Vietnamese Scene Text Detection Method based on Deep Learning

Huynh Van Huy [1], Nguyen Thi Thanh Tan[2], Ngo Quoc Tao [3]
[1] Lac Hong University, Vietnam
[2] Electric Power University, Vietnam
[3] Institute of Information Technology - Vietnam Academy of Science and Technology, Vietnam
Correspondence: Nguyen Thi Thanh Tan, tanntt@epu.edu.vn
Communication: received 18 Aug 2023, revised 28 Sep 2023, accepted 23 Nov 2023
Digital Object Identifier: 10.32913/mic-ict-research.v2024.n1.1242

Abstract: This paper proposes an efficient method for detecting Vietnamese text in outdoor scene images. Essentially, the text detection method presented here is based on the idea of utilizing deep learning network architectures to learn various geometric properties in order to reconstruct polygonal representations of text regions. The effectiveness of the method has been evaluated on four real-world outdoor scene image datasets, including the ICDAR 2015, Total-Text, VinText, and VnSceneText datasets. Experimental results show that the proposed method can detect text of various shapes and sizes with high and consistent accuracy. Specifically, the method achieved Precision, Recall, and Hmean scores of 87.53%, 86.94%, and 87.23%, respectively, on the test datasets, 84.32%, 88.17%, and 86.20% on a different dataset, 85.63%, 87.94%, and 86.77% on yet another dataset, and 85.14%, 87.23%, and 86.17% on the last dataset. The experimental results indicate that this approach is feasible for detecting Vietnamese text in outdoor scene images.

Keywords: *Scene text, scene images, text regions, segmentation, detection, features, mapping, accuracy, recall, harmonic mean, convolution, scale, batch, batch normalization.*

I. Introduction

The traditional problem of text or document recognition (OCR) has been the subject of research investment for several decades. The input for this problem mainly consists of document images acquired from scanning devices (scanners). It can be said that the traditional text recognition problem has been effectively addressed, and OCR systems commercially available on the market now meet and satisfy user requirements.

In recent times, stemming from practical needs, the application of text recognition has expanded into various fields [1, 2] exemplified by: Detecting and retrieving information (origin, composition, user recommendations, etc.) from signs, product packaging, and product labels to enhance customers' understanding of products or services in the commercial sector. Employing OCR to assist machines in reading and recognizing labels, and technical specifications on components in the industrial sector. Reading information on road signs, directional signs, and street names to support visually impaired individuals in navigating or developing autonomous vehicle systems. Additionally, integrating advanced technologies like text recognition, machine translation, and speech recognition for translating comprehensible content from signs, menus, and instructions into the user's (customer's) language is becoming increasingly popular in the tourism industry as well as daily life. The common characteristic of these expanded applications is that the input images are typically irregular in shape and captured or recorded using various handheld devices (smartphones, webcams, tablets, iPads, cameras, etc.) under entirely natural conditions (with no constraints on lighting, angles, shooting distances, as well as the area, tilt/tilt of the text-containing image, etc.). This class of input images is referred to by a general term called "scene images." Detecting text in scene images or videos is also known as scene text detection.

Compared to traditional OCR, text detection in scene images often involves significant complexity and faces numerous challenges due to the following factors: i) The images are highly complex due to containing various objects and different types of noise; ii) Text in images is typically non-standard (irregular) with diverse shapes, sizes, orientations, colors, and different font styles; iii) There is no uniformity among the images as they are captured from various angles, distances, lighting conditions, and different resolutions; iv) Text can be occluded by other objects in the image or by various factors such as blurriness, darkness, glare, or noise; v) Scene text detection tasks often require processing a large volume of images or videos, demanding

algorithms capable of handling data quickly and efficiently. This paper proposes an effective method for detecting Vietnamese scene text based on a deep learning approach. The main idea of the method is to use deep neural network architectures to learn various geometric properties for reconstructing polygonal representations of text regions. The proposed model is composed of four main components, including: (1) A backbone network for extracting features from the input image; (2) A feature fusion model constructed as a Feature Pyramid Network (FPN) to integrate and represent features at multiple scales, aiding in the detection of text regions of various sizes; (3) A context attention model structured as a deep convolutional network, trained to capture context information (long-range dependencies) of pixels to obtain more representative features and (4) A text instance segmentation algorithm in the final step, which is responsible for identifying and grouping boundary points into text regions. Our main contributions in this paper include:

- First, we have proposed an effective method for detecting Vietnamese text in scene images or videos with high accuracy, that is less affected by noise and the orientation of the text. The method is capable of adapting to and performing well with text of various shapes, including curved text.

- Second, we have incorporated a contextual attention mechanism that allows the model to learn how to create a weight matrix (referred to as the attention matrix). This matrix indicates the importance of each pixel in the image for text detection. Regions with higher attention values will be given more consideration by the model. This helps reduce the algorithm's processing time and diminishes the impact of unimportant factors in the image. As a result, it simplifies and enhances the efficiency of subsequent text segmentation and post-processing stages.

- Third, we have collected and constructed a dataset comprising 3000 Vietnamese scene images to facilitate research and evaluate the testing of algorithms for text detection and recognition in Vietnamese scene images. In addition, we have applied advanced image processing techniques such as transformations, synthesis, and image augmentation to enrich the training dataset for building and testing deep learning algorithms.

The remaining contents of the paper are presented as follows: Section 2 provides a summary of related approaches. The main idea and the implementation steps of the proposed method are detailed in Section 3. Section 4 presents the experimentation and evaluation process of the method's effectiveness. Some conclusions and avenues for further development are discussed in Section 5.

II. Related works

Numerous methods have been proposed to address the problem of scene text detection. Traditional approaches have focused on image processing techniques and basic machine learning algorithms, such as sliding window methods and connected component analysis [1]–[8]. In practice, these methods have exhibited several limitations and achieved low detection performance on real-world scene images. They can particularly fail in challenging scenarios, such as detecting curved or skewed text, text with many characters joined together, or text captured under non-uniform lighting conditions [1], [9].

