https://doi.org/10.2352/EI.2024.36.8.IMAGE-241

Self-Attention Enhanced Recognition: A Unified Model for Handwriting and Scene-Text Recognition with Improved Inference

Gaurav Patel[†], *Taewook Kim*[†], *Qian Lin*[‡], *Jan P. Allebach*[†], *and Qiang Qiu*[†]
[†] *Purdue University*
[‡] *HP Inc.*

Abstract

In this paper, we introduce a unified handwriting and scene-text recognition model tailored to discern both printed and handwritten text images. Our primary contribution is the incorporation of the self-attention mechanism, a salient feature of the transformer architecture. This incorporation leads to two significant advantages: 1) A substantial improvement in the recognition accuracy for both scene-text and handwritten text, and 2) A notable decrease in inference time, addressing a prevalent challenge faced by modern recognizers that utilize sequence-based decoding with attention.

Introduction

Image-based text recognition encompasses the identification and interpretation of printed and handwritten text from images. While printed text recognition has seen vast improvements over the years [1, 2, 26, 32, 39], handwritten text recognition continues to pose significant challenges due to its inherent variability, styles, and deformations. This discrepancy has driven research to devise models to bridge the gap between scene-text and handwriting recognition, aiming for robust recognizers that perform commendably on both fronts. The majority of modern text recognizers are rooted in sequence-based decoding methods, most notably those that leverage the attention mechanism. While these models have substantially elevated the state-of-the-art, they suffer from prolonged inference times, which poses limitations especially in near real-time or large-scale applications [3, 4, 5, 6, 7, 9, 22, 36, 40, 42, 43]. Furthermore, despite the successes in recognizing printed text, there is still ample room for improvement in the handwritten domain. This paper focuses on developing a unified handwriting and scene-text recognition model, proficient in discerning printed and handwritten text images with striking precision. Drawing inspiration from the pioneering transformer architecture [34], our model integrates a self-attention mechanism. First, we observe a significant improvement in the accuracy of recognizing scene text and handwritten text. Second, our model also improves the inference time of the recognition model, making it more suitable for near real-time scenario cases. By addressing the challenges associated with sequence-based decoding, we present a system that is not only more accurate but also markedly inculcates faster inference time.

Related Works

High-performance convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have aided Scene-Text recognition. Convolutional-Recurrent Neural Network (CRNN) [37] was the first use of CNN and RNN for scene-text recognition. The CNN features are extracted from the input text image and reconfigured with an RNN for robust sequence prediction. Several versions [3, 4, 22, 29, 36, 42] have been proposed to increase performance following CRNN. For example, transformation modules to normalize text images have been proposed for rectifying arbitrary text geometries [9, 15]. Furthermore, improved CNN feature extractors [8, 20, 30] have been incorporated for treating complicated text images with high intrinsic variability, for example, font style and cluttered backdrop. The adopted recognition model can be divided into 3 stages of operations namely, Feature Extraction, Sequence Modeling, and Prediction, as shown in Figure 1.

Methodology

Current state-of-the-art CNN-based text/handwriting recognition systems employ a sequence-to-sequence framework. They encode CNN image features using a sequence model, either LSTM or GRU, and then decode the final hidden state through an attention-based auto-regressive method [1]. Although this methodology is widely adopted, its sequential unrolling of the decoder model and prediction of each character individually make it relatively slow, limiting its use in real-time or near real-time applications. An alternative to this sequential decoding is the connectionist-temporal classification (CTC) [11]. The CTC directly produces a fixed sequence of probability distributions, allowing us to select the character with the highest probability at each time step and decode it into a string. However, directly using the CTC decoding sacrifices the sequential context that the attention-based decoder provided. To reintroduce this sequential context while leveraging the speed of CTC decoding, we integrate 'self-attention' inspired by the widely recognized transformer networks [34]. By applying CTC to the features extracted post self-attention layers, we incorporate contextual modeling and benefit from the speed of CTC decoding. This approach reduces inference time during decoding while minimally increasing the parameter count, yielding results superior to those from attention-based decoding. The architecture of a typical text-recognition system is illustrated in Figure 1.

