SAFE: Scale Aware Feature Encoder for Scene Text Recognition

Wei Liu, Chaofeng Chen, and Kwan-Yee K. Wong

Department of Computer Science, The University of Hong Kong
{wliu, cfchen, kykwong}@cs.hku.hk

Abstract. In this paper, we address the problem of having characters with different scales in scene text recognition. We propose a novel scale aware feature encoder (SAFE) that is designed specifically for encoding characters with different scales. SAFE is composed of a multi-scale convolutional encoder and a scale attention network. The multi-scale convolutional encoder targets at extracting character features under multiple scales, and the scale attention network is responsible for selecting features from the most relevant scale(s). SAFE has two main advantages over the traditional single-CNN encoder used in current state-of-the-art text recognizers. First, it explicitly tackles the scale problem by extracting scale-invariant features from the characters. This allows the recognizer to put more effort in handling other challenges in scene text recognition, like those caused by view distortion and poor image quality. Second, it can transfer the learning of feature encoding across different character scales. This is particularly important when the training set has a very unbalanced distribution of character scales, as training with such a dataset will make the encoder biased towards extracting features from the predominant scale. To evaluate the effectiveness of SAFE, we design a simple text recognizer named scale-spatial attention network (S-SAN) that employs SAFE as its feature encoder, and carry out experiments on six public benchmarks. Experimental results demonstrate that S-SAN can achieve state-of-the-art (or, in some cases, extremely competitive) performance without any post-processing.

1 Introduction

Scene text recognition refers to recognizing a sequence of characters that appear in a natural image. Inspired by the success [1] in neural machine translation, many of the recently proposed scene text recognizers [2,3,4,5,6] adopt an encoder-decoder framework with an attention mechanism. Despite the remarkable results reported by them, very few of them have addressed the problem of having characters with different scales in the image. This problem often prevents existing text recognizers from achieving better performance.

In a natural image, the scale of a character can vary greatly depending on which character it is (for instance, 'M' and 'W' are in general wider than 'i' and 'l'), font style, font size, text arrangement, viewpoint, *etc.* Fig. 1 shows some examples of text images containing characters with different scales. Existing text

recognizers [2,3,4,5,6,7,8] employ only one single convolutional neural network for feature encoding, and often perform poorly for such text images. Note that a single-CNN encoder with a fixed receptive field[1] (refer to the green rectangles in Fig. 1) can only effectively encode characters which fall within a particular scale range. For characters outside this range, the encoder captures either only partial context information of a character or distracting information from the cluttered background and neighboring characters. In either case, the encoder cannot extract discriminative features from the corresponding characters effectively, and the recognizer will have great difficulties in recognizing such characters.

问题：相同感受野下，
字符的尺度很不同
a)不同类型，大小和排版
b)同一张图片中不同的字符，
扭曲

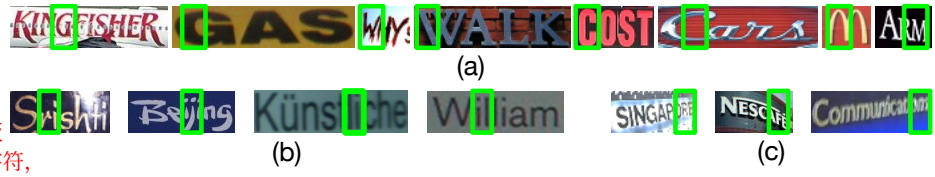

Fig. 1: Examples of text images having characters with different scales. The green rectangles represent the effective receptive field of a single-CNN encoder. (a) The character scale may vary greatly because of font style, font size, text arrangements, *etc.* (b) For characters in the same image with the same font style and font size, their scales may still vary depending on which characters they are. For instance, lower case letters 'l' and 'i' are usually much 'narrower' than other characters. (c) The distortion of text can also result in different character scales within the same image.

In this paper, we address the problem of having characters with different scales by proposing a simple but efficient scale aware feature encoder (SAFE). SAFE is designed specifically for encoding characters with different scales. It is composed of (i) a multi-scale convolutional encoder for extracting character features under multiple scales, and (ii) a scale attention network for automatically selecting features from the most relevant scale(s). SAFE has two main advantages over the traditional single-CNN encoder used in current state-of-the-art text recognizers. First, it explicitly tackles the scale problem by extracting scale-invariant features from the characters. This allows the recognizer to put more effort in handling other challenges in scene text recognition, like those caused by character distortion and poor image quality (see Fig. 2a). Second, it can transfer the learning of feature encoding across different character scales. This is particularly important when the training set has a very unbalanced distribution of character scales. For instance, the most widely adopted SynthR dataset [10] has only a small number of images containing a single character (see Fig. 2b). In order to keep the training and testing procedures simple and efficient, most of the previous methods [2,3,4,6,8] resized an input image to a fixed resolution before feeding it to the text recognizer. This resizing operation

[1] Although the receptive field of a CNN is large, its effective region [9] responsible for calculating each feature representation only occupies a small fraction.

will therefore result in very few training images containing large characters. Obviously, a text recognizer with a single-CNN encoder cannot learn to extract discriminative features from large characters due to limited training examples. This leads to a poor performance on recognizing text images with a single character (see Fig. 2c). Alternatively, [7,11] resized the training images to a fixed height while keeping their aspect ratios unchanged. This, however, can only partially solve the scale problem as character scales may still vary because of font style, text arrangement, distortion, *etc.* Hence, it is still unlikely to have a well-balanced distribution of character scales in the training set. Training with such a dataset will definitely make the encoder biased towards extracting features from the predominant scale and not able to generalize well to characters of different scales. Unlike the single-CNN encoder, SAFE can share the knowledge of feature encoding across different character scales and effectively extract discriminative features from characters of different scales. This enables the text recognizer to have a much more robust performance (see Fig. 2c).