For this reason, most methods proposed in recent years have adopted a deep learning approach. Fundamentally, these methods can be categorized into three main approaches: bounding-box regression-based methods, part-based methods, and segmentation-based methods. The first approach, bounding-box regression-based, utilizes regression models to predict bounding boxes for text regions. M. Liao and colleagues [10] adjusted the anchors and aspect ratios of the convolutional layers based on SSD [11] for text detection. In [12]-[13], authors applied quadrilateral regression to detect multi-oriented text. P. He and colleagues [14] proposed an attention mechanism to preliminary identify text regions. In [15], the authors introduced the RRD method, separating classification and regression. In this method, rotation-invariant features are used for classification, while rotation-sensitive features are employed for regression to achieve better performance, particularly for long and multi-oriented text. In [16], [17], authors presented anchor-free methods, applying pixel-level regression for multi-oriented text regions. L. Xie and colleagues [18] introduced an RPN (Region Proposal Network) architecture to address the scaling issue in scene text detection. Generally, regression-based methods often employ simple post-processing algorithms like Non-Maximum Suppression (NMS). However, most of these methods do not achieve the expected accuracy for input text with irregular shapes, such as curved or covered text.

For part-based text detection methods, the approach initially involves detecting small components of text regions and then linking them to form words' bounding boxes or text line bounding boxes. In [19], the authors discovered bounding boxes for text segments and predicted their connections to handle long text regions effectively. J. Tang et al. [20] proposed an instance-aware component grouping algorithm to separate text regions more efficiently and improve the linking algorithm to accommodate text regions with arbitrary shapes. These methods generally achieve good performance in detecting long text lines. However, the linking algorithms are relatively complex with

manually tuned hyperparameters, making it challenging to adjust the algorithms and adapt to the diversity of input data effectively. Segmentation-based methods often combine pixel-level predictions with post-processing algorithms to determine bounding boxes. Zhang et al. [21] detected multi-oriented text using semantic segmentation and MSER-based algorithms. Xue et al. [22] used contour lines to separate text regions. In [23] and [24], the authors detected arbitrarily-shaped text regions using an instance segmentation approach based on Mask R-CNN [25]. W. Wang et al. [26] proposed progressive scale expansion by segmenting text regions with different scale kernels. Tian et al. [27] introduced a pixel embedding mechanism to cluster pixels from segmentation results. In [10] and [27], the authors presented new post-processing algorithms for segmentation results, leading to the lower inference speed. Fast methods for text detection in scene images focus on accuracy and inference speed. Recently, Pengfei Wang and colleagues [28] proposed the SAST method for detecting arbitrary-shaped text in scene images. This method uses a context-aware multi-task training framework based on the Fully Convolutional Network (FCN) architecture to learn various geometric properties for polygon representation of text regions. Experimental results show that SAST can accurately detecting text with arbitrary shapes and quickly process image. However, this method has limitations in detecting extremely large or small characters and text in highly curved images. Zhu and colleagues [29] introduced the FCE (Fourier Contour Embedding) method to represent the outline of arbitrarily shaped text as compact Fourier series symbols. They built the FCENet architecture with a backbone network, feature pyramid networks (FPN), and a simple post-processing model using Inverse Fourier Transform (IFT) and the Non-Maximum Suppression (NMS) algorithm. Experimental results demonstrate that this method achieves high accuracy on the CTW1500 and Total-Text datasets. However, it may encounter challenges when the text and background lack clear contrast or are oversized. M. Liao and colleagues [30] proposed a text detection method for arbitrary-shaped text using the Differentiable Binarization algorithm and the Adaptive Scale Fusion (ASF) mechanism to combine information from different scales of images flexibly. . This method can effectively address text detection in distorted text, and text of various shapes and sizes, achieving high accuracy and fast processing. However, the method's performance may significantly decrease in the presence of heavy noise. Additionally, the method's effectiveness depends on the selected scale threshold, as it performs well when the chosen threshold is suitable for the input image.

For Vietnamese scene text recognition, apart from facing the challenges of scene text recognition in general as mentioned above, there are additional difficulties related to diacritics and tone marks in the Vietnamese language. These diacritics can lead to issues like character and line concatenation within the text, reducing accuracy in detection and recognition. To ensure accurate recognition, it's necessary to integrate specialized processing mechanisms suitable for the specific characteristics of the Vietnamese language. Practical surveys reveal that research outcomes of Vietnamese text detection and retrieval in scene images are still comparatively limited. Some recent notable research results in Vietnamese text recognition have been published [31], [32]. In [31], the authors primarily focused on the text recognition stage. They proposed an approach that incorporates a dictionary into both the training and inference phases of the text recognition system. The dictionary is used to generate a list of possible results and find the most suitable result for the text's display format. This method helps to handle ambiguous cases encountered in practice and improves the overall performance of text recognition frameworks. The authors have contributed to the research community by providing the VinText scene text dataset, which serves research and algorithm quality improvement. In [32], the authors presented experimental results of various methods including DBN, PMTD, PAN, FCENet, RCNN, SATRN, NRTR, and RS on the VinText dataset. Inheriting from existing approaches (EAST [16], SAST[28], DB [29]), our Vietnamese Scene Text Detector (VNSTD) method is proposed to address challenging issues in recognizing Vietnamese scene text in images. It focuses on two main objectives: the capacity of accurately detecting scene text of any shape or size with high precision and fast processing speed and performs well in handling Vietnamese diacritics and tone marks.

III. VIETNAMESE SCENE TEXT DETECTION METHOD

The basic steps in the scene text detection process of the proposed method are described in Figure 1 From any given input image, it first passes through a backbone network to automatically extract features. The extracted features are fed into a fusion model. The fusion model is specially designed to integrate and represent the features at various scales. This processing mechanism aims to address text regions of different sizes.

The output results obtained from the fusion model will continue to be passed to the context attention module to integrate distant dependencies of pixels in order to obtain features with better representational qualities, thereby enhancing the accuracy of the segmentation algorithm in the subsequent step. The results obtained from the context

Figure 1. Vietnamese Scene Text Detection method (VNSTD)

attention module are used as input for the text instance segmentation task. The text instance segmentation step is responsible for locating text regions (identifying the position and shape) in the input image. In the following section, we will describe the implementation steps of the proposed method in detail and with more specificity.

