LAL: Linguistically Aware Learning for Scene Text Recognition

Yi Zheng, Wenda Qin, Derry Wijaya, and Margrit Betke
Computer Science Department
Boston University
<yizheng,wdqin,wijaya,betke>@bu.edu

ABSTRACT

Scene text recognition is the task of recognizing character sequences in images of natural scenes. The considerable diversity in the appearance of text in a scene image and potentially highly complex backgrounds make text recognition challenging. Previous approaches employ character sequence generators to analyze text regions and, subsequently, compare the candidate character sequences against a language model. In this work, we propose a *bimodal* framework that *simultaneously* utilizes visual and linguistic information to enhance recognition performance. Our *linguistically aware learning (LAL) method* effectively learns visual embeddings using a rectifier, encoder, and attention decoder approach, and linguistic embeddings, using a deep next-character prediction model. We present an innovative way of combining these two embeddings effectively. Our experiments on eight standard benchmarks show that our method outperforms previous methods by large margins, particularly on rotated, foreshortened, and curved text. We show that the bimodal approach has a statistically significant impact. We also contribute a new dataset, and show robust performance when LAL is combined with a text detector in a pipelined text spotting framework.

CCS CONCEPTS

• **Computing methodologies** → **Object recognition**; *Natural language processing*.

KEYWORDS

text recognition, bimodal, visual, linguistic, deep learning

ACM Reference Format:
Yi Zheng, Wenda Qin, Derry Wijaya, and Margrit Betke. 2020. LAL: Linguistically Aware Learning for Scene Text Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413913

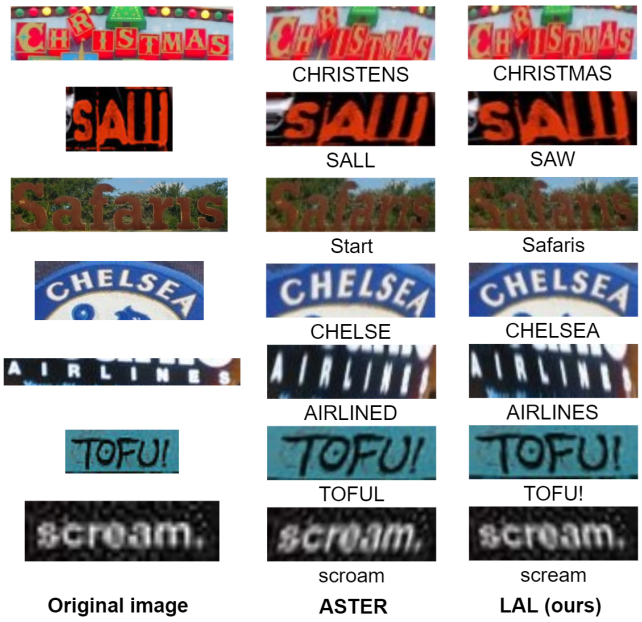

Figure 1: The comparison of the proposed linguistically aware learning (LAL) method and the rectification based method ASTER [35]. The second and third columns give the predictions of ASTER and LAL on the rectified images respectively. State-of-the-art methods may encounter difficulties when the appearance of the text is irregular ("CHRISTMAS", "SAW"), the text background is complex ("Safaris"), the characters ("!" vs. "l") are difficult to distinguish ("TOFU!"), or the image is blurred ("scream"). Our LAL approach addresses these problems.

1 INTRODUCTION

Scene text recognition is a research problem that has attracted significant interest due to its importance to various tasks, such as assisting the visually impaired, scene understanding for autonomous cars, and image retrieval. While Optical Character Recognition (OCR) techniques work well on printed document images, scene text recognition remains a challenging problem due to large variations in text appearance, layout, and background. "Regular text," i.e., text with horizontally aligned characters, can be recognized with convolutional [15] and recurrent neural networks [33]. Recognizing "irregular text" (Fig. 1) is more difficult. This includes *rotated text*, i.e., text that is not horizontally aligned with the image rows (Fig. 1, row 5), and *foreshortened text* due to perspective projection. Moreover, the text design itself may not be straight but contain characters aligned along a curve (Fig. 1, row 4) or have misaligned characters (Fig. 1, row 1). Images of such text designs may then also be rotated and foreshortened. Created without the ability to handle irregular text, the methods mentioned above often struggle in recognizing irregular text.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413913

Recognition of irregular text has been addressed by two main lines of research: (1) rectification based [34, 35] and (2) 2D feature map based [6, 20] approaches. Rectification based methods [34, 35] adapt Spatial Transformer Networks (STN) [16] to rectify irregular text images into regular shapes, and then recognize them with regular text recognizer. 2D feature map based methods learn 2D feature maps from the original input image without any rectification and generate characters sequentially in 2D space. For example, the method by Cheng et al. [6] encodes 2D space information from four directions to transform the 2D image feature maps to a 2D feature sequence. Li et al. [20] proposed a 2D attention mechanism that encodes the 2D feature map column by column to yield a holistic feature vector and then decodes this holistic feature vector by applying a 2D attention mechanism on the image feature maps.

Most state-of-the-art methods use solely visual information to recognize irregular text. A character sequence, however, is designed to convey linguistic information to the observer, and this linguistic information, in addition to the visual information, should be utilized. Our approach is therefore to fuse visual and linguistic information. Inspired by the Transformer method [37], which has made profound advances in solving various tasks in natural language processing (NLP), we propose a new, linguistically aware learning (LAL) approach that incorporates a transformer-based model.

The input of LAL are cropped images of text. In practice, a text recognizer like LAL must be used together with a text detector that localizes regions of text, crops them from a full scene image, and passes them to the text recognizer. The literature calls the combination of a text detector and a text recognizer a text spotting system [39]. The performance of text recognizers is typically only analyzed on perfectly cropped images of text, i.e., ground-truth bounding boxes or polygons. When text recognizers are combined with text detectors, however, they encounter more challenging inputs, i.e., images of text that may not be cropped perfectly or, if the text detector completely fails, images that do not even contain text. We propose a two-stage text spotting system, called LAL*, to show the practicality of LAL and its robust performance when combined with a text detector.