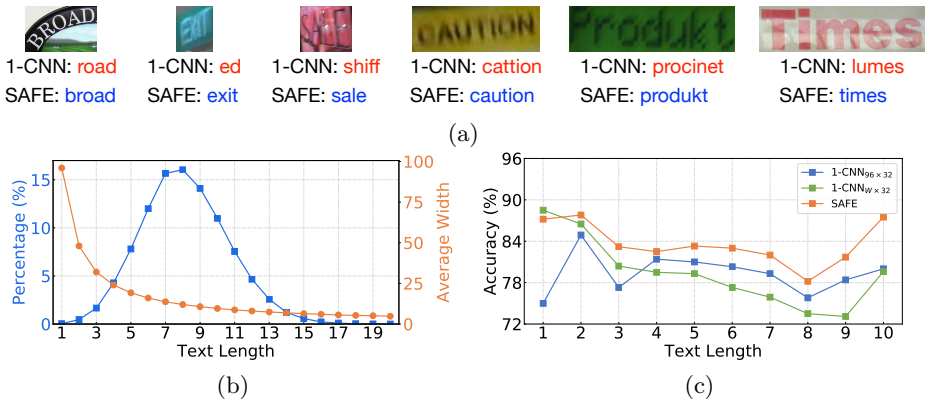

Fig. 2: Advantages of SAFE over the single-CNN encoder (1-CNN). (a) SAFE performs better than 1-CNN in handling distorted or low-quality text images. (b) The distribution of text lengths in SynthR [10] and the average character width in text images of different lengths after being resized to 96×32. (c) Detailed performance of recognizers with 1-CNN and with SAFE on recognizing text of different lengths. All the models are trained using SynthR. 1-CNN$_{96\times32}$ is trained by resizing the images to 96×32, whereas 1-CNN$_{W\times32}$ is trained by only rescaling images to a fixed height while keeping their aspect ratios unchanged.

To evaluate the effectiveness of SAFE, we design a simple text recognizer named scale-spatial attention network (S-SAN) that employs SAFE as its feature encoder. Following [5,11], S-SAN employs a spatial attention network in its LSTM-based decoder to handle a certain degree of text distortion. Experiments are carried out on six public benchmarks, and experimental results demonstrate that S-SAN can achieve state-of-the-art (or, in some cases, extremely competitive) performance without any post-processing.

2 Related Work

Many methods have been proposed for scene text recognition in recent years. According to their strategies of recognizing words in text images, previous text recognizers can be roughly classified into bottom-up and top-down approaches.

Recognizers based on the bottom-up approach first perform individual character detection and recognition using character-level classifiers, and then they integrate the results across the whole text image to get the final word recognition. In [12,13,14,15], traditional hand-crafted features (*e.g.*, HOG) were employed to train classifiers for detecting and recognizing characters in the text image. With the recent success of deep learning, many methods [16,17,18,19] used deep neural networks to train character classifiers. In [16], Wang *et al.* employed a CNN to extract features from the text image for character recognition. [17] first over-segmented the whole text image into multiple character regions before using a fully-connected neural network for character recognition. Unlike [16,17] which used a single character classifier, [18] combined a binary text/no-text classifier, a character classifier and a bi-gram classifier to compute scores for candidate words in a fixed lexicon. To recognise text without a lexicon, a structured output loss was used in their extended work [19] to optimize the individual character and N-gram predictors.

The above bottom-up methods require the segmentation of each character, which is a highly non-trivial task due to the complex background and different font size of text in the image. Recognizers [2,3,4,6,7,8,10,11,20,21] which are based on the top-down approach directly recognize the entire word in the image. In [10] and [20], Jaderberg *et al.* extracted CNN features from the entire image, and performed a 90k-way classification (90k being the size of a pre-defined dictionary). Instead of using only CNNs, [7,8,21] also employed recurrent neural networks to encode features of word images. All these three models were optimized by the connectionist temporal classification [22], which does not require an alignment between the sequential features and the ground truth labelling.

Following the success of the attention mechanism [1] in neural machine translation, recent text recognizers [2,3,4,5,6,11] introduced a learnable attention network in their decoders to automatically select the most relevant features for recognizing individual characters. In order to handle distorted scene text, [2,8] employed a spatial transformer network (STN) [23] to rectify the distortion of the entire text image before recognition. As it is difficult to successfully train a STN from scratch, Cheng *et al.* [6] proposed to encode the text image from multiple directions and used a filter gate to generate the final features for decoding. Unlike [2,6,8] which rectified the distortion of the entire image, a hierarchical attention mechanism was introduced in [5] to rectify the distortion of individual characters.

Different from [2,5,6,8] which focused on recognizing severely distorted text images, we address the problem of having character with different scales. This is a common problem that exists in both distorted and undistorted text recognition. The scale-spatial attention network (S-SAN) proposed in this paper belongs to the family of attention-based encoder-decoder neural networks. Unlike

previous methods [2,3,4,5,6,7,8] which employed only a single CNN as their feature encoder, we introduce a scale aware feature encoder (SAFE) to extract scale-invariant features from characters with different scales. This guarantees a much more robust feature extraction for the text recognizer. Although a similar idea has been proposed for semantic segmentation [24] to merge segmentation maps from different scales, it is the first time that the scale problem has been identified and efficiently handled for text recognition using an attention-based encoder-decoder framework. Better still, SAFE can also be easily deployed in other text recognizers to further boost their performance.

3 Scale Aware Feature Encoder

The scale aware feature encoder (SAFE) is conceptually simple: in order to extract discriminative features from characters with different scales, feature encoding of the text image should first be carried out under different scales, and features from the most relevant scale(s) are then selected at each spatial location to form the final feature map for the later decoding. SAFE is thus a natural and intuitive idea. As illustrated in Fig. 3, SAFE is composed of a multi-scale convolutional encoder and a scale-attention network. Throughout this paper, we omit the bias terms for improved readability.

