SCAN: Sequence-character Aware Network for Text Recognition

Heba Hassan[1], Marwan Torki[2] and Mohamed E. Hussein[2,3]
[1]*Dept. of Computer Science and Engineering, Egypt-Japan University of Science and Technology, Egypt*
[2]*Dept. of Computer and Systems Engineering, Alexandria University, Egypt*
[3]*Information Sciences Institute, University of Southern California, U.S.A.*

Keywords: Text Recognition, Multi-task Learning.

Abstract: Text recognition continues to be a challenging problem in the context of text reading in natural scenes. Bearing in mind the sequential nature of text, the problem is usually posed as a sequence prediction problem from a whole-word image. Alternatively, it can also be posed as a character prediction problem. The latter approach is typically more robust to challenging word shapes. Attempting to find the sweet spot that attains the best of the two approaches, we propose Sequence-Character Aware Network (SCAN). SCAN starts by locating and recognizing the characters, and then generates the word using a sequence-based approach. It comprises two modules: a semantic-segmentation-based character prediction, and an encoder-decoder network for word generation. The training is done over two stages. In the first stage, we adopt a multi-task training technique with both character-level and word-level losses and trainable loss weighting. In the second stage, the character-level loss is removed, enabling the use of data with only word-level annotations. Experiments are conducted on several datasets for both regular and irregular text, showing state of the art performance of the proposed approach. It also shows that the proposed approach is robust against noisy word detection.

1 INTRODUCTION

Dealing with text recognition as a sequence recognition problem has its advantages as it leverages the sequential nature of text. However, the approach falls short when it comes to text with challenging shapes. Several methods have been proposed to deal with text of challenging shapes. In most recent works the text image is first rectified (Shi et al., 2018; Luo et al., 2019; Yang et al., 2019), and then sequence recognition is applied on the rectified image. Another group of methods handle irregular text shapes by starting with character prediction. The characters are then sorted to recognize the word (Lyu et al., 2018; Liao et al., 2019). In addition to handling irregular shapes, this approach has the advantage of being more robust against noisy text localization. In the proposed model, we adopt character prediction to handle irregular text shapes without the need for complex preposessing, such as rectification, while maintaining robustness against text localization errors.

Mere ordering of detected characters could lead to missing crucial sequential information in text. To address this issue, our proposed model uses a two-stage model. The first stage is a semantic segmentation network that is responsible for character prediction. It produces a pixel-level map with the locations and classes of the characters. The second stage is an encoder-decoder network that processes the map produced by the first stage and generates the final word, as shown in Figure 2. The character segmentation stage is a high-resolution network (HRNet) (Sun et al., 2019) for semantic segmentation. HRNet has proved efficiency in many recent semantic segmentation work as it maintains a high resolution representation of the image through the network. The word generation stage receives a map with the same size as the input image with the number of channels equals to the number of character classes plus the background class. A set of convolutional layers are used to encode the image into a sequence and then an LSTM decoder is used to produce the final result.

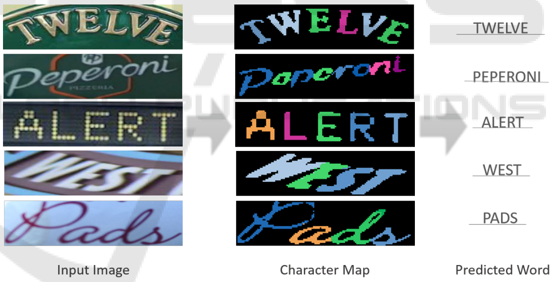

Figure 1: Examples taken from irregular text datasets, which include rotated, curved, distorted, and multi-font text. Our model first outputs a character map, then the map is used to generate a word.