The main contributions of this paper are:

– We designed the scene text recognition model LAL that learns a robust text representation by combining image and language information using an innovative multi-network approach. To the best of our knowledge, we are the first to propose to leverage word-analysis techniques from the NLP community in this way to solve the recognition task studied by the computer vision community.

– Experimental analysis of the efficacy of LAL on eight datasets and comparison with ten previous methods show that, with help of linguistic information, LAL outperforms state-of-the-art methods by large margins. We also show that the use of bimodality has a statistically significant impact.

– We constructed a text spotting system, called LAL*, by combining LAL with an existing text detector, which yields robust recognition performance on a widely used benchmark.

2 RELATED WORK AND MOTIVATION OF OUR APPROACH

This section focuses on discussing previous works on recognizing **irregular text**. Early methods [29, 38, 42] detect and recognize each individual character and then group characters into words. Mistakes in character detection and classification led to limited recognition performance. 1D sequence-to-sequence (seq2seq) approaches [19, 33, 34], inspired by speech recognition, were then introduced, which extract 1D features and writing direction from the input image and transform the text into sequences of characters. These methods fail to recognize rotated or curved text.

The methods by Shi et al. [34, 35] transform the text image into a canonical shape with a spatial transformer network (STN) and recognize the rectified text image using a 1D attentional seq2seq model. Instead of rectifying the original input image, the method by Liu et al. [23] detects and rectifies the individual characters recurrently. Other methods [6, 20, 27, 41] bypass rectification and instead recognize irregular text directly from the input image. Cheng et al. [6] adapted the sequence-based model with the image feature extracted from four directions to recognize arbitrarily-oriented text. Other methods [20, 27, 41] handle irregular text by applying a 2D attention mechanism on feature maps.

Scene text recognition methods [6, 20, 27, 34, 35, 41] utilize sequence generators that sequentially attend to certain regions on either 1D or 2D feature maps, following the character order in the text. They still suffer from losing some visual information due to pooling in the CNN or attention drift in the RNN, thus being inherently biased towards horizontally aligned text. To address this issue, we propose a bimodal solution that adopts the self-attention mechanism [37] and applies it to text images. It enables character features, obtained visually, to also encode the underlying linguistic information and supports the sequence generator to predict characters without any additional supervision.

The seq2seq model is one of the most common models in NLP to deal with various text-related tasks. Initially, RNNs were used for NLP seq2seq modeling [28]; then "Long Short Term Memory" (LSTM) [14][36] models were preferred. Recently, a seq2seq model called "Transformer" [37], which involves a self-attention mechanism, has been shown to outperform LSTM models for NLP tasks such as machine translation.

By connecting the vision system with a seq2seq language model trained on a large corpus, we discovered that NLP features provide helpful information that improves the performance of the whole text recognition system. Specifically, our language model is a model that predicts every subsequent character of an incomplete word, given all characters before the one to be predicted. Since we use a seq2seq framework, we tried all three above-mentioned models for our task (RNN, LSTM, and Transformer). We chose a Transformer as our final model, as it provides the lowest perplexity (perplexity reveals whether a model performs well in certain NLP tasks such as language modeling).

3 METHOD

We now describe our bimodal scene text recognition model LAL in full detail, including our next character prediction model. We also discuss the limitation of training data used by previous works and

how we modified and supplemented the data, creating the dataset SynthText*, which boosts the performance of rectification of the scene text recognizer.

3.1 Architecture of LAL

The architecture of our bimodal feature learning model for text recognition, LAL, is shown in Fig. 2 (a). It consists of a rectifier, encoder, attention decoder described in this section and a next-character prediction (NCP) model described in Section 3.2.

Rectifier: To address the non-linear spatial arrangement of characters and perspective distortion issues in scene text images, we included a rectification module as the first component in LAL. The architecture for rectification is inherited by Shi et al. [35]. LAL uses a 6-layer CNN to predict 20 control points to localize the text on the input image and attempts to rectify the input images to axis-aligned form with a Thin-Plate-Spline (TPS) approach [3]. Some distortions may still remain (e.g., UNITED in Fig. 2). To maximize the efficacy of the rectification module, we created a new synthetic training dataset by modifying a widely-used public synthetic dataset, as discussed in detail in Section 5.1.

Encoder: The encoder (Fig. 2 (b)) processes the rectified image through a 45-layer ResNet [13] that captures local patterns and textures, followed by two layers of a Bidirectional LSTM (BiLSTM). Each layer consists of a pair of LSTMs with 256 hidden units. The BiLSTM captures long-range dependencies of the feature sequence in both directions and outputs the new feature sequence $H = [h_1, ..., h_L]$, where L is the maximum word length (a fixed parameter, determined in advance).

Attention Decoder: The decoder (Fig. 2 (c)) retrieves the feature sequence from the encoder to generate a sequence of characters. We propose a linguistically-aware attentional sequence-to-sequence model [2, 7] to align target and label. It works iteratively for L steps, producing a character sequence of length L, denoted by $(c_1, ..., c_L)$. At time step t, the output character c_t is

$$c_t = \text{softmax}(W_{out} s_t + b_{out}), \tag{1}$$

where s_t is the hidden state at time step t, and W_{out} and b_{out} are trainable weights. A Gated Recurrent Neural Network (GRU) [8] computes the hidden state

$$s_t = GRU(\text{concatenate}(g_t, l_t), s_{t-1}), \tag{2}$$

based on the previous state s_{t-1} and a concatenation of two embeddings g_t and l_t, which encode the visual and linguistic information, respectively. The "glimpse vector" g_t is calculated by

$$g_t = \sum_{i=1}^{L} \exp(\alpha_{t,i}, h_i), \tag{3}$$

where α_t is the vector of attentional weights:

$$\begin{aligned} \alpha_{t,i} &= \exp(e_{t,i}) / \sum_{i'=1}^{n} \exp(e_{t,i'}), \\ \text{with} \quad e_{t,i} &= w^T \tanh(W_s \boldsymbol{s}_{t-1} + W_h \boldsymbol{h}_i + b), \end{aligned} \tag{4}$$

where w, W_s, W_h, and b are trainable weights. Using the embedding l_t to encode linguistic information is our innovative contribution. State-of-the-art methods [20, 34, 35] embed the previous output c_{t-1} when computing state s_t, but we found that this is insufficient to represent language priors. To capture the dependencies between output characters from a linguistic view, LAL uses a next-character prediction (NCP) model, described in section 3.2, to compute the linguistic embedding

$$l_t = NCP(c_0, ..., c_{t-1}), \tag{5}$$

taking a character sequence $c_0, ..., c_{t-1}$ as input, which is a prefix in a word detection model that processes characters from the left of the text to its right.

