

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

Minghao Li^{1*}, Tengchao Lv², Lei Cui², Yijuan Lu³,
Dinei Florencio³, Cha Zhang³, Zhoujun Li¹, Furu Wei²

¹Beihang University

²Microsoft Research Asia

³Microsoft Azure AI

{liminghao1630, lizj}@buaa.edu.cn

{tengchaolv, lecu, yijlu, dinei, chazhang, fuwei}@microsoft.com

Abstract

Text recognition is a long-standing research problem for document digitalization. Existing approaches for text recognition 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 both printed and handwritten text recognition tasks. The code and models will be publicly available at <https://aka.ms/TrOCR>.

1 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 in 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 DB-Net (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 approaches 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 approaches 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., 2021a), especially the recent self-supervised image pre-training (Bao et al., 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

Work done during internship at Microsoft Research Asia.

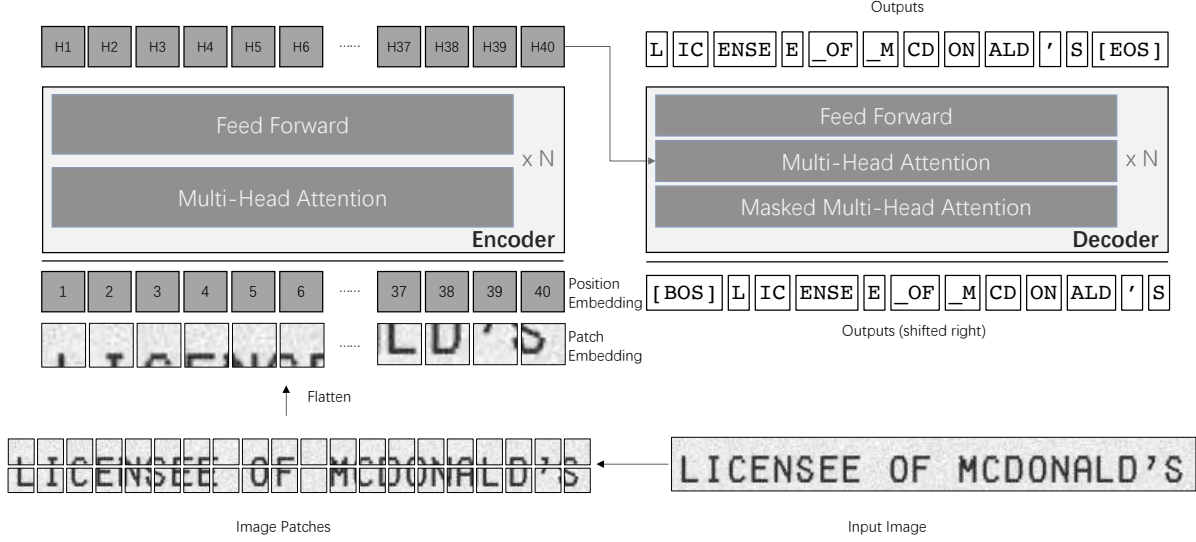


Figure 1: Model Architecture of TrOCR, where an encoder-decoder model is designed with a pre-trained image Transformer as the encoder and a pre-trained text Transformer as the decoder.

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., 2021a; Bao et al., 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), 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. Second, TrOCR does not require any sophisticated convolutional network for the backbone, 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 both printed and handwritten text recognition tasks 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.

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 accuracy 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 model and code will be publicly available at <https://aka.ms/TrOCR>.

2 TrOCR

2.1 Model Architecture

TrOCR is built up with the Transformer architecture, including image Transformer for extracting the visual features and 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 while paying attention to the encoder output and the previous generation.

2.1.1 Encoder

The encoder accepts an input image $x_{\text{img}} \in \mathbb{R}^{3 \times H_0 \times W_0}$, 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 . After that, the patches are flattened into vectors and linearly projected to D -dimension vectors, which are the patch embeddings and 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., 2021b), we keep the special token “[CLS]” that is usually used for the image classification task. 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. Then, the input sequence is passed through a stack of identical encoder layers. Each Transformer layer has a multi-head self-attention module and a fully connected feed-forward network. Both of these two parts are followed by residual connection and layer normalization.

The attention mechanism is to distribute different attention on the values and output the weighted sum of them, where the weights of the values are computed by the corresponding keys and the queries. For the self-attention modules, all of the queries, keys and values come from the same sequence. The matrix of the attention output is computed as:

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

The scaling factor of $\frac{1}{\sqrt{d_k}}$ is applied to avoid the extremely small gradients of the softmax function, where the d_k is the dimension of queries and keys. The multi-head attention is to project the queries, keys and values h times with different learnable weights of projection, which allows the model to jointly gather the information from different representation subspaces.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where } \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

Different from 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 enables the model to pay different attention to either the whole image or the independent patches.

