SCATTER: Selective Context Attentional Scene Text Recognizer

Ron Litman*, Oron Anschel*, Shahar Tsiper, Roee Litman, Shai Mazor and R. Manmatha
Amazon Web Services
{litmanr, oronans, tsiper, rlit, smazor, manmatha}@amazon.com

Abstract

Scene Text Recognition (STR), the task of recognizing text against complex image backgrounds, is an active area of research. Current state-of-the-art (SOTA) methods still struggle to recognize text written in arbitrary shapes. In this paper, we introduce a novel architecture for STR, named Selective Context ATentional Text Recognizer (SCATTER). SCATTER utilizes a stacked block architecture with intermediate supervision during training, that paves the way to successfully train a deep BiLSTM encoder, thus improving the encoding of contextual dependencies. Decoding is done using a two-step 1D attention mechanism. The first attention step re-weights visual features from a CNN backbone together with contextual features computed by a BiLSTM layer. The second attention step, similar to previous papers, treats the features as a sequence and attends to the intra-sequence relationships. Experiments show that the proposed approach surpasses SOTA performance on irregular text recognition benchmarks by 3.7% on average.

1. Introduction

We address the task of reading text in natural scenes, commonly referred to as Scene Text Recognition (STR). Although STR has been active since the late 90's, only recently accuracy reached a level that enables commercial applications, this is mostly due to advances in deep neural networks research for computer vision tasks. Applications for STR include, among others, recognizing street signs in autonomous driving, company logos, assistive technology for the blind and translation apps in mixed reality.

Text in natural scenes is characterised by a large variety of backgrounds, and arbitrary imaging conditions that can lead to low contrast, blur, distortion, low resolution, uneven illumination and other phenomena and artifacts. In addition, the sheer magnitude of possible font types and sizes add another layer of difficulty that STR algorithms must overcome. Generally, recognizing scene text can be divided into two main tasks - text detection and text recognition. Text detection is the task of identifying the regions in a natural image, that contain arbitrary shapes of text. Text recognition deals with the task of decoding a cropped image that contains one or more words into a digital string of its contents.

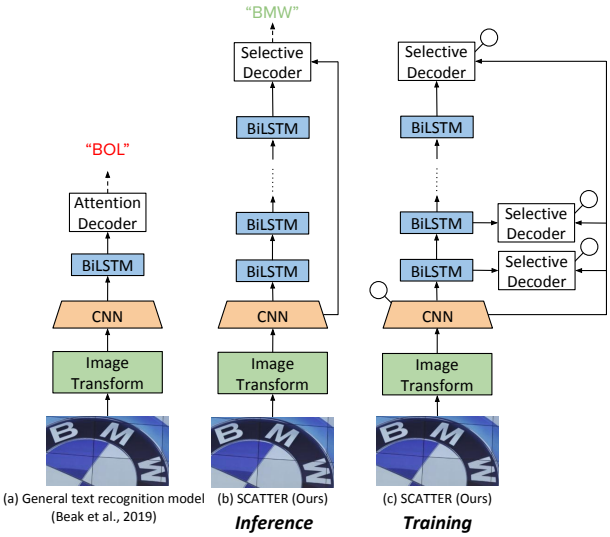


Figure 1: **The proposed SCATTER training and inference architecture**. We introduce intermediate supervision combined with selective-decoding to stabilize the training of a deep BiLSTM encoder (The circle represents where a loss function is applied). Decoding is done using a selective-decoder that operates on visual features from the CNN backbone and contextual features from the BiLSTM encoder, while employing a two-step attention.

In this paper, we propose a method for text recognition; we assume the input is a cropped image of text taken from a natural image, and the output is the recognized text string within the cropped image. As categorized by previous works [1, 16], text images can be divided into two categories: *Irregular text* for arbitrarily shaped text (e.g. curved text), as seen in Fig. 1, and *regular text* for text with nearly horizontally aligned characters (examples are provided in the supplementary material).

Traditional text recognition methods [37, 38, 42] detect and recognize text character by character, however, these methods have an inherent limitation – they do not utilize

*Authors contribute equally.

sequential modeling and contextual dependencies between characters.

Modern methods treat STR as a sequence prediction problem. This technique alleviates the need for character-level annotations (per-character bounding box) while achieving superior accuracy. The majority of these sequence-based methods rely on Connectionist Temporal Classification (CTC) [31, 7], or attention-based mechanisms [33, 16]. Recently, Baek et al. [1] proposed a modular four-step STR framework, where the individual components are interchangeable allowing for different algorithms. This modular framework, along with its best performing component configuration, is depicted in Fig. 1 (a). In this work, we build upon this framework and extend it.

While accurately recognizing regular scene text remains an open problem, recent irregular STR benchmarks (e.g., ICD15, SVTP) have shifted research focus to the problem of recognizing text in arbitrary shapes. For instance, Sheng et al. [30] adopted the Transformer [35] model for STR, leveraging the transformers ability to capture long-range contextual dependencies. The authors in [16] passed the visual features from the CNN backbone through a 2D attention module down to their decoder. Mask TextSpotter [17] unified the detection and the recognition tasks with a shared backbone architecture. For the recognition stage, two types of prediction branches are used, and the final prediction is selected based on the output of the more confident branch. The first branch uses semantic segmentation of characters, and requires additional character-level annotations. The second branch employs a 2D spatial attention-decoder.