1. Backbone network architecture

The backbone network is responsible for automatically extracting features from the input image. In the proposed model, we use the ResNet-50 network architecture Figure 2(a)

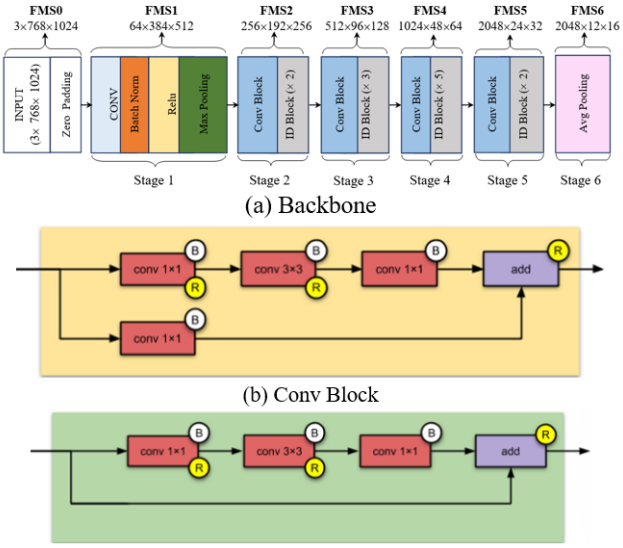


Figure 2. The backbone network architecture

This is the best-rated network architecture at the current time. The remarkable difference between ResNet-50 and traditional CNN models is the use of shortcut connections. This mechanism not only makes training deeper models easier (eliminating the vanishing gradient problem) but also reduces the dependence on the number of layers. This architecture consists of 50 layers: Zero Padding, Convolution, Max Pooling, Convolution Blocks, Identity Blocks, Average Pooling, and Flattening. In this context, the multiplication symbol (×) in the ID block implies multiple stacked ID blocks (for example, ID block (×2) means two stacked ID blocks). Each convolution layer in ResNet is applied with Batch Normalization.

The network execution process is divided into 6 stages, denoted from stage 1 to stage 6. The result of each stage is a set of feature maps. Specifically, with 6 stages, we obtain 6 sets of feature maps (denoted as FMS1 to FMS6). In the proposed method, the input image consists of 3 channels and is normalized to a size of 768×1024. Each channel is considered a feature map in FMS0. Zero padding is of size 3×3. Stage 1 takes the original input as a 3×768×1024 image. The architecture of stage 1 consists of 3 convolution layers (CONV) with 64 filters (size 7×7) and a stride of 2×2, BatchNorm normalization, and MaxPooling (3×3). The output of stage 1 (FMS1) comprises 64 feature maps with a size of 384×512. The feature maps' output from Stage 1 continues to be provided as input for Stage 2. The architecture of stage 2 includes 01 Convolutional block and 02 identity blocks, with 256 filters (size 2×2) and a stride of 2×2. The output of stage 2 (FMS2) consists of 256 feature maps (size 192×256). The feature maps from FMS2 continue to be provided as input for stage 3. The architecture of stage 3 consists of 01 convolutional block and 03 Identity blocks, with 1024 filters (size 2×2) and a stride of 2×2. The output of stage 3 (FMS3) comprises 512 feature maps, with a size of 96×128. The feature maps from FMS3 continue to be provided as input for stage 4. The architecture of stage 4 consists of 01 convolutional block and 05 identity blocks, using 1024 filters (size 2×2) with a stride of 2×2. The output of stage 4 (FMS4) comprises 1024 feature maps, with a size of 48×64. The feature maps from FMS4 are further provided as input to stage 5. The architecture of stage 5 consists of 01 convolutional block and 2 Identity blocks, using 2048 filters (size 2×2) with a stride of 2×2. The output of Stage 5 (FMS5) comprises 2048 feature maps, with a size of 24×32. The feature maps from FMS5 are further provided as input to Stage 6. The architecture of stage 6 consists of 01 Convolutional block and 02 identity blocks, using 2048 filters (size 2×2) with a stride of 2×2. The output of stage 6 (FMS6) comprises 2048 feature maps, with a size of 12×16.

Some representative output results (randomly selected) of the stages are illustrated in Figure 3. In this figure, the input image is depicted, and the notation (FMSi: C m/n) corresponds to the order of the stage and the order of the channel and the total channels (the number of output feature maps) of each stage (for example, the notation FMS1: C 39/64 means the feature map 39 (channel 39) out of a total

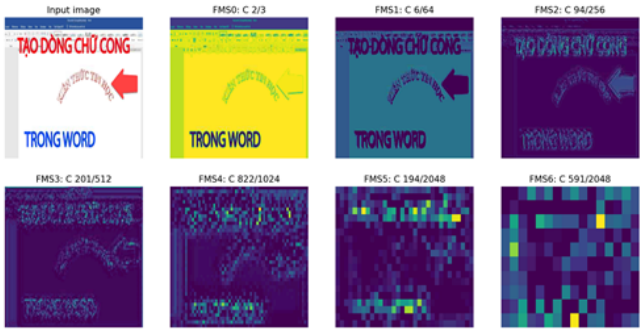


Figure 3. Some representative results of the stages

of 64 output feature maps of stage 1).

The structure of each Conv Block Figure 2(b) consists of four convolutional layers (Conv) and one merge layer (Add). In this structure, the symbol B stands for Batch normalization (applying the Batch normalization technique), and R stands for Relu (using the Relu activation function). Similarly, the structure of each ID block consists of three normalized convolutional layers (Batch normalization) and one merge layer (Add).

2. Fusion model

The Fusion model is described in detail in Figure 4. This model takes as input the 07 feature maps (from FMS0 to FMS6) obtained from the backbone network. This model

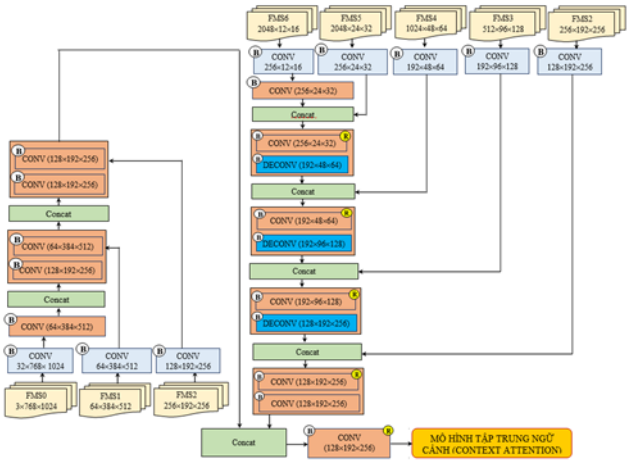


Figure 4. Fusion model

integrates and represents features at multiple scales to detect text regions of varying sizes. Essentially, the architecture of the model is constructed as a Feature Pyramid Network (FPN). This network is structured by two pathways: the bottom-up and the top-down pathways. In each layer of the network, three values correspond to the number of channels, height, and width of the output feature maps.

