TASuRe: Text Aware Super-Resolution

Elena Filonenko
Engineering Cybernetics Department
The National University of Science and Technology MISIS
Moscow, Russia
elena@filonenko.net

Alexander Filonenko
Research and Development
IREX
Murrieta, USA
alexander@filonenko.net

Kang-Hyun Jo
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, South Korea
acejo@ulsan.ac.kr

2022 International Workshop on Intelligent Systems (IWIS) | 979-8-3503-9893-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWIS56333.2022.9920903

***Abstract*—Recognition of text on low-resolution (LR) images is a challenging task. Traditional interpolation methods, as well as general super-resolution approaches, do not recover the shape of text character robustly. In this work, we propose a text-aware super-resolution neural network called TASuRe. Text awareness is interwoven in the proposed network that contains a text rectification part and text recognition auxiliary module. The training procedure is built around character shape restoration by adding a binary mask to the input image and using a specialized loss that penalties the network for missing gradients on the border of characters. Experiments on the real LR images have shown that the proposed network can deal with hard cases better than convolutional competitors.**

***Index Terms*—scene text recognition, super-resolution, deep learning**

I. Introduction

Text recognition (TR) remains a widely discussed topic in the research community as well as in the industry. Robust TR is required in many fields from the digitalization of old books and receipts to real-time text translation in the wild. Recognition of well-structured scanned documents with a uniform background achieves human-like performance, but TR in many cases struggles with low-resolution (LR) images. The traditional way to increase image size is to apply some kind of interpolation; however, none of them provide good image quality and results in blurred images. There is a need for a super-resolution (SR) system that increases the size of an image with text while improving recognition accuracy.

We propose a neural network TASuRe that takes advantage of the text in the images by using a text recognition branch to push the weights of the network to generate the SR more suitable for the third-party text recognizers. We also keep the inference part of the network fully convolutional to allow the use of hardware optimizations. We apply text rectification, binary mask, and shape-specific loss function to improve the performance of the proposed network on text images.

II. Related Work

Methods used for the SR task can be divided into two categories:

- Traditional methods that do not use neural networks.
- Methods based on deep learning.

Traditional methods do not achieve state-of-the-art accuracy and thus are not considered in this paper.

A. General Super-Resolution

General deep learning methods can be applied to the text image SR. They show performance better than bicubic interpolation, but still cannot provide robust results on text images. SRCNN [1] is the general SR that uses only convolutions with different kernel sizes to refine the image upscaled by bicubic interpolation. LapSRN [2] uses general convolutions and transpose convolutions to acquire features and images at higher resolutions. SRResNet [3] uses residual blocks and allows four times upscaling. RDN [4] introduces the long skip connection among groups of residual blocks and uses 1×1 convolution to fuse the features from different residual blocks.

B. Super-Resolution for Text Images

TSRN [5] is built on SRResNet with two modifications: residual blocks were replaced by sequential residual blocks to extract the image features and the authors added the text rectification module. TSRN also uses a loss function that considers the boundaries of objects.

TBSRN-5 [6] consists of three parts. In the first part, an LR image with text is rectified by a spatial transformer network (STN) [7] to tackle the misalignment problem. The image is then processed by a set of blocks with transformers and upsampled to an SR text image by pixel shuffling [8]. In the second part, the attention maps of the HR image and the SR image are processed by an absolute difference loss. The third part uses a pretrained transformer to predict the characters and provide additional loss.

III. Method

The general structure of the proposed TASuRe is shown in Fig. 1. TASuRe is divided into 3 main parts: backbone, super-resolution branch, and text recognition brach. SR and TR branches share the same backbone. While the proposed

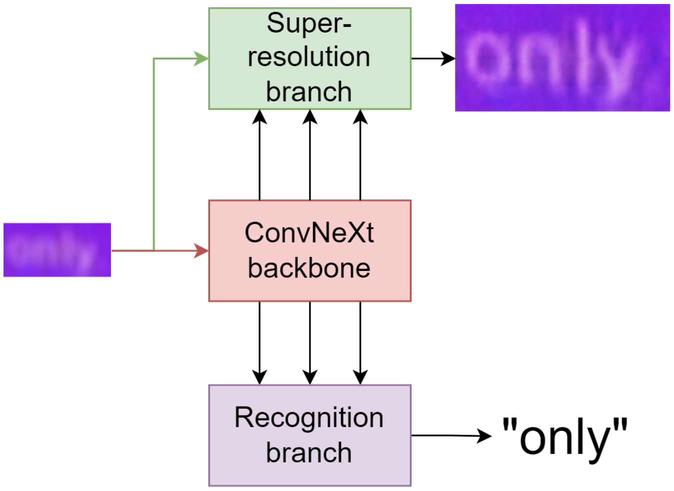


Fig. 1. General structure of the proposed architecture.