Feature Extraction

At the foundational layer of the recognition model lies the feature extraction phase. In this stage, an input image, denoted as X, is abstracted through a Convolutional Neural Network (CNN), resulting in a visual feature map $V = \{v_i\}, i = 1, ..., I$, where I

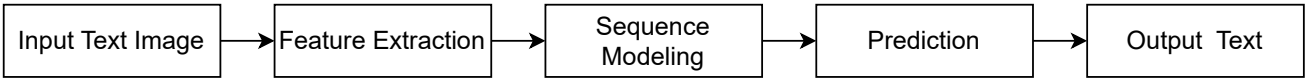

Figure 1. *Block diagram depicting the different modules present in the text recognition system.*

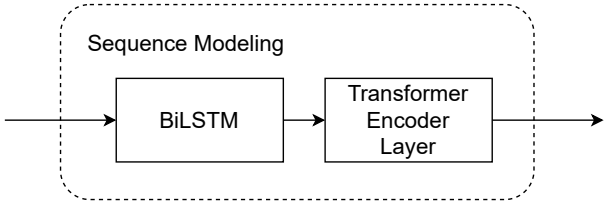

Figure 2. *Sequence modeling stage block diagram.*

represents the column count within the feature map. Intriguingly, each column of this map correlates with a unique receptive field spanning the horizontal axis of the input image. It's noteworthy that the character present within each receptive field is ascertained using its corresponding column vectors. For this purpose, we employ the ResNet architecture [13] — a profound CNN design enriched with residual connections, facilitating the efficient training of considerably deep networks. We develop a network architecture inspired by FAN [8], which seamlessly integrates the ResNet module [13]. This architecture comprises of 29 trainable layers. Furthermore, we incorporate dropout layers to inherently enforce regularization, as suggested by [33].

Sequence Modeling

After the feature extraction stage the ouptut is reshaped to have a sequence of features V. That is, each column in a feature map $v_i \in V$ serves as an individual element in the sequence of frames. This sequence, however, may suffer from a lack of contextual information. As a result, some prior studies [8, 29, 30] have used Bidirectional LSTM (BiLSTM) to improve the sequence, V, using a sequence model as $H = \text{Seq}(V)$, following the feature extraction step, where, Seq(.) denotes the sequence modeling function.

In our model, while the BiLSTM layer captures sequential information, the subsequent integration of the *Transformer* encoder [34] brings substantial enhancements. This encoder, illustrated in Figure 2, introduces a sophisticated attention mechanism on the input sequential features (H) that excels at recognizing long-range dependencies, something BiLSTM might struggle with. By employing a stack of $n = 2$ identical transformer layers, we ensure deeper and refined sequence representations. Each of these layers incorporates a **Multi-Head Attention** mechanism, enabling the model to focus on multiple sequence segments concurrently. This translates to a more versatile and comprehensive understanding of the input. Complementing this, the position-wise **Feed-Forward Network** ensures that the positional context of tokens is well-respected, enriching the overall sequence interpretation beyond what a standalone BiLSTM layer could achieve.

Transformer Encoder Layer

- **Attention in the context of Transformers.** The mapping of a query and a set of key-value pairs to an output can be characterized as an attention function, where the query, keys, values, and output are all vectors. The result is a weighted sum of the values, with the weight allocated to each value determined by the query's similarity to the relevant key [34].

