Proposing Vietnamese Text Recognition Algorithm Combining CRAFT and VietOCR

Phat Nguyen Huu*, Thanh Tran Ngoc*, and Quang Tran Minh†,‡
*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam
†Department of Information Systems, Faculty of Computer Science and Engineering, HCMUT, Ho Chi Minh City, Vietnam
‡Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh City, Vietnam
Email: phat.nguyenhuu@hust.edu.vn; thanh.tn182791@sis.hust.edu.vn; quangtran@hcmut.edu.vn

2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS) | 978-1-6654-5248-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCAIS56082.2022.9990349

***Abstract*—Letter recognition based on artificial neural networks is currently achieving promising results. In this paper, we propose a method that combines character-region awareness for text detection (CRAFT) and VietOCR to recognize letters and Vietnamese texts effectively based on finding characters in words and affinity among them. The algorithm uses the tool to crop each character to put it into VietOCR to recognize letters. The results show that the algorithm achieves an accuracy of 89.84% with 10 epochs. We find that the proposed method has accuracy and execution time depending on the variety of input data based on the obtained results.**

***Index Terms*—CRAFT, VietOCR, long short-term memory, image-to-text, image processing.**

I. Introduction

Handwriting recognition has attracted much attention in the field of articial intelligence (AI) because of its many applications such as instant translation, image retrieval, scene parsing, and geolocation. Recently, scene text detectors based on deep learning (DL) have shown a lot of promise. The core method is to train the network to recognize the bounding box word level. The disadvantage is that it is difficult to recognize the folded box or the text is too long. In addition, the character level bounding box has many advantages when concatenating and recognizing them in the top-down direction. However, most datasets currently do not provide character-level annotation, and the workload to get the level ground truth is extremely expensive.

A transformer is a recognition model that is extremely popular today and is applied to handle many problems related to translation or word recognition. VietOCR is a library created based on a transformer structure that achieves good accuracy and is suitable for many types of data.

Our paper has three main points. Firstly, we propose to combine two methods character-region awareness for text detection (CRAFT) and VietOCR to recognize letters and handwriting. Second, we have built a dataset suitable for the Vietnamese language. Third, we also built an online Vietnamese recognition program with an accuracy of up to 89.94%.

The rest of the paper includes four parts and is organized as follows. Section I will discuss the related work. In Section II, we present the proposal system. Section III will evaluate the proposed system and analyze the results. In the final section, we give conclusions and future research directions.

II. Related work

Character level text detector [1]–[6]: This method often uses text blocks filtered by maximally stable extremal regions (MSER) [4], [6] to detect the level text. However, if we use MSER to identify individual characters, it is limited the detection for certain situations such as scenes with low contrast, curvature, and light reflections. The authors [2] used map prediction of characters along with text regions and associative orientations to request caption character levels. Therefore, seglink searches for text grids and associates these segments with a link prediction instead of an explicit character-level prediction.

Although mask textspotter predicts a character-level probability map, it is used for text recognition instead of detecting individual characters. This method is inspired by the idea of words which uses a weakly-supervised framework to train the character level detector. However, one drawback of words is that the character representation is formed in a set of rectangular anchors. Therefore, it makes distortion when the character perspective is produced by different camera angles. Furthermore, it is constrained by the performance of the backbone structure since a single shot detector (SSD) is limited by the number of anchor boxes and their size.

Segmentation-based text detectors [7]–[11]: An approach aimed at finding text regions at the pixel level based on segmentation jobs. These approaches detect text by estimating word-bound regions such as multi-scale fully convolutional neural network (FCN) [7]. Comprehensive prediction and pixel link [8] have also been suggested using segmentation. Single-shot text detector (SSTD) [9] attempted to benefit from both regression and segmentation approaches by using an attention mechanism to enhance the area associated with the text through object-level background noise reduction. Recently, text snake [10] has been proposed to detect text instances by predicting text area and center-line along with geometry properties.

End-to-end detector [12]–[15]: An end-to-end approach trains detection and recognition modules simultaneously to improve detection accuracy by leveraging results. Fast-oriented text spotting (FOTS) [12] and explicit alignment and attention (EAA) [13] combine common detection and recognition methods and train them in an end-to-end manner. Mask textspotter

978-1-6654-5248-9/22/$31.00 ©2022 IEEE

[14] took advantage of their merging model to treat the recognition task as a semantic segmentation problem. The train module recognition helps the text detector to handle text-like backgrounds well. Most methods detect text with word level. However, determining ranges for a word to detect is not straightforward as words can be separated by various criteria such as meaning, space, or color. In addition, word segmentation boundaries cannot be rigorously defined. Therefore, the word segment has no distinct semantic meaning. The uncertainty in the word annotation reduces the meaning of the ground truth for both regression and segmentation approaches.