Multi-Scale Convolutional Encoder. The multi-scale convolutional encoder is responsible for encoding the original text images under multiple scales. The basic component of the multi-scale convolutional encoder takes the form of a backbone convolutional neural network. Given a single gray image $\mathbf{I}$ as input, a multi-scale pyramid of images $\{\mathbf{X}_s\}_{s=1}^{N}$ is first generated. $\mathbf{X}_s$ here denotes the image at scale s having a dimension of $W_s \times H_s$ (width$\times$height), and $\mathbf{X}_1$ is the image at the finest scale. In order to encode each image in the pyramid, N backbone CNNs which share the same set of parameters are employed in the multi-scale convolutional encoder (see Fig. 3). The output of our multi-scale convolutional encoder is a set of N spatial feature maps

$$\{\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_N\} = \{\mathrm{CNN}_1(\mathbf{X}_1), \mathrm{CNN}_2(\mathbf{X}_2), \cdots, \mathrm{CNN}_N(\mathbf{X}_N)\}, \quad (1)$$

where CNN_s and $\mathbf{F}_s$ denote the backbone CNN and the output feature map at scale $s \in \{1, 2, \cdots, N\}$ respectively. $\mathbf{F}_s$ has a dimension of $W'_s \times H'_s \times C'$ (width$\times$height$\times$#channels).

Unlike dominant backbone CNNs adopted by [2,4,6,7,8] which compress information along the image height dimension and generate unit-height feature maps, each of our backbone CNNs keeps the spatial resolution of its feature map $\mathbf{F}_s$ close to that of the corresponding input image $\mathbf{X}_s$. Features from both text and non-text regions, as well as from different characters, therefore remain distinguishable in both the width and height dimensions in each feature map. In order to preserve the spatial resolution, we only apply two 2×2 down-sampling layers (see Fig. 3) in each of our backbone CNNs. Each feature map $\mathbf{F}_s$ is therefore sub-sampled only four times in both width and height (i.e., $W'_s = W_s/4$ and

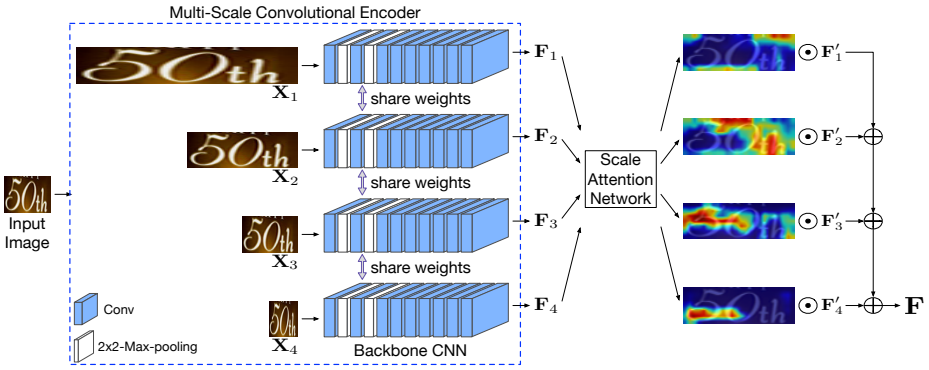

Fig. 3: The architecture of SAFE ($N = 4$). As most of the text images contain only a single word, the scale problem along the height dimension is not as severe as that along the width dimension. In our implementation, we resize the input images to 4 different resolutions, namely 192×32, 96×32, 48×32 and 24×32 respectively. The saliency maps in the scale attention network represent the scale attention for the feature maps extracted under 4 different resolutions. $\odot$ and $\oplus$ denote element-wise multiplication and summation respectively.

$H'_s = H_s/4$). More implementation details related to our backbone CNNs can be found in Section 5.2.

Scale Attention Network. The scale attention network is responsible for automatically selecting features from the most relevant scale(s) at each spatial location to generate the final scale-invariant feature map $\mathbf{F}$. In the proposed scale attention network, the N convolutional feature maps $\{\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_N\}$ outputted from the multi-scale convolutional encoder are first up-sampled to the same resolution

$$\{\mathbf{F}'_1, \mathbf{F}'_2, \cdots, \mathbf{F}'_N\} = \{u(\mathbf{F}_1), u(\mathbf{F}_2), \cdots, u(\mathbf{F}_N)\}, \tag{2}$$

where $u(\cdot)$ is an up-sampling function, and $\mathbf{F}'_s$ is the up-sampled feature map with a dimension of $W' \times H' \times C'$ (width$\times$height$\times$#channels). Through extensive experiments, we find that the performances of different up-sampling functions (*e.g.*, bilinear interpolation, nearest interpolation, *etc.*) are quite similar. We therefore simply use bilinear interpolation in our implementation.

After obtaining the up-sampled feature maps, our scale attention network utilizes an attention mechanism to learn to weight features from different scales. Since the size of the characters may not be constant within a text image (see Fig. 3), the computation of attention towards features from different scales takes place at each spatial location. In order to select the features from the most relevant scale(s) at the spatial location (i, j), N scores are first computed for

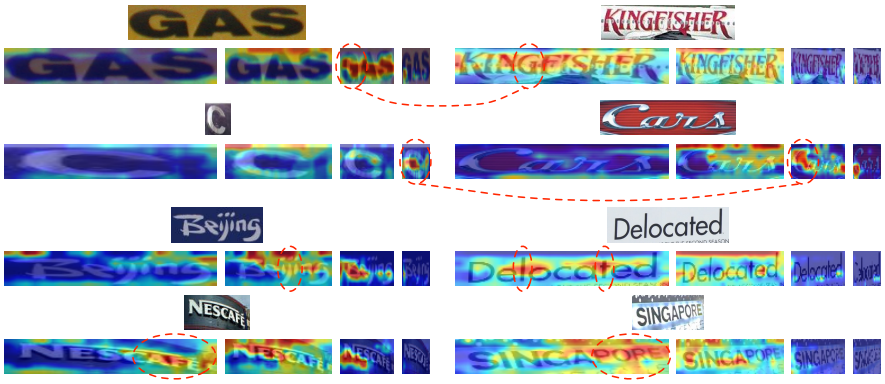

Fig. 4: Visualization of scale attention in the proposed SAFE. The saliency maps of the scale attention are superimposed on the images of the corresponding scales.