Table 1: NCP model predictions of the next character given prefixes of the word "united" (only the 5 most likely predictions are shown here). Row 1 shows that the letter "u" is typically followed by a consonant in the English language, and most likely by the letter "n." The correct letter "n" is ranked first (correct predictions are shown in bold). ⟨eos⟩ is an indicator symbol meaning "end of sequence."

NCP Input Prefix	NCP Ranked Predictions				
	1*st*	2*nd*	3*rd*	4*th*	5*th*
u	**n**	s	p	k	.
un	**i**	d	t	c	l
uni	v	**t**	o	q	f
unit	**e**	s	⟨eos⟩	y	.
unite	**d**	⟨eos⟩	s	r	m
united	⟨**eos**⟩	.	,	’	:

3.2 Next-Character Prediction Model

We here describe how we designed the next-character prediction model (NCP) of our LAL method to effectively and efficiently extract linguistic information from sequential characters. The input that LAL first passes into NCP is a prediction of the first character of the text c_0, which LAL makes solely based on its visual analysis. Given this first character c_0, NCP can then predict a ranked list of characters $c_1^{1st}, c_1^{2nd}, \ldots$ that most likely follow as the second character of the text (see Table 1). NCP passes an embedding, i.e., the vector l_1, which represents the various choices of the second character c_1, to the attention decoder of LAL, where it is combined with visual information to recognize c_1 in the prefix (c_0, c_1). As more and more characters are given as input, NCP is able to obtain more information about the word and becomes more accurate in predicting the next character. Because of this increase of prediction accuracy, NCP provides the attention decoder of the LAL model increasingly helpful information through its hidden state l_t. NCP accepts 70 different characters (letters, digits, special symbols) as possible next characters. At the end of the word, NCP suggests an ⟨eos⟩ character for "end of sequence" or a punctuation mark. NCP also accepts a "padding" symbol ⟨pad⟩ that allows the model to be trained with fixed-length inputs.

The architecture of NCP is a Transformer model [37]. It contains four major components (Fig. 2(d)) that are described next:

Embedding Layer. This first layer of NCP receives a character sequence from the attention decoder of LAL as its input, and converts it into a 70-element sequence of indices, where each index represents a specific character. This is done with a look-up table

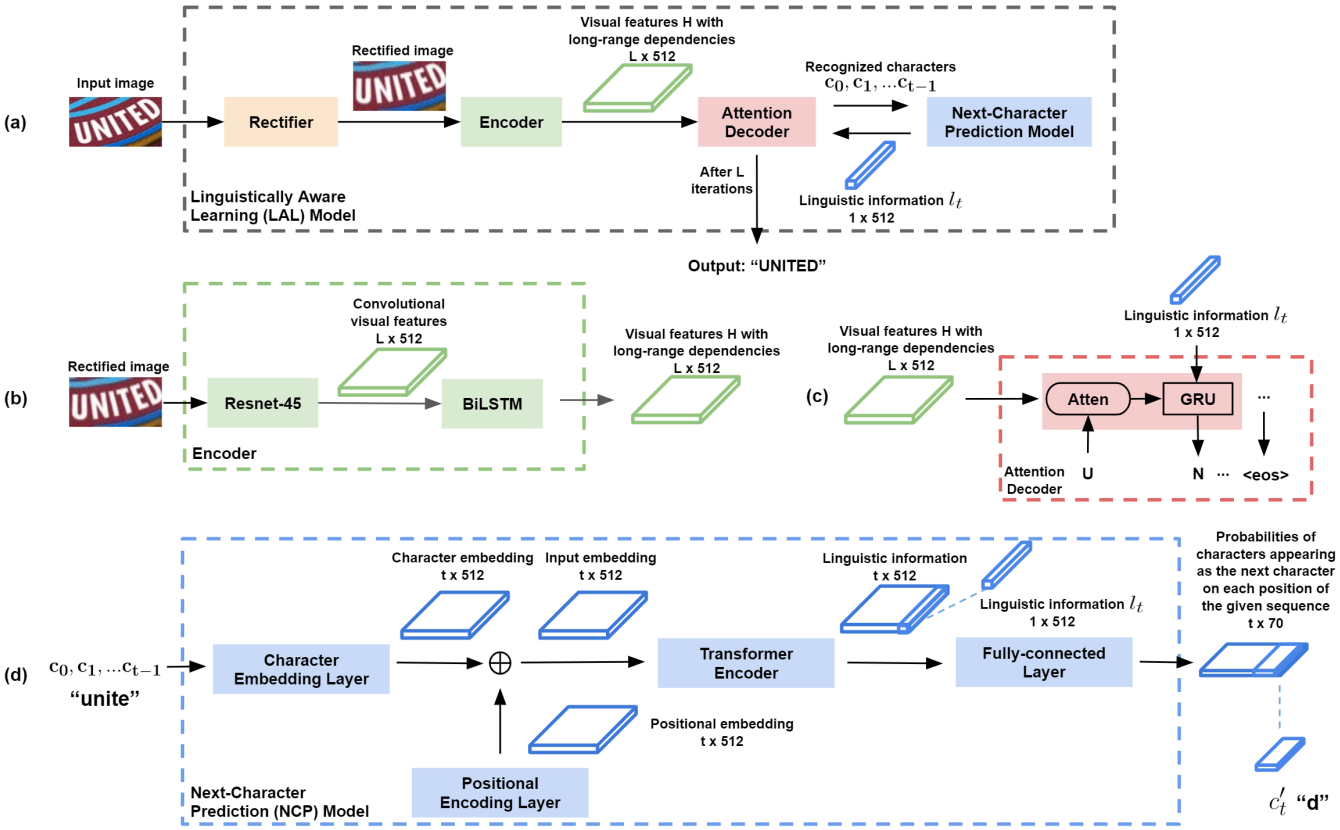

Figure 2: (a) Overview of the proposed LAL method. (b) Encoder. (c) Attention Decoder. (d) Next-Character Prediction Model. $\oplus$ - addition operator.

that is represented as a trainable weight matrix. The resulting index sequence can be considered as a character embedding vector.