Convolutional recurrent neural network (CRNN) [16]–[22]: The network architecture of CRNN consists of three components, namely convolution, recurrent, and bottom-up transcription layers. At the end of the CRNN, the convolutional layers automatically extract a feature sequence from each input image. On top of the convolutional neural networks (CNN), a recurrent neural network (RNN) is built to predict each frame of the sequence feature which is output by the convolutional layers. The transcription layer at the top of the CRNN is used to translate the per-frame predictions by recurrent layers into a sequence of labels. Although CRNN covers various types of network architectures, it can be trained with a loss function.

III. Proposal system

A. Overview

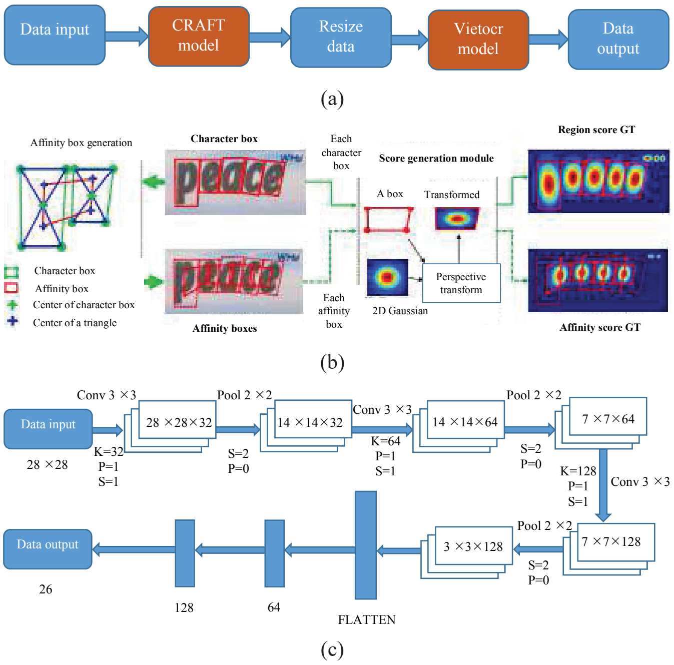
Figure 1 is an overview of the proposed recognition system based on [23]. First, the input data goes through the CRAFT model. The model then detects the frames of the words and extracts the positions on the image. The positions are processed through the image cropping algorithm. The system will continue to process it through VietOCR which is a model used to recognize words after having an image of each word.

B. Details of proposed system

1) CRAFT: CRAFT uses a VGG-16 network backbone with batch normalization. The model of using skip connections at decoding is similar to U-net in collecting low-level features. The output of the network is two channels, namely a region and an affinity score. The region score represents the probability that the input pixel is the center of the letter and the affinity score represents the probability of a space between two adjacent letters. CRAFT uses a Gaussian heat map to train both output channels.

To ensure integrity, the 2D Gaussian image is calculated to bring the correct angle of view. The region score of the output image is calculated by converting the input data to a Gaussian heat-map image. The affinity score alone is calculated by drawing two opposite triangles that pass through the centroid of each character box. It then connects the four centroids of the triangles between two adjacent boxes and performs a Gaussian heat map.

In this step, we generate the region and affinity score using the Gaussian heat map. Figure 1(b) depicts the process of creating a ground-truth label for synthetic data. It can recognize long and large sentences effectively despite using small receptive fields. On the other hand, other approaches such as box regression need a large receptive field in a particular case.



(a)

(b)

(c)

Fig. 1. a) Diagram of the proposed model, b) process of creating a ground-truth label, and c) proposed network.

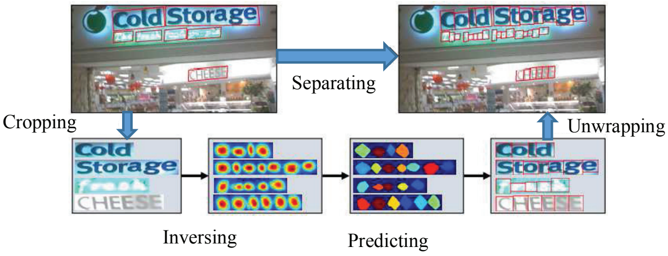


Fig. 2. The process of annotating character level [23].