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

$$\begin{aligned} h_i &= \text{Proj}(\text{Emb}(\text{Token}_i)) \\ \sigma(h_{ij}) &= \frac{e^{h_{ij}}}{\sum_{k=1}^V e^{h_{ik}}} \quad \text{for } j = 1, 2, \dots, V \end{aligned}$$

The embedding from the decoder is projected from the model dimension to the dimension of the vocabulary size V . The probabilities over the vocabulary are calculated by the softmax function and we use beam search to get the final output.

2.2 Model Initialization

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

2.2.1 Encoder Initialization

The DeiT (Touvron et al., 2021a) and BEiT (Bao et al., 2021) models are used for the encoder initial-

ization in the TrOCR models. DeiT trains the image Transformer with ImageNet as the sole training set, and tries different hyper-parameters and data augmentation to make the model in a data-efficient manner. Moreover, they distill the knowledge of a strong image classifier to a distilled tokens in the initial embeddings, which leads to a competitive result comparing 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. For each image, it 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. Essentially, the structure of BEiT is the same as the image Transformer and lacks the distilled token when comparing with DeiT.

2.2.2 Decoder Initialization

We use the RoBERTa 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. When loading the RoBERTa models to the decoders, the structures do not exactly match. For example, the encoder-decoder attention layers are absent in the RoBERTa models. To address this, we initialize the decoders with the RoBERTa models and the absent layers are randomly initialized.

2.3 Task Pipeline

The pipeline of the text recognition task in this work is described that given the textline images, the model extracts the visual features and predict 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 normally indicates the end of a sentence. During training, we rotate the sequence backward by one position and move the “[EOS]” token to the beginning. The rotated 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 “[EOS]” token to predict the output iteratively, while continuously taking the newly

generated output as the next input.

2.4 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 which differs by the used dataset. In the first stage, we synthesize a large dataset consisting of hundreds of millions of printed textline images with their corresponding text content and pre-train the TrOCR models on that. In the second stage, we build two relatively small datasets which correspond to printed and handwritten downstream tasks, containing about millions of textline images each. Subsequently, we pre-train two separate models on the printed data and the handwritten data of the second stage, both initialized by the first-stage model.

2.5 Fine-tuning

The pre-trained TrOCR models are fine-tuned on printed and handwritten text recognition tasks. The output of the TrOCR models are based on Byte Pair Encoding (BPE) (Sennrich et al., 2015) and does not rely on any task-related vocabularies.

2.6 Data Augmentation

To enhance the variety of the pre-training data and the fine-tuning data, we leverage the data augmentation. In total, seven kinds of image transformations (including keeping the original input image) are taken in this work. For each sample, we randomly decide which image transformation to take with equal possibilities. We augment the input images with random rotation (-10 to 10 degrees), Gaussian blurring, image dilation, image erosion, downscaling, underlining or keeping the original.

3 Experiments

3.1 Data

3.1.1 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 the PDF files into page images, extract the textlines and their cropped images. In total, the first-stage pre-training dataset contains 684M textlines.

Stage	Text Type	#Samples
First	Printed	684M
Second	Printed	3.3M
Second	Handwritten	17.9M

Table 1: Statistics of synthetic data for pre-training. "Stage" indicates the pre-training pipeline.

For the second stage, we use 5,427 handwritten fonts¹ to synthesize handwritten textline images by the TRDG², 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. The OCR engine returns the texts with the two-dimensional coordinates and extra information of the input images, like the orientation. We correct the orientation to the vertical, crop the textlines from the whole receipt images, rotate the textline images if not horizontal, and prune them. We also use TRDG to synthesize 1M printed textline images with two receipt fonts and the built-in printed fonts. The printed dataset for the second-stage pre-training consists of 3.3M textlines.

3.1.2 SROIE Task 2

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 train and test set of SROIE. In this work, since the text detection is not included, 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.

3.1.3 IAM Handwriting Database

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 dataset³: 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.

¹The 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>

3.2 Settings

The TrOCR models are built upon the Fairseq (Ott et al., 2019) which is a popular sequence modeling toolkit. For the encoders, 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 are from the UniLM’s official repository⁴. For the decoders, we use the RoBERTa models provided by the Fairseq and the corresponding dictionary. 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 BPE 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. Both the DeiT_{BASE} and the BEiT_{BASE} has 12 layers with 768 hidden sizes and 12 heads while the BEiT_{LARGE} has 24 layers with 1024 hidden sizes and 16 heads. We use 6 layers, 512 hidden sizes and 8 attention heads for the base decoders while 12 layers, 1,024 hidden sizes and 16 heads for the large decoders. For this task, the higher-level layers of the decoder (language model part) are more important, so we only use the last half of all layers from the corresponding RoBERTa model, which are the last 6 layers for the RoBERTa_{BASE} and the last 12 layers for the RoBERTa_{LARGE}.