TABLE I
BACKBONE STRUCTURE

Network	Number of blocks	Number of channels
TASuRe-L	[3, 3, 27]	[192, 384, 768]
TASuRe-S	[3, 3, 27]	[96, 192, 384]
TASuRe-T	[3, 3, 9]	[96, 192, 384]

TABLE II
MODEL SIZE

Network	Number of Parameters
TASuRe-L	141.6M
TASuRe-LS	135.4M
TASuRe-S	39.9M
TASuRe-T	18.3M

network has no purpose to recognize text, the text recognition branch is designed to influence the adjustment of backbone parameters during training.

Each convolution in the recognition and SR branches is followed by GeLU activation. If there is batch normalization, then it is placed before the activation function. The final convolution of the SR branch is followed by the sigmoid function.

A. Input Image Preprocessing

In most text images, characters are represented by the same color that is different from the background color or texture. By following the approach in [5], the binary mask that roughly separates text from the background is added as a fourth channel of the input image. The mask results from the binarization of the image with a threshold computed by calculating the average grayscale of the red, green, and blue (RGB) channels.

Text recognition systems require text to be horizontal to achieve higher recognition accuracy. TASuRe achieves this by preprocessing the input LR image via a Spatial Transformer Network (STN) [7] to rectify the deformed text. The rectified text is then fed to SR and backbone branches.

B. Backbone

The ConvNeXt branch is the main component of the proposed architecture since it shares the signal between the upper and lower branches of the proposed architecture. All the blocks are adapted from [9]. A low-resolution image with a dimension of $64\times32\times4$ (width, height, RGB+binary mask channels) is fed to the input of this backbone.

1) Downsample layers: In the original ConvNeXt structure, the first downsample block named the ”patchify stem” is the 4×4 non-overlapping convolution with stride 4. This block is shown in Fig. 2 as *Conv downsample 0*. Next downsample blocks in [9] are 2×2 non-overlapping convolutions with stride 2. ConvNeXt was developed to work with large image sizes. Text super-resolution tasks are usually performed on low-resolution images. Excessive reduction of the tensor size can lead to worse performance. We have tested different ways of performing stem operation (*Conv downsample 0*): 4×4 convolution with stride 4, 2×2 convolution with stride 2, and 7×7 convolution with stride 1 and padding 3. The best results were achieved by the latter configuration. Utilization of *Conv downsample 1* and *Conv downsample 2* decreased the performance on test datasets; therefore, these operations were replaced in the final model by 1×1 convolutions with stride 1 to expand the number of channels for the next stage.

2) ConvNeXt blocks: ConvNeXt0, *ConvNeXt1*, and *ConvNeXt2* are the sets of ConvNeXt blocks with an increasing number of channels. Only the first three of four original stages of blocks are used in TASuRe. There are four different types of the proposed network that are based on the backbone variant choice. The difference among *TASuRe-L*, *TASuRe-S*, and *TASuRe-T* is the number of blocks and channels in each stage. *TASuRe-LS* is the same as *TASuRe-L* but with the recognition branch turned off during training. Table I contains details of the number of blocks and channels in each stage. The number of parameters varies in a wide range and is shown in Table II.

C. Super-Resolution Branch

The purpose of the SR branch is the reconstruction of the input image at a higher resolution. In this work, the image is upscaled two times.