The bottom-up pathway serves as an encoder network to extract features. As the resolution and feature map size increase, the number of filters (corresponding to the number of output feature maps) and contextual information also increase. The bottom-up pathway yields 128 feature maps (corresponding to 128 output channels). Figure 5(a) illustrates an output result of this pathway (C: 100/128 means feature map 100 out of a total of 128 feature maps - displayed randomly). The top-down pathway acts as a

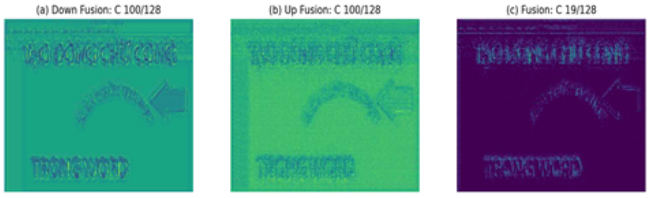


Figure 5. Fusion model

decoder network to construct the object. The execution process of the top-down pathway is the reverse of the bottom-up pathway: as it descends, the resolution increases, the feature map size grows, and the number of filters (number of output feature maps) decreases. Figure 5(b) illustrates an output result of this pathway. The final processing step involves concatenating the bottom-up and top-down pathways using a convolutional layer to produce an output consisting of 128 feature maps, each with a size of 192×256×128 (H = 192, W = 256, C = 128, where H is the height, W is the width, and C is the number of channels) - Figure 5(c). These feature maps are further provided to the context attention model to obtain better representative features, thereby enhancing the accuracy of text detection/segmentation in the subsequent stage.

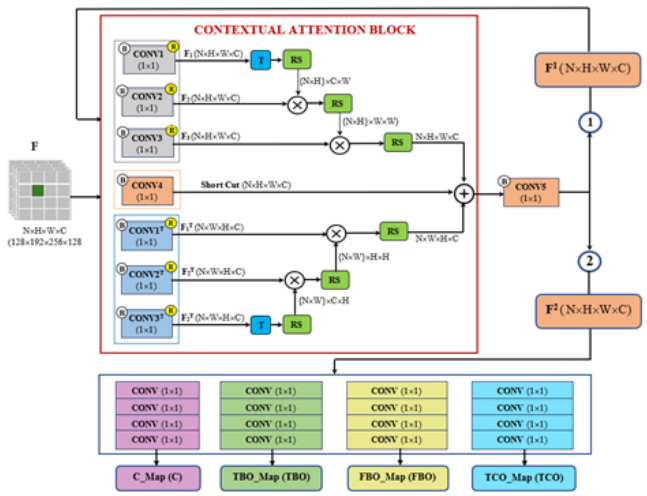


Figure 6. The context attention model

3. Contextual attention

The contextual attention model (Figure 6) inherits the self-attention mechanism from [28], [33]. The aim of the contextual attention model is to aggregate contextual information (at a higher level) to enhance the representation of feature maps, thereby improving the accuracy of text detection/segmentation in the subsequent step. The specific steps of the contextual attention model are described in Figure 6. The Contextual Attention Block computes attention matrices from horizontal or vertical features and integrates contextual information through matrix multiplication between attention matrices and the original features. The transformations used in the contextual attention block include:

Transpose operation of the feature map	T
Reshape the feature map	RS
Multiply two feature maps	⊗
Concatenate two feature maps	⊕

To reduce the computational cost, the contextual attention mechanism here only considers the similarity of each position in the feature map with other positions in the same column (vertical context) or row (horizontal context). The flowchart shown in Figure 6 demonstrates that the contextual attention block is applied twice consecutively to capture both near and far dependencies for each pixel. The specific process is as follows: Starting from the input feature map F (obtained from the fusion model), with a size of N×H×W×C, where N, H, W, and C represent the number of input feature maps, height, width, and the number of channels in each feature map (specifically, N = 128, H = 192, W = 256, C = 128), the steps for collecting horizontal contextual information include

- B1 Perform 3 parallel 1×1 convolution layers (CONV1, CONV2, CONV3) to obtain 3 feature maps,F_1, F_2 and F_3
- B2. Transpose F_1 and reshape the result to size $N \times H \times W \times C$.
- B3. Multiply F_2 with the transposed feature map obtained in B2 and reshape the result to size $N \times H \times W \times W$
- B4 Multiply F_3 with the product feature map obtained in B3 and reshape the result to size $N \times H \times W \times C$. The collection of vertical contextual information is performed based on the transposed feature maps. Specifically, three parallel convolution layers (CONV1T, CONV2T, CONV3T) are executed to obtain three corresponding transposed feature maps (F_1^T, F_2^T, F_3^T) with a shape of $N \times W \times H \times C$. The specific steps are as follows:
- B5. Transpose F_3^T and reshape the result to size $N \times W \times C \times H$.
- B6 Multiply F_2^T with the transposed feature map obtained in B5 and reshape the result to size N×W×H ×H.
- B7 Multiply F_1^T with the product feature map obtained in B6 and reshape the result to size $N \times W \times H \times C$.

The shortcut path is utilized to preserve local features. The final step involves concatenating the vertical contextual feature maps, horizontal contextual feature maps, and the short/cut path feature maps to obtain the output feature map F_1 and reduce the number of output channels using a 1×1 convolution layer.

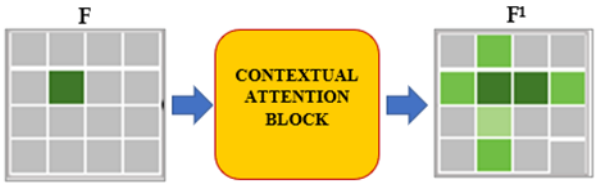


Figure 7. Apply the contextual attention on the feature map F

This mechanism allows the contextual attention block to aggregate contextual information in the vertical and horizontal directions for each pixel. The result of applying the first round of contextual attention on the input feature map F is depicted in Figure 7.
Since the convolution layers (CONV1, CONV2, CONV3) and $(CONV1^T, CONV2^T, CONV3^T)$ share weights, continuing to apply the contextual attention block on the feature map F1 will enable each pixel to capture more distant dependencies from all other pixels. Figure 8 shows that after applying the contextual attention block for the second time, the long-range dependencies for each pixel have been enhanced. In other words, the feature map F^2 is richer in contextual information than to F^1.