Hassan, H., Torki, M. and Hussein, M.
SCAN: Sequence-character Aware Network for Text Recognition.
DOI: 10.5220/0010321106020609
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 602-609
ISBN: 978-989-758-488-6

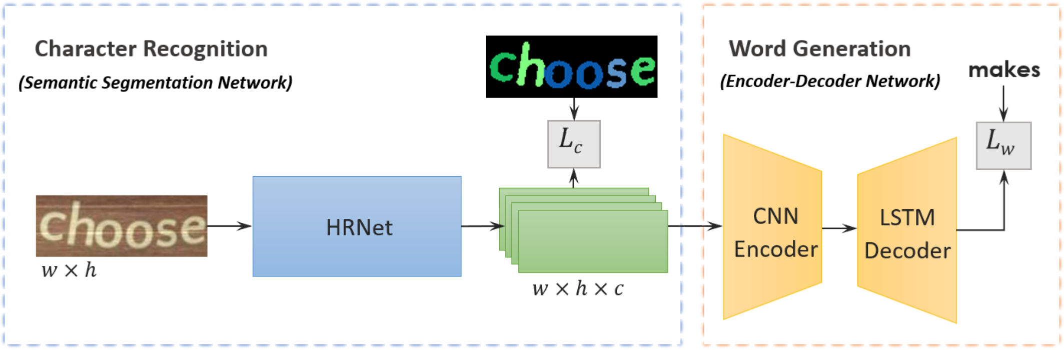

Figure 2: The proposed model architecture: A character map is first produced with a semantic segmentation network (HRNet (Sun et al., 2019), in particular) with a character-level loss L_c, and then the map is used to generate a word using an encoder-decoder network with a word-level loss L_w.

Our training procedure is done over two stages. The first stage is a multi-task training stage, where the two parts of character segmentation and word generation are trained together with both losses contributing to the final loss of the network. We use the adaptive weighting technique proposed in (Kendall et al., 2018) to ensure efficient training with the two losses. The second stage is the word-level stage where only one loss is used and the training is done with data with only word-level annotations. To overcome the absence of accurate ground truth for character segmentation, a pseudo ground truth is generated. The generated ground truth is used to train the character segmentation network, which is further supervised by the word generation network.

The main contribution of this paper could be summarized as follows:

- We propose Sequence-Character Aware Network (SCAN) for text recognition that combines both character and sequence awareness, which has proven to be robust for both regular and irregular text recognition. Our approach, while being simple and intuitive, outperforms other –more complex– recognition techniques.
- We deploy a state-of-the-art semantic segmentation network (HRNet), which yields a lightweight and efficient semantic segmentation for character prediction.
- We also deploy a multi-task training setting with adaptive loss weighting. Our experimental evaluation shows the effectiveness our training procedure.
- Finally, we propose a new technique for pseudo ground truth generation, which proves to be more efficient than using only the character bounding boxes as ground truth.

2 RELATED WORK

The techniques for text recognition could be divided into two main streams; namely character-based and sequence-based. Early techniques in text recognition used the characters as the main detection block. In such techniques, conventional features are deployed to detect the characters, which are then sorted to form a word. For example, in (Wang et al., 2011; Wang and Belongie, 2010), characters are detected using a multi-scale sliding-window search along with a classifier to classify the character windows, the positively classified of which are later grouped into words. In some other works, binarization techniques, such as Maximally Stable Extremal Regions (MSER) (Neumann and Matas, 2010) or Stroke Width Transform (SWT) (Epshtein et al., 2010), are used to detect the characters, relying on common texture and shape characteristics of text.

With the success of deep learning techniques, Jaderberg et al. (Jaderberg et al., 2016) considered the word recognition problem as a classification problem using a Convolutional Neural Network (CNN) with a dictionary of 90k words as the classes. Later, many approaches addressed the problem as a sequence recognition problem, considering the cropped word image to be the input sequence and the desired word to be the output sequence (Su and Lu, 2014; Shi et al., 2016a; He et al., 2016b). Adding attention further boosted the performance of the sequence-based techniques, as illustrated in (Lee and Osindero, 2016; Cheng et al., 2017). To have a network that is more robust to irregularly warped words, e.g. oriented, curved, or skewed, rectification is employed to invert the warping applied to a word before recognizing it.

