International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 3
ISSN: 2088-5334

Intelligent Deep Learning Empowered Text Detection Model from Natural Scene Images

S. Kiruthika Devi [a,*], Subalalitha C.N [a]

[a] *Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, India*
*Corresponding author: *kiruthis2@srmist.edu.in*

***Abstract*— The scene Text Recognition process has become a hot research topic and a challenging task owing to the complicated background, varying light intensities, colors, font styles, and sizes. Text extraction from natural scene images encompasses two main processes: text detection and text recognition. The latest advancements in Machine Learning (ML) and Deep Learning (DL) concepts can effectually automate the text detection and recognition process by training the model properly. In this view, this paper presents an Automated DL empowered Text Detection model from Natural Scene Images (ADLTD-NSI). The ADLTD-NSI technique includes two important processes: text detection and text recognition. Firstly, a single shot detector (SSD) with Inception-v2 as a baseline model is employed for text detection, an object detector based on the VGG-16 framework for feature map extraction followed by six convolution layers. Secondly, Convolutional Recurrent Neural Network (CRNN) technique is utilized for the text recognition process. Besides, the recurrent layers in the CRNN model utilize long short-term memory (LSTM) for encoding the sequence of feature vectors. Lastly, Connectionist Temporal Classification (CTC) loss is applied to predict text labels equivalent to the sequences from the recurrent layers. A wide range of experiments was carried out on benchmark COCO datasets, and the results are examined in several aspects. The experimental outcomes showcased the better performance of the ADLTD-NSI technique over the other compared methods with a maximum accuracy of 96.78%.**

***Keywords*— Deep learning; natural scene images; text detection; text recognition; COCO dataset; CRNN model; CTC loss.**

Manuscript received 14 Jul. 2021; revised 17 Dec. 2021; accepted 17 Jan. 2022. Date of publication 30 Jun. 2022.

I. Introduction

Nowadays, Scene text recognition has become advanced research since it has abundant semantic data [1]. Text reading technique is embedded in several vision tasks like automatic license-plate recognition, autonomous driving, scene understanding, robot navigation, and assistant technologies for the blind. The computer Vision methodology is used for obtaining required data from an image or the sequences of images by inspecting the image. The extracted essential data can be utilized for many purposes[2]. Text extraction from natural scene images consists of 2 main functionalities, i.e., Text recognition and Text detection [3]. The study as a primary section emphasizes text recognition from natural scene images that is complex to execute in real-time situations because of the variances in color, font style, and the illumination of darkness and light, as shown in Fig.1.

In various scene images, the texts are embedded with many distorted objects. Hence the fundamental problem exists in extracting this text from the image's background [4]. Text extraction includes image Objects Detection, pre-processing, Text Classification, Text Reconstruction, and Text Object Detection. This proposed work focuses on 3 components that finds every object embedding in the image, identifies the text object alone leave after the non-text one and lastly the text classification. In Scene text detection, the primary importance must be provided for the text recognition and with fine-tuned results. Several techniques are available for text recognition stage for improving the entire quality of results [5]. Fig. 2 shows the steps involved in text detection and recognition.

Fig. 1 Examples of variance of scene text

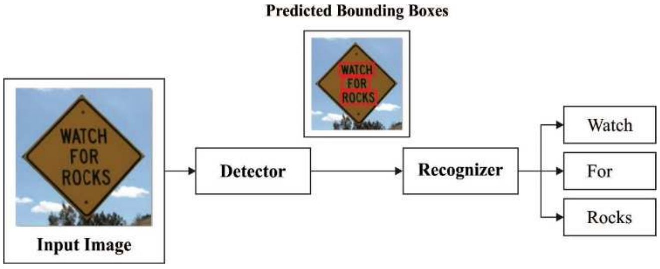

Fig. 1 Steps in Text Detection and Recognition

Researchers frequently deliberate text reading as sequence-to-sequence tasks. Most state-of-the-art text reading models useNN and RNN with an attention decoder or a connectionist temporal classification transcription [6]. Reading text from the natural scene is complex; due to the complex backgrounds, variation in text size, distortion, perspectivity, and so on, a simple sequence to sequence detector might have problems with the irregular text. So, a better model is needed to handle various forms of text in complex backgrounds with good speed in a real-time scenario. This task can be achievable with the rapid development of computer vision with deep learning algorithms. The proposed method is based on a single-stage algorithm called SSD with Inception-v2 for text detection and text recognition CRNN with LSTM model. For text detection, the SSD model is trained with COCOText.v2 dataset and for text recognition, the CRNN model is trained on the MJsnth word dataset.