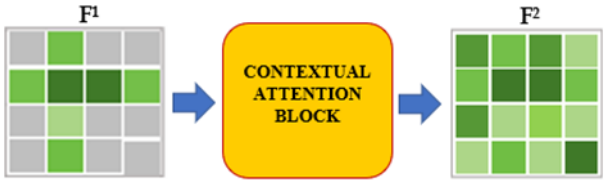


Figure 8. Apply the contextual attention on the feature map F^1

This processing mechanism also helps alleviate issues arising from limited receptive capabilities when handling more challenging text, such as longer text. Figure 9 illustrates the result of applying the contextual attention block to a specific input image. The feature map F^2 from

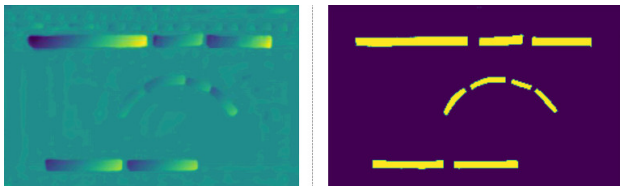


Figure 9. Appling the contextual attention model

the contextual attention will be further used to predict high level features (knowledge) of text regions to enhance the accuracy of the text segmentation algorithm. In this approach, four main types of high-level features are of particular interest: the centerline of text regions, boundary points, corner points of the text polygon, and the offset of points on the centerline from the boundary points.

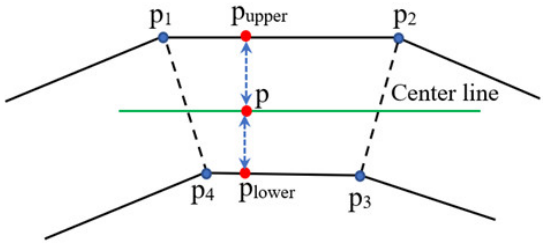


Figure 10. Predict centerline feature map

The extraction of these high level features is handled by four parallel blocks within a Fully Convolutional Network (FCN), as described at the end of Figure 6. Each block consists of four 1×1 convolution layers.The output of each block corresponds to a specific type of high-level feature map that needs to be determined. Specifically, the first convolution block allows for the prediction of the feature map representing the center line of text regions (C) - Figure 10. The second convolution block allows predicting the feature map TBO_ Map (TBO Figure 6). This map defines the upper and lower boundary points (p_upper, p_lower) for each point p in the centerline feature map C and estimates the offset between them (Figure 11). The third convolution

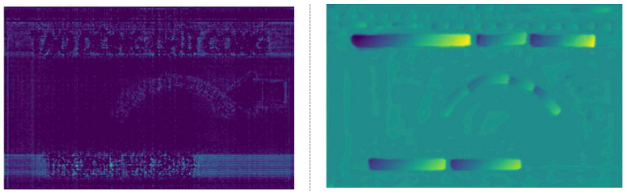


Figure 11. The offset of p and the pair of points (p_upper, p_lower)

block allows predicting the FBO_Map (FBO Figure 6). This map identifies the four corner points of each text region (e.g., four vertices p1, p2, p3, p4 in Figure 11) and estimates the offset of these four points to the pixels in the boundary map C. The fourth convolution block allows predicting the TCO_Map (TCO), determining the offset between the pixels in the centerline map C and the center of the text region's bounding box.

4. Text instance segmentation

Text instance segmentation is the final processing step in the text detection pipeline. The input to this step consists of 128 feature maps (of size $H\times W\times C$, where H=192, W=256, C=128) from the F^2 set obtained from the contextual attention model in the previous stage.
This involves determining the positions of text regions, represented by their enclosing polygons, in the input image. Building upon ideas from [16], [28], [34], [35], the text instance segmentation algorithm is executed as follows:

- B1. Identify the center line of the text region.
- B2. Sample center points along the center line.
- B3. Extract the boundary points of the text region.
- B4. Group the boundary points in a clockwise direction to form text regions.

a) Input image b) Text centerline

Figure 12. Determining the text centerline

Where the text's centerline is regarded as a scaled-down representation of the text region. Figure 12(b) shows the centerline of the input text line in Figure 12(a). Determining the text's centerline is done based on the centerline feature map C (constructed from the contextual attention model mentioned above).
The method of sampling center points is performed from left to right along the centerline, with center points taken at equal intervals (Figure 13). Based on the sampled center

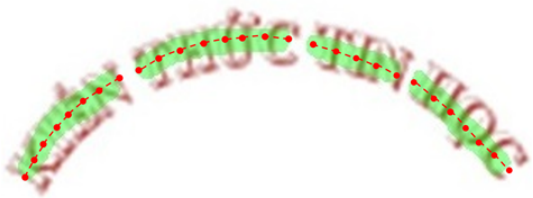


Figure 13. Sampling the center points

points, the following processing step will proceed to determine the pairs of boundary points (upi and loi) Figure 14.

Finally, by connecting the boundary points clockwise, we

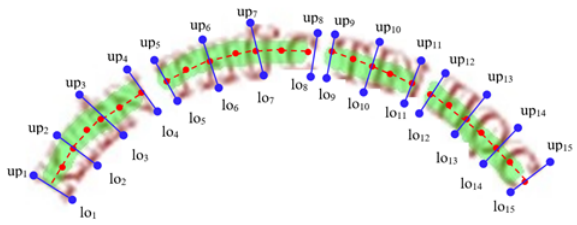


Figure 14. Determine the boundary point pairs

will obtain the complete bounding box (position and shape) of the text regions (Figure 15).

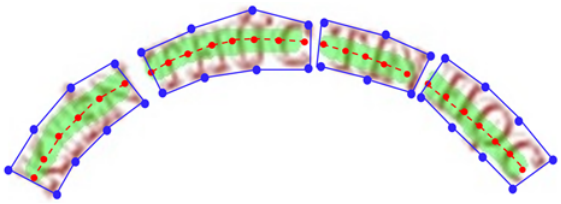


Figure 15. Determine the boundary point pairs

IV. Experimentation

1. Experimental environment

We use the Python 3 environment to implement the algorithm and experimental model. The experimental computer is configured with an Intel Core i7/9700 4.7 GHz (Max Turbo Frequency) processor, 32 GB RAM. The computer is also equipped with Nvidia Tesla K80 12 GB $\times$ 2 GDDR5 graphics card.