The input image is processed by *Conv2D* 7×7 convolution with stride 1, padding 3, and 32 output channels. Decreasing the receptive field of this operation or replacing 7×7 convolution with a faster set of three 3×3 convolutions led to worse output image quality. After the *Conv2D*, a batch normalization is applied. To combine low and high-level features, outputs of each ConvNeXt stage are concatenated with the output of the Conv2D. In the final model, outputs of *Conv2D*, *ConvNeXt0*, *ConvNeXt1*, and *ConvNeXt2* have the same width and height. For the intermediate versions of the model with actual downsampling used in experiments, we applied the max

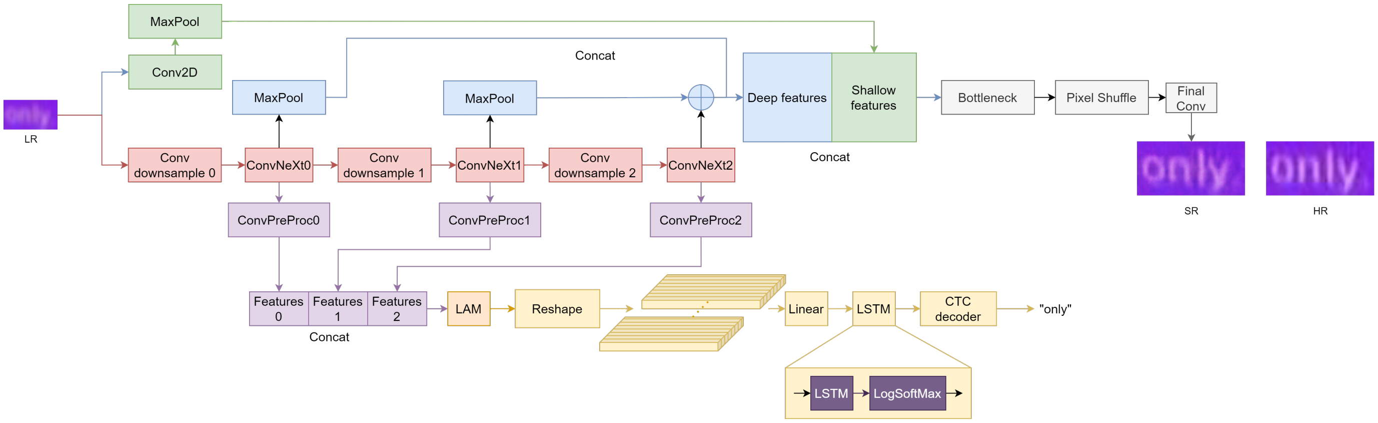


Fig. 2. Detailed structure of the proposed TASuRe architecture.

pooling operation to match the width and height of all the tensors.

Two approaches were tested as the upscaling method: a set of transposed convolutions and Pixel Shuffle [8]. The first one allowed achieving the lower loss values but increased the number of parameters and led to early overfitting of the network with lower test accuracy. Pixel Shuffle, on the other hand, does not introduce additional parameters by itself. Therefore, Pixel Shuffle is chosen for the proposed architecture. The *Bottleneck* with 1×1 convolution re-combines the channels of the concatenated tensors into $3*scale^2$ output channels. Where *scale* is the difference in image sizes between the input and output of the network. In this work, *scale* equals 2. The output of Pixel Shuffle tends to have a grid pattern that is smoothened by the final 9×9 convolution with stride 1, padding 4, and 3 output channels. In general, convolutions with a large kernel size tend to have too many parameters; however, in the case of the final convolution with three input and output channels, the number of parameters is as small as 729.

D. Recognition Branch

The hypothesis for adding the recognition branch to the architecture was that this branch should influence the weights of the backbone in a manner that one adds the awareness of the recognition accuracy to the super-resolution branch. In other words, the backbone should emphasize the features that lead to the improvement of the recognition, i. e. the shape and consistency of the characters rather than of the background. It can be seen in Fig. 4 that the quality of the characters is better than the one of the background.

The recognition branch combines the signal of *ConvNeXt0*, *ConvNeXt1*, and *ConvNeXt2* by applying to each of them the 1×1 convolution with number of channels equals to the channels number in *ConvNeXt2* divided by 3. The output is then processed by batch normalization and concatenated. In the variants of the model with width and height reduction, the transposed convolution is utilized instead of the 1×1 convolution.

The proposed network uses the LAM [10] variant of attention to better utilize the knowledge between the different levels of the features. The concatenated tensors are passed through LAM. The result of LAM is downscaled by max pooling with kernel size 2 and stride 2 and vectorized by combining the height and channel dimensions. The vectorized tensor is compressed to 256 values by a fully-connected *Linear* layer.