For real data, we perform annotation on each word, crop it, and use the model to guess their character to create a character-level box. The whole process is performed as shown in Fig. 2. First, the word level is cut from the original image. The model then predicts the region score. Next, we use the watershed algorithm to separate the text area. Finally, we use the inverse transform from the cropping step.

If the model is trained with incorrect region scores, the output will be wrong in this method. To avoid the problem, we must evaluate the quality of each ground truth. For the sample work level w of the training data, $R(w)$ and $l(w)$ will be the performed bounding box region and the length, respectively. Through the character separation process, it is possible to predict the bounding boxes and the length of each character $l^c(w)$ and it is then calculated the confidence score

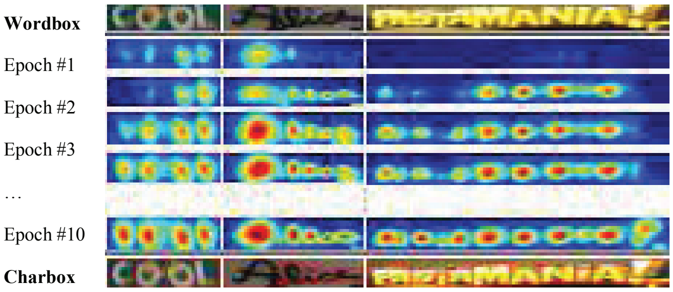


Fig. 3. Result of training process over 10 epochs [23].

as

$$s_{conf}(w)=\frac{(l(w)-min(l(w),|l(w)-l^{c}(w)|)}{l(w)}, \qquad (1)$$

and the pixel-wise map is

$$S_c\left(p\right)=\begin{cases} S_{conf}\left(w\right),\ if\ p\in R\left(w\right) \\ 1,\quad otherwise \end{cases}, \qquad (2)$$

where p is pixel of $R(w)$ by

$$Loss\left(L\right)=\sum_{p}S_c\left(p\right).(||S_r\left(p\right)-S_r^{*}\left(p\right)|\,|_2^2 \\ +||S_a\left(p\right)-S_a^{*}\left(p\right)||_2^2, \qquad (3)$$

where $S_r^*(p)$ and $S_a^*\left(p\right)$ are ground-truth region score and affinity score, respectively. $S_r\left(p\right)$ and $S_a(p)$ are region score and affinity score predicted, respectively. When it is trained, the model will predict characters more and more accurately leading to an increase in the confidence score $S_{conf}(w)$. Figure 3 shows the region score during training.

After training the model, the images are detected. They are then passed to the model whose output is the character region score represented by pixel coordinates. We perform to cut each character and then pass through the VietOCR model to perform identification.

2) VietOCR: The most famous of models is bidirectional encoder representations from transformers (BERT) [24], [25]. The model uses to learn the best representations of words. It has created a major turning point for natural language processing (NLP). Google has also adopted BERT for finding data. The main idea of the transformer is still to apply the attention body on a more complex level.

The general architecture of the transformer model consists of two main parts similar to other machine translation models, namely encoder and decoder. An encoder is used to learn the expression vector of a sentence with the expectation that this vector carries the perfect information. A decoder performs the function of converting the other representation vector into the target language.

One of the advantages of the transformer is that this model is capable of parallel processing of words. Encoders of the transformer model are a type of feed-forward neural network consisting of many different encoder layers. Each encoder layer will process words simultaneously. On the other hand, words must be processed sequentially in the long short-term memory (LSTM) model. In addition, the transformer model also processes the input sentence in two directions without having to stack an additional LSTM as in the bidirectional LSTM architecture [26]–[28].

The transformer encoder of the model can consist of many similar encoder layers. Each encoder layer consists of two main components, namely multi-head attention and feed-forward network. In addition, it has both a skip connection and a normalization layer.

A decoder performs the function of decoding the source sentence vector into the target sentence. Therefore, the decoder will receive information from the encoder as two vector keys and values. The architecture of a decoder is very similar to that of an encoder. Besides, it has multi-head attention in the middle used to learn the relationship between the translated and original words.

Similar to many other models, a fully connected layer is needed to convert the output from the previous layer into a matrix with dimensions equal to the number of words to be predicted. Softmax then calculates the probability of the appearing word. The loss function is cross-entropy.