2. Experimental Data

Both English and Vietnamese belong to the Latin alphabet system. The Vietnamese alphabet almost encompasses the entire English alphabet. In reality, Vietnamese foreign context documents often contain many English terminologies (even half English and half Vietnamese, for example, on signs, directions, advertisements, etc.). Therefore, to evaluate the method's effectiveness, the authors conducted experiments on both Vietnamese and English datasets. Specifically, four experimental datasets were used, including:

- ICDAR 2015. This dataset was collected for the ICDAR 2015 text recognition competition [36], including 1000 scene text images for training and 500 for testing. The text images were collected using Google Glass, and the text in the images is very diverse and random (Figure 16). This image set has been annotated at the word level.

Figure 16. ICDAR 2015 dataset

- Total-Text. The Total-Text dataset [37] is an outdoor scene image dataset containing curved text with various orientations (multi-oriented). This dataset comprises a total of 1255 training images and 300 testing images (Figure 17). All images in this dataset have been labeled at the word level.

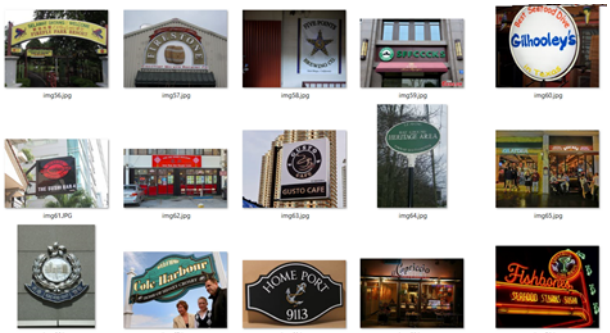


Figure 17. Total-Text dataset

- VinText. The VinText Vietnamese scene text dataset was collected by the VinAI Institute [31], comprising a total of 1200 training images and 300 testing images (Figure 18).

Figure 18. VinText dataset

The images in this dataset were collected in the Hanoi region and its surrounding areas, featuring diverse content,

including images of advertisement signs, street names, shop names, offices, text on various modes of transportation, and more. All images in this dataset have been labeled at the word level.

- VNSceneText. This is a dataset of scene text images collected directly by the authors in the streets of Ba Ria / Vung Tau and Ho Chi Minh City under completely natural conditions using smart phone devices (Iphone 7, Iphone 12, Oppo Reno 8). The dataset includes 3000 images, with 2400 images for training and 600 images for testing. The text in this dataset is very diverse in terms of types (images taken from advertising signs, traffic signs, street names, signs on buildings, vehicles, and images of various other types of documents).

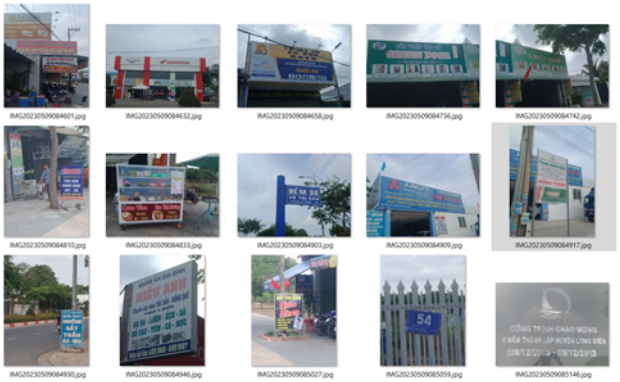
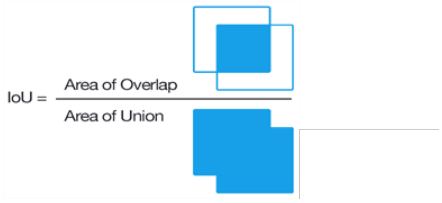


Figure 19. VNSceneText dataset

3. Evaluation Metrics

In the context of object detection in general and text detection in scene images, IoU (Intersection over Union) is commonly used to determine the accuracy of an object or text region detection. IoU is defined as the area of intersec-



tion between the detected text region and the corresponding ground truth text region (stored in the ground truth file) over the area of their union. A text region is considered detected correctly if its corresponding IoU (Intersection over Union) score is not less than a predefined threshold Γ. To assess the effectiveness of the method, we use the Precision, Recall, and H_mean metrics:

- Precision is the ratio of the number of correctly detected text regions to the total number of detected text regions: $Precsion = \frac{T_P}{T_P+F_P}$

- Recall is the ratio of the number of correctly detected text regions to the total number of actual text regions: $Recall = \frac{T_P}{T_P+F_N}$

- H_mean is the harmonic mean between the precision and recall of the detection results: Where T_P (True Positive) is the number of correctly detected text regions in the image, F_P (False Positive) is the number of text regions falsely detected (not actual text but detected as text), and F_N is the text regions that were not detected (missed).

$$H_mean = \frac{2xPrecisionxRecall}{Precision+Recall}$$

To evaluate the runtime performance of the algorithms, we use the FPS (Frames Per Second) metric, defined as the number of image frames processed in one second

4. Experimental Results

The experimental process was conducted sequentially on each dataset. For each dataset, we first trained the network using the training data. To ensure the stability of the model, from the training data of each model, we applied data augmentation techniques such as changing the image scale (random-scale), rotating the image (random-rotate), blurring the image (random-blur), adjusting image contrast (random-contrast), and flipping the image. For each image in the training set, we typically generated an additional 10-15 images for training purposes.

During the training process, the four components of the model (backbone network, aggregation model, contextual attention model, and text instance segmentation) are trained sequentially. In particular, the backbone network is trained using transfer learning with pre /trained weights from ImageNet [38]. The fundamental parameters for training the network models are set as follows: Epoch: 5000; Batch size (per GPU): 16; Num-workers: 4; Optimizer: Adam, beta1: 0.9, beta2: 0.999; Learning-rate: 0.001; Regularizer: L2 normalization.

To obtain visually informative evaluation results, we compared the performance of the proposed method with the EAST [16], SAST [28], FCENet [29], and DB++ [30] methods, as mentioned above. Specifically, for the ICDAR 2015 and Total-Text datasets, we compared the experimental results of the proposed method with the results published in [16], [28], [29], [30]. For the VinText and VNSceneText datasets, to facilitate a fair comparison, we first trained EAST, SAST, FCENet, and DB++ models directly on the VinText training and VNSceneText training datasets using transfer learning with pre-trained weights from [37]. Subsequently, we evaluated and compared the performance of these methods on the VinText and VNSceneText testing datasets.