Nowadays, recognizers based on transformers produce the best recognition accuracy. However, they need much more data than the older approaches. Due to the lack of large datasets, the long short-term memory (LSTM) [11] was chosen for the proposed model. The input and hidden sizes of the LSTM are 256. The output size of the LSTM is the length of the vocabulary. The output of the LSTM is converted to text via Connectionist Temporal Classification (CTC) decoder [12].

The text recognition branch is only active during training. The inference is done only by backbone and super-resolution branch.

IV. Experiments

A. Dataset

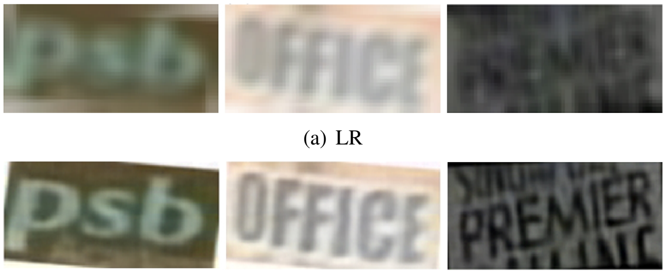
The LR images in vast majortity of text super-resolution dataset are generated by resizing the HR images. This kind of LR images does not represent the real-life situation.

When the proposed TASuRe-L is trained on the downscaled LR images of ICDAR 2013, ICDAR 2015, The Street View Text, The IIIT 5K-word, and TextZoom [5], it can achieve the image quality comparable to the ground truth HR images. The performance of this model is shown in Table III as *TASuRe-L, synthetic LR* and it is not compared to other works that use the real LR images due to the unfair nature of the training process on synthetic LR. SR methods trained on such data may actually learn how to reverse the interpolation rather than restore the image by taking into consideration noise, compression artifacts, lens distortion, etc. The examples of such image restoration ability can be seen in Fig. 3. TextZoom solves this interpolation bias by ensuring that LR and HR images were taken by physical cameras with lenses of different focal lengths. Therefore, TextZoom was used in this work to train and test the proposed method and compare the performance to competitors.

TABLE III
RECOGNITION ACCURACY AND IMAGE QUALITY COMPARISON

Network	**CRNN**			**MORAN**			**SSIM**			**PSNR**		
	Easy	***Medium***	***Hard***	***Easy***	***Medium***	***Hard***	***Easy***	***Medium***	***Hard***	***Easy***	***Medium***	***Hard***
HR	76.0	74.7	64.63	87.3	81.6	70.4						
Bicubic	36.4	21.1	21.1	60.6	37.9	30.8	0.7884	0.6254	0.6592	22.35	18.98	19.39
SRCNN	38.7	21.6	20.9	63.2	39.0	30.2	0.8379	0.6323	0.6791	23.48	19.06	19.34
LapSRN	37.5	21.8	20.9	64.6	44.9	32.2	0.8556	0.6480	0.7087	24.58	18.85	19.77
SRResNet	37.4	21.6	21.2	60.7	42.9	32.6	0.8681	0.6406	0.6911	23.48	19.06	19.34
RDN	41.6	24.4	23.5	61.7	42	31.6	0.8249	0.6427	0.7113	22.27	18.95	19.70
TSRN	37.8	22	21.0	**70.1**	**53.3**	37.9	**0.8897**	**0.6676**	0.7302	**25.07**	18.86	19.71
TBSRN-5	**54.2**	40.6	32.7	-	-	-	0.8660	0.6533	**0.7490**	23.82	19.17	19.68
TASuRe-L	54.1	**44.2**	**35.6**	65.7	52.9	**40.6**	0.8408	0.6556	0.7196	23.74	19.43	20.08
TASuRe-LS	41.3	31.5	29.3	52.4	40.7	33.3	0.7989	0.6395	0.6952	22.45	19.36	19.89
TASuRe-S	41.8	35.3	29.7	54.7	46.8	36.2	0.8059	0.6535	0.7071	22.66	19.37	19.98
TASuRe-T	41.2	29.0	32.2	55.0	46.2	35.7	0.7912	0.6481	0.7001	22.48	**19.63**	**20.15**
TASuRe-L, synthetic LR	77.3	74.7	64.3	87.7	81.7	70.4	0.9788	0.9780	0.9764	33.78	35.09	33.50

The TextZoom dataset is originally divided into easy, medium, and hard subsets. The training is performed using all the subsets simultaneously.


