi et al. 2018; Zhan and Lu 2019) addressed this problem by processing the input images with an initial rectification step. For example, thin-plate spline transformation (Shi et al. 2016; Baek et al. 2019; Litman et al. 2020; Shi et al. 2018) is applied to find a smooth spline interpolation between a set of fiducial points and normalize the text region to a predefined rectangle, while (Zhan and Lu 2019) proposed an iterative rectification network to model the middle line of scene texts as well as the orientation and boundary of textlines. (Baek et al. 2019; Diaz et al. 2021) proposed universal architectures for comparing different recognition models.

Handwritten Text Recognition

(Memon et al. 2020) gave a systematic literature review about the modern methods for handwriting recognition. Various attention mechanisms and positional encodings are compared in the (Michael et al. 2019) to address the alignment between the input and output sequence. The combination of RNN encoders (mostly LSTM) and CTC decoders (Bluche and Messina 2017; Graves and Schmidhuber 2008; Pham et al. 2014) took a large part in the related works for a long time. Besides, (Graves and Schmidhuber 2008; Voigtlaender, Doetsch, and Ney 2016; Puigcerver 2017) have also tried multidimensional LSTM encoders. Similar to the scene text recognition, the seq2seq methods and the scheme for attention decoding have been verified in (Michael et al. 2019; Kang et al. 2020; Chowdhury and Vig 2018; Bluche 2016). (Ingle et al. 2019) addressed the problems in building a large-scale system.

Conclusion

In this paper, we present TrOCR, an end-to-end Transformer-based OCR model for text recognition with pre-trained models. Distinct from existing approaches, TrOCR does not rely on the conventional CNN models for image understanding. Instead, it leverages an image Transformer model as the visual encoder and a text Transformer model as the textual decoder. Moreover, we use the wordpiece as the basic unit for the recognized output instead of the character-based methods, which saves the computational cost introduced by the additional language modeling. Experiment results show that TrOCR achieves state-of-the-art results on printed, handwritten and scene text recognition with just a simple encoder-decoder model, without any post-processing steps.

References

Atienza, R. 2021. Vision Transformer for Fast and Efficient Scene Text Recognition. *arXiv preprint arXiv:2105.08582*.

Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S. J.; and Lee, H. 2019. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4715–4723.

Baek, J.; Matsui, Y.; and Aizawa, K. 2021. 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 (CVPR)*, 3113–3122.

Bao, H.; Dong, L.; and Wei, F. 2021. BEiT: BERT Pre-Training of Image Transformers. arXiv:2106.08254.

Bautista, D.; and Atienza, R. 2022. Scene Text Recognition with Permuted Autoregressive Sequence Models. *arXiv preprint arXiv:2207.06966*.

Bhunia, A. K.; Chowdhury, P. N.; Sain, A.; and Song, Y.-Z. 2021a. Towards the Unseen: Iterative Text Recognition by Distilling From Errors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14950–14959.

Bhunia, A. K.; Sain, A.; Kumar, A.; Ghose, S.; Chowdhury, P. N.; and Song, Y.-Z. 2021b. Joint Visual Semantic Reasoning: Multi-Stage Decoder for Text Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14940–14949.

Bleeker, M.; and de Rijke, M. 2019. Bidirectional scene text recognition with a single decoder. *arXiv preprint arXiv:1912.03656*.

Bluche, T. 2016. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *Advances in Neural Information Processing Systems*, 29: 838–846.

Bluche, T.; and Messina, R. 2017. Gated convolutional recurrent neural networks for multilingual handwriting recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, 646–651. IEEE.

Cai, H.; Sun, J.; and Xiong, Y. 2021. Revisiting Classification Perspective on Scene Text Recognition.

Chowdhury, A.; and Vig, L. 2018. An efficient end-to-end neural model for handwritten text recognition. *arXiv preprint arXiv:1807.07965*.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 702–703.

Cui, M.; Wang, W.; Zhang, J.; and Wang, L. 2021. Representation and Correlation Enhanced Encoder-Decoder Framework for Scene Text Recognition. In Lladós, J.; Lopresti, D.; and Uchida, S., eds., *Document Analysis and Recognition – ICDAR 2021*, 156–170. Cham: Springer International Publishing. ISBN 978-3-030-86337-1.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Diaz, D. H.; Qin, S.; Ingle, R.; Fujii, Y.; and Bissacco, A. 2021. Rethinking Text Line Recognition Models. arXiv:2104.07787.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. arXiv:1905.03197.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*.

Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; and Zhang, Y. 2021. 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 (CVPR)*, 7098–7107.

Gao, Y.; Chen, Y.; Wang, J.; Tang, M.; and Lu, H. 2019. Reading scene text with fully convolutional sequence modeling. *Neurocomputing*, 339: 161–170.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376.

Graves, A.; and Schmidhuber, J. 2008. Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in neural information processing systems*, 21: 545–552.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2315–2324.