evaluating the importance of features from the N different scales

$$\begin{bmatrix} f_1(i,j) \\ f_2(i,j) \\ \vdots \\ f_N(i,j) \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{F}'_1(i,j) \\ \mathbf{F}'_2(i,j) \\ \vdots \\ \mathbf{F}'_N(i,j) \end{bmatrix} \tag{3}$$

where $f_s(i,j)$ is the score of the C'-dimensional feature vector $\mathbf{F}'_s(i,j)$ at location (i,j) of the corresponding feature map $\mathbf{F}'_s$ at scale s, and $\mathbf{W}$ is the parameter matrix. The proposed scale attention network then defines the scale attention at location (i,j) as

$$\omega_s(i,j) = \frac{\exp(f_s(i,j))}{\sum_{s'=1}^{N} \exp(f_{s'}(i,j))}, \tag{4}$$

where $\omega_s(i,j)$ is the attention weight for the feature vector $\mathbf{F}'_s(i,j)$. Finally, the scale-invariant feature map $\mathbf{F}$ can be computed as

$$\mathbf{F}(i,j) = \sum_{s=1}^{N} \omega_s(i,j)\mathbf{F}'_s(i,j). \tag{5}$$

The final feature map $\mathbf{F}$ has the same dimension as each up-sampled feature map $\mathbf{F}'_s$. Note that the scale attention network together with the multi-scale convolutional encoder are optimized in an end-to-end manner using only the recognition loss. As illustrated in Fig. 4, although we do not have the ground truth to supervise the feature selection across different scales, the scale attention network can automatically learn to attend on the features from the most relevant scale(s) for generating the final feature map $\mathbf{F}$.

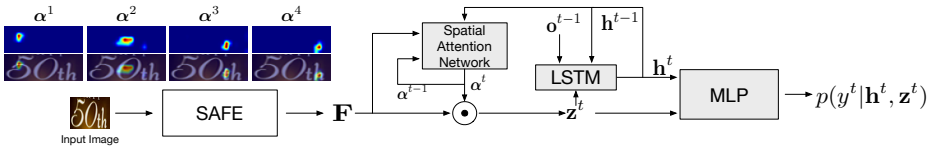

Fig. 5: The architecture of S-SAN. The saliency maps represent the spatial attention at different decoding steps. $\odot$ denotes element-wise multiplication.

4 Scale-Spatial Attention Network

To demonstrate the effectiveness of SAFE proposed in Section 3, we design a scale-spatial attention network (S-SAN) for scene text recognition. S-SAN uses SAFE as its feature encoder, and employs a character aware decoder composed of a spatial attention network and a LSTM-based decoder. Fig. 5 shows the overall architecture of S-SAN.

4.1 SAFE

To extract scale-invariant features for recognizing characters with different scales, S-SAN first employs SAFE proposed in Section 3 to encode the original text image. By imposing weight sharing across backbone CNNs of different scales, SAFE keeps its number of parameters to a minimum. Comparing with previous encoders adopted by [2,3,4,5,6,7,8,25] for text recognition, SAFE not only keeps its structure as simple as possible (see Table 1), but also effectively handles characters with different scales. This enables S-SAN to achieve a much better performance on recognizing text in natural images (see Table 3 and Table 4).

4.2 Character Aware Decoder

The character aware decoder of S-SAN is responsible for recurrently translating the scale-invariant feature map $\mathbf{F}$ into the corresponding ground truth labelling $\mathbf{y} = \{y^1, y^2, ..., y^T, y^{T+1}\}$. Here, T denotes the length of the text and y^{T+1} is the end-of-string (eos) token representing the end of the labelling. In this section, we refer to the process of predicting each character y^t as one time/decoding step.

Spatial Attention Network. The function of the spatial attention network is to generate a sequence of context vectors from the scale-invariant feature map $\mathbf{F}$ for recognizing individual characters. Following [5,11], our spatial attention network extends the standard 1D attention mechanism [2,3,4,6] to the 2D spatial domain. This allows the decoder to focus on features at the most relevant spatial locations for recognizing individual characters, and handle a certain degree of text distortion in natural images. At a particular decoding step t, we compute

the context vector $\mathbf{z}^t$ as a weighted sum of all feature vectors in $\mathbf{F}$, i.e.,

$$\mathbf{z}^t = \sum_{i=1}^{W'} \sum_{j=1}^{H'} \alpha^t(i,j)\mathbf{F}(i,j), \tag{6}$$

where $\alpha^t(i,j)$ is the weight applied to $\mathbf{F}(i,j)$ as determined by the spatial attention network. For the recognition of a single character at decoding step t, the context vector $\mathbf{z}^t$ should only focus on features at the most relevant spatial locations (i.e., those corresponding to the single character being recognized). To compute $\alpha^t(i,j)$ at the current decoding step t , we first exploit a simple CNN to encode the previous attention map $\boldsymbol{\alpha}^{t-1}$ into

$$\mathbf{A}^{t-1} = \mathrm{CNN}_{\mathrm{spatial}}(\boldsymbol{\alpha}^{t-1}). \tag{7}$$