Positional Encoding Layer. The second layer of NCP adds information of the absolute and relative position of a character in the embedding. We use the sinusoidal position encoding proposed by Vaswanit et al. [37]. The resulting vector is added to the embedding vector and passed into the Transformer Encoder.

Transformer Encoder. The transformer encoder is a stack of 6 identical layers, each having one multi-head attention sublayer, followed by another sublayer, which is a feed-forward network [37]. In our example of the embedding vector of the character sequence "united" (Fig. 2(d)), the attention sublayers evaluate, based on embedding vectors that represent characters, how important each of the characters in "unite" is to predict possible subsequent characters, including "d." The multi-head attention sublayers compute attentions in parallel $m = 8$ times, each time with different parameters, then concatenate them and linearly transform them into a representation that is passed to the feed-forward network, which produces the embedding vector l_t (Eq. 5).

The methodological contribution of our work is to have envisioned that the embedding vector l_t, which represents linguistic information in the text image, can be passed into the GRU of the Attention Decoder (Fig. 2(d)), which processes the visual information of the text image. We have thus found an innovative way to compute a bimodal analysis of the text image at each step t through the character sequence.

Fully-connected layer. This layer transforms the embedding vector l_t linearly into a 70-element output vector, where each element represents the probability that a particular character is present at step t in the character sequence. This layer is not employed when LAL is in use mode (i.e., when we test LAL) because LAL directly works with the embedding l_t. It is only needed for training the NCP model, as the probabilities are used to compute the loss between prediction and ground truth in backpropagation.

Training NCP. The NCP model is pretrained and frozen during the training of LAL. To train NCP, we used the cross-entropy loss and stochastic gradient decent (SGD) with a learning rate of 0.05. The batch size for training is 32. NCP is trained on a subset of words in the "enwiki" Wikipedia dataset [10]. We chose a subset to reduce the training time. We only train on words that occur in the collected enwiki corpus at least 240 times (for comparison, the word "coffee" appears 2,341 times, the word "is" 1,474,645 times). We consider 240 to be suitable cutoff of infrequent words, yielding a size of the training set that is small enough to ensure that the training process

can converge in a reasonable time. In particular, it took 6 hours to train NCP with a RTX 2080 TI GPU.

Testing NCP. To evaluate how well NCP predicts a character sequence $C = c_0, \ldots, \langle \text{eos} \rangle$, we used the

$$\text{perplexity} = e^{H(C_{gt}, C)}, \tag{6}$$

where $H(C_{gt}, C)$ is the cross-entropy between the ground truth character sequence C_{gt} and predicted sequence C. Perplexity can be considered to measure the inverse of the probability of predicting the next character. Since the number of characters that can be predicted is 70, i.e., 26 letters, 10 digits, and various special characters, an inverse of the uniform probability would be 70. We thus aim for a perplexity that is lower than 70. We randomly chose 1,000 words from "enWiki" as testing data. We repeated the testing process 10 times and computed the average perplexity of each character in a word. The testing result shows that the average perplexity of our trained NCP model is 2.57, which is favorably much lower than our bound of 70.

4 DATASET: SYNTHTEXT*

Supervised training of scene text recognition models based on deep CNNs or RNNs requires a very large number of labeled training data since these models contain millions of parameters. Not only are inadequate numbers of data insufficient to train deep learning models, they also limit how variations in text font, size, and position in natural images can be represented. Consequently, two large datasets have been created synthetically and become widely used by researchers to train text recognizers: **Syn90k** [15] is a 9-million synthetic dataset generated based on 90k generic English words. It has a large vocabulary but the images are monochrome (Fig. 3 top), so it is used with **SynthText** [12] to provide realistic colored scene images (Fig. 3 middle).

Figure 3: Samples from Syn90k [15] (top row), SynthText [12] (middle) and our SynthText* (bottom). For SynthText and SynthText*, text images are cropped from a synthetic background image with text.

We found that SynthText has a relative paucity of curved text instances. Our solution is to use the SynthText generating engine and the same background images along with their segmentation and depth masks provided by Gupta et al. [12] and generate additional, realistic curved text images. We refer to the enhanced dataset as **SynthText*** (Fig. 3 bottom). The modifications are as follows:

(1) We increased the proportion of curved text to about 50% by combining non-curved and curved text images after synthesis. We estimate that the portion of curved words of SynthText ranges between 20%–25%, given that the engine only renders a sample of text containing single words with less than 10 characters as a curve.

(2) To render text as a curve, SynthText places characters one by one symmetrically around the original point following a parabolic trajectory. To give us more flexibility in creating curved words that appear in real-word text designs (e.g., half-circle designs), we replaced the parabolic trajectory with an elliptic trajectory.

(3) We randomly added some motion blur since we found that motion blur is common in the training data of the scene text recognition benchmark datasets (see Section 5).

5 EXPERIMENTS

To verify the effectiveness of the proposed method, we conducted experiments on seven widely-used benchmarks. We also performed a detailed analysis of the ability of LAL to process linguistic information provided by NCP. We conducted an ablation study to evaluate the design choice we made for LAL to process text solely in the left-to-right direction (as opposed to bidirectional). Finally, we report the results of our text spotting system LAL*.

5.1 Implementation Details

We trained LAL on Synth90K and SynthText* jointly from scratch. Images were resized to 64×256 before entering the rectification network, which outputs images of size 32×100 as the input to the recognition network. We used ADADELTA [43] to optimize our model with a batch size of 512 for 200K iterations. Empirically, we set the initial learning rate to 1 and decreased it to 0.1 and 0.01 at iteration 100K and 150K, respectively. No dropout was used. The decoder can recognize 70 character classes, including 10 digits (0–9), 26 lower-case English characters (a–z, the case is ignored), 32 ASCII punctuation marks, and the symbols ⟨eos⟩ and ⟨pad⟩. Since LAL recognizes 70 distinct characters, we set the input and output sizes of the NCP model to 70 to match. The maximum length of words to be trained is $L = 20$ (for longer words, only the first 20 characters are kept).