3.2.1 Baselines

We take the CRNN model (Shi et al., 2016a) 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 implementation⁵ and initialized the parameters by the provided pre-trained model.

3.3 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 definition

⁴<https://github.com/microsoft/unilm>

⁵<https://github.com/meijieru/crnn.pytorch>

Encoder	Decoder	Precision	Recall	F1
DeiT _{BASE}	RoBERTa _{BASE}	69.28	69.06	69.17
BEiT _{BASE}	RoBERTa _{BASE}	76.45	76.18	76.31
ResNet50	RoBERTa _{BASE}	66.74	67.29	67.02
DeiT _{BASE}	RoBERTa _{LARGE}	77.03	76.53	76.78
BEiT _{BASE}	RoBERTa _{LARGE}	79.67	79.06	79.36
ResNet50	RoBERTa _{LARGE}	72.54	71.13	71.83
From Scratch		36.60	36.97	36.78
CRNN		28.71	48.58	36.09
Tesseract OCR		57.50	51.93	54.57

Table 2: Ablation study on the SROIE dataset, where all the models are trained using the SROIE dataset only.

Model	Recall	Precision	F1
TrOCR _{BASE}	96.37	96.31	96.34
TrOCR _{LARGE}	96.59	96.57	96.58
H&H Lab (Shi et al., 2016a)	96.35	96.52	96.43
MSOLab (Sang and Cuong, 2019)	94.77	94.88	94.82
CLOVA OCR (Baek et al., 2019)	94.3	94.88	94.59

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>).

of the precision, recall and f1 score are described as the following:

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

The IAM dataset is evaluated by the case sensitive Character Error Rate (CER). CER represents the number of insertions (i), substitutions (s) and deletions (d) of characters, aka the Levenshtein distance, and normalized by the number of the characters (n) in the reference text. It is described as:

$$CER = [(i + s + d)/n] \times 100$$

3.4 Results

3.4.1 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 RoBERTa_{BASE} and the large decoders initialized

by RoBERTa_{LARGE}. To evaluate the improvement from the initialization of the pre-trained models, we also experiment with the option that all the parameter is randomly initialized, while both of the encoder and the decoder are the base settings. 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.

Table 2 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 RoBERTa_{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 two settings on the subsequent experiments: **TrOCR_{BASE}** (total parameters=334M) consists of the encoder of BEiT_{BASE} and the decoder of RoBERTa_{LARGE} while **TrOCR_{LARGE}** (total parameters=558M) consists of the encoder of BEiT_{LARGE} and the decoder of RoBERTa_{LARGE}.

3.4.2 SROIE Task 2

Table 3 shows the results of the TrOCR models and the current SOTA methods on the leaderboard

Model	Architecture	Training Data	External LM	CER
TrOCR _{BASE}	Transformer	Synthetic + IAM	No	3.42
TrOCR _{LARGE}	Transformer	Synthetic + IAM	No	2.97
(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., 2020)	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

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

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 et al., 2016a) 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.

3.4.3 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 TrOCR_{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., 2020), 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 bet-

ter results, illustrating after pre-training the Transformer structures are competent to extract visual features well. 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 achieve a new state-of-the-art CER of 2.97. Without leveraging any extra human-labeled data, TrOCR even get comparable results with the methods in (Diaz et al., 2021) using the additional internal human-labeled dataset.

4 Related Work

4.1 Scene Text Recognition

For text recognition, the most popular approaches are usually based on the CTC-based models. (Shi et al., 2016a) 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., 2020; Wang et al., 2019; Sheng et al., 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., 2020), 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 BiLSTM. 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 et al., 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., 2016b; Baek et al., 2019; Litman et al., 2020; Shi et al., 2018; Zhan and Lu, 2019) address this problem by processing the input images with an initial rectification step. For example, thin-plate spline transformation (Shi et al., 2016b; 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. The framework of the former work consists of four stages, which are the transformation stage, feature extraction stage, sequence modeling stage and prediction stage. In contrast, the latter focuses more on encoder/decoder comparison.

4.2 Handwritten

(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 et al., 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; Poulos and Valle, 2017; Chowdhury and Vig, 2018; Bluche, 2016). (Ingle et al., 2019) addressed the problems in building a large-scale system.

