Deep Learning-Based Framework For Text Detection and Recognition In Natural Images

Djouher Akrour[1], Mohamed Akram Khelili[2], Imene Aloui[2] and Azeddine Aissaoui[3]

[1]LESIA laboratory / Department of Computer Science, Biskra University, PB 145 RP, 07000 Biskra, Algeria.
[2]LINFI laboratory / Department of Computer Science, Biskra University, City communal 197 Biskra, Algeria
[3]Centre de Recherche Scientifiques et Techniques sur les Régions Arides, Campus Universitaire, Université Mohamed Khider, Biskra, Algeria

Abstract

Detecting and recognizing text in natural images is a critical task for extracting meaningful information, yet it remains highly challenging due to the variability and complexity of unstructured text in real-world scenarios. Traditional image processing techniques often rely on handcrafted features, which struggle to adapt to the diverse and unpredictable nature of text in the wild. To address these limitations, this paper leverages advancements in deep learning to develop a robust framework capable of adaptive feature learning, text extraction, and digitization. The proposed method utilizes YOLOv5 for precise localization of text-rich regions, followed by an LSTM-based module to segment text into individual characters. These characters are subsequently processed by a Capsule Network-based recognition module, ensuring accurate text recognition. A semantic post-processing step is incorporated to further enhance the system's overall performance. Experimental evaluations conducted on popular benchmark datasets demonstrate that the proposed framework significantly outperforms existing state-of-the-art methods, achieving superior accuracy and efficiency in both text detection and recognition tasks.

Keywords

Capsule Network, Yolov5, LSTM, Text detection, Text recognition, Semantic recognition

1. Introduction

The detection and recognition of unstructured text in natural scene images have garnered significant attention in the computer vision community, driven by their broad applicability across diverse real-world scenarios, including robotics [1, 2]and cognitive systems [3, 4]. Unlike traditional printed documents, the extraction of text from unstructured, real-world scenes presents unique challenges due to the high variability of text patterns, diverse fonts, and complex background environments. These challenges have spurred extensive research into scene text detection and recognition over the past decade, leading to the development of numerous methods and frameworks to address this problem [5, 6]. Approaches can be generally categorized into conventional machine learning techniques [7, 8, 9, 10] and advanced deep learning-based methods [11, 12, 13]. Traditional machine learning methods for scene text detection typically rely on handcrafted features combined with classifiers, which, although successful in certain controlled environments, struggle with the large variability and complexity of natural scene text. These methods often fail to generalize effectively across diverse settings, making them less reliable in real-world applications. In contrast, deep learning-based approaches have exhibited remarkable improvements by automatically learning hierarchical features from data, providing the flexibility and robustness needed for handling diverse and unstructured text [11, 12, 14]. This trend of leveraging deep neural networks has revolutionized the detection and recognition of scene text, pushing the boundaries of performance in both detection [15, 14] and recognition [16, 17] tasks. In related fields such as EEG signal classification and robotic vision, similar advancements have been achieved through the integration of deep learning models. For instance, in EEG-based classification, Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have been successfully used to classify brain activity patterns for various applications, including mental health diagnostics and control systems for robotics [18, 19]. In robotics, autonomous systems use deep learning for visual perception, enabling robots to perform complex tasks such as navigation in dynamic and unstructured environments, a field where advancements in computer vision are closely tied to real-time decision-making [13, 1].

Building on these advancements, this paper proposes a novel two-stage framework for scene text detection and recognition, incorporating state-of-the-art techniques from both the computer vision and deep learning fields. In the first stage, we employ a YOLOv5-based detection model, which allows for the efficient and precise identification of text regions in natural scene images. YOLOv5's ability to predict both bounding boxes and class probabilities in a single pass ensures high accuracy and real-time

ICYRIME 2025: 10th International Conference of Yearly Reports on Informatics, Mathematics, and Engineering. Czestochowa, January 14-16, 2025

djouher.akrour@univ-biskra.dz (D. Akrour);
mohamedakram.khelili@univ-biskra.dz (M. A. Khelili);
aloui.imene@univ-biskra.dz (I. Aloui);
azeddine.aissaoui@gmail.com (A. Aissaoui)

performance, making it suitable for practical deployment in robotics and mobile applications. In the second stage, a Latin text recognition module is introduced, which combines character segmentation via an LSTM network and text recognition using a Capsule Network (CapsNet) to capture complex spatial relationships between characters and words. The system is further enhanced by a semantic post-processing step that applies grammatical corrections and evaluates word similarity using metrics such as Levenshtein distance and cosine similarity.

The primary contributions of this work are as follows: First, we present a robust end-to-end system for scene text detection and recognition tailored for Latin scripts. Second, we propose an efficient one-stage text detector based on a Fully Convolutional Network (FCN), which handles multi-scale text detection without introducing excessive computational overhead. Third, we introduce an innovative recognition module that integrates LSTM and CapsNet, achieving comparable performance to state-of-the-art systems in text recognition tasks.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work, highlighting significant advances in scene text detection, EEG classification, and robotic applications. Section 3 presents the details of the proposed framework. Section 4 outlines the experimental setups and performance evaluations, while Section 5 concludes with a summary and future directions.