Most of the aforementioned STR methods perform a sequential modeling step using a recursive neural network (RNN) or other sequential modeling layers (e.g., multi-head attention [30]), usually in the encoder and/or the decoder. This step is performed to convert the **visual feature** map into a **contextual feature** map, which better captures long-term dependencies. In this work, we propose using a stacked block architecture for repeated feature processing, a concept similar to that used in other computer-vision tasks such as in [40] and later in [27, 26]. The authors above showed that repeated processing used in conjunction with intermediate supervision could be used to increasingly refine predictions.

In this paper, we propose the **S**elective **C**ontext **AT**tentional **TE**xt **R**ecognizer (SCATTER) architecture. Our method, as depicted in Fig. 1, utilize a stacked block architecture for repetitive processing with intermediate supervision in training, and a novel selective-decoder. The selective-decoder receives features from two different layers of the network, namely, visual features from a CNN backbone and contextual features computed by a BiLSTM layer, while using a two-step 1D attention mechanism. Figure 2 shows the accuracy levels computed at the intermediate auxiliary decoders, for different stacking arrangements, thus demonstrating the increase in performance as additional blocks are added in succession. Interestingly, training with additional blocks in sequence leads to an improvement in the accuracy of the intermediate decoders as well (compared to training with a shallower stacking arrangement).

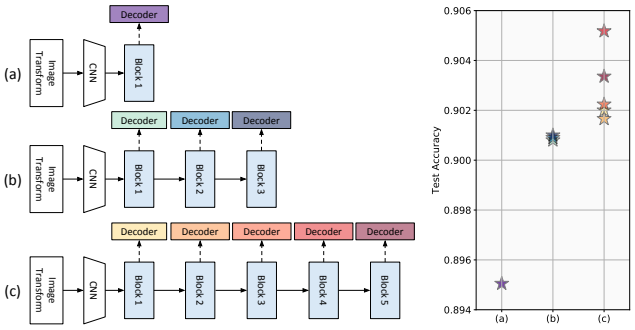


Figure 2: Average test accuracy (IIT5K, SVT, IC03, IC13,IC15, SVTP, CUTE) at intermediate decoding steps, compared across different network depths used in training. Given a computation budget, improved results can be obtained by first training a deeper network, and then running only the first decoder(s) during inference.

This paper presents two main contributions:

1. We propose a repetitive processing architecture for text recognition, trained with intermediate selective decoders as supervision. Using this architecture we train a deep BiLSTM encoder, leading to SOTA results on irregular text.
2. A selective attention decoder, that simultaneously decodes both visual and contextual features by employing a two-step attention mechanism. The first attention step figures out which visual and contextual features to attend to. The second step treats the features as a sequence and attends the intra-sequence relations.

2. Related Work

STR has attracted considerable attention over the past few years [34, 36, 3, 20]. Comprehensive surveys for scene text detection and recognition may be found in [43, 1, 21]. As mentioned above, STR may be divided into two categories: regular and irregular texts (further examples are provided in the supplementary material). Earlier papers [37, 38, 42], focused on regular text and used a bottom-up approach, which involved segmenting individual characters with a sliding window, and then recognizing the characters using hand-crafted features. A notable issue with the bottom-up approaches above, is that they struggle to use contextual information; instead they rely on accurate character classifiers. Shi et al. 2015 [31] and He et al. [11] considered words as sequences of varying lengths, and employed RNNs to model the sequences without explicit char-

acter separation. Shi et al. 2016 [32] presented a successful end-to-end trainable architecture using the sequence approach, without relying on character level annotations. Their solution employed a BiLSTM layer to extract the sequential feature vectors from the input feature maps, these vectors are then fed into an attention-Gated Recurrent Unit (GRU) module for decoding.

The methods mentioned above introduced significant improvements in STR accuracy on public benchmarks. Therefore, recent work has shifted focus to the more challenging problem of recognizing irregularly shaped text, hence promoting new lines of research. Topics such as input rectification, character-level segmentation, 2D attentional feature maps and self attention have emerged, pushing the envelope on irregular STR. Shi et al. 2018 [33] rectified oriented or curved text based on a Spatial Transformer Network (STN). Liu et al. 2018 [19] introduced a Character-Aware Neural Network (Char-Net) to detect and rectify individual characters. A combination of a CTC-Attention mechanism within an encoder-decoder framework, that was used for speech recognition tasks, was used for STR in [45], showing the benefits of joint CTC-Attention learning. The authors in [7] proposed two supervision branches to tackle explicit and implicit semantic information. In [4] a gate was inserted to the recurrent decoder, for controlling the transmission-weight of the previous embedded vector, demonstrating that context is not always needed for decoding. The authors of Mask TextSpotter [17] unified text detection and text recognition in an end-to-end fashion. For recognition they used two separate branches, a branch that uses visual (local) features and a branch which utilizes contextual information in the form of 2D attention.