Additionally, to assess the effectiveness of the aggregation model, we conducted experiments and compared the model's performance with the case where the fusion model is not used (denoted as 'Not Fusion'). Specifically, in 'Not Fusion,' the input images are passed through the backbone network for feature extraction, the contextual attention mechanism is applied directly to the feature maps obtained from the backbone, and text instance segmentation is performed. The experimental results of the methods on the ICDAR 2015 dataset are presented in detail in Table I1.

TABLE I
THE EXPERIMENTAL RESULTS ON THE ICDAR 2015 DATASET

Method	Precision	Recall	H_mean	FPS
EAST [16]	78.33	83.27	80.72	13.2
SAT [28]	87.09	86.72	86.91	–
FECNet[29]	84.02	85.01	84.60	-
DB++ [30]	90.90	83.90	87.30	10
Not Fusion	84.07	85.23	85.12	6.3
VNSTD	87.53	86.94	87.23	8.4

From the experimental results, it is evident that the fusion model not only significantly enhances the accuracy of text detection, especially for arbitrarily shaped and sized text, but also optimizes the processing time for the contextual attention model.

Figure 20. Some results on the ICDAR 2015 dataset

Figure 20 illustrates some results of the proposed method on the ICDAR 2015 dataset. This dataset focuses on evaluating the adaptability of algorithms to real-world images captured in urban environments with natural lighting and diversity: indoor and outdoor images, close-up and distant shots, images with single or multiple text lines, and text in arbitrary shapes. The experimental results of the methods on the Total-Text dataset are presented in detail in Table II.

TABLE II
THE EXPERIMENTAL RESULTS ON THE TOTAL-TEXT DATASET

Method	Precision	Recall	H_mean	FPS
EAST [16]	–	–	-	–
SAT [28]	76.86	83.77	80.17	–
FECNet[29]	79.80	87.40	83.40	–
DB++ [30]	88.90	83.20	86.00	28.0
Not Fusion	77.26	84.02	80.34	20.5
VNSTD	84.32	88.17	86.20	26.6

Figure 21 shows some results of the algorithm on the Total-Text dataset. This dataset is focused on evaluate the algorithms' adaptability to diverse text images, including short, long, curved, vertical, horizontal, and various other text types.

Figure 21. Some results on the Total-Text dataset

The experimental results show that the DB++ algorithm and the proposed VNSTD algorithm achieve higher accuracy than the others. In the case of not using the fusion

model, the model's accuracy significantly decreases. This indicates that fusion and representing features at various scales can effectively handle text with different shapes and sizes.

The experimental results of the methods on the VinText dataset are presented in detail in Table III.

TABLE III
THE EXPERIMENTAL RESULTS ON THE VINTEXT DATASET

Method	Precision	Recall	H_mean	FPS
EAST [16]	71.24	73.38	72.30	11
SAT [28]	83.12	85.03	84.06	8.5
FECNet[29]	83.23	84.57	83.89	9.2
DB++ [30]	84.05	83.90	83.97	8.3
Not Fusion	83.26	84.04	83.32	6.2
VNSTD	85.63	87.94	86.77	7.9

Some results of the proposed method on the VinText dataset are presented in Figure 22.

Figure 22. Some results on the VinText dataset

TABLE IV
THE EXPERIMENTAL RESULTS ON THE VNSCENETEXT DATASET

Method	Precision	Recall	H_mean	FPS
EAST [16]	70.73	72.16	71.44	10.03
SAT [28]	83.25	84.17	83.70	10.60
FECNet[29]	81.23	82.68	81.95	11.50
DB++ [30]	84.05	81.02	82.50	9.30
Not Fusion	82.23	83.40	82.27	7.60
VNSTD	85.14	87.23	86.17	8.80

Figure 23 displays some results of the proposed method on the VNSceneText dataset.

Figure 23. Some results on the VNSceneText dataset

V. CONCLUSION

In this paper, we propose an efficient approach to address the problem of Vietnamese text detection in scene images using deep learning. The proposed method is based on the idea of utilizing deep neural network architectures to learn various geometric attributes to reconstruct the polygonal representation of text regions.

The proposed network architecture consists of four main components: (1) The backbone network (resnet-50) for feature extraction from the input image; (2) The fusion model, constructed as a Feature Pyramid Network (FPN) to integrate and represent features at multiple scales, aiding in the detection of text regions of various sizes; (3) The contextual attention model, structured as a deep convolutional network, trained to capture contextual information (both short-range and long-range dependencies) at each pixel to obtain more representative features; (4) The text instance segmentation model, implemented using semantic segmentation, is responsible for determining the positions

and bounding boxes of all text regions in the input image based on the detection of center lines, center points, and boundary points for grouping them into corresponding text regions.

Notably, during the clustering process in step 4, the algorithm integrates low-level features (pixel-level) with high-level object knowledge to adapt and effectively address complex text shapes (curved, slanted, irregular font sizes). Experimental results show that this is a feasible approach to tackle the problem of Vietnamese text detection in outdoor images.

ACKNOWLEDGEMENT

We would like to sincerely thank the Basic Research and Technology Development Task 'Research on Improving Image Retrieval Efficiency with Relevance Feedback Using Graph Convolutional Neural Network', project code: CSCL02.05/23-24;3, for their support during this research.

REFERENCES

[1] Zobeir Raisi, Mohamed A. Naiel, Paul Fieguth, Steven Wardell, John Zelek, "Text Detection and Recognition in the Wild: A Review", https://doi.org/10.48550/arXiv.2006.04305.

[2] S. Long, X. He, and C. Yao. Scene text detection and recognition: The deep learning era. *Int. J. Comput. Vision*, pp.1–24, 2020.

[3] S. M. Hanif and L. Prevost, "Text detection and localization in complex scene images using constrained adaboost algorithm," *in Proc. Int. Conf. on Doc. Anal. and Recognit*, pp 1-5, 2009.

[4] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," *in Proc. Int. Conf. on Comp. Vision*, pp.1457–1464, 2001.

[5] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven,"PhotoOCR: Reading text in uncontrolled conditions," *in Proc. IEEE Int. Conf. on Comp. Vision*, pp. 785–792, 2013.

[6] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan, "Text flow: A unified text detection system in natural scene images," *in Proc. IEEE Int. Conf. on Comp. Vision*, pp.4651–4659, 2015.