2. Related work

The detection and recognition of scene text have garnered substantial attention in the computer vision domain due to their significance in numerous real-world applications. Over the years, various methods have been proposed to tackle the challenges associated with scene text detection and recognition, which have been thoroughly reviewed in several comprehensive surveys and analyses [20, 21]. These methods can be broadly classified into two main categories: text detection and text recognition.

Scene text detection approaches can be divided into traditional machine learning-based methods and modern deep learning-based methods. Traditional approaches rely heavily on handcrafted features and techniques such as sliding windows and connected components to detect text in natural scene images [22, 23, 24, 25, 26]. Although these methods have shown promising results, they often suffer from a high rate of false positives when applied to complex and diverse real-world scenarios. In contrast, deep learning-based methods have emerged as the dominant approach, offering improved accuracy and robustness [11, 27, 14, 28]. Deep learning-based text detection methods can be further categorized into two-stage and one-stage strategies. Two-stage approaches, such as Faster R-CNN [29], rely on regional proposals and have inspired advanced models like Connectionist Text Proposal Network (CTPN) [30], R2CNN [31], and RRPN [32, 33]. For example, TextFuseNet [34, 35] uses multi-level feature representations and multi-path fusion to enhance text detection, achieving high accuracy but with significant computational overhead. On the other hand, one-stage approaches eliminate the region proposal phase and directly estimate candidate text regions from feature maps. Networks such as YOLO [36, 37, 38], SSD [39], and their derivatives have demonstrated exceptional efficiency. For instance, Gupta et al. [40] integrated YOLO with a random-forest classifier to reduce false positives, while He et al. [41] incorporated an attention mechanism in SSD to suppress background noise. Similarly, TextBoxes [11] and its extension, TextBoxes++ [42], addressed varying text aspect ratios and arbitrary orientations, respectively, while SegLink [15] used SSD to segment text into smaller components linked into complete instances. EAST [14] directly employed a fully convolutional network (FCN) for efficient text region detection without unnecessary intermediate steps, followed by thresholding and non-maximum suppression for refinement.

Text recognition methods are generally classified into sequence-based, word-based, and character-based approaches. Sequence-based approaches represent text as a sequence of characters. For example, CRNN [43] combines convolutional and recurrent neural networks to extract feature sequences and model contextual information. Similarly, Shi et al. [16] integrated a spatial transformer network with a sequence recognition network to robustly recognize irregular text. Word-based approaches, such as Jaderberg et al.'s method [17], focus on recognizing entire words by training convolutional neural networks on synthetic word datasets. While these methods have achieved state-of-the-art performance, they are often constrained by a predefined vocabulary. Character-based approaches, on the other hand, detect and recognize individual characters before assembling them into words. For instance, Minetto et al. [44] utilized histograms of oriented gradients for character description and recognition, while Yao et al. [45] proposed Strokelets, a robust multi-scale representation capturing character structures at different levels. This approach offers greater flexibility and is not limited by text length, making it suitable for complex scenarios.

These advancements in both text detection and recognition have significantly contributed to the development of more robust and efficient systems, laying a strong foundation for further research in this domain.

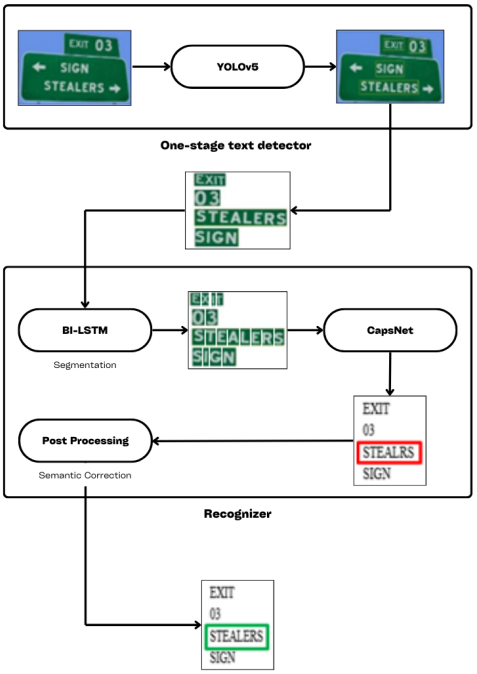

Figure 1: General architecture of proposed model.

3. Proposed model

The proposed model, as illustrated in Figure 1, consists of two imperative component including text detector and text recognizer. Firstly, candidate text region is localized from input image using one-stage text detector based on YOLOv5. Following that, text image is segmented into set of individual character patches using BILSTM-based segmentation technique. Then, these patches pass one-by-one to the capsule network which help to accurately recognize each character. The Set of recognized characters form complete word which pass by Post-Processing module to apply semantic correction in order to enhance the accuracy and effectiveness of recognizer component. More details about each component are described below.