In contrast to sequence-based approaches, character-based approaches rely on detecting characters and then sorting them to form words. Such techniques offer natural handling of challenging text shapes. In (Lyu et al., 2018; Liao et al., 2019), Fully Convolutional Networks (FCNs) are used for semantic segmentation to detect the character locations and general orientation. The detected characters are then sorted from left to right to construct a word.

Each of the sequence-based and character-based approaches has its points of strength. The sequence-based approaches exploit the sequential nature of text, adding a language sense to the model, and hence leading to better word recognition. On the other hand, character-based approaches exhibit better capabilities in handling background clutter, noisy text localization, and irregular text shapes. In this paper, we are attempting to get the best of both streams. Our method uses character segmentation as a first stage, which can also be seen as a type of supervised spatial attention, focusing the sequence-based stage on the right character locations despite the possible irregularity in the text shape. Subsequently, the word is obtained via a sequence-based stage, which adds a sense of a language to the model, and hence adds extra word recognition accuracy.

3 PROPOSED MODEL

As shown in Figure 2, the model has two modules, the first is the character segmentation module, and the second is word generation module. In this section, we describe the two components in SCAN and the different stages of the training process.

3.1 Semantic Segmentation for Character Prediction

This part of the network is a semantic segmentation network that outputs a map with size $w \times h \times c$, where c is the number of classes. The HRNet-V1 networks architecture is adopted, which, as explained in (Sun et al., 2019), maintains high resolution representation by fusing high resolution features with low resolution features at each stage in the network. The network has four blocks. In each block, the low resolution features are up-sampled and added to the high resolution features while the high resolution features are down-sampled and added to the low resolution ones in a fully connected manner. The down-sampling is performed via a 2-strided convolution while the up-sampling is performed using bilinear interpolation. The last layer has a Sigmoid activation with the number of channels equal to the number of classes. The network has a cross entropy loss, which we denote as L_c for *character-level loss*, that is calculated per pixel between the ground-truth and the predicted semantic segmentation map, normalized by the total number of pixels.

Table 1: Architecture of the Encoder-Decoder Network.

Layer	Output Size	Configuration
Input	$64 \times 256 \times 38$	–
Conv	$32 \times 256 \times 64$	$[3 \times 3, 64]$, s: 2×1
Conv	$16 \times 128 \times 64$	$[3 \times 3, 64]$, s: 2×2
Conv	$8 \times 128 \times 128$	$[3 \times 3, 128]$, s: 2×1
Conv	$4 \times 64 \times 128$	$[3 \times 3, 128]$, s: 2×2
Conv	$2 \times 64 \times 256$	$[3 \times 3, 256]$, s: 2×1
Conv	$1 \times 32 \times 512$	$[3 \times 3, 512]$, s: 2×2
Bi-LSTM	32×512	512 unit
LSTM	32×512	512 unit
FC	32×38	38

3.2 Encoder-decoder Network for Word Generation

The word generation stage of the model takes the output from the character segmentation stage as its input, which is a feature map of the same size as the input image $h \times w \times c$, with c being the number of classes. The encoder is a set of convolutional layers that shapes the image in the form of a 1D sequence. The decoder is a bidirectional LSTM layer followed by an LSTM layer with self attention to produce the final output. The final layer has a Softmax activation with cross entropy loss, which we denote as L_w for *word-level loss*. The configuration for the encoder and the decoder are shown in Table 1.