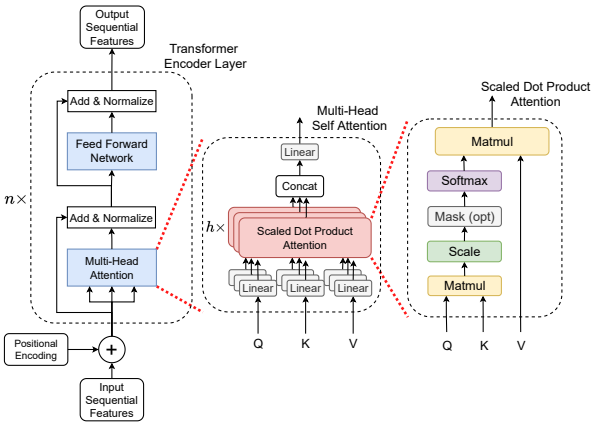

Figure 3. *Transformer encoder layer. Here n and h denote the number of transformer encoder layers and number of heads in the self-attention step, respectively.*

- **Scaled Dot-Product Attention.** The input consists of three vectors as inputs, namely, queries, keys and values of dimensions d_q, d_k, and d_v, where $d_q = d_k$. In practice, the attention function computes simultaneously on a series of queries, which are grouped into a matrix Q. In matrices K and V, the keys and values are also grouped together. The output matrix is computed in the following way,

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \tag{1}$$

- **Multi-Head Attention.** The Multi-Head Attention linearly projects the queries, keys, and values h times with separate, learned linear projections to d_k, d_k, and d_v dimensions, respectively. The attention function is applied in parallel on each of these projected versions of queries, keys, and values, providing d_v-dimensional output values. Multi-head attention allows the model to simultaneously attend to information from various representation subspaces at various locations.

$$\text{Multi-Head}(Q,K,V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_n)W^O, \tag{2}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{3}$$

where, $W_i^Q \in \mathbb{R}^{d_{model} \times d_q}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, and $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$ are the projection parameters. In our implementation, we employ $h = 2$ parallel attention layers or heads, and the values $d_k = d_v = d_{model}/h =$

128. A schematic architectural summary of the Scaled Dot-Product Attention and the Multi-Head Attention is provided in Figure 3.

Prediction

In the Prediction stage, the module predicts a sequence of characters from the input H (i.e., $Y = y_1, y_2, ...$). We use the Connectionist temporal classification (CTC) [11] decoding to get the final machine-readable text. Even though a fixed number of features are given, CTC allows for predicting a non-fixed number of sequences. CTC's main approach is to predict a character for each column ($h_i \in H$) and to change the whole character sequence into a non-fixed stream of characters by eliminating repeated characters and blanks.

Connectionist Temporal Classification (CTC). CTC takes a in sequence of features H of a fixed length T and produces a probability of an hypothesis π, defined as,

$$p(\pi|H) = \prod_{t=1}^{T} y^t_{\pi_t}, \tag{4}$$

where, $y^t_{\pi_t}$ denotes the probability of predicting the character at the t-th time step. After that, a mapping function M maps π to Y by removing repeated characters and blank spaces. due to many-to-one nature of the mapping function M, the conditional probability of Y is defined as sum of the probability of all π that map to Y,

$$p(Y|H) = \sum_{\pi:M(\pi)=Y} p(\pi|H), \tag{5}$$

at test time, we use greedy selection. We take the highest probability character π_t at each time step t to map π to Y,

$$Y^* \approx M(\arg\max_{\pi} p(\pi|H)). \tag{6}$$

Recurrent architecture such as Attention-based LSTM, while adept at capturing sequential details, face computational bottlenecks due to their sequential nature, slowing down inference times—a major limitation for real-time or near real-time applications. Conversely, transformer models, using self-attention, process multiple sequence features concurrently, leveraging matrix multiplication's parallelism for faster inference. As a result, transformers outperform LSTMs in both computation speed and prediction (decoding) accuracy.

Objective Function

Let $T = \{(X_i, Y_i)\}_{i=1}^{N}$, denote the training dataset, where X_i is the training image and Y_i is the corresponding word label. The training is done by minimizing the negative log-likelihood of the conditional probability of word labels as the objective function as follows,

$$L = -\sum_{(X_i,Y_i)\in T} \log(p(Y_i|X_i)). \tag{7}$$

The modules of the framework are trained end-to-end using the above-described objective, which calculates a cost given an image and its word label. Additionally, while training the recognition model, we used data-augmentation to impose variations to the images such that the trained model is robust and generalizes efficiently. We use random Linear Contrast, Gaussian Blur, Cropping, Sharpening, Affine transforms, etc.