We implemented our method under the framework of PyTorch [30]. The model was trained on two NVIDIA 1080 ti graphics cards with 12 GB memory. The training speed is about 1.4 iterations/s, taking less than 40 hours to reach convergence. The inference speed is 26.7 frames per second.

5.2 LAL Performance on 7 Benchmark Datasets Compared to 10 State-of-the-Art Methods

Seven benchmark datasets are widely used for evaluation of scene text recognition models. According to recognition difficulty and geometric layout, we divide them into two groups, "Regular" and "Irregular."

Regular: IIIT5K-Words (IIIT5K) [29] is a dataset for scene text recognition. It consists of 5,000 images, of which 3,000 images are used for the test. Each image associates with a 50-word lexicon and a 1,000-word lexicon respectively. Street View Text (SVT) [38] consists of 647 word images cropped from Google Street View for testing. Each image associates with a 50-word lexicon. ICDAR2003 (IC03) [26] contains 867 cropped text images taken in a mall. ICDAR 2013 (IC13) [18] has 1,015 cropped word images for testing. No lexicon is provided.

Irregular: ICDAR 2015 (IC15) [17] has 2,077 cropped word images for test. No lexicon is provided. CUTE80 (CT) [32] contains 288 high-resolution curved word images. Street View Text Perspective (SVTP) [31] contains 238 street images, which were cropped to

Table 2: Scene text recognition accuracy (%) on seven benchmark datasets, as reported in the literature. These are lexicon-free results which means the predicted words were not corrected by dictionary matching. "Regular" means the datasets consist of horizontally aligned text, and "irregular" means they include foreshortened, rotated, and curved text.

Method	Dimension of Feature Map	Rectification	Regular				Irregular		
			IIIT5K	SVT	IC03	IC13	IC15	SVTP	CT
CRNN [33]	1D	no	78.2	80.9	-	86.7	-	-	-
RARE [34]	1D	no	81.9	81.9	-	-	-	71.8	59.2
STAR-Net [24]	1D	no	83.3	83.6	-	89.1	-	73.5	-
FAN [5]	1D	no	87.4	85.9	94.2	93.3	-	-	-
ASTER [35]	1D	yes	93.4	89.5	-	91.8	76.1	78.5	79.5
ESIR [44]	1D	yes	93.3	**90.2**	-	-	76.9	79.6	83.3
Attentive Text Recognition [41]	2D	no	-	-	-	-	-	75.8	69.3
AON [6]	2D	no	87.0	82.8	91.5	-	68.2	73.0	76.8
CA-FCN [22]	2D	no	92.0	82.1	-	91.4	-	-	79.9
SAR [20]	2D	no	91.5	84.5	-	-	69.2	76.4	83.3
LAL without NCP	1D	yes	94.4	87.5	92.5	93.8	76.4	79.5	84.7
LAL	1D	yes	**95.0**	89.8	**94.3**	**95.1**	**79.0**	**82.9**	**87.8**

yield 645 word images with a great variety of viewpoints. SVTP is specifically designed for perspective text recognition.

LAL Performance. We report the performance of LAL and 10 existing methods, listed in Table 2, on the above described 7 datasets. We divided the methods into two groups based on the dimensionality of feature maps and report whether a method uses rectification. To evaluate the impact of our bimodal approach, we also tested LAL without NCP (i.e., processing only visual information). LAL obtains the highest accuracy numbers compared to the ten existing methods on six of seven benchmarks. For three of the four datasets with regular text, our model achieves accuracy levels above 94%, improving over the state-of-the-art. On the three datasets with irregular text, LAL improves upon the best existing method with a margin of 3.3 percentage points (pp) on average. LAL without NCP is consistently less accurate than LAL (2 pp on average). Since the size of the datasets varies from 288 to 3,000 samples, to analyze whether the difference between LAL with and without NCP is statistically significant, we used the N-1 Chi-Squared test [4, 11] and report p-values (Table 3). Applying the conventional threshold for declaring statistical significance to be a p-value of less than 0.05, we found that using the bimodal approach is statistically significant on the larger datasets IIIT5k and IC15 (over 2,000 samples).

5.3 Experiments with Highly Challenging Data

Given the high accuracy rates for recognizing regular text, ongoing research efforts should focus on irregular text. However, the benchmarks for irregular text have only about a total of 3,000 samples (while regular text data sets have about 5,500 samples), and include only machine-printed text. To increase the number of challenging images and the level of difficulty, we looked for a dataset that included hand-written text, which commonly occurs in daily life. We decided to focus on the 11,532-image "Text-containing Protest Image Dataset" (TPID) [45], which is based on a subset of the UCLA Protest Image Dataset [40], a collection of social media images that can be used to analyze protest activities in street scenes. The 816 scene images used to create TPID contain mostly hand-made protest signs with text that is hand-written. TPID contains 11,532 cropped

Table 3: P-value ($\times 10^{-2}$) on seven benchmark datasets, computed for LAL with and without NCP to show that NCP significantly improves recognition accuracy. The same training procedures were used.

Dataset	IIIT5K	SVT	IC03	IC13	IC15	SVTP	CT
P-value	**2.59**	9.45	6.06	10.42	**2.21**	5.84	13.81

word images and annotations with ground-truth polygons and textual representations of all the words on every protest sign. Figure 4 shows a random sample of these hand-written word images, as well as as examples from the "irregular text" datasets.

Figure 4: Top: selected samples of IC15, SVTP and CT. Bottom: selected samples of TPID, most text is hand-written.

We compare LAL against two representative models from two main lines of research, ASTER [35] (a rectification based model) and SAR [20] (a 2D feature map based model), which we re-implemented. To enable fair comparisons, we used the same optimization and pre-processing procedures and the same training dataset (Syn90k and SynthText* combined). The results on four test datasets are given in Table 4.