3.3 Model Training

3.3.1 Ground Truth Generation

The ground truth mask for the characters is generated using the bounding boxes provided in the SynthText (Gupta et al., 2016) dataset as shown in Figure 3 by the following steps. First, A rectangular slice is taken from the image containing one character, guided by the ground truth bounding box. For each slice we perform simple thresholding using Otsu's method (Otsu, 1979) .We then concatenate the characters in one binary image. The image is then multiplied by another binary image formed from the ground truth boxes to eliminate some of the noise outside the characters, and to give each character its label. As a final

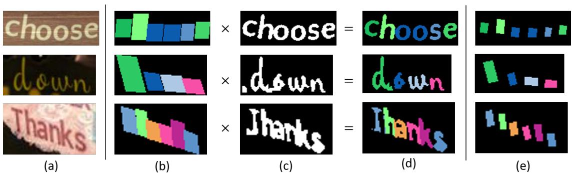

Figure 3: Ground truth generation: (a) image from SynthText; (b) Bounding Boxes ground truth; (c) Binary image after character thresholding; (d) Final ground truth used for training, which is the output of multiplying (b) and (c); (e)Rotated rectangles ground truth.

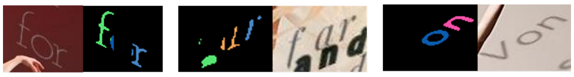

Figure 4: Examples of masks with IOU less than 30% with the bounding box mask.

verification step, we calculate the intersection over union (IOU) for the generated binary mask and the boxes ground truth and eliminate the image if the IOU is less than 30%. Some examples of images eliminated by this procedure could be seen in Figure 4. The number of eliminated images is about 1% of the number of images extracted from the dataset.

The obtained ground truth is considered a pseudo ground truth that is further guided by the second part of the network in the second stage of the training, as explained below. Having an accurate segmentation for each character is not the main goal of our network. However, having a more expressive ground truth would help both the semantic segmentation and the encoder-decoder networks perform better, as shown in Section 4. We also experimented with another way of generating ground truth, which is simply labeling each character with a rotated rectangle with the size of 50% of the size of the rotated rectangle surrounding the bounding box as shown in Figure 3. However, our approach performs better with the segmented character ground truth.

3.3.2 Multi-task Training Stage

One of the most important parameters that affect the training of a network with multiple losses is the weights assigned to each of the losses. The loss function would be the weighted sum of the losses in the system. However, due to the variation in the range of each loss and the uncertainty of each of the losses, the weights assigned to different losses must be fine tuned. We started by investigating several weight assignments for the two losses and observed the big effect it could have on the network's training behavior and performance.

A smarter way is to have the weight learnt by the network. In (Kendall et al., 2018), the use of learnable

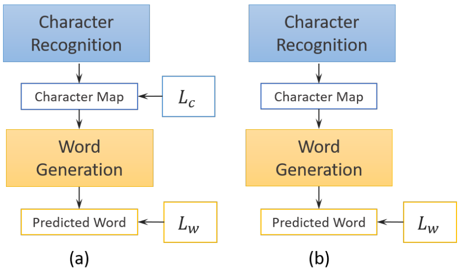

Figure 5: Training Stages: (a) multi-task training with two losses L_c and L_w; (b) word-level training with one loss L_w.

loss weights is investigated based on the homoscedastic uncertainty of each of the tasks in the network. The loss L for each task is mapped to another loss L^n that is formulated as:

$$L^n = \frac{1}{2\sigma^2}L + \log\sigma \qquad (1)$$

where σ is a the task uncertainty to be learnt by the network, and the second term works as a regularizer. According to (Kendall et al., 2018), this is found to be more stable than learning a linear weight multiplied by the loss. Our model has two losses, the character-level loss L_c, and the word-level loss L_w, as shown in Figure 5, the total loss L_t is formulated as:

$$L_t = \frac{1}{2\sigma_c^2}L_c + \frac{1}{2\sigma_w^2}L_w + \log\sigma_c + \log\sigma_w \qquad (2)$$

3.3.3 Word-level Training Stage

The need for ground truth for character locations is one of the biggest limitations in character-based text recognition, which is only available in synthetic data. In this stage of training, the network has only one output and one loss L_w as shown in Figure 5. In this stage, we can use data that only has word level annotations. Here, the semantic segmentation output is only supervised by the final word output, which helps the output of the character segmentation stage take a form that increases the accuracy of the sequence network.