VietOCR is a Vietnamese language recognition library using the transformerOCR model. The model consists of two main parts, namely encoder and the decoder. An encoder is used to learn the representation vector of a sentence and this vector carries the perfect information. A decoder performs the function of converting the other representation vector into the target language.

VietOCR consists of two main steps, embedding layer with position encoding and sinusoidal position encoding.

1) Embedding layer with position encoding

   Position encoding is used to put information about the position of words into the transformer model. First, words are represented by a vector using a word embedding matrix with several lines equal to the size of the vocabulary set. The words are then found in this matrix and concatenated into rows of a 2D-dimensional matrix containing the semantics of each word. Since the transformer processes words in parallel, the model cannot know the position of words with just word embedding. Therefore, several mechanisms are needed to put the word position information into the input vector.

   The basic way can be used as follows. The position of words is represented by a sequence of consecutive numbers from $0, 1, 2, 3, ..., n$. However, the problem is that when the sequence is long, the number can be quite large and the model will have difficulty predicting sentences. To solve the problem, we can renormalize this sequence of numbers between 0 and 1 by dividing by n. However, the distance between two consecutive words will depend on the length of the string. In a fixed distance, we will not be able to calculate how many words it contains. This means that the meaning of position encoding will be different depending on the length of the sentence.

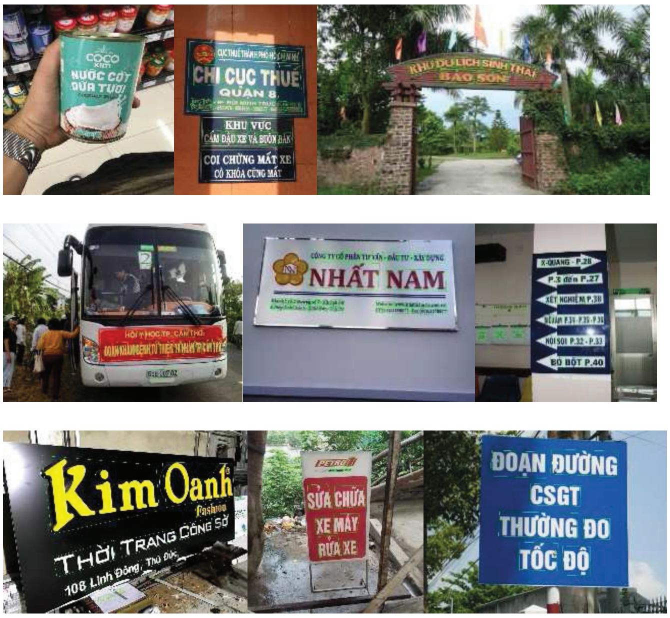


Fig. 4. An example of the dataset.

2) Sinusoidal position encoding
The positions of the words are encoded with an embedding-sized vector and added directly to it. Specifically, we use the sine function at the even position ($i = 2k$) and the cosine function for the odd position ($i = 2k+1$) to calculate the value at that dimension as follows:

$$p_t^i = f(t)^i = \begin{cases} \sin(w_k t) \ , & if\ i = 2k \\ \cos(w_k t) \ , & if\ i = 2k+1 \end{cases} , \qquad (4)$$

where

$$w_k = \frac{1}{10000^{\frac{2k}{d}}} , \qquad (5)$$

where d is the size of model and k is the position of word.
More details can be seen in [29]–[31].

IV. Simulation and Result

A. Dataset

In this paper, we use the dataset of VinsAI containing 3000 images [32]–[36]. Besides, we also built the dataset by ourselves. The result is as shown in Fig. 4.

The alphabetical dataset used for training is shown in Fig. 5.

Besides, the parameters of the network are shown in Tab. I.

B. Result of CRAFT

We use the synthetic dataset to train for 50,000 iterations and optimize ADAM during training. The ratio of synthetic data is 1:5 to ensure that the character area is properly separated during fine-tuning. The crops, rotations, and color variations techniques are also used. To perform the model with 10 self-created images, the accuracy result is 86.70%. The results are shown in Fig. 6.