A comparison of the accuracy values on the three benchmarks reported in Table 2 (last three columns) and Table 4 show that we must have successfully re-implemented ASTER and SAR, since our implementation yields higher accuracy values than the values reported in the literature. The higher accuracy values also indicate that Synthtext* boosts the performance of ASTER and SAR on

Table 4: Text recognition results of LAL on three irregular text benchmarks and TPID (accuracy percentage). All models were trained using Syn90k and SynthText* jointly.

Model	IC15	SVTP	CT	TPID
ASTER	76.4	79.5	84.7	68.3
SAR	75.7	78.0	84.7	68.4
LAL	**79.0**	**82.9**	**87.8**	**70.3**

Table 5: Recognition accuracy of LAL without NCP on eight benchmarks. Optimization and pre-processing as before.

Regular Text Datasets	IIIT5K	SVT	IC03	IC13
Syn90k+SynthText	93.9	89.8	92.5	94.1
Syn90k+SynthText*	94.4	87.5	92.5	93.8
Irregular Text Datasets	**IC15**	**SVTP**	**CT**	**TPID**
Syn90k+SynthText	76.0	78.3	79.5	66.6
Syn90k+SynthText*	76.4	79.5	84.7	68.3

irregular text. LAL improves the accuracy on TPID by 1.9 pp (last column of Table 4), with a p-value of 9×10^{-4}, which indicates we are 99.91% confident that this improvement of LAL on TPID over the other models has statistical significance.

5.4 Detailed Analysis of LAL: Ablation Studies

Sections 5.2 and 5.3 show that LAL outperforms prior scene text recognition methods on eight public benchmarks. In this section, we study (1) the contribution of the NCP model through a thorough quantitative comparison against LAL without NCP, (2) the effect of SynthText* on rectification, and (3) the left-to-right order of character processing by the attention decoder.

LAL with NCP vs. LAL without NCP: Unlike prior state-of-the-art methods [34, 35], which use embeddings of single characters, NCP embeds sequential characters using linguistic knowledge learned from a large-scale text dataset, such as enWiki. This additional linguistic information is valuable since it extends the recognition ability from visual-only to visual-&-linguistic. With the help of this bimodal information, the performance of the text recognizer is significantly improved. An example is shown in Fig. 5.

When predicting "e" in "hotel" in the image in Fig. 5, due to perspective distortion and shadows, LAL with CV only, i.e., LAL that only uses visual information, predicts "i" with a low probability of 38.15%. However, NCP predicts "e" with a probability of 51.89%. With this additional linguistic information, LAL (CV+NCP) is able to predict the correct character with a probability of 58.83%, which is a 20.68 pp increase over LAL with CV only.

Another observation is that at early time steps, NCP predicts characters with a low probability. This is reasonable since the beginning of a word could be followed by various characters, and so the visual information plays a more important role to determine the character. At later time steps, NCP predicts characters with a high probability, and so NCP contributes more to predict the character.

Effect of SynthText* on Training LAL's Rectifier. We created the dataset SynthText* so we can train the neural net in the LAL rectifier module on curved text instances (see section 4). We

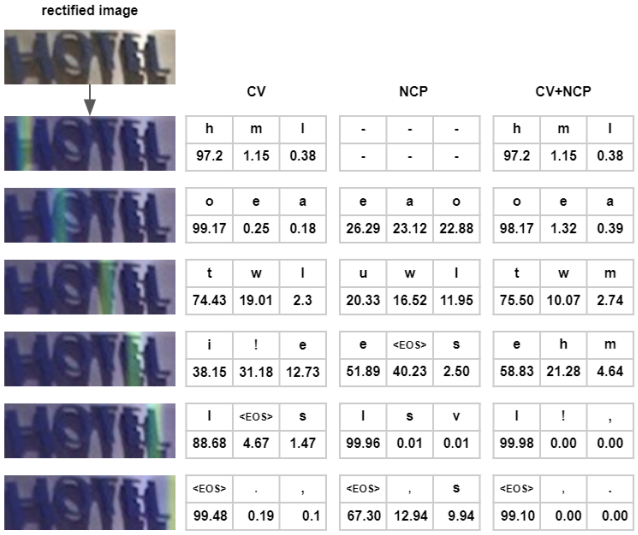

Figure 5: Visualization of the 1D attention weights (left) and probabilities (%) of the top three predicted characters for 3 versions of LAL at each time step. "CV" means the predicted character is based on visual information only. "NCP" means the predicted character is based on previous sequential characters. "CV+NCP" represents the predicted character based on the combination of visual and linguistic information.

were motivated to create SynthText* because previous training datasets, Syn90k and SynthText, contain few curved text instances. To disentangle the effects of rectification and linguistic information and enable a fair comparison of training with our new SynthText* versus the existing SynthText, we used LAL without NCP. We tested on eight benchmarks and report superior performance of training the SynthText* in six of eight benchmarks in Table 5. Specifically, we make two observations from Table 5: First, the recognition accuracy of LAL when trained on SynthText and SynthText* is similar for tests on regular text datasets (IIIT5K, SVT, IC03, IC13) with few curved text instances. This finding is reassuring because the design goal of SynthText* was to improve the training of the rectification module, and so, for regular datasets (that do not require rectification), there should not be a large recognition difference between using Synthtext* and Synthtext. Second, SynthText* boosts the performance on irregular text data (IC15, SVTP, CT, TPID) as much as 1.7 pp over 11,532 samples (we report the improvement with 99.7% confidence).

The increase of recognition accuracy on irregular text between Tables 2 and 4 suggests that previous state-of-the-art methods could also benefit from training on SynthText*.

Ablation Studies on Attention Decoder. The decoder of LAL captures output dependencies in the left-to-right (L2R) order. We observed that the NCP model can have difficulties deciding on the first few characters of a word (see section 5.4). It is worth checking if this problem could be alleviated by including a decoder that works in the right-to-left (R2L) order, which would capture dependencies in the opposite direction and presumably become more accurate in predicting the leftmost characters of a word. To compare the impact

Table 6: Recognition accuracy of LAL with decoders in different directions.