4 EXPERIMENTS

In this section, we provide a brief description of the datasets used for training and evaluation. Next, the implementation and training details are presented. We also describe the evaluation criteria used to evaluate the model. We then present the results obtained on different datasets and ablation studies conducted to demonstrate the contribution of the proposed model's components.

Table 2: Word recognition accuracy for public datasets, both regular and irregular. ”50” means lexicon of size 50, ”1k” means lexicon of size one thousand, and ”None” means no lexicon. GT1 refers to the usage of rotated-rectangles ground truth. GT2 refers to the usage of segmented-character ground truth. En is an ensemble of the models trained with the two types of ground truth. Rot refers to using a rotation strategy at test time.

Method	**Regular**						**Irregular**		
	IIIT5k			SVT		IC13	IC15	SVT-P	CUTE80
	50	1k	None	50	None	None	None	None	None
(Wang et al., 2011)	-	-	-	57.0	-	-	-	-	-
(Mishra et al., 2012)	64.1	57.5	-	73.2	-	-	-	-	-
(Wang et al., 2012)	-	-	-	70.0	-	-	-	-	-
(Yao et al., 2014)	80.2	69.3	-	75.9	-	-	-	-	-
(Jaderberg et al., 2016)	97.1	92.7	-	95.4	80.7	90.8	-	-	-
(He et al., 2016a)	94.0	91.5	-	93.5	-	-	-	-	-
(Lee and Osindero, 2016)	96.8	94.4	78.4	96.3	80.7	90.0	-	-	-
(Shi et al., 2016b)	96.2	93.8	81.9	95.5	81.9	88.6	-	71.8	59.2
(Shi et al., 2016a)	97.8	95.0	81.2	97.5	82.7	89.6	-	66.8	54.9
(Yang et al., 2017)	97.8	96.1	-	95.2	-	-	-	75.8	69.3
(Cheng et al., 2017)	99.3	97.5	87.4	97.1	85.1	93.3	70.6	71.5	63.9
(Liu et al., 2018)	-	-	92.0	-	85.5	91.1	74.2	78.9	-
(Liao et al., 2019)	**99.8**	98.8	92.0	98.8	86.4	91.5	-	-	79.9
(Cheng et al., 2018)	99.6	98.1	87.0	96.0	82.8	-	68.2	73.0	76.8
(Shi et al., 2018)	99.6	**98.8**	93.4	**99.2**	**93.6**	91.8	76.1	78.5	79.5
(Luo et al., 2019)	97.9	96.2	91.2	96.6	88.3	92.4	68.8	76.1	77.4
(Huang et al., 2019)	99.6	**98.8**	**94.5**	97.1	90.0	93.9	75.3	79.8	84.7
(Yang et al., 2019)	99.5	**98.8**	94.4	97.2	88.9	**94.2**	78.7	80.8	87.5
(Qiao et al., 2020)	-	-	93.8	-	89.6	92.8	**80.0**	80.8	83.6
(Wang et al., 2020)	-	-	94.3	-	89.2	**94.2**	**80.0**	74.5	84.4
SCAN (GT1)	99.5	98.7	91.4	99.0	87.1	92.2	74.2	77.5	85.0
SCAN (GT2)	99.5	98.6	92.6	**99.2**	89.3	93.3	78.5	81.0	85.7
SCAN (En)	99.6	98.6	93.2	99.1	90.0	**94.2**	77.8	81.3	86.8
SCAN (En+Rot)	99.6	98.7	93.7	**99.2**	90.7	**94.2**	**80.0**	**81.7**	**87.8**

4.1 Datasets

There are several datasets used in text recognition: two large synthetic datasets, and several real datasets, which contain both regular and irregular text images.