Zhu et al. [7] employed federated learning with a DCN for performing variable-length text string detection using a huge amount. In this work, they have developed a federated learning architecture using PySyft and Tensorflow. The result shows that federated text detection modules could attain equivalent or greater accuracy than modules trained on DL architecture. The optimal character accuracy is attained using TFF at 49.20% on a five-client distributed dataset. Zhang et al. [8] proposed an automatic STR (AutoSTR) that could tackle the problems above by seeking data-based backbones. Especially, they display both selections on the operation, and the downsampling paths are highly significant in the search space design of NAS. Also, no present NAS methods could deal with the spatial limitation on the path. They proposed 2 phase search algorithms that decouple operation and downsampling paths to search the provided spaces effectively.

Zhang et al. [9] proposed a novel SaHAN for recognizing scene texts. Simulated by feature pyramid networks, they exploit the inherent pyramidal framework of a DCN for retaining multiscale features for flexible receptive areas. Later, they constructed a hierarchical attention decoder that executes the attention method twice/multiscale features to collect the fine-grained data for prediction. Sang and Cuong [10] proposed a new CRNN for recognizing text. In particular, they adopted an Efficient Net structure to extract deep features and proposed a multi-head attention method for improving character localization. STAN includes sequential transformation networks, and attention-based detection networks are projected for recognizing a common scene text [11]. The sequential transformation network corrects irregular text by decomposing the task to a sequence of patch-wise basic transformations, succeeded by a grid projection module for smoothing the junctions among adjacent patches.

Mirza et al. [12] presented a complete architecture for detecting and recognizing textual content in video frames. Particularly, they aim cursive script takes Urdu text as an analysis. Scripts of the recognized textual content are detected by a CNN, whereas for detection, they proposed an UrduNet, an integration of CNN and LSTM networks. Aberdam et al. [13] proposed architecture for SeqCLR of visual depictions that they employ for recognizing text. To demonstrate the sequence-to-sequence framework, every feature map is separated into different cases where the contrastive loss is calculated. This process allows us to compare in a sub-word level wherefrom all the images. They extracted many multiple negative and positive pair's instances. To produce efficient visual depictions for recognizing text, they further suggested new custom projection heads, augmentation heuristics, and distinct encoder frameworks.

Harizi et al. [14] presented a novel CNN based system to read scene text. They investigated combining the character detection model succeeded by the word detection model for achieving the entire system goals. The initial model analyses characters within multiscale images based on the power of the convolution network and the FCN for recognizing characters. Luo et al. [15] have proposed a multi-object rectified attention network for handling irregular text. Xinyu Zhou et al. [16] have proposed a pipeline model called EAST that can handlarbitrary oriented and different shaped text lines. This model was trained on MSRA-TD500, COCO-Text, and ICDAR 2015 datasets.

There are multiple current techniques available for the scene text reading; even then, the art's state-of-models are observed to have several shortcomings. The challenges exist in text recognition in real-time due to the various environmental disturbance along with irregular text patterns in terms of size, language, different font, color, distortion of images, photographic angle etc.,

This paper presents an automated DL empowered text detection model from natural scene images (ADLTD-NSI). The ADLTD-NSI technique includes two important processes: text detection and text recognition. At the initial stage, a single shot detector (SSD) with Inception-v2 as a baseline model is employed for text detection. Then, Convolutional Recurrent Neural Network (CRNN) method is utilized for the text recognition process. In addition, the recurrent layers in the CRNN model exploit long short-term memory (LSTM) for encoding the sequence of feature vectors. Lastly, Connectionist Temporal Classification (CTC) loss is employed to forecast the text labels equivalent to the sequences from the recurrent layers. A set of simulations is carried out to validate the enhanced detection and recognition performance, and the results are examined in terms of different measures.

The rest of the paper is structured as follows: Section 2 describes the implementation details of the proposed ADLTD-NSI technique. The performance evaluation of the proposed model on the Mjsnth dataset and its performance comparison with other deep learning models for both text detection and text recognition are described in section 3. Finally, the conclusion of this simulative experimental analysis and future direction of enhancement are discussed in Section 4.