Training dataset

We utilize a combination of synthetically generated, real-world scene-text and handwriting datasets to train our model.

Synthetically Generated Dataset. To augment the variability of the dataset, we use SynthTIGER [41] to create text-images using handwriting and printed text fonts. The recognition model is trained using synthetically generated data. Figure 4 displays several illustrations of synthetic word images that are generated and indicate SynthTIGER and other contemporary synthetic text-image generator MJ [14] and ST [12]. **Real-world Scene Text**

Figure 4. *Example images synthesized from MJ [22], ST [23], and SynthTIGER [24].*

Datasets. Additionally, we utilize a blend of pre-existing datasets that represent printed or non-handwritten text, named *RealSTR*. The details of the dataset composition in *RealSTR* are as follows:

- **Street View Text (SVT) [38]**: Consists of images from GoogleStreet View where the texts were embedded in street pictures. It includes 257 training images and 647 evaluation images.
- **IIIT5k-Words (IIIT) [24]**: The data was gathered by querying Google image searches for terms like "billboards" and "movie posters." It includes 2,000 training and 3,000 evaluation images.
- **ICDAR2013 (IC13) [17]**: The dataset was produced for the ICDAR Robust Reading competition in 2013. It includes 848 training images and 1,015 evaluation images.
- **ICDAR2015 (IC15) [16]**: Many of them have perspective texts and some of them are hazy, as they were collected by persons wearing Google Glass. It has a total of 4,468 training images and 2,077 evaluation images.
- **COCO-Text (COCO) [21]**: The MS COCO dataset was used to construct this dataset. COCO contains many occluded or low-resolution texts due to the fact that the MS COCO dataset was not designed to capture text.
- **RCTW [31]**: The dataset was created for **R**eading **C**hinese **T**ext in the **W**ild competition. Thus many are Chinese text.
- **Uber-Text (Uber) [45]**: Bing Maps Streetside is used to collect Uber-Text (Uber). Many are home numbers, while others are signs with text.
- **MLT19 [25]**: The dataset is designed to recognize text in multiple languages. Arabic, Latin, Chinese, Japanese, Korean, Bangla, and Hindi are among the seven languages.
- **ReCTS [44]**: The dataset was created for the **Re**ading **C**hinese **T**ext on **S**ignboard competition. It contains many irregular texts arranged in various layouts or written with unique fonts.

Table 1. Recognition Performance on Benchmark Datasets: Displayed values indicate the word-level accuracy (%). Notably, the *RealSTR* dataset encompasses a compilation of several distinct benchmarks, including SVT [38], IIIT [24], IC13 [17] IC15 [16], COCO-Text [21], RCTW [31], Uber-Text [45], MLT19 [25], and ReCTS [44] training set. Top-performing results are emphasized in bold for clarity and ease of comparison.

Training Data Combination	Scene-Text/Printed Text Datasets						Handwriting Datasets				
	IIIT	SVT	IC13	IC15	SVTP	CUTE	IAM	LVDB	HP-OCW	HP-NW	
										Notepad	Whiteboard
SynthTiger	80.56	81.60	91.72	64.60	70.23	75.69	43.54	61.84	60.67	63.73	71.84
GNHK	18.10	7.41	26.60	7.69	3.87	5.55	46.68	43.96	31.70	57.94	55.24
SynthTiger + GNHK	78.43	75.42	87.68	59.25	65.73	66.66	55.35	64.34	60.09	74.61	73.45
SynthTiger + GNHK + RealSTR	**84.16**	**85.16**	**94.29**	**75.61**	**76.28**	**78.81**	**60.94**	**69.42**	**66.24**	**80.50**	**79.50**