We then evaluate a relevancy score at each spatial location as

$$r^t(i,j) = \mathbf{w}^{\mathrm{T}}\tanh[\mathbf{M}\mathbf{h}^{t-1} + \mathbf{U}\mathbf{A}^{t-1}(i,j) + \mathbf{V}\mathbf{F}(i,j)], \tag{8}$$

where $\mathbf{h}^{t-1}$ is the previous hidden state of the decoder (explained later), $\mathbf{M}$, $\mathbf{U}$ and $\mathbf{V}$ are the parameter matrices, and $\mathbf{w}$ is a parameter vector. Finally, we normalize the scores to obtain the attention weight

$$\alpha^t(i,j) = \frac{\exp(r^t(i,j))}{\sum_{i'=1}^{W'}\sum_{j'=1}^{H'}\exp(r^t(i',j'))}. \tag{9}$$

LSTM-based Decoder. The actual decoder of S-SAN takes the form of a long short-term memory layer and a multi-layer perceptron (MLP). Let L denote the set of 37-class (26 letters + 10 digits + eos) case-insensitive characters in our task. At the decoding step t, the LSTM-based decoder defines a probability distribution over L as

$$\mathbf{h}^t = \mathrm{LSTM}(\mathbf{h}^{t-1}, \mathbf{o}^{t-1}, \mathbf{z}^t) \tag{10}$$

$$p(y^t|\mathbf{h}^t, \mathbf{z}^t) = \mathrm{SoftMax}(\mathbf{W}_y \begin{bmatrix} \mathbf{h}^t \\ \tanh(\mathbf{W}_z\mathbf{z}^t) \end{bmatrix}), \tag{11}$$

where $\mathbf{h}^{t-1}$ and $\mathbf{h}^t$ denote the previous and current hidden states respectively, $\mathbf{o}^{t-1}$ is the one-hot encoding of the previous character y^{t-1}, $\mathbf{W}_y$ and $\mathbf{W}_z$ are the parameters of two linear layers in the MLP, and SoftMax denotes the final layer of the MLP for outputting the probability. The probability of the sequential labelling is then given by the product of the probability of each label, i.e.,

$$p(\mathbf{y}|\mathbf{I}) = \prod_{t=1}^{T+1} p(y^t|\mathbf{h}^t, \mathbf{z}^t). \tag{12}$$

5 Datasets and Implementation Details

5.1 Datasets

Following [2,3,5,7,8,19,20], S-SAN is trained using the most commonly adopted dataset released by Jaderberg *et al.* [10] (referred to as SynthR). This dataset contains 8-million synthetic text images together with their corresponding word labels. The distribution of text lengths in SynthR is shown in Fig. 2b. The evaluation of S-SAN is carried out on the following six public benchmarks.

- **IIIT5K** [26] contains 3,000 word test images collected from the Internet. Each image has been associated to a 50-word lexicon and a 1K-word lexicon.
- **Street View Text (SVT)** [13] contains 647 test word images which are collected from Google Street View. Each word image has a 50-word lexicon.
- **ICDAR-2003 (IC03)** [27] contains 860 text images for testing. Each image has a 50-word lexicon defined by Wang *et al.* [13]. A full lexicon is constructed by merging all 50-word lexicons. Following [13], we recognize the images having only alphanumeric words (0-9 and A-Z) with at least three characters.
- **ICDAR-2013 (IC13)** [28] is derived from ICDAR-2003, and contains 1,015 cropped word test images without any pre-defined lexicon. Previous methods [2,3,5] recognise images containing only alphanumeric words with at least three characters, which results in 857 images left for evaluation. In Section 6, we refer to IC13 with 1,015 images as **IC13-L** and that with 857 images as **IC13-S**, where L and S stand for large and small respectively.
- **Street View Text Perspective (SVT-P)** [29] contains 639 test images which are specially picked from the side-view angles in Google Street View. Most of them suffer from a large perspective distortion.
- **ICDAR Incidental Scene Text (ICIST)** [30] contains 2,077 text images for testing. Many of them are severely distorted and blurry. To make a fair comparison with [4,25], we also discard the images with non-alphanumeric characters in ICIST, which results in 1,811 images for testing. Like IC13, we refer to ICIST with 2,077 images as **ICIST-L** and that with 1,811 images as **ICIST-S** in Section 6.

5.2 Network Architecture

Scare Aware Feature Encoder. The detailed architecture of the backbone CNN in SAFE is illustrated in Table 1. The basic backbone CNN consists of nine convolutional layers. The stride and padding size of each convolutional layer are both equal to 1. All the convolutional layers use ReLU as the activation function. Batch normalization [31] is used after every convolutional layer. In our implementation, the multi-scale pyramid of images used in the multi-scale convolutional encoder has 4 scales, with images having a dimension of 192×32, 96×32, 48×32 and 24×32 respectively. The dimensions of the corresponding feature maps $\{\mathbf{F}_s\}_{s=1}^{4}$ are $48 \times 8 \times 512$, $24 \times 8 \times 512$, $12 \times 8 \times 512$ and $6 \times 8 \times 512$

Table 1: Comparison of encoders among different text recognizers. 'STN', 'BLSTM' and 'SCAN' denote the spatial transformer network [23], the bi-directional LSTM and our scale attention network respectively.

Model	STN	Backbone CNN					BLSTM	SCAN	Encoder Size
		Unit 1	Unit 2	Unit 3	Unit 4	Unit 5			
CRNN [7]	0	$[3,64]\times1$ (2,2,2,2)	$[3,128]\times1$ (2,2,2,2)	$[3,256]\times2$ (2,2,1,2)	$[3,512]\times2$ (2,2,1,2)	$[2,512]\times1$	2	0	8.3M
RARE [2]	1								16.5M
STAR-Net [8]	1	$[3,64]\times5$ (2,2,2,2)	$[3,128]\times4$ (2,2,2,2)	$[3,256]\times4$ (1,2,1,2)	$[3,512]\times4$ (1,2,1,2)	$[3,512]\times1$	1	0	14.6M
Char-Net [5]	1	$[3,64]\times3$ (2,2,2,2)	$[3,128]\times2$ (2,2,2,2)	$[3,256]\times6$	$[3,256]\times4$	$[3,512]\times3$	0	0	12.8M
FAN [4] Bai *et al.* [25]	0	$[3,32]\times1$ $[3,64]\times1$ (2,2,2,2)	$[3,128]\times3$ (2,2,2,2)	$[3,256]\times5$ (2,2,1,2)	$[3,512]\times11$	$[3,512]\times6$ $[2,512]\times2$	1	0	45.8M
1-CNN	0	$[3,64]\times1$ (2,2,2,2)	$[3,128]\times1$ (2,2,2,2)	$[3,256]\times3$	$[3,512]\times3$	$[3,512]\times1$	0	0	9.03M
SAFE								1	9.04M
SAFE_{res}	0	$[3,64]\times2$ (2,2,2,2)	$[3,128]\times3$ (2,2,2,2)	$[3,256]\times5$	$[3,512]\times9$	$[3,512]\times7$	0	1	38.9M