II. Materials and Methods

The working process involved in the ADLTD-NSI technique is demonstrated in Fig. 3. The proposed ADLTD-NSI technique operates on two major processes: SSD with Inception v2 based text detection and CRNN based text recognition. The detailed working of these processes is described in the succeeding sections.

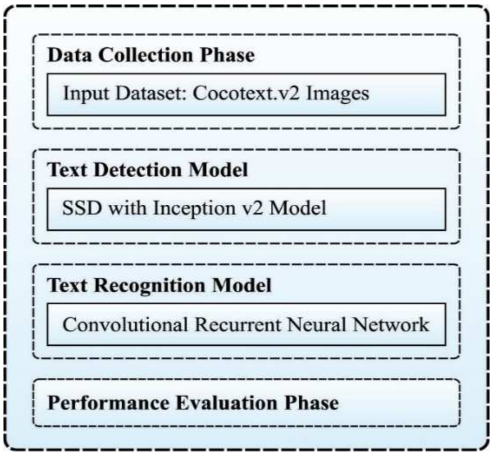
Data Collection Phase
Input Dataset: Cocotext.v2 Images
Text Detection Model
SSD with Inception v2 Model
Text Recognition Model
Convolutional Recurrent Neural Network
Performance Evaluation Phase

Fig. 3 Process in Proposed Model

A. Phase I: Text Detection Process

In the initial phase, the textual data in the natural scene image is detected by using SSD with Inception v2 model as shown in the Fig.4. The SSD 96. In the proposed ADLTD-NSI model, for extracting the feature map, the inception-V2 model is used base network instead of VGG -16 framework, with further convolutional layers that generate different feature maps of distinct for detecting the word-text embedded in the natural scenes. Thus, SSD techniques enable extra aspect ratios to generate default bounding boxes.

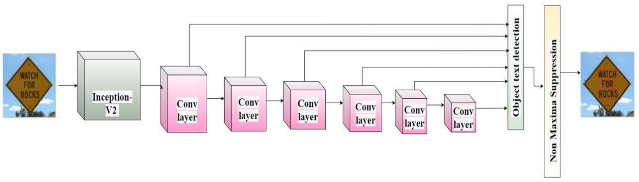

Fig. 4 SDD model for text Detection

The reason for using the InceptionV2 module instead of the VGG model was to enable the proposed ADLTD-NSI to reach a faster detection and high accuracy since it has a deeper structure than VGG modules. It also utilizes fewer variables than VGG modules because of the inception models, which are made up of multiple connected convolutional layers. In the Inception V2 framework, the 5×5 convolution is substituted with the two 3×3 convolutions. Also, it reduces computation time and hence increases computation speed as a 5×5 convolution is 2.78 costly compared to 3×3 convolutions. Thus, utilizing two 3×3 layers rather than 5×5 surges the framework efficiency. Also, this structure transforms nXn factorization to 1xn and nx1 factorization. As discussed, a 3×3 convolution could be transformed to 1×3 succeeded by 3×1 convolution that is 33% less expensive based on computation complexity than 3×3. For handling the problems of representational bottleneck, the feature banks of the model have been extended rather than making it deeper. This will avoid the loss of data during training.

After extracting the feature map with Inception-V2 and producing bounding boxes that are named default bounding box. Then by using Non-Maximum Suppression single bounding box per text is done. Afterward, it upgrades the whole variables based on the computed loss value with the Back Propagation method. Thus, attempting for learning an optimal filter structure which could find the features of text and generalization the training instance for reducing the loss values, hence achieving a higher accuracy at the evaluation stage [17].

The loss value is computed as combination of confidence of the forecasted bound box and the accuracy of position. An entire loss value (localization loss + confidence loss) can be calculated using equation (1).

$$L(x,c,l,g)=\frac{1}{N}\left(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)\right) \quad (1)$$

where N relates to the amount of corresponding default box. When there is no matching $(N=0)$, the entire loss is defined as 0 directly. The $\alpha =$ value is balanced of 2 kinds of losses, and it can be equivalent to one in the cross-validation stage [18]. When the center position of the boxes represents as cx, cy, the default box d, width w, and height as h ,the localization loss is calculated using equation (2).