More recent approaches have proposed leveraging various attention mechanisms for improved results. Li et al. [16] combined both visual and contextual features while utilizing a 2D attention within the encoder-decoder. Other researchers borrowed ideas from the Natural Language Processing (NLP) domain and adopted a transformer-based architecture [35]. One of them is Sheng et al. [30], that used a self-attention mechanism for both the encoder and the decoder.

Our method differs from above approaches, by being the first to utilize a stacked block architecture for text recognition. Namely, we show that repetitive processing for text recognition, trained with intermediate selective decoders as supervision (similar to [40, 27, 26]), increasingly refines text predictions.

3. Methodology

As presented in Fig. 3, our proposed architecture consists of four main components:

1. **Transformation:** the input text image is normalized using a Spatial Transformer Network (STN) [13].
2. **Feature Extraction:** maps the input image to a feature map representation while using a text attention module [7].
3. **Visual Feature Refinement:** provides direct supervision for each column in the visual features. This part refines the representation in each of the feature columns, by classifying them into individual symbols.
4. **Selective-Contextual Refinement Block:** Each block consists of a two-layer BiLSTM encoder that outputs contextual features. The contextual features are concatenated to the visual features computed by the CNN backbone. This concatenated feature map is then fed into the selective-decoder, which employs a two-step 1D attention mechanism, as illustrated in Fig. 4.

In this section we describe the training architecture of SCATTER , while addressing the differences between training and inference.

3.1. Transformation

The transformation step operates on the cropped text image X, and transforms it into a normalized image X'. We use a Thin Plate Spline (TPS) transformation, a variant of the STN, as used in [1]. TPS employs a smooth spline interpolation between a set of fiducial points. Specifically, it detects a pre-defined number of fiducial points at the top and bottom of the text region, and normalizes the predicted region to a constant predefined size.

3.2. Feature Extraction

In this step a convolutional neural network (CNN) extracts features from the input image. We use a 29-layer ResNet as the CNN's backbone, as used in [5]. The output of the feature encoder is 512 channels by N columns. Specifically, the feature encoder gets an input image X' and outputs a feature map $F = [f_1, f_2, ..., f_N]$. Following the feature map extraction, we use a text attention module, similar to [7]. The attentional feature map can be regarded as a visual feature sequence of length N, denoted as $V = [v_1, v_2, ..., v_N]$, where each column represents a frame in the sequence.

3.3. Visual Feature Refinement

Here, the visual feature sequence V is used for intermediate decoding. This intermediate supervision is aimed at refining the character embedding (representations) in each of the columns of V, and is done using CTC based decoding. We feed V through a fully connected layer that outputs a sequence H of length N. The output sequence is fed into a CTC [8] decoder to generate the final output. The CTC decoder transforms the output sequence tensor into a conditional probability distribution over the label sequences, and then selects the most probable label. The transcription pro-

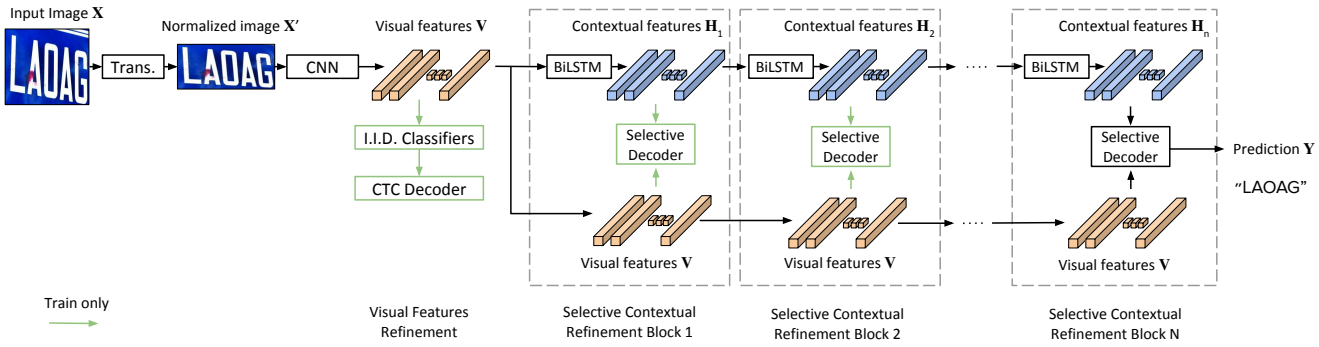


Figure 3: The proposed SCATTER architecture introduces, context refinement, intermediate supervision (additional decoders), and a novel selective-decoder.

cedure is given by

$$l = B(\arg\max_{\pi} p(\pi|H)), \tag{1}$$

where the probability of π is defined as

$$p(\pi|H) = \prod_{t=1}^{N} y^t_{\pi_t}. \tag{2}$$

Here $y^t_{\pi_t}$ is the probability of generating the character π_t at time stamp t, and B is a mapping function that removes all repeated characters and blanks. The CTC algorithm assumes that the columns are conditionally independent, and at each time stamp the output is a single character probability score. The loss for this branch, denoted by L_{CTC}, is the negative log-likelihood of the ground-truth conditional probability, as in [31].