(a) LR

(b) SR

Fig. 3. Example of the restoration ability of the proposed model TaSuRe-L when trained on the downscaled LR images with additional rotation and Gaussian blur. LR are low-resolution images. SR are the super-resolution images.

B. Training process

All the HR images were resized to 64×128 pixels and LR images were resized to to 32×64 pixels.

The batch size differs for each variant of TASuRe and was chosen to fit a Nvidia RTX 3090.

The optimization was performed by the amsgrad version of Adam with 0.0005 learning rate, β_1 = 0.5, β_2 = 0.999, with no weight decay.

Augmentations for both LR and HR images were the random channel shuffle and image inversion (subtraction pixel values from 255).

1) Loss functions: The recognition branch of the model was trained using the CTC loss [12] L_c.

A loss for the super-resolution part consists of two parts. The first one is the mean squared error (MSE) L_{mse}. The second one is the gradient profile loss [5] L_{gp} that aims to emphasize the correctness of the boundary of objects.

The combined loss is the combination:

$$L = L_c * \lambda_c + L_{mse} * \lambda_{mse} + L_{gp} * \lambda_{gp}, \tag{1}$$

where $\lambda_c = 0.004$, $\lambda_{mse} = 1$, and $\lambda_{gp} = 1$.

2) Benchmark: Two text recognition models are used as the benchmark. SR images are fed to each of the benchmark models as input.

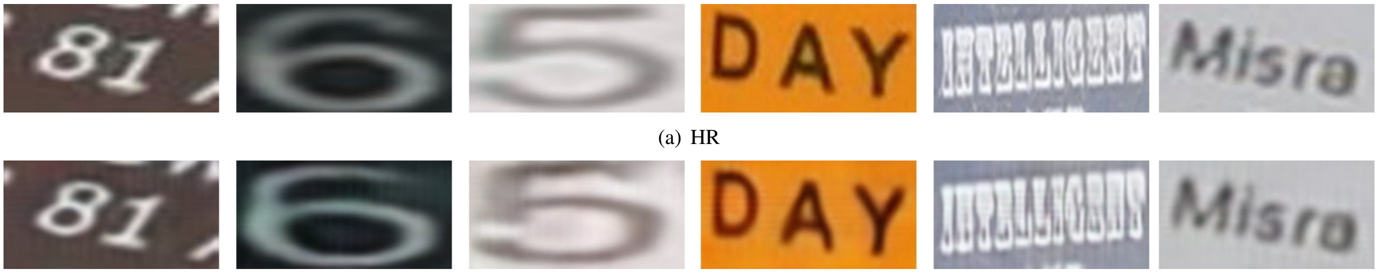
The first one is CRNN [13] that is the traditional network that computes features by its convolutional layers, passes them to a bidirectional LSTM, and gets the final text by CTC decoder. TASuRe uses LSTM that should prepare the SR image to be recognizable better by the network that also utilizes the same module.

The second recognition model is MORAN [14] that is more recent and advanced. It rectifies the images to improve the accuracy on distorted images. MORAN utilizes attention to further improve its performance. As encoder and decoder it uses LSTM and gated recurrent unit (GRU) [15].

3) Performance: The proposed method was compared to other works which use the same training and test data. Comparison results are summarized in Table III. Bold values are the best results. Underscore values and the second best results. The HR row is the performance on CRNN and MORAN on the ground truth HR images.

SSIM is the structural similarity index measure and *PSNR* is the peak signal-to-noise ratio applied to the HR and SR images.