$$L_{loc}(x,l,g)$$

$$=\sum_{i\in Pos_{m\in\{cx,cy,w,h\}}}^{N}\sum x_{ij}^{k}\, smooth_{L1}\left(l_i^m-\hat{g}_j^m\right) \quad (2)$$

in which:

$$smooth_{L1}(x)=\begin{cases}0.5x^2 & if\ |x|<1\\ |x|-0.5 & otherwise,\end{cases}$$

$$\hat{g}_j^{cx}=\left(g_j^{cx}-d_i^{cx}\right)/d_i^w\ \hat{g}_j^{cy}=\left(g_i^{cy}-d_i^{cy}\right)/d_i^h$$

$$\hat{g}_j^w=\log\frac{g_i^w}{d_i^w}\ \hat{g}_j^h=\log\frac{g_i^h}{d_i^h}.$$

Also, the confidence loss (c) is computed as two class SoftMax loss using equation (3)

$$L_{conf}(x,c)=-\sum_{i\in Pos}^{N}x_{ij}^{p}\, log\hat{c}_i^p-\sum_{i\in Neg}^{N}\log\left(\hat{c}_i^0\right) \quad (3)$$

in which:

$$\hat{c}_i^p=\frac{\exp\left(c_i^p\right)}{\sum_p\exp\left(c_i^p\right)}.$$

Finally, the image with detected text is fed into the next phase for text recognition.

B. Phase II: Text Recognition Process

Once the text is detected, the next phase is to recognize the text using the CRNN model. The text extraction model will initially extract the text areas respective to the coordinate points of the text bounding boxes created by the object text detection process. The text regions are scaled to a predefined aspect ratio, and the scaled text regions are fed as input to the convolution layer of CRNN model. The CRNN model involves the convolution, recurrent, and transcription layers, as shown in Fig.5.

The convolution layer is used to extract the feature map from the processed textual region of the static feature. But as far as text recognition is considered, the contextual feature is very important because it aids the learning from the previous stage. So, for the RNN layer, the static feature obtained from the convolution layer is fed as an input to produce the context feature as an output. But, this RNN model has the disappearing gradient due to the backpropagation process resulting in lesser storage of contextual features. In order to remember the previous state, this RNN layers model uses the bi-directional LSTM cell to store the contextual feature. Each LSTM cell possesses three gates: input, output, and forget gates. The input and output gates store the previous context for a long time. LSTM cells follow a comfortable strategy with the use of "forget" gates to select what to forget particularly. The state of LSTM memory unit uses the following equations:

$$i_t = \sigma \left(W_{xi} + W_{hi}h_{t-1} + \mathrm{W}_{ci}c_{t-1} + b_i\right)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + \mathrm{W}_{cf}c_{t-1} + b_f)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$

$$o_t = \Gamma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \otimes tanh(c_t)$$

The subscripts resemble the initials of what every matrix denotes (i.e., W_{hf} is the hidden forget weight matrix). Moreover, f, i, o, and c signify the forget, input, output, and cell gate vectors [19]. Therefore, the LSTM technique is mostly related to the RNN method but with LSTM cells' inclusion. Lastly, in the transcription layer, the Connectionist Temporal Classification (CTC) is used to predict text levels respective to the sequences from the recurrent layer.

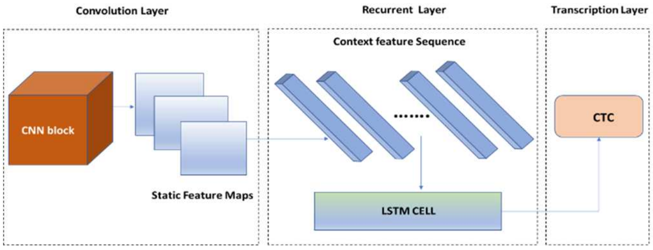

Fig. 5 CRNN model for text recognition

III. Results and Discussion

The presented method is simulated using Python 3.6.5 on the benchmark COCOText.v2 dataset from https://bgshih.github.io/cocotext/#h2-explorer contains 63,686 images and 2,39,506 annotated text for scene text detection. MJsynth word dataset is used for training the text recognition model the CRNN. The MJsynth word dataset consists of nine million images covering 90,000 English words. Few sample images from the COCOText.v2 dataset are demonstrated in Fig. 6. Besides, a sample visualization detection results analysis of the presented method is provided in Fig. 7. The figures showcased that the proposed technique has effectually detected the text in all the applied images.