Direction	IIIT5K	SVT	IC03	IC13	IC15	SVTP	CT
L2R	**95.0**	**89.8**	**94.3**	**95.1**	**79.0**	**82.9**	**87.8**
R2L	94.5	87.5	91.9	94.6	76.7	78.7	85.1
L2R+R2L	94.5	88.1	93.5	93.9	77.3	80.9	85.4

of processing direction on the decoder, inspired by Shi et al. [35], we built three model variants: (1) L2R recognizes text in the left-to-right order; (2) R2L recognizes text in the right-to-left order; (3) L2R+R2L consists of both and outputs the result with the highest recognition probability from L2R or R2L. All models, including reverse NCP, which was trained on reverse words in "enwiki," were trained from scratch with the same protocol, described in sections 3.2 and 5.1. Recognition accuracy is shown in Table 6.

It is notable that the NCP model can learn the patterns of the English language in different ways, L2R, R2L, or bidirectional. However, L2R outperforms the other models on all the datasets. For the English language, linguistic form and structure provide the key linguistic information for text recognition, and L2R can represent English better than R2L.

In the bidirectional approach, the result with the highest probability returned by L2R or R2L is selected. However, both L2R and R2L could produce incorrect results with higher probabilities than the correct result. There is no guarantee that such a bidirectional approach could outperform either L2R or R2L because it actually learns two kinds of linguistic information in two directions instead of a unified one.

5.5 Results of Text Spotting System LAL*

We constructed a two-stage text spotting (TS) system, called LAL*. "Two-stage" means that the system consists of a detector and a recognizer in a pipeline. For LAL*, we use LAL as the text recognizer and CRAFT [1], obtained from its official Github repository [9], as the text detector. CRAFT effectively detects arbitrarily-oriented, curved, or deformed text in scene images. We compare LAL* to two text spotting systems, which are representative for different types of text spotting systems, FOTS and ASTER* (Table 7). FOTS [25] is an end-to-end trainable model that can detect and recognize text by sharing convolutional features. ASTER* is a two-stage text spotting system constructed with TextBoxes [21] and ASTER [35].

We trained and tested LAL*, FOTS, and ASTER* on the ICAR 2015 dataset [17], a common benchmark for text spotting, which includes 1,000 training and 500 testing images. More specifically, for LAL*, we only fine-tuned the detection component, CRAFT, with the ICDAR 2015 data, keeping our recognition component, LAL, fixed (after training on Syn90k and SynthText*). Both FOTS and ASTER*, however, used the 1,000 ICDAR 2015 training images. The ICDAR 2015 dataset provides four difficulty levels for testing, defined by lists of words (lexica) that the text spotting system can use for reference in the test phase: “Strong” (100 words per-image including all words that appear in the image), “Weak” (all words that appear in the entire test set), “Generic” (Syn90k), and "No" (no lexicon). It also identifies two testing protocols (word spotting and end-to-end). From the results shown in Table 7, we conclude

Table 7: Text spotting (TS) results (F-measure %) on ICDAR 2015 for two evaluation protocols “Word Spotting” and “End-to-End” and four testing levels, “Strong” (S), “Weak” (W), “Generic” (G), and "No lexicon" (N).

TS	Word Spotting				End-to-End			
System	S	W	G	N	S	W	G	N
FOTS	84.7	79.3	63.3	-	81.1	75.9	60.8	-
ASTER*	75.2	71.3	**67.6**	-	70.6	67.3	**64.0**	-
LAL*	**85.6**	**81.6**	64.2	**73.4**	**81.6**	**78.3**	61.9	**71.3**

that our text spotting system, LAL*, beats the results reported in the literature for FOTS and ASTER* in 4 of 6 cases. Notably, LAL* without lexicon lookup at test time beats LAL* with lookup in the generic lexicon by 9.4 pp. The lookup, designed by the ICDAR 2015 challenge organizers to simplify the spotting task, here fails to help and instead has the opposite effect on recognition accuracy. We found that some correctly recognized words were changed to incorrect words that appear in the lexicon.

6 CONCLUSIONS

Recognition of irregularly shaped text in scene images has been addressed in the computer vision community mostly as a single-modality problem, processing visual information only. The paper instead proposes the scene text recognition method LAL with an explicit linguistic-based approach. By utilizing visual and linguistic dependencies through a self-attention mechanism, LAL is able to sequentially predict characters. If the networks in LAL that process visual information struggle to predict correct characters, the networks that process linguistic information make up for it. We showed that this effect of bimodality has statistical significance.

LAL sets a new bar for state-of-the-art performance on text recognition benchmarks, achieving accuracy levels that are substantially higher than those of previous methods (an average of 3.3 percentage points on irregular text datasets). LAL also shows better performance on the highly challenging TPID than previous methods.

The strong performance of our text spotting system LAL*, when tested on data unseen to LAL, shows that LAL is robust and generalizes well without any fine-tuning, and can be effectively included in a two-stage text spotting system.

Our finding that training with our new downloadable dataset SynthText* has the potential to increase the recognition accuracy of previous methods on irregular text is important because our new dataset could propel the research efforts by other teams. Similarly, to support others in benefiting from using our approach to combine processing bimodal information, we open-source our code and SynthText* at https://github.com/ivc-yz/LAL.

ACKNOWLEDGMENTS

We acknowledge grants by NSF (1838193) and ONR (MURI N00014-19-1-2571, associated with AUSMURIB000001) and the Boston University Hariri Institute for Computing for partial support.

REFERENCES

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019. Character Region Awareness for Text Detection. In CVPR. Computer Vision Foundation / IEEE, 9365–9374.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. http://arxiv.org/abs/1409.0473 cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

[3] F. L. Bookstein. 1989. Principal warps: thin-plate splines and the decomposition of deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 6 (June 1989), 567–585. https://doi.org/10.1109/34.24792

[4] Ian Campbell. 2007. Chi-squared and Fisher-Irwin tests of two-by-two tables with small sample recommendations. Statistics in medicine 26 19 (2007), 3661–75.

[5] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. 2017. Focusing Attention: Towards Accurate Text Recognition in Natural Images. CoRR abs/1709.02054 (2017). arXiv:1709.02054 http://arxiv.org/abs/1709.02054

[6] Zhanzhan Cheng, Xuyang Liu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. 2017. Arbitrarily-Oriented Text Recognition. CoRR abs/1711.04226 (2017). arXiv:1711.04226 http://arxiv.org/abs/1711.04226

[7] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. 2015. Attention-Based Models for Speech Recognition. CoRR abs/1506.07503 (2015). arXiv:1506.07503 http://arxiv.org/abs/1506.07503

[8] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. CoRR abs/1412.3555 (2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

[9] CRAFT 2019. Official implementation of Character Region Awareness for Text Detection (CRAFT). https://github.com/clovaai/CRAFT-pytorch.