3.4. Selective-Contextual Refinement Block

The features extracted by the CNN are limited to its receptive field, and may suffer due to the lack of contextual information. To mitigate this drawback, we employ a two-layer BiLSTM [9] network over the feature map V, outputting $H = [h_1, h_2, ..., h_n]$. We concatenate the BiLSTM output with the visual feature map, yielding $D = (V, H)$, a new feature space.

The feature space D is used both for selective decoding, and as an input to the next Selective-Contextual Refinement block. Specifically, the concatenated output of the jth block can be written as $D_j = (V, H_j)$. The next $j+1$ block uses H_j as input to the two-layer BiLSTM, yielding H_{j+1}, and the $j+1$ feature space is updated such that $D_{j+1} = (V, H_{j+1})$. The visual feature map V does not undergo any further updates in the Selective-Contextual Refinement blocks, however we note that the CNN backbone is updated with back-propagated gradients from all of the selective-decoders. These blocks can be stacked together as many times as needed, according to the task or accuracy

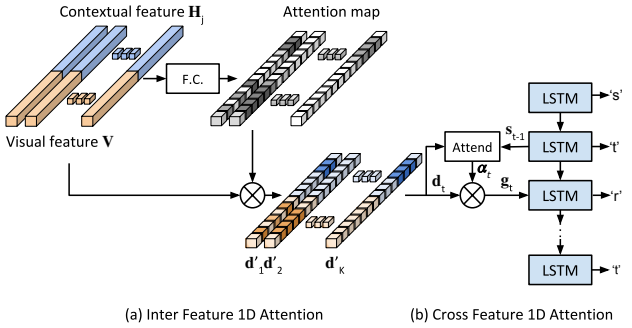


Figure 4: Architecture of the Two-Step Attention Selective-Decoder.

levels required, and the final prediction is provided by the decoder from the last block.

3.4.1 Selective-Decoder

We employ a two-step attention mechanism, as illustrated in Fig. 4. First, we use a 1D self attention operating on the features D. A fully connected layer is used to compute an attention map from these features. Next, an element-wise product is computed between the attention map and D, yielding the attentional features D'. The decoding of D' is done with a separate attention-decoder, such that for each t-time-step the decoder outputs y_t, similar to [5, 2].

Decoding starts by computing the vector of attentional weights, $\alpha_t \in R^N$:

$$e_{t,i} = w^T \tanh(W s_{t-1} + V d'_i + b) \tag{3}$$

$$\alpha_{t,i} = \exp(e_{t,i}) / \sum_{i^*=1}^{n} e_{t,i^*}, \tag{4}$$

where b, w, W, V are trainable parameters, s_{t-1} is the hidden state of the recurrent cell within the decoder at time t, and $d\prime$ is a column of $D\prime$. The decoder linearly combines the

columns of D' into a vector G, which is called a glimpse:

$$g_t = \sum_{i=1}^{n} \alpha_{t,i} d'_i \,. \tag{5}$$

Next, a recurrent cell of the decoder is fed with

$$(x_t, s_t) = \mathrm{RNN}\Big(s_{t-1}, \big(g_t, f(y_{t-1})\big)\Big) \,, \tag{6}$$

where $(g_t, f(y_{t-1}))$ denotes the concatenation between g_t and the one-hot embedding of y_{t-1}.

The probability for a given character $p(y_t)$ can now be recovered by:

$$p(y_t) = \mathrm{softmax}(W_o x_t + b_o) \,. \tag{7}$$

The loss for the jth block is the negative log-likelihood, denoted as $L_{\mathrm{Attn},j}$, as in [5, 2].

3.5. Training Losses

The objective function is given by

$$L = \lambda_{\mathrm{CTC}} \cdot L_{\mathrm{CTC}} + \sum_{j=1} \lambda_j L_{\mathrm{Attn},j} \,, \tag{8}$$

where L_{CTC} is the loss function of the CTC decoder and $\sum_{j=1} \lambda_j L_{\mathrm{Attn},j}$ is the sum of the losses from all of the Selective-Contextual Refinement blocks, as defined above. The λ notation depicts a hyper-parameter used to balance the trade-off between the different supervisions, and specifically, λ_{CTC}, λ_j are empirically set to 0.1, 1.0 respectively for all j.

3.6. Inference

Once training is done, for test time we remove all of the intermediate decoders, as they are used only for additional supervision and refinement of intermediate feature. The visual features V are processed by the BiLSTM layers in all blocks, and are also fed directly, via a skip connection, to the final selective-decoder. The final selective decoder is used to predict the output sequence of characters. A visualization of these changes can be seen in Fig. 3, where all the green colored operations are disabled during inference, and in Fig. 1 (b).

4. Experiments

In this section we empirically demonstrate the effectiveness of our proposed framework. We begin with a brief discussion regarding the datasets used for training and testing, and then describe our implementation and evaluation setup. Next, we compare our model against state-of-the-art methods on public benchmark datasets, including both regular and irregular text. Finally, we address the computational cost of our method.

4.1. Datasets

In this work, all SCATTER models are trained on three synthetic datasets. The model is evaluated on four regular scene-text datasets: ICDAR2003, ICDAR2013, IIIT5K, SVT, and three irregular text datasets: ICDAR2015, SVTP and CUTE.