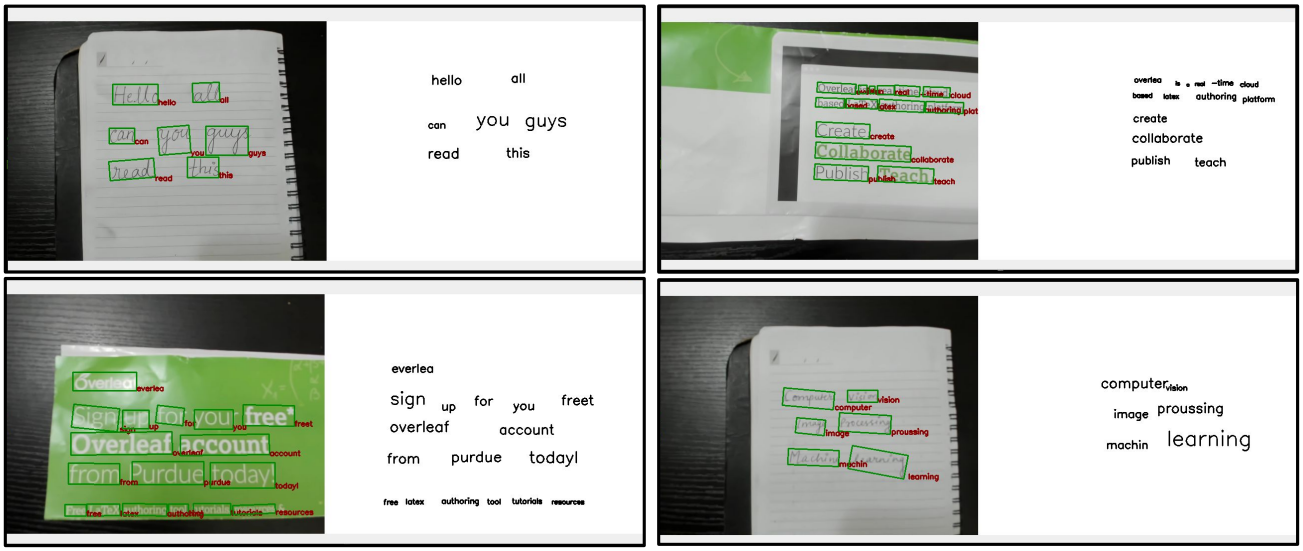

Figure 5. *Examples of end-to-end frame-level recognition using a web camera, adjacent to detection results [18] on a white backdrop we embed the recognized text*

Real-world Handwriting Dataset. We also utilize the GoodNotes Handwriting Collection (GNHK) dataset [19]. This dataset comprises unrestricted camera-captured images of handwritten English text sourced from diverse regions globally. It contains 87 document images, encompassing 172,936 characters, 39,026 texts, and 9,363 lines.

Evaluation Dataset

In addition to SVT [38], IIIT [24], IC13 [17], IC15 [16] test set, we evaluate on SVTP [27], CUTE (CT) [28] for scene-text and printed text. For handwriting we evaluated on IAM [23], LectureVideoDB [10] and our in-house collected datasets HP-OCW and HP-NW. The details of the dataset used for evaluation are as follows:

- **SVTP [27]**: Similar to SVT, SVTP is gathered via Google Street View. Unlike SVT [38], SVTP has a large number of texts from different view points. It includes 645 images for testing.
- **CUTE (CT) [28]**: The dataset pertains to curved text. The images were taken with a digital camera or downloaded from the internet. It includes 288 cropped images for evaluation.
- **IAM [23]**: This English handwritten text dataset, which was divided into writer-independent training, validation, and test, included 657 different writers.
- **LectureVideoDB (LVDB) [10]**: The dataset comprises of over 5000 frames from lecture videos annotated for text detection and recognition. The dataset is benchmarked using existing state-of-the-art methods for scene text detection and scene text/ handwriting recognition.
- **HP-Open Course Ware Dataset (HP-OCW)**: We manually collected and labeled Lecture videos from MIT Open Course Ware using the free and open-source annotation application LabelMe [35] to assess the performance of our system. We gathered 433 lectures from different institutions' video lectures. Then, from 50 video clips, we took 10 consecutive frames and labeled them.
- **HP-Notepad Whiteboard Dataset (HP-NW)**: Additionally, we collected 50 image frames from GNHK test set and 50 from YouTube videos that feature handwritten text on whiteboards or notebooks. The text regions are annotated with word-labels and used for evaluation. The majority of the words in this dataset are handwritten.