5 Conclusion and Future Work

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 word-piece 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 accuracy on both printed text and handwritten text recognition. With just a simple encoder-decoder model, without any post-processing steps.

For future research, we treat TrOCR as a research framework where a variety of visual and textual pre-trained models are allowed for plug and play in terms of architectures and model sizes, in order to support cloud/edge computation. Furthermore, we will investigate more efficient data syn-

thesis and augmentation strategies that can work well with the pre-trained encoders and decoders, so as to further improve the text recognition accuracy. We are also interested in extending the TrOCR to address the multilingual text recognition problems.

References

- Rowel Atienza. 2021. Vision transformer for fast and efficient scene text recognition. *arXiv preprint arXiv:2105.08582*.
- Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. 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*, pages 4715–4723.
- Hangbo Bao, Li Dong, and Furu Wei. 2021. [Beit: Bert pre-training of image transformers](#).
- Maurits Bleeker and Maarten de Rijke. 2019. Bidirectional scene text recognition with a single decoder. *arXiv preprint arXiv:1912.03656*.
- Théodore Bluche. 2016. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *Advances in Neural Information Processing Systems*, 29:838–846.
- Théodore Bluche and Ronaldo Messina. 2017. Gated convolutional recurrent neural networks for multilingual handwriting recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 646–651. IEEE.
- Arindam Chowdhury and Lovekesh Vig. 2018. An efficient end-to-end neural model for handwritten text recognition. *arXiv preprint arXiv:1807.07965*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Daniel Hernandez Diaz, Siyang Qin, Reeve Ingle, Yasuhisa Fujii, and Alessandro Bissacco. 2021. [Re-thinking text line recognition models](#).
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. [Unified language model pre-training for natural language understanding and generation](#).
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.
- Yunze Gao, Yingying Chen, Jinqiao Wang, Ming Tang, and Hanqing Lu. 2019. Reading scene text with fully convolutional sequence modeling. *Neurocomputing*, 339:161–170.
- Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on machine learning*, pages 369–376.
- Alex Graves and Jürgen Schmidhuber. 2008. Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in neural information processing systems*, 21:545–552.
- R Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C Papat. 2019. A scalable handwritten text recognition system. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 17–24. IEEE.
- Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. 2020. Pay attention to what you read: Non-recurrent handwritten text-line recognition. *arXiv preprint arXiv:2005.13044*.
- Praveen Krishnan and C. V. Jawahar. 2016. [Generating synthetic data for text recognition](#).
- Junyeop Lee, Sungrae Park, Jeonghun Baek, Seong Joon Oh, Seonghyeon Kim, and Hwalsuk Lee. 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*, pages 546–547.
- Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. 2019. [Real-time scene text detection with differentiable binarization](#).
- Ron Litman, Oron Anschel, Shahar Tsiper, Roe Litman, Shai Mazor, and R Manmatha. 2020. Scatter: selective context attentional scene text recognizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11962–11972.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#).
- Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. 2020. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668.
- Johannes Michael, Roger Labahn, Tobias Grüning, and Jochen Zöllner. 2019. Evaluating sequence-to-sequence models for handwritten text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1286–1293. IEEE.

- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition*, pages 285–290. IEEE.
- Jason Poulos and Rafael Valle. 2017. Attention networks for image-to-text.
- Joan Puigcerver. 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, pages 67–72. IEEE.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.
- Dinh Viet Sang and Le Tran Bao Cuong. 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*, pages 285–290.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Fenfen Sheng, Zhineng Chen, and Bo Xu. 2019. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 781–786. IEEE.
- Baoguang Shi, Xiang Bai, and Cong Yao. 2016a. 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.
- Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016b. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176.
- Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2018. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048.
- Bolan Su and Shijian Lu. 2014. Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, pages 35–48. Springer.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021a. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021b. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Paul Voigtlaender, Patrick Doetsch, and Hermann Ney. 2016. Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 228–233. IEEE.
- Siwei Wang, Yongtao Wang, Xiaoran Qin, Qijie Zhao, and Zhi Tang. 2019. Scene text recognition via gated cascade attention. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1018–1023. IEEE.
- Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. 2020. Decoupled attention network for text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12216–12224.
- Ross Wightman. 2019. *Pytorch image models*. <https://github.com/rwightman/pytorch-image-models>.
- Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12110–12119.
- Fangneng Zhan and Shijian Lu. 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*, pages 2059–2068.
- Jiaxin Zhang, Canjie Luo, Lianwen Jin, Tianwei Wang, Ziyang Li, and Weiying Zhou. 2020. Sahan: Scale-aware hierarchical attention network for scene text recognition. *Pattern Recognition Letters*, 136:205–211.