3.1. One-stage text detector

Yolov5 was chosen as our scene text detector for several key reasons. First, it integrates the Cross Stage Partial Network (CSPNet) [46] with Darknet, forming CSPDarknet as its backbone. This design enhances inference speed and accuracy while reducing computational complexity by merging feature maps from different network stages. Second, Yolov5 employs the Path Aggregation Network (PANet) [47] to improve information flow. PANet uses an enhanced Feature Pyramid Network (FPN) structure with a shorter bottom-up path to better propagate low-level features, aiding the model's performance on unseen data and improving text scaling. Additionally, adaptive feature pooling ensures valuable information is passed through each feature level, enhancing localization accuracy for text detection. Finally, Yolov5's detection heads generate three different feature map sizes, enabling multi-scale predictions and enabling the detector to handle text of varying sizes under challenging real-world conditions.

3.2. Text Recognition System

In this section, we introduce the second stage of our framework which consists of three modules:

3.2.1. Segmentation module

After detecting the text using Yolov5, two layers of LSTM have been used with 256 units to learn long-ranges temporal dependencies. The LSTM architecture consists of three gates called input, forget, and output gates, connected with memory cells which make the LSTM stores the previous context for long time. The input gate consists of encoding information by applying hyperbolic tangent function $(tanh)$ on the active cell (x_t) and the previous cell output (h_{t-1}) in order to generate vector of values between -1 and $+1$. Meanwhile, the forget gate used (x_t) and (h_{t-1}) to be multiplied with weight's matrices and added to the bias, then passed to the activation function which resulted binary values. Where the 0 means that the cell information will be cleaned, however, the 1 means that the cell information will be stored for the future use. The output gate applies $sigmoid$ and $tanh$ function to active cell (x_t) and the previous cell output (h_{t-1}), then, multiply them with the vector values generated in the input gate to produce an output that will be passed to the next cell.

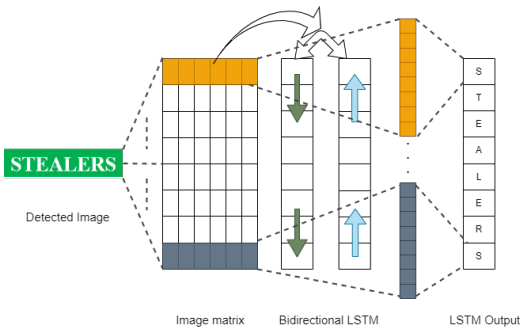

Figure 2: BI-LSTM architecture for segmentation module.

In our work, we used bidirectional LSTM, as shown in figure 2, to context information from each vector of the

detected words by applying the forward and the backward LSTM. The first one is used to analyze a vector of forward hidden states $\overrightarrow{S} = \{\overrightarrow{S_1}, \overrightarrow{S_2}, ..., \overrightarrow{S_t}\}$, which is only dependent on the left neighbors at each time t. while, the backward LSTM is used for analyzing a vector of backward hidden states $\overleftarrow{S} = \{\overleftarrow{S_1}, \overleftarrow{S_2}, ..., \overleftarrow{S_t}\}$, which is only dependent on the right neighbors at each time t. In the last step, the result of forward and backward should be concatenate to represent character's segment at each vector $S_t = [\overrightarrow{S_t}; \overleftarrow{S_t}]$. The output of the segmentation is sequence of character's image which will be faded to CapsNet after convert it to binary image.

3.2.2. Recognition module

Here, we tend to apply the same CapsNet structure employed previously in [6] and modifying it according to our purpose. Figure 3 depicts the overall CapsNet structure used for scene text recognition.

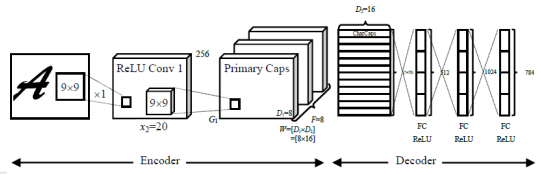

Figure 3: An overview of proposed CapsNet architecture. [6]

The CapsNet structure is composed of an encoder and a decoder, former of which comprises of:

- *ReLU Convolutional Layers:* The layer has 256 kernels each with a bias term, stride of 1, size of $9\times9\times1$ followed by the rectified linear activation $(ReLU)$. This layer used as lower-level feature extractors and outputs $20 \times 20 \times 256$ tensor.
- *PrimaryCaps layers:* The 8 capsule layer applies $9 \times 9 \times 256$ convolutional kernels, with stride 2, to the $20 \times 20 \times 256$ input tensor. This layer produce combination of the above feature outputs and generates $6 \times 6 \times 8 \times 8$ tensor.
- *CharCaps Layers:* These 70 capsule layers are used for the generation of the loss function and transformational weight matrix.