The configurations of all the convolutional layers in the backbone CNN follow the format of $[kernel\ size, number\ of\ channels] \times number\ of\ layers$. Cells with a gray background indicate convolutional blocks with residue connections. For the max-pooling layers, the kernel size and strides in both width and height dimensions follow the format of $(kernel_w, kernel_h, stride_w, stride_h)$.

respectively. In our scale attention network, the spatial resolutions of the up-sampled[2] feature maps $\{\mathbf{F}'_s\}_{s=1}^{4}$ are all 24×8 (i.e., same as that of $\mathbf{F}_2$ which is extracted from $\mathbf{X}_2$ with a dimension of 96×32).

Character Aware Decoder. The character aware decoder employs a LSTM layer with 256 hidden states. In the spatial attention network, $\text{CNN}_{\text{spatial}}(\cdot)$ is one 7×7 convolutional layer with 32 channels. In Eq. (8), $\mathbf{h}^{t-1}$, $\mathbf{A}(i,j)$ and $\mathbf{F}(i,j)$ are vectors with a dimension of 256, 32 and 512 respectively, and $\mathbf{w}$ is a parameter vector of 256 dimensions. Consequently, the dimensions of $\mathbf{M}$, $\mathbf{U}$ and $\mathbf{V}$ are 256×256, 256×32 and 256×512 respectively. The initial spatial attention map is set to zero at each position.

5.3 Model Training and Testing.

Adadelta [32] is used to optimize the parameters of S-SAN. During training, we use a batch size of 64. The proposed model is implemented using Torch7 and trained on a single NVIDIA GTX1080 GPU. It can process about 150 samples per second and converge in five days after about eight epochs over the training dataset. S-SAN is trained in an end-to-end manner only under the supervision of the recognition loss (i.e., the negative log-likelihood of Eq.(12)). During testing, for unconstrained text recognition, we directly pick the label with the highest probability in Eq. (11) as the output of each decoding step. For constrained recognition with lexicons of different sizes, we calculate the edit distances between the prediction of the unconstrained text recognition and all words in the

[2] We obtain $\mathbf{F}'_1$ by actually down-sampling $\mathbf{F}_1$.

Table 2: Ablation study of the proposed SAFE.

Model	Image Size	IIIT5K	SVT	IC03	IC13-S	IC13-L	SVT-P	ICIST-S	ICIST-L
1-CNN$_{24\times32}$	24×32	71.0	70.9	82.8	80.2	79.3	54.6	56.1	51.0
1-CNN$_{48\times32}$	48×32	81.3	81.8	90.5	87.6	86.2	68.8	66.3	60.3
1-CNN$_{96\times32}$	96×32	82.6	83.2	91.0	89.7	87.7	69.9	67.1	61.7
1-CNN$_{192\times32}$	192×32	82.1	83.2	90.6	89.5	87.2	68.5	65.1	59.5
1-CNN$_{W\times32}$	$W \times 32$	83.6	82.2	91.3	90.1	89.4	63.3	62.4	57.4
S-SAN	48×32, 96×32	84.2	84.7	92.3	89.7	88.3	71.6	70.5	64.4
	96×32, 192×32	83.7	84.9	92.4	**91.1**	88.9	72.1	68.8	62.9
	24×32, 48×32 96×32	85.0	84.4	92.0	90.7	89.8	72.9	70.7	64.7
	48×32, 96×32 192×32	85.0	84.5	91.9	90.7	89.7	71.8	70.3	64.4
	24×32, 48×32 96×32, 192×32	**85.2**	**85.5**	**92.9**	**91.1**	**90.3**	**74.4**	**71.8**	**65.7**

lexicon, and take the one with the smallest edit distance and highest recognition probability as our final prediction. In Table 3 and Table 4, 'None' indicates unconstrained scene text recognition (*i.e.*, without imposing any lexicon). '50', '1K' and 'Full' denote each scene text recognized with a 50-word lexicon, a 1,000-word lexicon and a full lexicon respectively.

6 Experiments

Experiments are carried out on the six public benchmarks. Ablation study is first conducted in Section 6.1 to illustrate the effectiveness of SAFE. Comparison on the performance of S-SAN and other methods is then reported in Section 6.2, which demonstrates that S-SAN can achieve state-of-the-art performance on standard benchmarks for text recognition.

6.1 Ablation Study

In this section, we conduct an ablation study to demonstrate the effectiveness of SAFE. Note that all the models in the ablation study are trained using the commonly adopted SynthR dataset, and we only use the accuracies of unconstrained text recognition for comparison.

To demonstrate the advantages of SAFE over the single-CNN encoder (1-CNN), we first train five versions of text recognizers which employ single backbone CNN for feature encoding. The architecture of the backbone CNNs and the decoders in the single-scale recognizers are identical to those in S-SAN. As illustrated in Table 2, we train the recognizers with the single-CNN encoder under different experimental settings, which resize the input text image to different resolutions. For simplicity, we directly use the encoder with the resolution of the training images to denote the corresponding recognizer. For example, 1-CNN$_{24\times32}$ stands for the single-CNN recognizer trained with images having a resolution of 24×32. As resizing images to a fixed resolution during training

Table 3: Text recognition accuracies (%) on six pubic benchmarks. *[20] is not strictly unconstrained text recognition as its outputs are all constrained to a pre-defined 90K dictionary.