[10] enwiki 2020. Wikimedia downloads: enwiki datasets. https://dumps.wikimedia.org/backup-index.html.

[11] Karl Pearson F.R.S. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 50, 302 (1900), 157–175. https://doi.org/10.1080/14786440009463897

[12] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. 2016. Synthetic Data for Text Localisation in Natural Images. CoRR abs/1604.06646 (2016). arXiv:1604.06646 http://arxiv.org/abs/1604.06646

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. CoRR abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.

[15] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. CoRR abs/1406.2227 (2014), 10. http://arxiv.org/abs/1406.2227

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. 2015. Spatial Transformer Networks. CoRR abs/1506.02025 (2015). arXiv:1506.02025 http://arxiv.org/abs/1506.02025

[17] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman K. Ghosh, Andrew D. Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. 2015. ICDAR 2015 competition on Robust Reading. In ICDAR. IEEE Computer Society, 1156–1160.

[18] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernández Mota, Jon Almazán, and Lluís-Pere de las Heras. 2013. ICDAR 2013 Robust Reading Competition. In ICDAR. IEEE Computer Society, 1484–1493.

[19] Chen-Yu Lee and Simon Osindero. 2016. Recursive Recurrent Nets with Attention Modeling for OCR in the Wild. CoRR abs/1603.03101 (2016). arXiv:1603.03101 http://arxiv.org/abs/1603.03101

[20] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. 2018. Show, Attend and Read: A Simple and Strong Baseline for Irregular Text Recognition. CoRR abs/1811.00751 (2018). arXiv:1811.00751 http://arxiv.org/abs/1811.00751

[21] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. 2017. TextBoxes: A Fast Text Detector with a Single Deep Neural Network. In AAAI. AAAI Press, 4161–4167.

[22] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2018. Scene Text Recognition from Two-Dimensional Perspective. CoRR abs/1809.06508 (2018). arXiv:1809.06508 http://arxiv.org/abs/1809.06508

[23] Wei Liu, Chaofeng Chen, and Kwan-Yee K. Wong. 2018. Char-Net: A Character-Aware Neural Network for Distorted Scene Text Recognition. In AAAI. AAAI Press, 7154–7161.

[24] Wei Liu, Chaofeng Chen, Kwan-Yee K. Wong, Zhizhong Su, and Junyu Han. 2016. STAR-Net: A SpaTial Attention Residue Network for Scene Text Recognition. In BMVC. BMVA Press.

[25] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. 2018. FOTS: Fast Oriented Text Spotting with a Unified Network. CoRR abs/1801.01671 (2018). arXiv:1801.01671 http://arxiv.org/abs/1801.01671

[26] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. 2003. ICDAR 2003 Robust Reading Competitions. In Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2 (ICDAR '03). IEEE Computer Society, USA, 682.

[27] Pengyuan Lyu, Zhicheng Yang, Xinhang Leng, Xiaojun Wu, Ruiyu Li, and Xiaoyong Shen. 2019. 2D Attentional Irregular Scene Text Recognizer. CoRR abs/1906.05708 (2019). arXiv:1906.05708 http://arxiv.org/abs/1906.05708

[28] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In INTERSPEECH. ISCA, 1045–1048.

[29] Anand Mishra, Karteek Alahari, and C. V. Jawahar. 2012. Scene Text Recognition using Higher Order Language Priors. In BMVC. BMVA Press, 1–11.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[31] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. 2013. Recognizing Text with Perspective Distortion in Natural Scenes. In ICCV. IEEE Computer Society, 569–576.

[32] Anhar Risnumawan, Palaiahnakote Shivakumara, Chee Seng Chan, and Chew Lim Tan. 2014. A robust arbitrary text detection system for natural scene images. Expert Syst. Appl. 41 (2014), 8027–8048.

[33] B. Shi, X. Bai, and C. Yao. 2017. 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 (Nov 2017), 2298–2304. https://doi.org/10.1109/TPAMI.2016.2646371

[34] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016. Robust Scene Text Recognition with Automatic Rectification. In CVPR. IEEE Computer Society, 4168–4176.

[35] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai. 2019. ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 9 (Sep. 2019), 2035–2048. https://doi.org/10.1109/TPAMI.2018.2848939

[36] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. LSTM Neural Networks for Language Modeling. In INTERSPEECH. ISCA, 194–197.

[37] 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. 5998–6008.

[38] Kai Wang, Boris Babenko, and Serge J. Belongie. 2011. End-to-end scene text recognition. In ICCV. IEEE Computer Society, 1457–1464.

[39] Qitong Wang, Yi Zheng, and Margrit Betke. 2020. A method for detecting text of arbitrary shapes in natural scenes that improves text spotting. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020 Workshops. 10.

[40] Donghyeon Won, Zachary C. Steinert-Threlkeld, and Jungseock Joo. 2017. Protest Activity Detection and Perceived Violence Estimation from Social Media Images. In ACM Multimedia. ACM, 786–794.

[41] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C. Lee Giles. 2017. Learning to Read Irregular Text with Attention Mechanisms. In IJCAI. ijcai.org, 3280–3286.

[42] C. Yao, X. Bai, and W. Liu. 2014. A Unified Framework for Multioriented Text Detection and Recognition. IEEE Transactions on Image Processing 23, 11 (Nov 2014), 4737–4749. https://doi.org/10.1109/TIP.2014.2353813

[43] Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. CoRR abs/1212.5701 (2012).

[44] Fangneng Zhan and Shijian Lu. 2019. ESIR: End-To-End Scene Text Recognition via Iterative Image Rectification. In CVPR. Computer Vision Foundation / IEEE, 2059–2068.

[45] Yi Zheng, Qitong Wang, and Margrit Betke. 2020. Deep Neural Network for Semantic-based Text Recognition in Images. CoRR abs/1908.01403 (2020), 9. http://arxiv.org/abs/1908.01403