Whereas, Decoder consists of three Fully Connected layers (FC).

The loss function is calculated for correct and incorrect CharCaps, primarily defined as 1 if the correct label corresponds with the character of this particular CharCap and 0 otherwise. A zero-loss event is initiated either when a probability of right or wrong prediction is greater than m^+ or less than m^-, respectively. For each CharCaps capsule, k, the incurred loss is as follows:

$$L_k = T_k \max\left(0, m^+ - \|v_k\|\right)^2 + \lambda(1 - T_k) \max\left(0, \|v_k\| - m^-\right)^2 \quad (1)$$

where $T_k = 1$ if an image of class k is present and $m^+ = 0.9$ and $m^- = 0.1$, we use $\lambda = 0.5$.

The 8×16 transformation matrix W_{ij} maps the 8-dimensional capsule input space to a 16-dimensional capsule output space for each class j in relation to the capsule output of the previous layer u_i. The predicted vector $\hat{u}_{j|i}$ is expressed by a matrix operation between the weight matrix W_{ij} and u_i.

$$\hat{u}_{j|i} = W_{ij} u_i \quad (2)$$

The final output v_j for class j is computed using novel vector-to-vector nonlinearity squashing function as:

$$v_j = \frac{\|S_j\|^2}{1 + \|S_i\|^2} \frac{S_j}{\|S_i\|} \quad (3)$$

where:

$$S_j = \sum_i C_{ij}\, \hat{u}_{j|i} \quad (4)$$

with C_{ij} coupling coefficients measuring the probability of primary capsule i probabilistically triggering capsule j. S_j representing the weighted sum shrinked by the squashing function.

3.2.3. Post-processing module

In this module, English lexicon, Levenshtein distance [48] and cosine similarity [49] metrics are adopted to grammatically check the resulted word from CapsNet. The main purpose of use such metrics is to determine the required number of changes (inserting, deleting or replacing a character in word) and enhancing recognizer computational efficiency by reducing the number of words that will be treated by cosine metric. Figure 4 depicts the overall architecture of post-processing module.

The word generated by CapsNet pass firstly to the lexicon for selecting the set of words that have the same Stem of the input word. Then, this set of words will be handled one-by-one by the two metrics mentioned before. Finally, the word with the highest cosine similarity is chosen as the correct word.

Levenshtein [48] is based on calculating the distance matrix between the components of two words. The first step is to create matrix of shape $(m+2, n+2)$ where m and n are the size of the two words. The first two lines represent the first word and indices respectively, and the first two columns represent the second word and indices respectively. Then, the matrix should be completed with

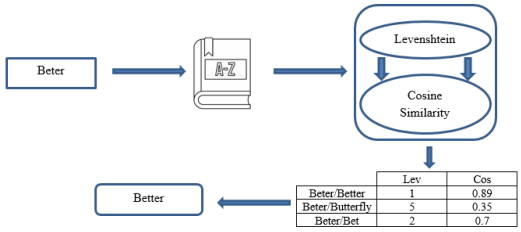

Figure 4: An overview of the proposed post-processing module.

0. For instance, the matrix of the word "beter" and "better" will look like:

$$\begin{pmatrix} & & b & e & t & e & r \\ & & 1 & 2 & 3 & 4 & 5 \\ b & 1 & 0 & 0 & 0 & 0 & 0 \\ e & 2 & 0 & 0 & 0 & 0 & 0 \\ t & 3 & 0 & 0 & 0 & 0 & 0 \\ t & 4 & 0 & 0 & 0 & 0 & 0 \\ e & 5 & 0 & 0 & 0 & 0 & 0 \\ r & 6 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

After that, we have to compare between the characters of the two words, character by character in each row and each column. The value of comparison in the point (x, y) will be the minimum of three values $[(x-1, y)+1]$, $[(x-1, y-1)]$, and $[(x, y-1)+1]$. The output of this matrix will be:

$$\begin{pmatrix} & & b & e & t & e & r \\ & & 1 & 2 & 3 & 4 & 5 \\ b & 1 & 0 & 1 & 2 & 3 & 4 \\ e & 2 & 1 & 0 & 1 & 2 & 3 \\ t & 3 & 2 & 1 & 0 & 1 & 2 \\ t & 4 & 3 & 2 & 1 & 1 & 2 \\ e & 5 & 4 & 3 & 2 & 1 & 2 \\ r & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}$$

As we see in the resulting matrix, the positions (5, 4) and (6, 5) have the value 1 which are incorrect because the letter "e" in the position (5, 0) is equal to the letter "e" in the position (0, 4). In addition to that, the Levenshtein distance between the two words is 1 which means that there is missing character in the second word. Using Levenshtein distance allows recognizer to select three most identical words from the set of words who will be next treated by the cosine metric.