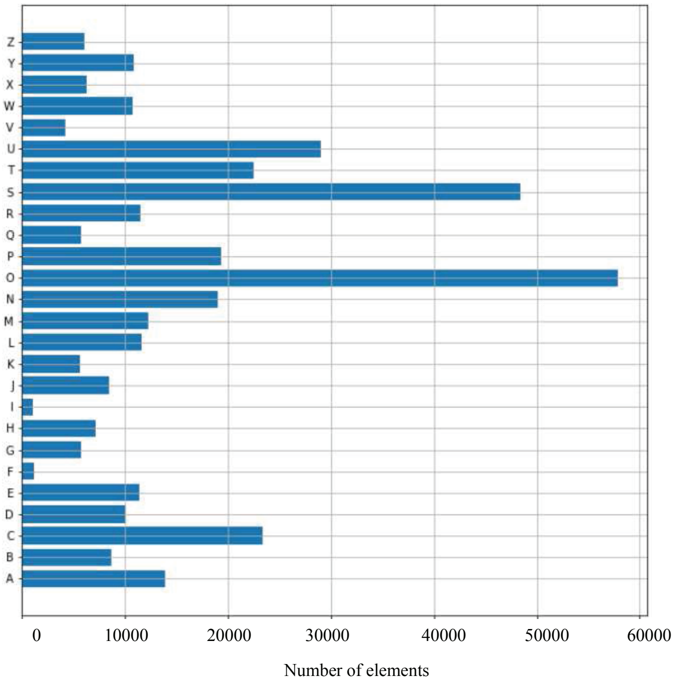
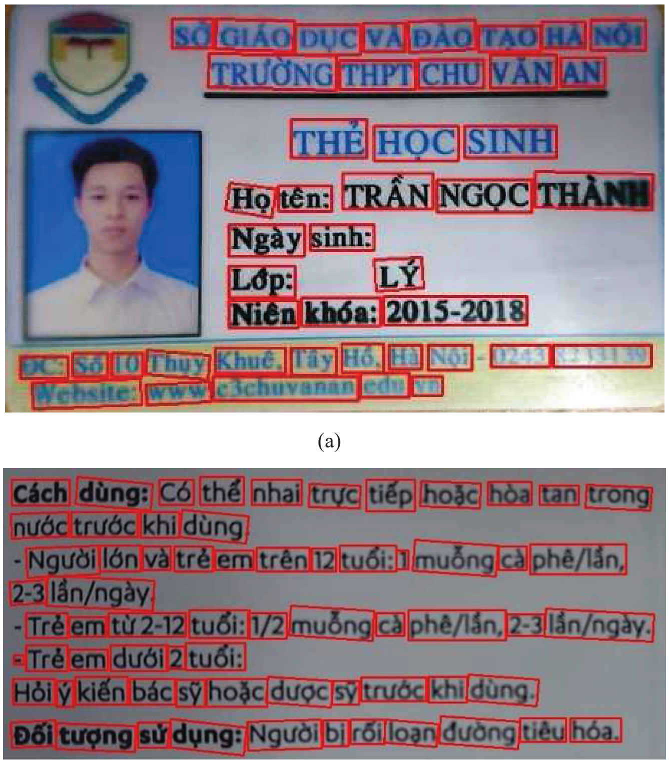


Fig. 5. Statistics on the amount of character for training.


SỞ GIÁO DỤC VÀ ĐÀO TẠO HÀ NỘI
TRƯỜNG THPT CHU VĂN AN
THẺ HỌC SINH
Họ tên: TRẦN NGỌC THÀNH
Ngày sinh:
Lớp: LÝ
Niên khóa: 2015-2018
Cách dùng: Có thể nhai trực tiếp hoặc hòa tan trong nước trước khi dùng
- Người lớn và trẻ em trên 12 tuổi: 1 muỗng cà phê/lần, 2-3 lần/ngày
- Trẻ em từ 2-12 tuổi: 1/2 muỗng cà phê/lần, 2-3 lần/ngày.
- Trẻ em dưới 2 tuổi:
Hỏi ý kiến bác sỹ hoặc dược sỹ trước khi dùng.
Đối tượng sử dụng: Người bị rối loạn đường tiêu hóa.

(a)

(b)

Fig. 6. Results of testing for CRAFT with corresponding accuracy a) **86.71%** and b) **90.94%**.

We export the “weight.pth” file to save the weight of the model after training. We build a program to create a text file that stores the pixel coordinates of the boxes of each image

TABLE I
STATISTICAL DATA IS COLLECTED FROM OUR CAMERA.