Method	IIIT5K			SVT		IC03			IC13-S	IC13-L	SVT-P	ICIST-L
	50	1K	None	50	None	50	Full	None	None	None	None	None
ABBYY [13]	24.3	-	-	35.0	-	56.0	55.0	-	-	-	-	-
Wang *et al.* [13]	-	-	-	57.0	-	76.0	62.0	-	-	-	-	-
Mishra *et al.* [26]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-	-	-
Wang *et al.* [16]	-	-	-	70.0	-	90.0	84.0	-	-	-	-	-
PhotoOCR [17]	-	-	-	90.4	78.0	-	-	-	-	87.6	-	-
Phan *et al.* [29]	-	-	-	73.7	-	82.2	-	-	-	-	-	-
Almazán *et al.* [33]	91.2	82.1	-	89.2	-	-	-	-	-	-	-	-
Lee *et al.* [34]	-	-	-	80.0	-	88.0	76.0	-	-	-	-	-
Yao *et al.* [15]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-	-	-
Jaderberg *et al.* [18]	-	-	-	86.1	-	96.2	91.5	-	-	-	-	-
Su *et al.* [35]	-	-	-	83.0	-	92.0	82.0	-	-	-	-	-
Jaderberg *et al.* [19]	95.5	89.6	-	93.2	71.7	97.8	97.0	89.6	-	81.8	-	-
Jaderberg *et al.* [20]	97.1	92.7	-	95.4	80.7*	**98.7**	**98.6**	**93.1***	-	**90.8***	-	-
R^2AM [3]	96.8	94.4	78.4	96.3	80.7	97.9	97.0	88.7	90.0	-	-	-
CRNN [7]	97.8	95.0	81.2	**97.5**	82.7	98.7	98.0	91.9	-	89.6	66.8	-
SRN [2]	96.5	92.8	79.7	96.1	81.5	97.8	96.4	88.7	87.5	-	-	-
RARE [2]	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6	-	71.8	-
STAR-Net [8]	97.7	94.5	83.3	95.5	83.6	96.9	95.3	89.9	-	89.1	73.5	-
Char-Net [5]	-	-	83.6	-	84.4	-	-	91.5	90.8	-	73.5	60
S-SAN	**98.4**	**96.1**	**85.2**	97.1	**85.5**	98.5	97.7	92.9	**91.1**	90.3	**74.4**	**65.7**

and testing would result in the training dataset having very few images containing large characters (see Fig. 2b), we also train a version of the single-CNN recognizer by resizing the image to a fixed height while keeping its aspect ratio unchanged (denoted as 1-CNN$_{W\times 32}$ in Table 2). In order to train the recognizer with a batch size larger than 1, we normalize all the images to have a resolution of 192×32. For an image whose width is smaller than 192 after being resized to a height of 32, we pad it with the mean value to make its width equal to 192. Otherwise, we directly resize the image to 192×32 (there are very few text images with a width larger than 192). From the recognition results reported in Table 2, we see that S-SAN with SAFE outperforms the other recognizers with a single-CNN encoder by a large margin.

We also evaluate the scale selection of SAFE by using different combinations of rescaled images in the pyramid. As shown in Table 2, S-SAN with four scales (192×32, 96×32, 48×32 and 24×32) in the pyramid has the best performance on all the six public benchmarks.

6.2 Comparison with Other Methods

The recognition accuracies of S-SAN are reported in Table 3. Compared with the recent deep-learning based methods [2,3,5,7,8,19,20], S-SAN can achieve state-of-the-art (or, in some cases, extremely comparative) performance on both constrained and unconstrained text recognition.

Table 4: Text recognition accuracies (%) of models using more training data.

Method	IIIT5K			SVT		IC03			IC13-L	SVT-P	ICIST-S	ICIST-L
	50	1K	None	50	None	50	Full	None	None	None	-	None
AON [6]	**99.6**	98.1	87.0	96.0	82.8	98.5	97.1	91.5	-	73.0	-	68.2
FAN [4]	99.3	97.5	87.4	97.1	85.9	**99.2**	97.3	94.2	93.3	-	70.6	66.2
Bai *et al.* [25]	99.5	97.9	88.3	96.6	87.5	98.7	97.9	94.6	**94.4**	-	73.9	-
S-SAN	99.0	97.9	90.5	97.1	87.0	98.5	97.6	93.7	91.9	77.1	76.9	70.7
S-SAN$_{\mathbf{res}}$	99.3	**98.3**	**91.5**	**98.6**	**89.6**	99.1	**98.0**	**94.9**	93.8	**81.6**	**80.0**	**73.4**

In particular, we compare S-SAN against RARE [2], STAR-Net [8] and Char-Net [5], which are specifically designed for recognizing distorted text. In order to handle the distortion of text, RARE, STAR-Net and Char-Net all employ spatial transformer networks (STNs) [23] in their feature encoders. From the recognition results in Table 3, we find S-SAN outperforms all the three models by a large magain on almost every public benchmark. Even for the datasets IIIT5K and SVT-P that contain distorted text images, S-SAN without using any spatial transformer network can still achieve either extremely competitive or better performance. In order to explore the advantages of S-SAN comprehensively, we further report the complexity of these three models. In our implementation, S-SAN has 10.6 million parameters in total, while the encoders of RARE, STAR-Net and Char-Net already have about 16.5 million, 14.8 million and 12.8 million parameters respectively, as reported in Table 1.