- **SynthText:** It is a large synthetically generated dataset that was introduced in (Gupta et al., 2016). It provides ground truth boxes for both words and characters. We used the word bounding boxes to crop images of words, and the character boxes were used for training in the Multi-task stage.
- **Synth90K:** It is a synthetic dataset with images of words generated for text recognition task (Jaderberg et al., 2016). It contains about 9 million images generated from a dictionary of 90k words.
- **COCO-Text:** COCO-Text (Veit et al., 2016) is the largest dataset for text that contains real images with bounding boxes for words extracted from the original COCO dataset. It has 42618 images for training, 9896 images for validation, and 9837 images for testing.
- **ICDAR2013:** This dataset has 848 images for training and 1015 images for testing (Karatzas et al., 2013).
- **IIIT5k-Words:** This dataset (Mishra et al., 2012) has both front-end, perspective, and curved text with 3000 images for testing and 2000 for training. The images are associated with a 50-word lexicon and a 1000-word lexicon.
- **Street View Text (SVT):** This dataset (Wang et al., 2011) contains 647 images cropped from Google's street view dataset. Each image is associated with a 50-word lexicon.
- **CUTE:** This dataset is mainly for curved text (Risnumawan et al., 2014). The dataset contains 288 images with no associated lexicon.
- **ICDAR2015:** This dataset consists of mostly irregular text (Karatzas et al., 2015). It contains 4468 images for training and 2077 images for testing.
- **Street View Text Perspective (SVTP):** Images in the SVTP dataset (Quy Phan et al., 2013) were

cropped from Google street view dataset's images with perspective view.

4.2 Implementation and Training

The configuration and implementation details for HRNet follow the original paper (Sun et al., 2019). For the word generation part, the configuration for the convolutional encoder and the LSTM decoder are shown in Table 1. All the images are resized to the size 64 × 256. The number of classes used are 38 classes for letters and numbers, one class for special characters, and a background class. The training starts with the **multi-task training** with images extracted from SynthText dataset, using approximately 6 million images. After training for two epochs, training is switched to **word-level training** with around 500k images from SynthText, 500k images from Synth90K, and around 49k real images from the training sets in COCO-text, IIIT5K, ICDAR13 and ICDAR15 datasets. The initial learning rate is set to 10^{-3} and reduced on plateau by a factor of 0.8 until it reaches 10^{-5}

4.3 Evaluation Criteria

We use the word-accuracy as the main evaluation metric for both the character segmentation and the word recognition modules. For the character segmentation stage, after word-level training, we follow a simple technique by applying connected component analysis on character pixels. Each blob is considered a character and is given the label of the majority of its pixels. The characters are then sorted from left to right to form a word. The output of the second stage is a set of characters that readily form a word.

4.4 Experimental Results and Comparison with Other Methods

The proposed SCAN network is evaluated over six public datasets that contain regular and irregular text. Compared to the most recent prior methods, the network achieves better results in most of the datasets. As shown in Table 2, our network performs well for both regular and irregular datasets, achieving the best results in SVT, IC13, SVTP, and CUTE80. The effect of the proposed ground truth generation method could be observed from rows 1 and 2 in SCAN's results. From these results, we see that using the ground truth with segmented character shapes performs better than using the rotated rectangles ground truth. We also obtain even better performance with an ensemble of the two models as shown in the **SCAN (En)** row (for ensemble) in the table. The output for the ensemble is obtained by summing the two outputs from the two models. We also evaluate the test datasets with a rotation strategy as shown in the row **SCAN (En+Rot)**. In this strategy, we rotate the image in the range from -50 to 50 degrees with a step of 10 degrees. For each rotation angle, we estimate the confidence of the prediction as the sum of the most-likely-character probabilities for each pixel, excluding pixels classified as background. We then choose the prediction with the highest confidence.