Model	Input	Output
conv2d_4 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_4 (Maxpooling2)	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 4, 4, 128)	73856
max_pooling2d_5 (Maxpooling2)	(None, 2, 2, 128)	0
flatten_1 (Flatten)	(None, 512)	0
dense_3 (Dense)	(None, 64)	32832
dropout (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 128)	8320

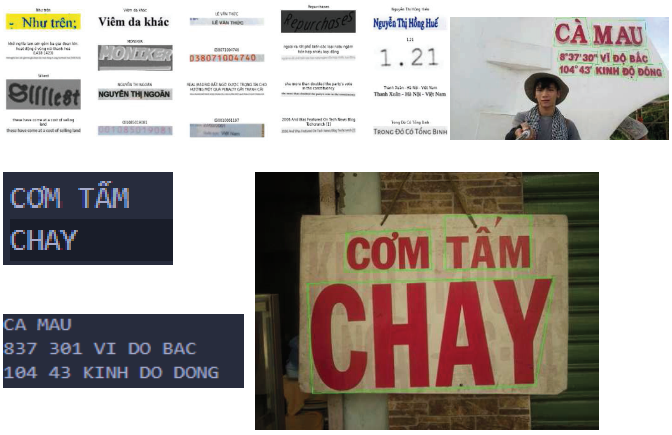


Fig. 7. Results of testing for VietOCR with accuracy as **89.84%**.

using the libraries NumPy, PIL, OS, and CV2.

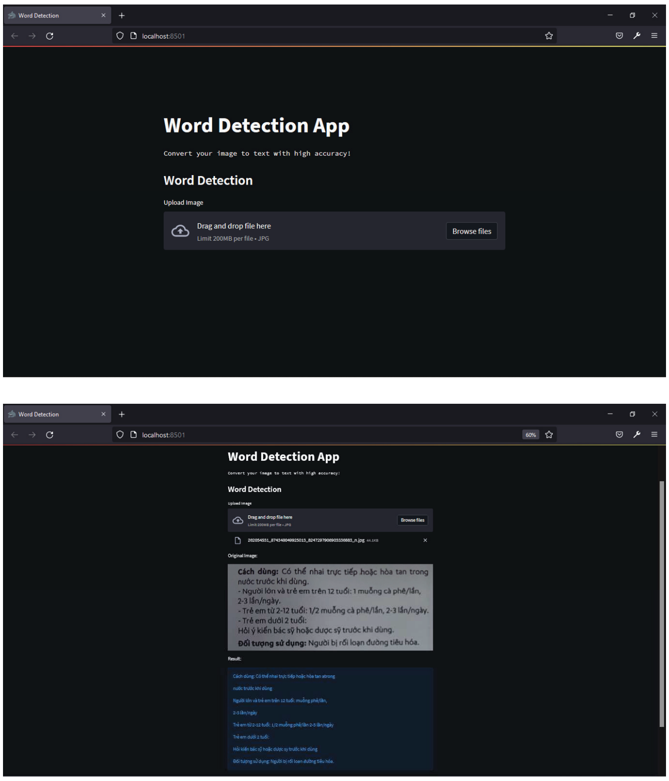
We use python with NumPy, CV2, and OS libraries. The input of the function is the output of the CRAFT. The text file is created to store the pixel coordinates of each box. First, we use CV2 to read the input image. We will next create a mask that is a dimensionless matrix img.shape" [0:2]. Finally, we use "cv2.drawContours file to create the boxes onto the mask. The output image is the CV2 object after bitwise. It is put between the original image and the mask. This is a 2-dimensional array with each dimension corresponding to the length and width to be cut.

C. Result of VietOCR

VietOCR gives very good results when the text is complete and with little noise [37]. We perform with 20 images and the accuracy is up to 89.45%. The results are shown in Fig. 7.

Besides, we also use the Streamlight framework written in python to build the system. The website system has a file import function and you can drag and drop files to them. It will display the original image and text after detection. The results are shown in Fig. 8.

We also evaluate the accuracy of the method using CRAFT combining VietOCR with the mentioned dataset. The results give an average accuracy of 89.84%.



Word Detection App

Convert your image to text with high accuracy!