Table 2. Evaluation results on word level accuracy (%) on the current SOTA TRBA [1] and our method for inference speed and parameter comparison.

Method	Test Set			Inference Time (ms)	No. of Parameters (M)
	IAM	LVDB	HP-OCW		
TRBA [1]	70.13	68.94	53.45	1.006	49.60
Ours	83.43	75.08	63.90	0.685	49.66

Experiments and Observations

Quantitative Observations. The strategic use of self-attention layers into our model architecture significantly enhances the model's performance, enabling to surpass the results achieved by the existing state-of-the-art CNN backbone SOTA method, TRBA [1]. A noteworthy aspect of our model's enhancement is not just the improvement in accuracy but also its efficiency. We markedly reduce the time it takes for the model to make a prediction (inference time) by about 38%, while only slightly increasing the model's size by about 0.12%, in terms of a number of parameters. To provide a clear and detailed comparative analysis, we detail our findings in Table 2. This table displays the word-level accuracy (expressed in percentages) of the models when evaluated on the IAM [23], LVDB [10] and HP-OCW datasets. The inference time of the recognition model is computed on an Nvidia RTX 2080 GPU.

Qualitative Observations. In Figure 5, we present a comprehensive visual representation showcasing the end-to-end recognition capabilities of our proposed method for both handwriting and printed text recognition. This includes both handwritten, which can vary widely in style, structure, as well as printed text images that encompass a range of fonts, layouts, and complexities. These example images serve to demonstrate the efficacy and versatility of our the the proposed method. The images of handwritten and printed texts captured in Figure 5 are taken using the Logitech C920x HD Pro webcam.

Conclusion

In this paper, we examined the limitations of current CNN-based text and handwriting recognition systems that hinge on sequence-to-sequence frameworks. We presented an approach that amalgamates the rapidity of CTC decoding with the contextual modeling provided by the self-attention mechanisms, derivative of transformer networks. The resultant methodology, as evidenced, not only reduces inference time considerably but also surpasses the performance metrics of conventional attention-based decoders. Our system's architecture showcases a promising avenue towards near real-time text-recognition applications, merging efficiency with accuracy.

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, 2019.

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, 2019.

[3] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In *CVPR*, 2018.

[4] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, and Yi-Zhe Song. Towards the unseen: Iterative text recognition by distilling from errors. In *ICCV*, 2021.

[5] Ayan Kumar Bhunia, Shuvozit Ghose, Amandeep Kumar, Pinaki Nath Chowdhury, Aneeshan Sain, and Yi-Zhe Song. Metahtr: Towards writer-adaptive handwritten text recognition. In *CVPR*, 2021.

[6] Ayan Kumar Bhunia, Aneeshan Sain, Amandeep Kumar, Shuvozit Ghose, Pinaki Nath Chowdhury, and Yi-Zhe Song. Joint visual semantic reasoning: Multi-stage decoder for text recognition. In *ICCV*, 2021.

[7] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[8] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, 2017.

[9] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In *CVPR*, 2018.

[10] Kartik Dutta, Minesh Mathew, Praveen Krishnan, and C. V. Jawahar. Localizing and recognizing text in lecture videos. In *ICFHR*, 2018.

[11] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006.

[12] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] M Jaderberg, K Simonyan, A Vedaldi, and A Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *NeurIPS*, 2014.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, 2015.

[16] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. Icdar 2015 competition on robust reading. In *ICDAR*, 2015.

[17] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazàn Almazàn, and Lluís Pere de las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013.