Cosine Similarity is based on calculating the cosine angle of words' vectors [49]. After constructing the vector of the two words (W_1, W_2), the cosine similarity is calculated as follows:

$$\cos(W_1, W_2) = \frac{W_1 \times W_2}{\|W_1\| \times \|W_2\|}$$

$$= \frac{\sum_{i=1}^{n} W_{1i} \times W_{2i}}{\sqrt{\sum_{i=1}^{n} W_{1i}^2} \times \sqrt{\sum_{i=1}^{n} W_{2i}^2}} \quad (5)$$

$$\cos(beter, better) = 0.89$$

The values of the cosine similarity will be arranged between 0 and 1 where values closer to 1 indicate that the words more similar.

4. Experiments and results

4.1. Datasets

To evaluate the performance and versatility of our proposed text detection and recognition framework, we conduct experiments using four challenging benchmark datasets: ICDAR2013 [50], ICDAR2015 [51], MSRA-TD500 [52], and ICDAR2017-MLT [53]. The ICDAR2013 dataset is widely recognized as the standard benchmark for horizontal text detection. It includes 229 training images and 233 testing images, with word-level annotations provided for each image. Similarly, the ICDAR2015 dataset comprises 1000 training images and 500 testing images, featuring various accidental scene text instances annotated with quadrangular bounding boxes. The MSRA-TD500 dataset contains 300 training images and 200 test images, incorporating both English and Chinese text. The text areas in this dataset are arbitrarily oriented, and annotations are provided at the sentence level, making it particularly challenging for text detection models. The ICDAR2017-MLT dataset is a more complex and diverse collection, consisting of 7200 training images, 1800 validation images, and 9000 testing images. This dataset includes multi-oriented, multi-script, and multilingual scene text instances with line-level and word-level annotations, significantly increasing the difficulty of the detection task. For the evaluation of text recognition, we use a modified version of the EnglishFnt dataset from the Chars74K collection [54], which has also been used in previous works [6]. This dataset is employed for training the Long Short-Term Memory (LSTM) network for word segmentation. To assess the effectiveness of our text detection and recognition system, we adopt the standard evaluation metrics, including precision (P), recall (R), and F-measure (F), to quantify detection and recognition performance.

4.2. Evaluation

4.2.1. Text detection

To assess the effectiveness of our framework in detecting horizontal and long text, we compare its performance with state-of-the-art text detection methods on the ICDAR2013 and MSRA-TD500 datasets. On the ICDAR2013 benchmark, our detector outperforms other methods by

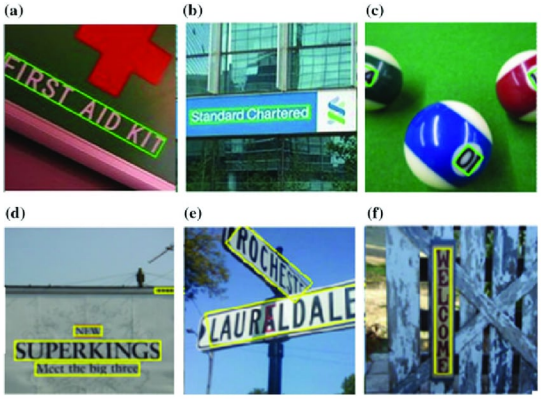

Figure 5: Examples of text detection results of our detector.

at least 1%, except for TextFuseNet [34]. On the MSRA-TD500 dataset, our detector achieves a precision of 89.5%, improving upon the SRPN+VGGDet [55] method, which has a precision of 87.3%. This improvement demonstrates the superiority of our framework in detecting long scene text using a single fully connected network. We also validate the performance of our detector on multilingual text detection using the ICDAR2017-MLT dataset. Except for DB-ResNet-50 [56], our detector delivers the highest precision, confirming that our Yolov5-based framework effectively handles the diverse text shapes across different languages. For multi-oriented text detection on the ICDAR2015 dataset, our method achieves an F-measure of 55.7% and precision of 76%. Compared to one-stage methods such as SegLink [57], EAST [14], TextBoxes++ [42], and RRD [58], our precision is 7.3%, 11.2%, and 9.6% lower, respectively, but 2.9% higher than SegLink. This indicates that while our detector does not surpass others in precision for multi-oriented text, it still performs competitively. Additionally, the use of multi-branch detection improves detection accuracy. By generating feature maps of three different sizes (18×18, 36×36, 72×72) and fusing them, our detector effectively utilizes both shallow and deep features. This enables it to capture rich details and semantic information, enhancing its ability to handle text of varying sizes. Overall, our experimental results demonstrate that the proposed text detector achieves comparable performance to state-of-the-art methods. It effectively detects horizontal, long, multilingual, and multi-oriented text in natural images, as illustrated in Figure 5. Despite the varying styles of images, the results highlight the detector's ability to accurately identify text with diverse shapes, orientations, sizes, and languages.

4.2.2. Text Recognition

The segmentation results demonstrate an impressive 94% accuracy when training our LSTM model on the Chars74K dataset. This improvement highlights the ability of LSTM to learn long-range temporal dependencies by utilizing both the forward and backward aspects of the LSTM architecture. The model effectively captures the features of both previous and future characters within the image, enhancing segmentation of the box image into sub-images, which are then passed to the CapsNet model.