The training dataset is a union of three datasets:

MJSynth (MJ) [12] is a synthetic text in image dataset which contains 9 million word box images, generated from a lexicon of 90K English words.

SynthText (ST) [10] is a synthetic text in image dataset, designed for scene-text detection, and recognition. We use a variant of the SynthText dataset composed of 5.5M samples, as used in [1]. This variant does not include any non-alphanumeric characters.

SynthAdd (SA) [16] is a synthetic text in image dataset, that contains 1.2 million word box images. This dataset was generated using the same synthetic engine as in ST, aiming to mitigate the lack of non-alphanumeric characters (e.g., punctuation marks) in the other datasets.

All experiments are evaluated on the seven real-word STR benchmark datasets described below. As in many STR manuscripts (e.g, [33, 1, 16]) the benchmark datasets are commonly divided into regular and irregular text, according to the text layout.

Regular text datasets include the following: **IIIT5K** [25] consists of 2000 training and 3000 testing images that are cropped from Google image searches. **SVT** [37] is a dataset collected from Google Street View images and contains 257 training and 647 testing word-box cropped images. **ICDAR2003** [23] contains 867 word-box cropped images for testing. **ICDAR2013** [15] contains 848 training and 1,015 testing word-box cropped images.

Irregular text datasets include the following: **ICDAR2015** [14] contains 4,468 training and 2,077 testing word-box cropped images, all captured by Google Glass, without careful positioning or focusing. **SVTP** [28] is a dataset collected from Google Street View images and consists of 645 cropped word-box images for testing. **CUTE 80** [29] contains 288 cropped word-box images for testing, many of which are curved text images.

4.2. Implementation Details

As baseline, we use the code of Baek et al.[1] [1], and our architectural changes are implemented on top of it. All experiments are trained and tested using the PyTorch[2] framework on a Tesla V100 GPU with 16GB memory. As for the training details, we do not perform any type of pre-training. We train using the AdaDelta optimizer, and the following training parameters are used: a decay rate of 0.95, gradient clipping with a magnitude of 5, a batch size of 128 (with a

[1] https://github.com/clovaai/deep-text-recognition-benchmark
[2] https://pytorch.org/

Table 1: Scene text recognition accuracies (%) over seven public benchmark test datasets (number of words in each dataset are shown below the title). No lexicon is used. In each column, the best performing result is shown in **bold** font, and the second best result is shown with an underline. Average columns are weighted (by size) average results on the regular and irregular datasets. "*" indicates using both word-level and character-level annotations for training.

Method	Regular test dataset					Irregular test dataset			
	IIIT5K	SVT	IC03	IC13	**Average**	IC15	SVTP	CUTE	**Average**
	3000	647	867	1015	5529	2077	645	288	3010
CRNN (2015) [31]	78.2	80.8	89.4	86.7	81.8	-	-	-	-
FAN (2017) [5]*	87.4	85.9	94.2	93.3	89.4	70.6	-	-	-
Char-Net (2018) [19]*	83.6	84.4	91.5	90.8	86.2	60.0	73.5	-	-
AON (2018) [6]	87.0	82.8	91.5	-	-	68.2	73.0	76.8	70.0
EP (2018) [2]*	88.3	87.5	94.6	94.4	90.3	73.9	-	-	-
NRTR (2018) [33]	86.5	88.3	95.4	94.7	89.6	-	-	-	-
Liao et al. (2019) [18]	91.9	86.4	-	86.4	-	-	-	79.9	-
Baek et al. (2019) [1]	87.9	87.5	94.9	92.3	89.8	71.8	79.2	74.0	73.6
ASTER (2019) [33]	93.4	89.5	94.5	91.8	92.8	76.1	78.5	79.5	76.9
SAR (2019) [16]	91.5	84.5	-	91.0	-	69.2	76.4	83.3	72.1
ESIR (2019) [44]	93.3	90.2	-	91.3	-	76.9	79.6	83.3	78.1
MORAN (2019) [24]	91.2	88.3	95.0	92.4	91.7	68.8	76.1	77.4	71.2
Yang et al. (2019) [41]	94.4	88.9	95.0	93.9	93.7	78.7	80.8	87.5	79.9
Mask TextSpotter (2019) [17]*	**95.3**	91.8	95.0	**95.3**	**94.8**	78.2	83.6	**88.5**	80.0
SCATTER (1 Block)	92.9	89.2	96.5	93.8	93.2	81.8	84.5	85.1	82.7
SCATTER (2 Block)	93.5	89.2	95.9	94.7	93.6	81.5	86.2	86.8	83.0
SCATTER (3 Block)	93.9	89.3	96.1	94.6	93.7	**82.8**	85.7	83.7	83.4
SCATTER (4 Block)	93.4	90.3	**96.6**	94.3	93.7	82.0	**87.0**	86.5	83.5
SCATTER (5 Block)	93.7	**92.7**	96.3	93.9	94.0	82.2	86.9	87.5	**83.7**

sampling ratio of 40%, 40%, 20% between MJ, ST and SA respectively). We use data augmentation during training, and augment 40% of the input images, by randomly resizing them and adding extra distortion. Each model is trained for 6 epochs on the unified training set. For our internal validation dataset, we use the union of the IC13, IC15, IIIT, and SVT training splits, to select our best model, as done in [1]. All images are resized to 32×100 during both training and testing, following common practice. In this paper, we use 36 symbol classes: 10 digits and 26 case-insensitive letters. As for special symbols for CTC decoding, an additional "[UNK]" and a "[blank]" token are added to the label set. For the selective-decoders three special punctuation characters are added: "[GO]", "[S]" and "[UNK]" which indicate the start of the sequence, the end of the sequence and unknown characters (that are not alpha-numeric), respectively.