Fig. 6 Sample Images

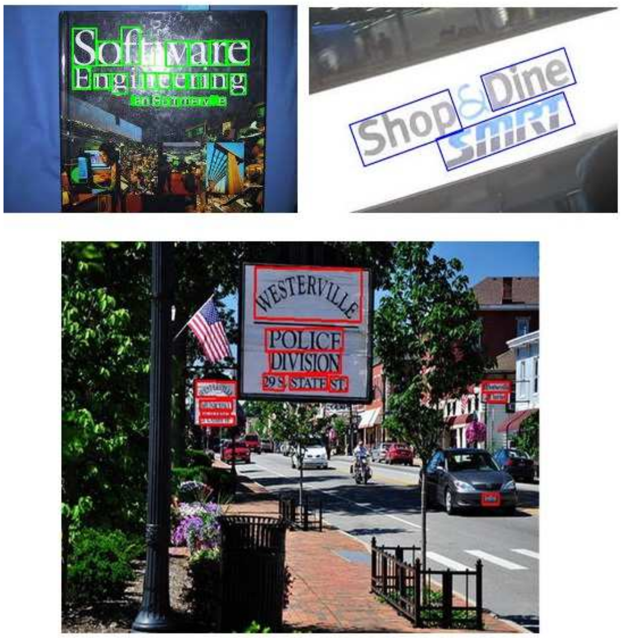

Fig. 7 Sample Text Detection on Test Images

Fig. 8 investigates the results analysis of the ADLTD-NSI technique regarding time, classification loss, and localization loss. The experimental results demonstrated that the ADLTD-NSI technique improved performance with minimal classification loss and localization loss.

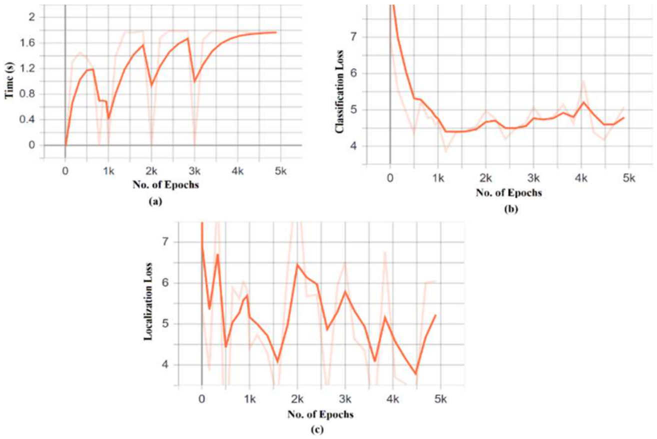

Fig. 8 Results analysis of ADLTD-NSI Technique. a) Time of Each Epochs b) Classification Loss c) Localization Loss

Another sample visualization recognition results analysis of the presented method is given in Fig. 9. The figures portrayed that the proposed method has successfully recognized all the text in the applied image.