Experimental results show that our CapsNet model, trained on Chars74K images, achieves a recognition rate of 92%. This indicates that our character recognition model significantly outperforms state-of-the-art methods, as presented in Table 1, and achieves comparable performance to the optimal methods.

Table 1
Recognition rate comparison of state-of-the-art methods on the Chars74K dataset

State-of-the-art methods	Recognition rate [%]
AlexNet [59]	77.77
GoogleNet [59]	88.89
Multiscale HoG Features [60]	80
ConvNet [60]	71.69
DCNN [61]	90.32
Proposed CapsNet architecture	**92**

Our results also demonstrate CapsNet's ability to handle a wide variety of character shapes and its robustness when dealing with datasets containing a larger number of classes (70 classes). Table 2 presents the accuracy, recall, and F1-score for a selection of characters from the Chars74K dataset. This significant improvement in performance is attributed to the complexity of the PrimaryCaps layers, which, by utilizing vectors during training, increase the model's capacity to represent character information and effectively capture various character attributes.

Table 2
Accuracy (Acc), Recall (Rec), and F1-score (F1) of Character Recognition (CapsNet)

Metric	0	2	9	I	P	Y	x	y	?
Acc [%]	78	99	99	89	91	92	80	96	97
Rec [%]	83	99	98	78	91	96	83	91	100
F1 [%]	81	99	98	83	91	94	81	93	99

Figures 6 and 7 illustrate the accuracy and loss of CapsNet during training and validation on the Chars74K dataset. Following the application of semantic correction in the post-processing module, the overall word recognition accuracy reaches an exceptional 98%.

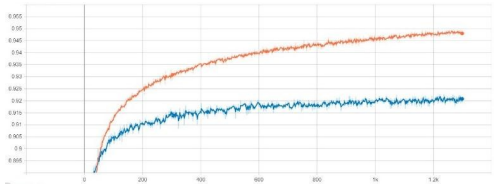

Figure 6: The accuracy of CapsNet in training and validation on Chars74K.

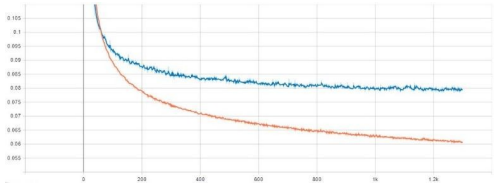

Figure 7: The loss of CapsNet in training and validation on Chars74K.

5. Conclusion

In this paper, we have presented a novel end-to-end system for extracting text from natural scene image. We introduced robust detector which can suitably localize and extracts the region where text is existing and this has an appreciable increase in accuracy while recognizing the texts. The proposed detector is resistant to backgrounds complexities and is insensitive to noise, scale change, variation of font and languages. Moreover, a modular Latin text recognition method is proposed to accurately recognize text in different situation. We additionally employed, in this work, CapsNet with dynamic routing for recognition of detected text. After devising the text detected to sub-images of individual characters using specific segmentation method based on BI-LSTM network; CapsNet is leveraged to diverse characters into tens of categories. Furthermore, we proposed semantic method as post processing step to improve the performance and the accuracy of the system in the full word recognition.

Experimental results on different popular text spotting benchmarks, including both regular and irregular datasets, prove that our proposed model can significantly outperform state-of-the-art methods in terms of detection and recognition with its efficiency and high accuracy. In future work, this system will be tested in Chinese or other languages. Future work will look also at improving our model to deal with the problems of false positives and partially detected text lines especially those belonging to arbitrarily-oriented and curved textual regions.

6. Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] W. Guettala, A. Sayah, L. Kahloul, A. Tibermacine, Real time human detection by unmanned aerial vehicles, in: 2022 International Symposium on iNnovative Informatics of Biskra (ISNIB), IEEE, 2022, pp. 1–6.

[2] A. Tibermacine, N. Djedi, Gene regulatory network to control and simulate virtual creature's locomotion (2015).

[3] N. Boutarfaia, S. Russo, A. Tibermacine, I. E. Tibermacine, Deep learning for eeg-based motor imagery classification: Towards enhanced human-machine interaction and assistive robotics, in: CEUR Workshop Proceedings, volume 3695, 2023, p. 68 – 74.

[4] N. Brandizzi, A. Fanti, R. Gallotta, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Unsupervised pose estimation by means of an innovative vision transformer, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 13589 LNAI, 2023, p. 3 – 20. doi:10.1007/978-3-031-23480-4_1.

[5] A. Tibermacine, W. GUETTALA, I. E. Tibermacine, Efficient one-stage deep learning for text detection in scene images, Electrotehnica, Electronica, Automatica (EEA) 72 (2024) 65–71.