At inference, we employ a similar mechanism to [16, 39, 22], where images with a height larger than their width, are rotated by 90 degrees clockwise and counter-clockwise respectively. The rotated versions are recognized alongside the original image. A prediction confidence score is calculated as the average decoder probabilities until the '[S]' token. We then choose the prediction with the highest confidence score as the final prediction. Unlike [33, 16, 30], we do not use beam-search for decoding, although the authors in [16] have reported it improves accuracy by approximately 0.5%, due to the added latency it incurs.

4.3. Comparison to State-of-the-art

In this section, we measure the accuracy of our proposed framework on several regular and irregular scene text benchmarks while comparing the results to the latest SOTA recognition methods. As seen in Table 1, our SCATTER architecture with 5 blocks outperforms the current SOTA, the Mask TextSpotter [17] algorithm, on irregular scene text benchmarks (i.e., IC15, SVTP, CUTE) by an absolute margin of 3.7% on average. Our approach provides an accuracy increase of **+4.0** pp (78.2% vs. 82.2%) on IC15, **+3.3** pp (83.6% vs. 86.9%) on SVTP, and is the second best to Mask TextSpotter [17] on the CUTE (88.5% vs. 87.5%) dataset. Additionally, the proposed method outperforms the other methods both on SVT and IC03 regular scene text datasets, and achieves comparable SOTA performance on the other regular scene text datasets (i.e.,IIIT5K and IC13).

To summarize, our model achieves the highest recognition score on 4 out of 7 benchmarks, and the second best score on 2 more benchmarks. Unlike other methods, which perform well on either regular or irregular scene text benchmarks, our approach is a top performer on all benchmarks.

We would like to briefly discuss key differences between Mask TextSpotter [17] and this work. The algorithm

Table 2: Ablation studies by changing the model hyper-parameters. We refer to our re-trained model using the code of Baek et al. 2019 as Baseline. Using intermediate supervision helps to boost results and enables stacking more blocks. Increasing the number of blocks has positive impacts on the recognition performance. * Regular Text and Irregular Text columns are weighted (by size) average results on the regular and irregular datasets respectively.

Sec	Method	CTC Decoder	Attention Decoders	Selective Decoders	LSTM Layers	N Blocks	Regular test dataset				Irregular test dataset			Regular Text*	Irregular Text*
							IIIT5K	SVT	IC03	IC13	IC15	SVTP	CUTE		
(a)	[1]	0	1	-	2	-	87.9	87.5	94.9	92.3	71.8	79.2	74.0	89.8	73.7
	Baseline	0	1	-	2	-	93.0	88.6	94.3	92.9	78.0	81.5	80.1	92.7	79.1
	Baseline	1	1	-	2	-	92.7	90.0	95.0	93.4	78.3	83.4	80.0	92.9	79.5
	SCATTER	0	-	1	2	1	93.1	89.3	95.7	93.7	80.6	84.0	86.5	93.1	81.8
	SCATTER	1	-	1	2	1	92.9	89.2	96.5	93.8	81.8	84.5	85.1	93.2	82.7
(b)	SCATTER	0	-	1	4	2	92.9	89.6	95.3	93.2	79.3	84.2	83.7	92.9	80.1
	SCATTER	0	-	2	4	2	93.1	90.7	94.9	93.6	81.6	84.7	85.8	93.2	82.6
	SCATTER	1	-	1	4	2	93.0	90.6	95.5	92.8	82.0	84.0	86.8	93.1	82.9
	SCATTER	1	-	2	4	2	93.5	89.2	95.9	94.7	81.5	86.2	86.8	93.6	83.0
(c)	SCATTER	1	-	3	6	3	93.9	89.3	96.1	94.6	82.8	85.7	83.7	93.7	83.4
	SCATTER	1	-	4	8	4	93.4	90.3	96.6	94.3	82.0	87.0	86.5	93.7	83.5
	SCATTER	1	-	5	10	5	93.7	92.7	96.3	93.9	82.2	86.9	87.5	94.0	83.7
	SCATTER	1	-	6	12	6	93.2	90.9	96.2	94.1	82.0	86.2	84.8	93.6	83.2

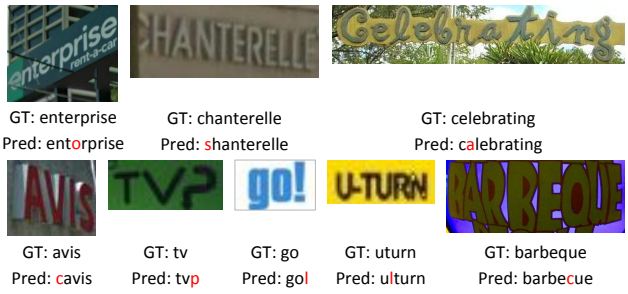


Figure 5: Failure cases of our model. "GT" stands for the ground-truth annotation, and "Pred" is the predicted result.

in [17] relies on annotations that contain character level annotations, information that our algorithm does not require. These annotations contribute an average increase of 0.9 pp, and 0.6 pp on regular and irregular text datasets as reported in the original paper. Hence, without the character level annotations, our model achieves slightly better results on regular text (93.9% vs. 94%) and significantly better results on irregular text (79.4% vs. 83.7%). Our approach, on the other hand, does not require these annotations, which are expensive and hard to annotate, especially for real-world data.