4.5 Effect of Word-level Training on Character Prediction

In the word-level training stage, the character segmentation loss is removed and the network learns using the final word-level loss only. The effect on the accuracy can be seen in Table 3. The accuracy enhancement with this training stage is obvious. However, it is also interesting to inspect the effect of applying only the word-level loss on the segmentation map. After word-level training stage, the characters in the semantic segmentation map start to take a more concise form, representing the character's silhouette rather than the exact character shape, and also the characters appear to be more separated, as shown in Figure 6. Both of these properties appear to make the segmentation results more suitable for word generation.

Table 3: Word accuracy after training with multi-task training and word-level training.

Training stage	IIIT5k	SVT	IC13	IC15	SVTP	CU80
Multi-task	90.6	81.8	90.1	69.4	70.4	78.8
Word-level	92.6	89.3	93.3	78.5	81.0	85.7

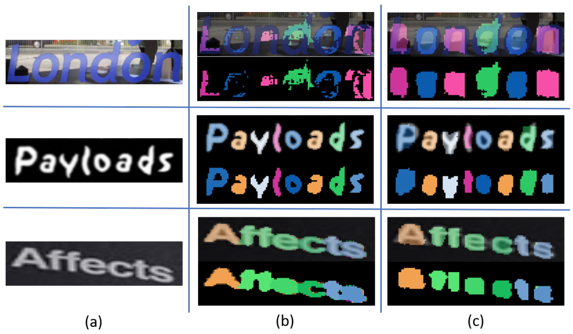

Figure 6: (a) Original image; (b) Character segmentation after the multi-task training stage; (c) Character segmentation after the word-level training stage.

Table 4: Word accuracy for the two outputs from both stages of the network.

Output	IIIT5k	SVT	IC13	IC15	SVTP	CU80
Sequence	93.7	90.7	94.2	80.0	81.7	87.8
Character	92.6	85.7	92.0	74.0	73.3	82.2

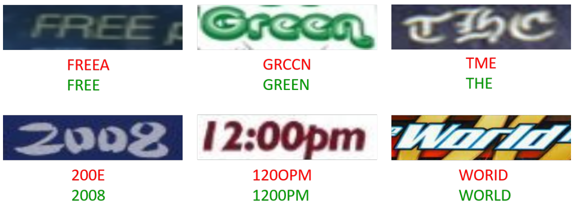

Figure 7: Images from IIIT5K dataset with predicted words, wrongfully predicted word by character sorting in red; corrected word by sequence prediction in green.

4.6 Effect of Encoder-decoder Network

We argued earlier that adding the encoder-decoder network to the model leverages the sequential characteristics of text. Here, we evaluate the word accuracy using the two outputs from both modules; character segmentation (character output) and word generation (sequence output). We can observe the effect of the sequence network on the accuracy in Table 4. This show the added value obtained by incorporating the sequence information into the model. In Figure 7, we present some of the cases where the sequence network corrected the output obtained from the semantic segmentation network.

4.7 Effect of Noisy Text Localization

One of the main points of strength in the proposed model is the ability to deal with noise in text detection. We conducted an experiment on the CUTE80 dataset to evaluate this characteristic of our model. The dataset has 80 images for text detection, from which 288 images are extracted for recognition. We used the detection dataset and cropped the words with a larger background by expanding the sides of the cropped rectangle with a random percentage of the side length, as shown in Figure 8. We refer to this new set of images with noisily localized words as the Noisy CUTE dataset. From Figure 8, it can be observed that the model still detects the characters and generate the correct word despite the added noise. The accuracy of our model on the CUTE dataset is 87.8%, while the accuracy on the Noisy CUTE dataset is 85.4%. With such a small difference in the recognition accuracy, the model proved to be robust against localization noise and background distraction.