[6] A. Tibermacine, S. M. Amine, An end-to-end trainable capsule network for image-based character recognition and its application to video subtitle recognition., ICTACT Journal on Image & Video Processing 11 (2021).

[7] F. Bonanno, G. Capizzi, A. Gagliano, C. Napoli, Optimal management of various renewable energy sources by a new forecasting method, in: SPEEDAM 2012 - 21st International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2012, p. 934 – 940. doi:10.1109/SPEEDAM.2012.6264603.

[8] F. Bonanno, G. Capizzi, G. L. Sciuto, C. Napoli, G. Pappalardo, E. Tramontana, A cascade neural network architecture investigating surface plasmon polaritons propagation for thin metals in openmp, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelli-

gence and Lecture Notes in Bioinformatics), volume 8467 LNAI, 2014, p. 22 – 33. doi:10.1007/978-3-319-07173-2_3.

[9] Y.-F. Pan, X. Hou, C.-L. Liu, Text localization in natural scene images based on conditional random field, in: 10th international conference on document analysis and recognition, IEEE, 2009, pp. 6–10.

[10] A. Tibermacine, D. Akrour, R. Khamar, I. E. Tibermacine, A. Rabehi, Comparative analysis of svm and cnn classifiers for eeg signal classification in response to different auditory stimuli, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–8.

[11] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: A fast text detector with a single deep neural network, in: Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.

[12] Z. Tian, W. Huang, T. He, P. He, Y. Qiao, Detecting text in natural image with connectionist text proposal network, in: European conference on computer vision, Springer, 2016, pp. 56–72.

[13] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: A flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 20–36.

[14] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 5551–5560.

[15] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2550–2558.

[16] B. Shi, X. Wang, P. Lyu, C. Yao, X. Bai, Robust scene text recognition with automatic rectification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4168–4176.

[17] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Synthetic data and artificial neural networks for natural scene text recognition, arXiv preprint arXiv:1406.2227 (2014).

[18] A. Tibermacine, I. E. Tibermacine, M. Zouai, A. Rabehi, Eeg classification using contrastive learning and riemannian tangent space representations, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–7.

[19] A. Tibermacine, N. Djedi, Neat neural networks to control and simulate virtual creature's locomotion, in: 2014 International Conference on Multimedia Computing and Systems (ICMCS), IEEE, 2014, pp. 9–14.

[20] H. Lin, P. Yang, F. Zhang, Review of scene text detection and recognition, Archives of computational methods in engineering 27 (2020) 433–454.

[21] M. Brisinello, R. Grbić, M. Vranješ, D. Vranješ, Review on text detection methods on scene images, in: international symposium ELMAR, IEEE, 2019, pp. 51–56.

[22] B. Nail, M. A. Atoussi, S. Saadi, I. E. Tibermacine, C. Napoli, Real-time synchronisation of multiple fractional-order chaotic systems: an application study in secure communication, Fractal and Fractional 8 (2024) 104.

[23] K. I. Kim, K. Jung, J. H. Kim, Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1631–1639.

[24] L. Neumann, J. Matas, Scene text localization and recognition with oriented stroke detection, in: Proceedings of the ieee international conference on computer vision, 2013, pp. 97–104.

[25] B. Nail, B. Djaidir, I. E. Tibermacine, C. Napoli, N. Haidour, R. Abdelaziz, Gas turbine vibration monitoring based on real data and neuro-fuzzy system, Diagnostyka 25 (2024).

[26] X.-C. Yin, X. Yin, K. Huang, H.-W. Hao, Robust text detection in natural scene images, IEEE transactions on pattern analysis and machine intelligence 36 (2013) 970–983.

[27] S. Russo, I. E. Tibermacine, A. Tibermacine, D. Chebana, A. Nahili, J. Starczewscki, C. Napoli, Analyzing eeg patterns in young adults exposed to different acrophobia levels: a vr study, Frontiers in Human Neuroscience 18 (2024). doi:10.3389/fnhum.2024.1348154.

[28] I. Naidji, A. Tibermacine, W. Guettala, I. E. Tibermacine, et al., Semi-mind controlled robots based on reinforcement learning for indoor application., in: ICYRIME, 2023, pp. 51–59.

[29] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE transactions on pattern analysis and machine intelligence 39 (2016) 1137–1149.

[30] S. Bouchelaghem, I. E. Tibermacine, M. Balsi, M. Moroni, C. Napoli, Cross-domain machine learning approaches using hyperspectral imaging for plastics litter detection, in: 2024 IEEE Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), IEEE, 2024, pp. 36–40.

[31] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, Z. Luo, R 2 cnn: Rotational region cnn for arbitrarily-oriented scene text detection, in: 24th International conference on pattern recognition (ICPR), IEEE, 2018, pp. 3610–3615.