Ingle, R. R.; Fujii, Y.; Deselaers, T.; Baccash, J.; and Popat, A. C. 2019. A scalable handwritten text recognition system. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 17–24. IEEE.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. In *Workshop on Deep Learning, NIPS*.

Kang, L.; Riba, P.; Rusiñol, M.; Fornés, A.; and Villegas, M. 2020. Pay attention to what you read: Non-recurrent handwritten text-line recognition. *arXiv preprint arXiv:2005.13044*.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. ICDAR 2015 competition on robust reading. In *ICDAR*.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and De Las Heras, L. P. 2013. ICDAR 2013 robust reading competition. In *ICDAR*.

Krishnan, P.; and Jawahar, C. V. 2016. Generating Synthetic Data for Text Recognition. arXiv:1608.04224.

Kudo, T.; and Richardson, J. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Lee, J.; Park, S.; Baek, J.; Oh, S. J.; Kim, S.; and Lee, H. 2020. On recognizing texts of arbitrary shapes with 2D self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 546–547.

Liao, M.; Wan, Z.; Yao, C.; Chen, K.; and Bai, X. 2019. Real-time Scene Text Detection with Differentiable Binarization. arXiv:1911.08947.

Litman, R.; Anschel, O.; Tsiper, S.; Litman, R.; Mazor, S.; and Manmatha, R. 2020. Scatter: selective context attentional scene text recognizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11962–11972.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.

Lyu, P.; Zhang, C.; Liu, S.; Qiao, M.; Xu, Y.; Wu, L.; Yao, K.; Han, J.; Ding, E.; and Wang, J. 2022. MaskOCR: Text Recognition with Masked Encoder-Decoder Pretraining. *arXiv preprint arXiv:2206.00311*.

Memon, J.; Sami, M.; Khan, R. A.; and Uddin, M. 2020. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, 8: 142642–142668.

Michael, J.; Labahn, R.; Grüning, T.; and Zöllner, J. 2019. Evaluating sequence-to-sequence models for handwritten text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1286–1293. IEEE.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Top-down and bottom-up cues for scene text recognition. In *CVPR*.

Mou, Y.; Tan, L.; Yang, H.; Chen, J.; Liu, L.; Yan, R.; and Huang, Y. 2020. Plugnet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, 158–174. Springer.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Pham, V.; Bluche, T.; Kermorvant, C.; and Louradour, J. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition*, 285–290. IEEE.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 569–576.

Puigcerver, J. 2017. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, 67–72. IEEE.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*.

Sang, D. V.; and Cuong, L. T. B. 2019. Improving CRNN with EfficientNet-like feature extractor and multi-head attention for text recognition. In *Proceedings of the Tenth International Symposium on Information and Communication Technology*, 285–290.

Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Sheng, F.; Chen, Z.; and Xu, B. 2019. NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 781–786. IEEE.

Shi, B.; Bai, X.; and 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.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4168–4176.

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

Su, B.; and Lu, S. 2014. Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, 35–48. Springer.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Voigtlaender, P.; Doetsch, P.; and Ney, H. 2016. Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 228–233. IEEE.

Wan, Z.; He, M.; Chen, H.; Bai, X.; and Yao, C. 2020. Textscanner: Reading characters in order for robust scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12120–12127.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *2011 International conference on computer vision*, 1457–1464. IEEE.

Wang, S.; Wang, Y.; Qin, X.; Zhao, Q.; and Tang, Z. 2019. Scene text recognition via gated cascade attention. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 1018–1023. IEEE.

Wang, T.; Zhu, Y.; Jin, L.; Luo, C.; Chen, X.; Wu, Y.; Wang, Q.; and Cai, M. 2020a. Decoupled Attention Network for Text Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

Wang, Y.; Xie, H.; Fang, S.; Wang, J.; Zhu, S.; and Zhang, Y. 2021. From Two to One: A New Scene Text Recognizer With Visual Language Modeling Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14194–14203.

Wightman, R. 2019. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models.

Yan, R.; Peng, L.; Xiao, S.; and Yao, G. 2021. Primitive Representation Learning for Scene Text Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 284–293.

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards Accurate Scene Text Recognition With Semantic Reasoning Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12110–12119.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *European Conference on Computer Vision*, 135–151. Springer.

Zhan, F.; and Lu, S. 2019. Esir: End-to-end scene text recognition via iterative image rectification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2059–2068.

Zhang, H.; Yao, Q.; Yang, M.; Xu, Y.; and Bai, X. 2020a. AutoSTR: Efficient backbone search for scene text recognition. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, 751–767. Springer.

Zhang, J.; Luo, C.; Jin, L.; Wang, T.; Li, Z.; and Zhou, W. 2020b. SaHAN: Scale-aware hierarchical attention network for scene text recognition. *Pattern Recognition Letters*, 136: 205–211.