In Fig. 5 we display failure cases of our method. The failure cases are mostly composed of blurry images, partial character occlusions, difficult lighting conditions, and miss-recognition of punctuation marks.

4.4. Computational Costs

During inference, only the selective decoder of the final block is kept active, as shown in Fig. 1. The total computational cost of our proposed architecture with a single block is 20.1 ms. The additional computational cost of each *intermediate* contextual refinement block during inference translates to a 3.2 ms per block. For a 5 block architecture (our best setup) this translates to a total increase of 12.8 ms and a total forward pass of 32.9 ms.

Furthermore, given a computational budget on inference time, performance can be improved by training the system with a large number of blocks and pruning them for inference. For example, an architecture trained with five blocks, and then pruned to a single block, is capable of outperforming an architecture solely trained with a single block. Figure 2(2c) demonstrates a network trained with five blocks and the average test accuracy of intermediate decoders. This showcases that pruning leads to an increase of **+0.4** pp and **+1.3** pp on regular an irregular datasets respectively (under the same computational budget). This novel feature of SCATTER allows for faster inference if needed and in some cases pruning can even boost results.

5. Ablation Experiments

In this section, we perform a series of experiments to better understand the performance improvements and analyze the impact of our key contributions. Throughout this section, we use a weighted-average (by the number of samples) of the results on the regular and irregular test datasets. For completeness, the first and second rows in Table 2 show the reported results in [1], and the improved results of our re-trained model of [1] with our custom training settings.

5.1. Intermediate Supervision & Selective Decoding

Section (a) of Table 2 shows an improvement in accuracy by adding the intermediate CTC supervision and the pro-

posed selective-decoder. Between row two and three of section (a) we add a CTC decoder for intermediate supervision which improves the baseline results by **+0.2** pp and **+0.4** pp on regular and irregular text respectively. The fourth row demonstrates the improvement compared to the baseline results by replacing the standard attentional decoder with the proposed selective-decoder, (**+0.4** pp and **+2.7** pp on regular and irregular text respectively).

Section (b) of Table 2 shows a monotonic increase in the accuracy using the SCATTER architecture with 4 BiLSTM layers by changing the number of intermediate supervisions (ranging from 1 to 3). The relative increase in accuracy of section (b) is **+0.7** pp and **+2.9** pp on regular and irregular text respectively.

5.2. Stable Training of a Deep BiLSTM Encoder

As mentioned in the introduction, previous papers use only a 2-layer BiLSTM encoder. The authors in [45] report a decrease in accuracy when increasing the number of layers in the BiLSTM encoder. We reproduce the experiment reported in [45] of training a baseline architecture with an increasing number of BiLSTM layers in the encoder (results are in the supplementary material). We observe a similar phenomena as in [45], i.e a reduction in accuracy when using more than two BiLSTM layers. Contrary to this discovery, Table 2 shows that the overall trend of increasing the number of BiLSTM layers in SCATTER , while increasing the number of intermediate supervisions improves accuracy. Recognition accuracy improves monotonically up to 10 BiLSTM layers, both for regular and irregular text datasets.

As evident from Table 2 (c), when training with more than 10 BiLSTM layers in the encoder, accuracy results slightly (**-0.4** pp and **-0.5** pp on regular and irregular text respectively) decrease on both regular and irregular text (similar phenomena was observed on the validation set). It is expected that increasing the network capacity leads to more challenging training procedures. Other training approaches might need to be considered to successfully train to convergence a very deep encoder. Such approaches may include incremental training, where we first train with a shallower network using a small number of blocks and incrementally stack more blocks during training.

Examples of intermediate predictions are seen in Table 3, showcasing SCATTER ability to increasingly refine text prediction.

5.3. Oracle Decoder Voting

In Table 4 the test accuracy is shown for the intermediate decoders on a SCATTER architecture trained with 5 blocks. The last row summarizes the potential results of an oracle, that for every test image chooses the correct prediction, if one exists in any of the decoders. If an optimal strategy to select between the different decoders predictions exists, the results on *all* datasets achieve a new state-of-the-art. The possible improvement in accuracy achievable by such an oracle ranges between **+0.8** pp, and up to **+5** pp across the datasets. A possible prediction selection strategy might be based on ensemble techniques, or a meta model that predicts which decoder to use for each specific image.

Table 3: The table shows example for when repeated processing used in conjunction with intermediate supervision increasingly refines text predictions.