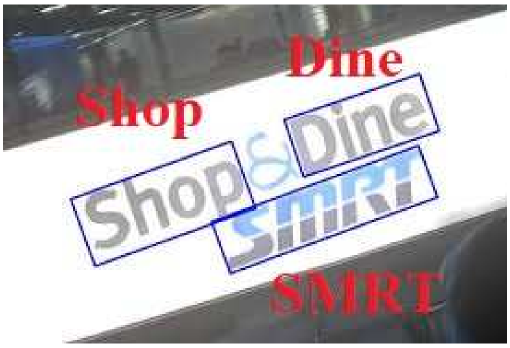

Fig. 9 Sample Text Recognition

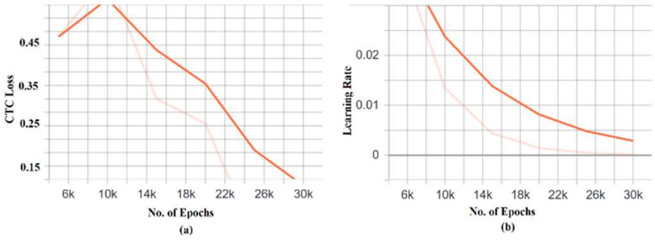

Fig. 10 a) CTC Loss b) Learning Rate

Fig. 10 demonstrates the CTC loss and learning rate analysis of the ADLTD-NSI technique. The figure displayed that the CTC loss and learning rate gets reduced with a rise in epoch count. To ensure the enhanced performance of the ADLTD-NSI technique, a brief comparison study is made interms of accuracy in Table 1 and Fig. 11 [20],[21]. From the obtained results, it is apparent that the Text Detection Using Least Square Support Vector Machine (TEDLESS) technique have attained an accuracy of 84.40%, because it works only for domain-specific data. Synthetically Supervised Feature Learning (SSFL) have attained a moderate accuracy of 87% due to the fact that it works good for distorted images only by increasing the learning feature parameter. Consequently, the Single Neural Network (STN-OCR) technique has gained moderate performance with an accuracy of 90.30%. At the same time, the Automatic Scene Text Recognition (AutoSTR) techniques have depicted near-optimal outcomes with a closer accuracy of 94.70% as it includes the rectification unit before the text recognition. The Scale-aware Hierarchal Attention Network (SaHAN) has achieved an accuracy of 95.20% because it concentrates on character-level scale variance and environmental distortion. However, the ADLTD-NSI technique has outperformed all the other techniques with a maximum accuracy of 96.78% because the SDD with Inception-v2 model has been used, which aids the detection of word text of any size with rid of vanishing gradient problem irrespective the layer depth. In addition to that, the text recognition by CRNN model offers character-level text recognition by maintaining the previous context of the text. Therefore, the proposed ADLTD-NSI technique can be an appropriate text detection and recognition tool for natural scene images.

TABLE I
RESULT ANALYSIS OF EXISTING WITH PROPOSED MODEL WITH RESPECT TO RECOGNITION ACCURACY (%)

Methods	**Accuracy (%)**
Proposed ADLTD-NSI	96.78
SSFL	87.00
AutoSTR	94.70
STN-OCR	90.30
SaHAN	95.20
TEDLESS	84.40

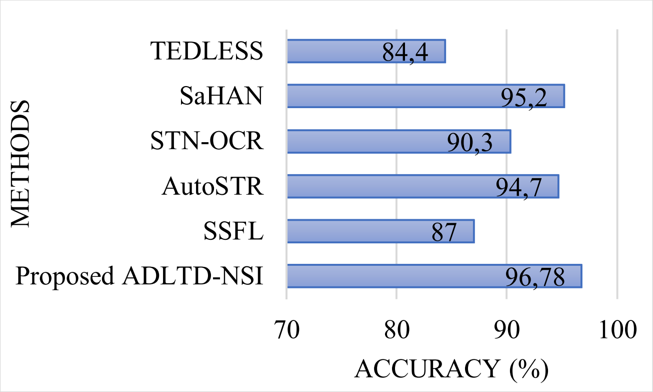

Fig. 11 Accuracy analysis of ADLTD-NSI model with existing techniques

IV. Conclusion

This paper has developed a novel ADLTD-NSI technique to detect and recognize textual data from natural scene images automatically. The proposed ADLTD-NSI technique operates on two major processes: SSD with Inception v2 based text detection and CRNN based text recognition. Besides, the use of Inception v2 as the baseline model instead of VGG model in the SSD architecture also assists in boosting the text detection performance in terms of speed and accuracy. In addition, the use of LSTM model in the recurrent layers of the CRNN model helps to improve the recognition outcome. In order to validate the enhanced detection and recognition performance, a group of simulations is carried out, and the results are examined in terms of different measures. The experimental results showcased the efficiency of the ADLTD-NSI technique compared to that of the other recent techniques. As a part of the future scope, the text detection and recognition performance can be increased by using hyperparameter optimizers.

References

[1] M. Ghosh, S. Chatterjee, H. Mukherjee, S. Sen, and S. M. Obaidullah, "Text/Non-text Scene Image Classification Using Deep Ensemble Network," in *Proceedings of International Conference on Advanced Computing Applications*, 2022, pp. 561–570.

[2] L. M. Francis and N. Sreenath, "TEDLESS – Text detection using least-square SVM from natural scene," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 3, pp. 287–299, 2020, doi: 10.1016/j.jksuci.2017.09.001.

[3] J. Diaz-Escobar and V. Kober, "Natural Scene Text Detection and Segmentation Using Phase-Based Regions and Character Retrieval," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/7067251.