[18] Taewook Kim, Gaurav Patel, Qian Lin, Jan P. Allebach, and Qiang Qiu. Generalizing handwriting and scene-text detection in images. In *Electronic Imaging*, 2024.

[19] Alex W. C. Lee, Jonathan Chung, and Marco Lee. Gnhk: A dataset for english handwriting in the wild. In *ICDAR*, 2021.

[20] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, 2016.

[21] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[22] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene

text recognizer. In *CVPR*, 2020.

[23] Urs-Viktor Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[24] Anand Mishra, Alahari Karteek, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

[25] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *ICDAR*, 2019.

[26] Gaurav Patel, Jan P. Allebach, and Qiang Qiu. Seq-ups: Sequential uncertainty-aware pseudo-label selection for semi-supervised text recognition. In *WACV*, 2023.

[27] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, 2013.

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

[29] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[30] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, 2016.

[31] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017.

[32] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *CVPR*, 2021.

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[35] Kentaro Wada. labelme: Image polygonal annotation with python. `https://github.com/wkentaro/labelme`, 2018.

[36] Zhaoyi Wan, Jielei Zhang, Liang Zhang, Jiebo Luo, and Cong Yao. On vocabulary reliance in scene text recognition. In *CVPR*, 2020.

[37] Jianfeng Wang and Xiaolin Hu. Convolutional neural networks with gated recurrent connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[38] Kai Wang, Boris Babenko, and Serge J. Belongie. End-to-end scene text recognition. *International Conference on Computer Vision*.

[39] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, 2019.

[40] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, 2019.

[41] Moonbin Yim, Yoonsik Kim, Han-Cheol Cho, and Sungrae Park. Synthtiger: Synthetic text image generator towards better text recognition models. In *ICDAR*, 2021.

[42] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, 2020.

[43] Fangneng Zhan, Shijian Lu, and Chuhui Xue. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In *ECCV*, 2018.

[44] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *ICDAR*, 2019.

[45] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *CVPR Workshops*, 2017.

Author Biography

Gaurav Patel *received his Bachelor of Technology (B.Tech.) degree in Electronics and Communication Engineering from National Institute of Technology Raipur, India in 2020. He is currently pursuing his Ph.D. in the Elmore Family School of Electrical and Computer Engineering at Purdue University. His research interest include Computer Vision, Machine Learning (Deep Learning), and Artificial Intelligence.*

Taewook Kim *is a Ph.D. student in Electrical and Computer Engineering at Purdue University, where he focuses on research in the fields of computer vision and machine learning. He earned his Bachelor's degree in Computer Engineering from Hongik University in South Korea, and his Master's degree in Computer Science and Engineering from Pohang University of Science and Technology (POSTECH) in South Korea.*

Qian Lin *is an HP Fellow working on computer vision and deep learning research. She is also an adjunct professor at Purdue University. She joined Hewlett-Packard Company in 1992. She received her BS from Xi'an Jiaotong University in China, her MSEE from Purdue University, and her Ph.D. in Electrical Engineering from Stanford University. She is inventor/co-inventor for 45 issued patents. She was awarded Fellowship by the Society of Imaging Science and Technology (IS&T) in 2012, Outstanding Electrical Engineer by the School of Electrical and Computer Engineering of Purdue University in 2013, and the Society of Women Engineers Achievement Award in 2021.*

Jan P. Allebach *is the Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. He was named Electronic Imaging Scientist of the Year by IS&T and SPIE, and was named Honorary Member of IS&T, the highest award that IS&T bestows. He has received the IEEE Daniel E. Noble Award, the IS&T/OSA Edwin Land Medal, the IS&T Johann Gutenberg Prize, is a Fellow of the National Academy of Inventors, and is a member of the National Academy of Engineering.*

Qiang Qiu *is a researcher with expertise in computer vision and machine learning, focusing on deep learning, image understanding, and representation learning. He holds a Ph.D. in Computer Science from the University of Maryland, College Park. He currently serves as an Assistant Professor at Purdue University and has previously worked as an Assistant Research Professor at Duke University.*