[32] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via

rotation proposals, IEEE transactions on multimedia 20 (2018) 3111–3122.
[33] B. Ladjal, I. E. Tibermacine, M. Bechouat, M. Sedraoui, C. Napoli, A. Rabehi, D. Lalmi, Hybrid models for direct normal irradiance forecasting: A case study of ghardaia zone (algeria), Natural Hazards 120 (2024) 14703–14725.
[34] J. Ye, Z. Chen, J. Liu, B. Du, Textfusenet: Scene text detection with richer fused features., in: IJCAI, volume 20, 2020, pp. 516–522.
[35] S. eddine Boukredine, E. Mehallel, A. Boualleg, O. Baitiche, A. Rabehi, M. Guermoui, A. Douara, I. E. Tibermacine, Enhanced performance of microstrip antenna arrays through concave modifications and cut-corner techniques, ITEGAM-JETIA 11 (2025) 65–71.
[36] A. Farhadi, J. Redmon, Yolov3: An incremental improvement, in: Computer vision and pattern recognition, volume 1804, Springer Berlin/Heidelberg, Germany, 2018, pp. 1–6.
[37] C. Napoli, V. Ponzi, A. Puglisi, S. Russo, I. Tibermacine, et al., Exploiting robots as healthcare resources for epidemics management and support caregivers, in: CEUR Workshop Proceedings, volume 3686, CEUR-WS, 2024, pp. 1–10.
[38] S. Russo, S. Ahmed, I. E. Tibermacine, C. Napoli, Enhancing eeg signal reconstruction in cross-domain adaptation using cyclegan, in: 2024 International Conference on Telecommunications and Intelligent Systems (ICTIS), IEEE, 2024, pp. 1–8.
[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, Springer, 2016, pp. 21–37.
[40] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2315–2324.
[41] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, X. Li, Single shot text detector with regional attention, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 3047–3055.
[42] M. Liao, B. Shi, X. Bai, Textboxes++: A single-shot oriented scene text detector, IEEE transactions on image processing 27 (2018) 3676–3690.
[43] B. Shi, X. Bai, 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 39 (2016) 2298–2304.
[44] R. Minetto, N. Thome, M. Cord, N. J. Leite, J. Stolfi, Thog: An effective gradient-based descriptor for single line text regions, Pattern recognition 46 (2013) 1078–1090.
[45] C. Yao, X. Bai, B. Shi, W. Liu, Strokelets: A learned multi-scale representation for scene text recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 4042–4049.
[46] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, Cspnet: A new backbone that can enhance learning capability of cnn, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390–391.
[47] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
[48] V. Lcvenshtcin, Binary coors capable or 'correcting deletions, insertions, and reversals, in: Soviet Physics-Doklady, volume 10, 1966, pp. 707–710.
[49] B. Li, L. Han, Distance weighted cosine similarity measure for text classification, in: Intelligent Data Engineering and Automated Learning–IDEAL, Springer, 2013, pp. 611–618.
[50] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. De Las Heras, Icdar 2013 robust reading competition, in: 2013 12th international conference on document analysis and recognition, IEEE, 2013, pp. 1484–1493.
[51] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al., Icdar 2015 competition on robust reading, in: 2015 13th international conference on document analysis and recognition (ICDAR), IEEE, 2015, pp. 1156–1160.
[52] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, Detecting texts of arbitrary orientations in natural images, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 1083–1090.
[53] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al., Icdar2017 robust reading challenge on multilingual scene text detection and script identification-rrc-mlt, in: 14th IAPR international conference on document analysis and recognition (ICDAR), volume 1, IEEE, 2017, pp. 1454–1459.
[54] T. E. de Campos, B. R. Babu, M. Varma, Character recognition in natural images, in: International conference on computer vision theory and applications, volume 1, SCITEPRESS, 2009, pp. 273–280.
[55] W. He, X.-Y. Zhang, F. Yin, Z. Luo, J.-M. Ogier, C.-L. Liu, Realtime multi-scale scene text detection with scale-based region proposal network, Pattern Recognition 98 (2020) 107026.
[56] M. Liao, Z. Wan, C. Yao, K. Chen, X. Bai, Real-time scene text detection with differentiable binarization, in: Proceedings of the AAAI conference on artificial

intelligence, volume 34, 2020, pp. 11474–11481.
[57] X. Chen, A. L. Yuille, Detecting and reading text in natural scenes, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, IEEE, 2004, pp. II–II.
[58] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, X. Bai, Rotation-sensitive regression for oriented scene text detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5909–5918.
[59] M. Soomro, M. A. Farooq, R. H. Raza, Performance evaluation of advanced deep learning architectures for offline handwritten character recognition, in: 2017 International Conference on Frontiers of Information Technology (FIT), IEEE, 2017, pp. 362–367.
[60] A. J. Newell, L. D. Griffin, Multiscale histogram of oriented gradient descriptors for robust character recognition, in: International conference on document analysis and recognition, IEEE, 2011, pp. 1085–1089.
[61] S. Arivazhagan, M. Arun, D. Rathina, Recognition of handwritten characters using deep convolution neural network., Journal of the National Science Foundation of Sri Lanka 49 (2021).