[4] X. Zhang, X. Gao, and C. Tian, "Text detection in natural scene images based on color prior guided MSER," *Neurocomputing*, vol. 307, pp. 61–71, 2018, doi: 10.1016/j.neucom.2018.03.070.

[5] S. Y. Arafat and M. J. Iqbal, "Urdu-Text Detection and Recognition in Natural Scene Images Using Deep Learning," *IEEE Access*, vol. 8, no. June, pp. 96787–96803, 2020, doi: 10.1109/ACCESS.2020.2994214.

[6] M. Liao et al., "Scene text recognition from two-dimensional perspective," *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 8714–8721, 2019, doi: 10.1609/aaai.v33i01.33018714.

[7] X. Zhu, J. Wang, Z. Hong, T. Xia, and J. Xiao, "Federated learning of unsegmented chinese text recognition model," *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, vol. 2019-November, no. 2018, pp. 1341–1345, 2019, doi: 10.1109/ICTAI.2019.00186.

[8] H. Zhang, Q. Yao, M. Yang, Y. Xu, and X. Bai, "AutoSTR: Efficient Backbone Search for Scene Text Recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12369 LNCS, pp. 751–767, 2020, doi: 10.1007/978-3-030-58586-0_44.

[9] J. Zhang, C. Luo, L. Jin, T. Wang, Z. Li, and W. Zhou, "SaHAN: Scale-aware hierarchical attention network for scene text recognition," *Pattern Recognition Letters*, vol. 136, pp. 205–211, 2020, doi: 10.1016/j.patrec.2020.06.009.

[10] D. V. Sang and L. T. B. Cuong, "Improving CRNN with EfficientNet-like feature extractor and multi-head atention for text recognition," *ACM International Conference Proceeding Series,* no. December, pp. 285–290, 2019, doi: 10.1145/3368926.3369689.

[11] Q. Lin, C. Luo, L. Jin, and S. Lai, "STAN: A sequential transformation attention-based network for scene text recognition," *Pattern Recognition*, vol. 111, p. 107692, 2021, doi: 10.1016/j.patcog.2020.107692.

[12] A. Mirza, O. Zeshan, M. Atif, and I. Siddiqi, "Detection and recognition of cursive text from video frames," *Eurasip Journal on Image and Video Processing*, vol. 2020, no. 1, 2020, doi: 10.1186/s13640-020-00523-5.

[13] A. Aberdam *et al*., "Sequence-to-Sequence Contrastive Learning for Text Recognition," 2020.

[14] R. Harizi, R. Walha, F. Drira, and M. Zaied, "Convolutional neural network with joint stepwise character/word modeling based system for scene text recognition," *Multimedia Tools and Applications*, 2021, doi: https://doi.org/10.1007/s11042-021-10663-z.

[15] C. Luo, L. Jin, and Z. Sun, "MORAN: A Multi-Object Rectified Attention Network for scene text recognition," *Pattern Recognition*, vol. 90, pp. 109–118, 2019, doi: 10.1016/j.patcog.2019.01.020.

[16] X. Zhou et al., "EAST: An efficient and accurate scene text detector," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2642–2651, 2017, doi: 10.1109/CVPR.2017.283.

[17] U. Alganci, M. Soydas, and E. Sertel, "Comparative research on deep learning approaches for airplane detection from very high-resolution satellite images," *Remote Sensing*, vol. 12, no. 3, 2020, doi: 10.3390/rs12030458.

[18] R. Suresh and N. Keshava, "A Survey of Popular Image and Text analysis Techniques," *CSITSS 2019 - 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution, Proceedings*, 2019, doi: 10.1109/CSITSS47250.2019.9031023.

[19] F. Zhang, J. Luan, Z. Xu, and W. Chen, "DetReco: Object-Text Detection and Recognition Based on Deep Neural Network," *Mathematical Problems in Engineering*, vol. 2020, 2020, doi: 10.1155/2020/2365076.

[20] Y. Liu, Z. Wang, H. Jin, and I. Wassell, "Synthetically Supervised Feature Learning for Scene Text Recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11209 LNCS, pp. 449–465, 2018, doi: 10.1007/978-3-030-01228-1_27.

[21] L. Chen and S. Li, "Improvement research and application of text recognition algorithm based on CRNn," *ACM International Conference Proceeding Series*, pp. 166–170, 2018, doi: 10.1145/3297067.3297073.