[7] H. Cho, M. Sung, and B. Jun, "Canny text detector: Fast and robust scene text localization algorithm," *in Proc. IEEE Conf. on Comp. Vision and Pattern Recognit*, pp.3566–3573, 2016

[8] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," *in Proc. IEEE Conf. on Comp. Vision and Pattern Recognit*, pp.2550–2558, 2017.

[9] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Comp. Sci*, vol. 10, no. 1, pp.19–36, 2016.

[10] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. "Textboxes: A fast text detector with a single deep neural network", *In AAAI Conf. on Artificial Intelligence*, 2017

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed. "SSD: single shot multibox detector", *In European Conf. Comput. Vision*, 2016.

[12] M. Liao, B. Shi, and X. Bai. "Textboxes++: A single-shot oriented scene text detector", *IEEE Trans. Image Processing*, 27(8), pp.3676–3690, 2018.

[13] Y. Liu and L. Jin. "Deep matching prior network: Toward tighter multi-oriented text detection", *In Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[14] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li " Single shot text detector with regional attention", *In Proc. Int. Conf. Comput. Vision*, pp.3047–3055, 2017.

[15] M. Liao, Z. Zhu, B. Shi, G. Xia, and X. Bai. "Rotation-sensitive regression for oriented scene text detection", *In Proc. Conf. Comput. Vision Pattern Recognition*, pp.5909–5918, 2018.

[16] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. "EAST: an efficient and accurate scene text detector", *In Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[17] W. He, X. Zhang, F. Yin, and C. Liu." Deep direct regression for multi-oriented scene text detection", *In Proc. Int. Conf. Comput. Vision*, 2017.

[18] L. Xie, Y. Liu, L. Jin, and Z. Xie. "Derpn: Taking a further step toward more general object detection", *In AAAI Conf. on Artificial Intelligence*, volume 33, pp.9046–9053, 2019.

[19] B. Shi, X. Bai, and S. J. Belongie." Detecting oriented text in natural images by linking segments", *In Proc. Conf. Comput. Vision Pattern Recognition*, 2017.

[20] J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, and X. Bai. "Seglink++: Detecting dense and arbitrary-shaped scene text by instanceaware component grouping", *Pattern recognition*, 2019.

[21] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. "Multioriented text detection with fully convolutional networks", *In Proc. Conf. Comput. Vision Pattern Recognition*, 2016.

[22] C. Xue, S. Lu, and F. Zhan. "Accurate scene text detection through border semantics awareness and bootstrapping", *In European Conf. Comput. Vision*, pp. 355–372, 2018.

[23] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai. "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes", *IEEE Trans. Pattern Anal. Mach. Intell*, 2019.

[24] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes", *In European Conf. Comput. Vision*, pp. 67–83, 2018.

[25] K. He, G. Gkioxari, P. Dollar, and R. Girshick." Mask R-CNN", *In Proc. Int. Conf. Comput. Vision*, pp. 2961–2969, 2017.

[26] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao. "Shape robust text detection with progressive scale expansion network", *In Proc. Conf. Comput. Vision Pattern Recognition*, pp. 9336–9345, 2019.

[27] Z. Tian, M. Shu, P. Lyu, R. Li, C. Zhou, X. Shen, and J. Jia. "Learning shape-aware embedding for scene text detection", *In Proc. Conf. Comput. Vision Pattern Recognition*, pp. 4234–4243, 2019.

[28] Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi " A Single-Shot Arbitrarily-Shaped Text Detector based on Context Attended Multi-Task Learning", *In Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, 2019.

[29] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, Wayne Zhang, "Fourier Contour Embedding for Arbitrary-Shaped Text Detection", *CVPR*, 2021.

[30] M. Liao, Z. Zou, Z. Wan, C. Yao and X. Bai, "Real-Time Scene Text Detection With Differentiable Binarization and Adaptive Scale Fusion", *in IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 45, no. 01, pp. 919-931, 2023.

[31] N. Nguyen et al., "Dictionary-guided Scene Text Recogni-

tion," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA*, pp. 7379-7388, 2021.

[32] N. T. Pham, V. D. Pham, Q. Nguyen-Van, B. H. Nguyen, D. N. Minh Dang and S. D. Nguyen, "Vietnamese Scene Text Detection and Recognition using Deep Learning: An Empirical Study," *6th International Conference on Green Technology and Sustainable Development (GTSD), Nha Trang City, Vietnam*, pp. 213-218, 2022.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need", *In Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 5998–6008, 2017.

[34] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes", *In Eur. Conf. Comp. Vis. (ECCV)*, pp. 20–36, 2018.

[35] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao, "Shape Robust Text Detection With Progressive Scale Expansion Network", *In IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR)*, pp. 9336–9345, 2019.

[36] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al, "ICDAR 2015 competition on robust reading", *In Int. Conf. Doc. Anal. Recognit. (ICDAR). IEEE*, pp. 1156–1160, 2015.

[37] Chee Kheng Ch'ng and Chee Seng Chan, "Total-Text: A comprehensive dataset for scene text detection and recognition", *In Int. Conf. Doc. Anal. Recognit. (ICDAR)*, Vol. 1, pp. 935–942, 2015.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database", *In IEEE Conf. Comp. Vis. Patt. Recognit. (CVPR). IEEE*, pp. 248–255, 2009.

Huynh Van Huy graduated from Hue University with a Bachelor's degree in Computer Science in 2005 and obtained a Master's degree in Information Technology from Lac Hong University in 2012. Currently, the author is a Ph.D. student in Computer Science at Lac Hong University.
Email: huynhvanhuy@gmail.com

Nguyen Thi Thanh Tan obtained her Bachelor's and Master's degrees in Computer Science and Information Technology from the Hanoi University of Science and Technology in 1999 and 2001, respectively. She earned her Ph.D. in Computer Science from the Vietnam Academy of Science and Technology. Currently, the author is working at the Electric Power University. Her primary research interests include Artificial Intelligence, Computer Vision, and Big Data Analysis. The author has published over 40 articles in reputable national and international journals.
Email: tanntt@epu.edu.vn

Ngo Quoc Tao obtained his Bachelor's degree in "Mathematical Assurance for Computers and Computing Systems" in 1997, an Associate Professor in Computer Science in 2002, and a Senior Researcher in 2018. Now, he is working at the Institute of Information Technology, Vietnam Academy of Science and Technology. He is conducting research in the fields of Pattern Recognition, Image Processing, Knowledge Technology, and Artificial Intelligence.
Email: nqtao@ioit.ac.vn