By comparing with RARE, STAR-Net and Char-Net, we can see that S-SAN is much simpler but can still achieve much better performance on recognizing distorted text. This is mainly because SAFE can effectively handle the problem of encoding characters with varying scales. On one hand, with the multi-scale convolutional encoder encoding the text images under multiple scales and the scale attention network automatically selecting features from the most relevant scale(s), SAFE can extract scale-invariant features from characters whose scales are affected by the distortion of the image. On the other hand, by explicitly tackling the scale problem, SAFE can put more attention on extracting discriminative features from distorted characters, which enables S-SAN a more robust performance when handling distortion of text images.

Note that the previous state-of-the-art methods [4,25] employ a 30-layer residual CNN with one BLSTM network in their encoders. We also train a deep version of our text recogniser (denoted as S-SAN$_{\text{res}}$), which employs a deep backbone CNN (denoted as SAFE$_{\text{res}}$ in Table 1). Besides, [4] and [25] were both trained using a 12-million dataset, which consists of 8-million images from SynthR and 4-million pixel-wise labeled images from [36]. Following [4,25], we also train S-SAN$_{\text{res}}$ using the 12-million dataset. The results of S-SAN$_{\text{res}}$ are reported in Table 4. As we can see from the table, S-SAN$_{\text{res}}$ achieves better results than [4,25] on almost every benchmark. In particular, S-SAN$_{\text{res}}$ significantly outperforms [4,25] on the most challenging dataset ICIST. Besides, unlike [4] which requires extra pixel-wise labeling of characters for training, S-SAN can be easily optimized using only the text images and their corresponding labels.

7 Conclusion

In this paper, we present a novel scale aware feature encoder (SAFE) to tackle the problem of having characters with different scales in the text images. SAFE is composed of a multi-scale convolutional encoder and a scale attention network. It can automatically extract scale-invariant features from characters with different scales. By explicitly handling the scale problem, SAFE can put more effort on handling other challenges in text recognition. Moreover, SAFE can transfer the learning of feature encoding across different character scales. This is particularly important for text recognizers to achieve a much more robust performance as it is nearly impossible to precisely control the scale distribution of characters in a training dataset. To demonstrate the effectiveness of SAFE, we design a simple but efficient scale-spatial attention network (S-SAN) for scene text recognition. Experiments on six public benchmarks illustrate that S-SAN can achieve state-of-the-art performance without any post-processing.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
3. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for ocr in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
4. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: Towards accurate text recognition in natural images. arXiv preprint arXiv:1709.02054v3 (2017)
5. Liu, W., Chen, C., Wong, K.Y.K.: Char-net: A character-aware neural network for distorted scene text recognition. In: AAAI Conference on Artificial Intelligence. (2018)
6. Cheng, Z., Liu, X., Bai, F., Niu, Y., Pu, S., Zhou, S.: Arbitrarily-oriented text recognition. arXiv preprint arXiv:1711.04226 (2017)
7. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)
8. Liu, W., Chen, C., Wong, K.K., Su, Z., Han, J.: Star-net: A spatial attention residue network for scene text recognition. In: British Machine Vision Conference. (2016)
9. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2016)
10. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: Workshop on Deep Learning, Advances in Neural Information Processing Systems. (2014)
11. Yang, X., He, D., Zhou, Z., Kifer, D., Giles, C.L.: Learning to read irregular text with attention mechanisms. In: Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. (2017)

12. Wang, K., Belongie, S.: Word spotting in the wild. In: European Conference on Computer Vision. (2010)
13. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: IEEE International Conference on Computer Vision. (2011)
14. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. (2012)
15. Yao, C., Bai, X., Shi, B., Liu, W.: Strokelets: A learned multi-scale representation for scene text recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. (2014)
16. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: IEEE International Conference on Pattern Recognition. (2012)
17. Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: Photoocr: Reading text in uncontrolled conditions. In: IEEE International Conference on Computer Vision. (2013)
18. Jaderberg, M., Vedaldi, A., Zisserman, A.: Deep features for text spotting. In: European Conference on Computer Vision. (2014)
19. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Deep structured output learning for unconstrained text recognition. In: International Conference on Learning Representations. (2015)
20. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International Journal of Computer Vision **116** (2016) 1–20
21. He, P., Huang, W., Qiao, Y., Loy, C.C., Tang, X.: Reading scene text in deep convolutional sequences. In: AAAI Conference on Artificial Intelligence. (2016)
22. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: International Conference on Machine Learning. (2006)
23. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Advances in Neural Information Processing Systems. (2015)
24. Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)
25. Bai, F., Cheng, Z., Niu, Y., Pu, S., Zhou, S.: Edit probability for scene text recognition. arXiv preprint arXiv:1805.03384 (2018)
26. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: British Machine Vision Conference. (2012)
27. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., Ashida, K., Nagai, H., Okamoto, M., Yamamoto, H., et al.: Icdar 2003 robust reading competitions: entries, results, and future directions. International Journal of Document Analysis and Recognition **7** (2005) 105–122
28. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Gomez i Bigorda, L., Robles Mestre, S., Mas, J., Fernandez Mota, D., Almazan Almazan, J., de las Heras, L.P.: Icdar 2013 robust reading competition. In: IEEE International Conference on Document Analysis and Recognition. (2013)
29. Phan, T., Shivakumara, P., Tian, S., Tan, C.: Recognizing text with perspective distortion in natural scenes. In: IEEE International Conference on Computer Vision. (2013)
30. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015

competition on robust reading. In: IEEE International Conference on Document Analysis and Recognition. (2015)
31. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. (2015)
32. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
33. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. IEEE Transactions on Pattern Analysis and Machine Intelligence **36** (2014) 2552–2566
34. Lee, C.Y., Bhardwaj, A., Di, W., Jagadeesh, V., Piramuthu, R.: Region-based discriminative feature pooling for scene text recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. (2014)
35. Su, B., Lu, S.: Accurate scene text recognition based on recurrent neural network. In: Asian Conference on Computer Vision. (2014)
36. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016)