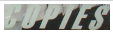
Test Image	Intermediate Decoder				Final Decoder	Ground Truth
	1	2	3	4		
	goptes	coptes	copies	copies	copies	copies
	bm_	bn_	bmy	bmy	bmw	bmw
	o0___	100__	100__	1000_	10000	10000

Table 4: Performance of each decoder and the oracle. The oracle performance is calculated given the ground truth.

Decoder	IIIT5K	SVT	IC03	IC13	IC15	SVTP	CUTE
CTC	89.6	84.6	93.5	90.9	69.9	77.4	77.8
$Attn_1$	93.5	90.3	95.9	93.9	82.9	86.4	**87.5**
$Attn_2$	93.5	90.4	96.3	94.1	82.6	86.0	86.8
$Attn_3$	**93.7**	91.0	95.7	94.3	82.7	85.1	87.2
$Attn_4$	**93.7**	91.2	**96.5**	**94.4**	**83.3**	85.8	86.8
Final	**93.7**	**92.7**	96.3	93.9	82.2	**86.9**	**87.5**
Oracle	96.3	95.1	97.1	95.5	87.1	89.9	90.3

6. Conclusions and Future Work

In this work we propose a stacked block architecture named SCATTER , which achieves SOTA recognition accuracy and enables stable, more robust training for STR networks that use deep BiLSTM encoders. This is achieved by adding intermediate supervisions along the network layers, and by relying on a novel selective decoder.

We also demonstrate that a repetitive processing architecture for text recognition, trained with intermediate selective decoders as supervision, increasingly refines text predictions. In addition, other approaches of attention could also benefit from stacked attention decoders, as our proposed novelty is not limited to our formulation of attention.

We consider two promising directions for future work. First, in Fig. 2 we show that training deeper networks and then pruning the last decoders (and layers) is preferable over training a shallower network. This could lead to an increase in performance given a constraint on computational budget. Finally, we see potential in developing an optimal selection strategy between the predictions of the different decoders for each image.

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. *arXiv preprint arXiv:1904.01906*, 2019. 1, 2, 3, 5, 6, 7

[2] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1508–1516, 2018. 4, 5, 6

[3] 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*, pages 71–79. ACM, 2018. 2

[4] Xiaoxue Chen, Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, and Canjie Luo. Adaptive embedding gate for attention-based scene text recognition. *arXiv preprint arXiv:1908.09475*, 2019. 3

[5] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5076–5084, 2017. 3, 4, 5, 6

[6] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5571–5579, 2018. 6

[7] Yuting Gao, Zheng Huang, and Yuchen Dai. Double supervised network with attention mechanism for scene text recognition. *arXiv preprint arXiv:1808.00677*, 2018. 2, 3

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006. 3

[9] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008. 4

[10] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016. 5

[11] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *Thirtieth AAAI conference on artificial intelligence*, 2016. 2

[12] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 5

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[14] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 5

[15] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. 5

[16] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8610–8617, 2019. 1, 2, 3, 5, 6

[17] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2, 3, 6, 7

[18] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8714–8721, 2019. 6

[19] Wei Liu, Chaofeng Chen, and Kwan-Yee K Wong. Char-net: A character-aware neural network for distorted scene text recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3, 6

[20] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: a spatial attention residue network for scene text recognition. In *BMVC*, volume 2, page 7, 2016. 2

[21] Shangbang Long, Xin He, and Cong Ya. Scene text

detection and recognition: The deep learning era. *arXiv preprint arXiv:1811.04256*, 2018. 2

[22] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, and Rong Xiao. Master: Multi-aspect non-local network for scene text recognition. *arXiv preprint arXiv:1910.02562*, 2019. 6

[23] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 682–687. Citeseer, 2003. 5

[24] Canjie Luo, Lianwen Jin, and Zenghui Sun. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019. 6

[25] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. 2012. 5

[26] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017. 2, 3

[27] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2, 3

[28] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 569–576, 2013. 5

[29] 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. 5

[30] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. *arXiv preprint arXiv:1806.00926*, 2018. 2, 3, 6

[31] 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*, 39(11):2298–2304, 2016. 2, 4, 6

[32] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016. 3

[33] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2, 3, 5, 6

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 3

[36] Jianfeng Wang and Xiaolin Hu. Gated recurrent convolution neural network for ocr. In *Advances in Neural Information Processing Systems*, pages 335–344, 2017. 2

[37] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011. 1, 2, 5

[38] Kai Wang and Serge Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer, 2010. 1, 2

[39] Peng Wang, Lu Yang, Hui Li, Yuyan Deng, Chunhua Shen, and Yanning Zhang. A simple and robust convolutional-attention network for irregular text recognition. *arXiv preprint arXiv:1904.01375*, 2019. 6

[40] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 2, 3

[41] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 9147–9156, 2019. 6

[42] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4042–4049, 2014. 1, 2

[43] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2014. 2

[44] Fangneng Zhan and Shijian Lu. Esir: End-to-end scene text recognition via iterative image rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2059–2068, 2019. 6

[45] Ling-Qun Zuo, Hong-Mei Sun, Qi-Chao Mao, Rong Qi, and Rui-Sheng Jia. Natural scene text recognition

based on encoder-decoder framework. *IEEE Access*, 7:62616–62623, 2019. 3, 8