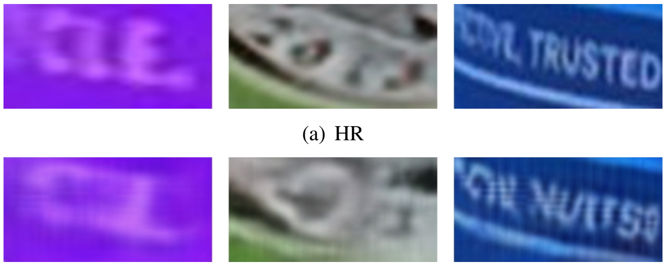
In terms of recognition accuracy, TASuRe-L achieves high values at medium and hard subsets for CRNN and hard subset for MORAN. For the rest of the subsets, it is almost on par with the competition. TASuRe-L does not outperform other works in SSIM and PSNR. These metrics measure the image quality for the whole image. It can be seen in Fig. 4 that background is not always smooth, but the shape of the characters is consistent. Therefore, SSIM and PSNR cannot be the only metrics to choose the best model during an early-stopping procedure. It is worth noting that TASuRe-LS achieves accuracy similar to general-purpose SR methods. A comparison of TASuRe-L and TASuRe-LS justifies the use of the recognition branch during training. Surprisingly, TASuRe-LS could not achieve the same SSIM and PSNR as TASuRe-L. Smaller variants TASuRe-S and TASuRe-T failed to compete with lightweight TSRN with MORAN, but they outperform it



(a) HR

(b) SR

Fig. 4. Examples of correct image reconstruction. HR are the ground truth images. SR are the output tensors of the proposed network.



(a) HR

(b) SR

Fig. 5. Examples of wrong image reconstruction. HR are the ground truth images. SR are the output tensors of the proposed network.

with CRNN. TBSRN-5 was tested by authors on CRNN only, and it works better on easy images than any of the TASuRe variants.

TASuRe-L works well in hard conditions when photos of text are taken in the wild. When the conditions are easy, the proposed model should be trained differently, like the *TASuRe-L, synthetic LR*.

The proposed model does not always work correctly with rotated and curved text as shown in Fig. 5 despite having the STN as image preprocessing.

V. Conclusion

In this paper, a model was proposed for the super-resolution of images with text, which shows the high quality of work in difficult conditions. The use of a binary mask and the gradient profile loss function increased the network's attention to the form of the text, rather than the content of the background. The use of the recognition branch has improved the suitability of the output image for use with third-party recognizers.

In future work, the recognition branch will be replaced by moden transformers to prepate the SR images for the next generation recognizers. Additional work will be done to decrease the complexity of the models to make them suitable for industrial application.

References

[1] C. Dong, C. C. Loy, K. He, and X. Tang, ”Learning a deep convolutional network for image super-resolution,” Computer Vision – ECCV. Lecture Notes in Computer Science, pp. 184–199, 2014.

[2] 27 W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, ”Deep Laplacian pyramid networks for fast and accurate super-resolution,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5835–5843, 2017.

[3] C. Ledig, L. Theis, F. Huszar et al. ”Photo-realistic single image super-resolution using a generative adversarial network,” IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690, 2017.

[4] Y. Zhang,Y. Tian, Y. Kong, B. Zhong, Y. Fu, ”Residual dense network for image super-resolution,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2472–2481, 2018.

[5] W. Wang, E. Xie, X. Liu, W. Wang, D. Liang, C. Shen, and X. Bai, ”Scene text image super-resolution in the wild,” Computer Vision – ECCV 2020, pp. 650–666, 2020

[6] J. Chen, B. Li and X. Xue, ”Scene Text Telescope: text-focused scene image super-resolution,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12021–12030, 2021.

[7] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, ”Spatial transformer networks,” NeurIPS, pp. 2017–2025, 2015

[8] W. Shi et al., ”Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874-188, 2016.

[9] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, ”A ConvNet for the 2020s,” CVPR 2022, in press.

[10] B. Niu et al., ”Single image super-resolution via a holistic attention network,” Computer Vision – ECCV 2020, pp. 191–207, 2020

[11] H. Sak, A. Senior, F. Beaufays, ”Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” CoRR, 2014

[12] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, ”Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” Proceedings of the 23rd international conference on Machine learning (ICML ’06), pp 369–376, 2006.

[13] B. Shi, X. Bai, and C. 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, pp. 2298–2304, 2017.

[14] C. Luo, L. Jin, and Z. Sun,”MORAN: a multi-object rectified attention network for scene text recognition,” Pattern Recognition, pp. 109–118, 2019.

[15] K. Cho et al., Learning phrase representations using RNN encoder–decoder for statistical machine translation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, 2014