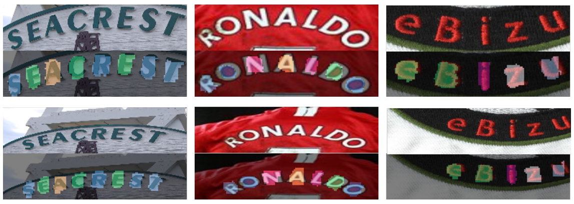

Figure 8: Character segmentation for images from CUTE dataset (top) and the Noisy CUTE dataset (bottom).

5 CONCLUSION

In this paper, we introduced a novel Sequence-Character Aware Network (SCAN) for text recognition that proved efficient for both regular and irregular text. Our method has the simplicity of character-based methods while benefiting from the added information provided by the sequence-based word generation, without the need for a complicated rectification process. The model has also proved its robustness against noisy text localization. Our future work will focus on developing our method into end-to-end text detection and recognition system, as well as applying our method to other languages.

REFERENCES

Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., and Zhou, S. (2017). Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5076–5084.

Cheng, Z., Xu, Y., Bai, F., Niu, Y., Pu, S., and Zhou, S. (2018). Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5571–5579.

Epshtein, B., Ofek, E., and Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970. IEEE.

Gupta, A., Vedaldi, A., and Zisserman, A. (2016). Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, P., Huang, W., Qiao, Y., Loy, C. C., and Tang, X. (2016b). Reading scene text in deep convolutional sequences. In *Thirtieth AAAI conference on artificial intelligence*.

Huang, Y., Luo, C., Jin, L., Lin, Q., and Zhou, W. (2019). Attention after attention: Reading text in the wild with cross attention. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 274–280. IEEE.

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20.

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., et al. (2015). Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE.

Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L. G., Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A., and De Las Heras, L. P. (2013). Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE.

Kendall, A., Gal, Y., and Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Lee, C.-Y. and Osindero, S. (2016). Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2231–2239.

Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., Yao, C., and Bai, X. (2019). Scene text recognition from two-dimensional perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8714–8721.

Liu, W., Chen, C., and Wong, K.-Y. K. (2018). Char-net: A character-aware neural network for distorted scene text recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Luo, C., Jin, L., and Sun, Z. (2019). Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118.

Lyu, P., Liao, M., Yao, C., Wu, W., and Bai, X. (2018). Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83.

Mishra, A., Alahari, K., and Jawahar, C. (2012). Scene text recognition using higher order language priors.

Neumann, L. and Matas, J. (2010). A method for text localization and recognition in real-world images. In *Asian Conference on Computer Vision*, pages 770–783. Springer.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Qiao, Z., Zhou, Y., Yang, D., Zhou, Y., and Wang, W. (2020). Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13528–13537.

Quy Phan, T., Shivakumara, P., Tian, S., and Lim Tan, C. (2013). Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 569–576.

Risnumawan, A., Shivakumara, P., Chan, C. S., and Tan, C. L. (2014). A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048.

Shi, B., Bai, X., and Yao, C. (2016a). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304.

Shi, B., Wang, X., Lyu, P., Yao, C., and Bai, X. (2016b). Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176.

Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., and Bai, X. (2018). Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*.

Su, B. and Lu, S. (2014). Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, pages 35–48. Springer.

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., and Wang, J. (2019). High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*.

Veit, A., Matera, T., Neumann, L., Matas, J., and Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.

Wang, K., Babenko, B., and Belongie, S. (2011). End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE.

Wang, K. and Belongie, S. (2010). Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer.

Wang, T., Wu, D. J., Coates, A., and Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308. IEEE.

Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., and Cai, M. (2020). Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224.

Yang, M., Guan, Y., Liao, M., He, X., Bian, K., Bai, S., Yao, C., and Bai, X. (2019). Symmetry-constrained rectification network for scene text recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9147–9156.

Yang, X., He, D., Zhou, Z., Kifer, D., and Giles, C. L. (2017). Learning to read irregular text with attention mechanisms. In *IJCAI*, volume 1, page 3.

Yao, C., Bai, X., and Liu, W. (2014). A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749.