Word Detection

Upload Image

Drag and drop file here
Limit 200MB per file • JPG

262054531_874348049925013_8247297906903336883_n.jpg 44.1KB

Original Image:

Cách dùng: Có thể nhai trực tiếp hoặc hòa tan trong nước trước khi dùng.
- Người lớn và trẻ em trên 12 tuổi: 1 muỗng cà phê/lần, 2-3 lần/ngày.
- Trẻ em từ 2-12 tuổi: 1/2 muỗng cà phê/lần, 2-3 lần/ngày.
- Trẻ em dưới 2 tuổi:
Hỏi ý kiến bác sỹ hoặc dược sỹ trước khi dùng.
Đối tượng sử dụng: Người bị rối loạn đường tiêu hóa.

Result:

Cách dùng: Có thể nhai trực tiếp hoặc hòa tan atrong
nước trước khi dùng
Người lớn và trẻ em trên 12 tuổi: muỗng phê/lần,
2-3 lần/ngày
Trẻ em từ 2-12 tuổi: 1/2 muỗng phê/lần 2-3 lần/ngày
Trẻ em dưới 2 tuổi:
Hỏi kiến bác sỹ hoặc dược sy trước khi dùng
Đối tượng sử dụng: Người bị rối loan đường tiêu hóa.

Fig. 8. Results when done on the online application.

V. CONCLUSION

The article proposed the method of using CRAFT combining VietOCR to recognize letters. We also built a system website to make letter recognition. The system results achieve an accuracy of up to 90%. However, the results are not optimized since the dataset is not large enough.

Therefore, we will update the Vietnamese dataset as well as enhance the preprocessing step to increase the accuracy of the proposed algorithm in the future.

ACKNOWLEDGMENT

We would like to thank Hanoi University of Science and Technology (HUST) and Ho Chi Minh City University of

Technology (HCMUT), VNU-HCM for the support of time and facilities for this study.

References

[1] Y. Baek, D. Nam, S. Park, J. Lee, S. Shin, J. Baek, C. Lee, and H. Lee, "Cleval: Character-level evaluation for text detection and recognition tasks," June 2020, pp. 2404–2412.

[2] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction." arXiv, 2016. [Online]. Available: https://arxiv.org/abs/1606.09002

[3] S. Khan, S. Thainimit, I. Kumazawa, and S. Marukatat, "Text detection and recognition on traffic panel in roadside imagery," in *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, 2017, pp. 1–6.

[4] H. Turki, M. Ben Halima, and A. Alimi, "Text detection based on mser and cnn features," Nov. 2017, pp. 949–954.

[5] J. Fan, T. Chen, and F. Zhou, "Bursts: A bottom-up approach for robust spotting of texts in scenes," *Journal of Visual Communication and Image Representation*, vol. 71, p. 102843, 2020.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004, british Machine Vision Computing 2002.

[7] D. He, X. Yang, C. Liang, Z. Zhou, A. G. Ororbia, D. Kifer, and C. L. Giles, "Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 474–483.

[8] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation." arXiv, 2018. [Online]. Available: https://arxiv.org/abs/1801.01315

[9] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3066–3074.

[10] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes." arXiv, 2018. [Online]. Available: https://arxiv.org/abs/1807.01544

[11] V. H. Tien, T. Huong, S. Van, and X. Hoangvan, "Improving tdwz correlation noise estimation: A deep learning based approach," *REV Journal on Electronics and Communications*, vol. 10, pp. 45–54, June 2020.

[12] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "Fots: Fast oriented text spotting with a unified network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5676–5685.

[13] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5020–5029.

[14] 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." arXiv, 2019.

[15] X. HoangVan and H.-H. Nguyen, "Enhancing quality for vvc compressed videos with multi-frame quality enhancement model," in *2020 International Conference on Advanced Technologies for Communications (ATC)*, 2020, pp. 172–176.

[16] Z. Dou, "The text captcha solver: A convolutional recurrent neural network-based approach," in *2021 International Conference on Big Data Analysis and Computer Science (BDACS)*, 2021, pp. 273–283.

[17] F. Sadaf, S. M. Taslim Uddin Raju, and A. Muntakim, "Offline bangla handwritten text recognition: A comprehensive study of various deep learning approaches," in *2021 3rd International Conference on Electrical Electronic Engineering (ICEEE)*, 2021, pp. 153–156.

[18] M. Strake, B. Defraene, K. Fluyt, W. Tirry, and T. Fingscheidt, "Fully convolutional recurrent networks for speech enhancement," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6674–6678.

[19] K. Mehrotra, M. K. Gupta, and K. Khajuria, "Collaborative deep neural network for printed text recognition of indian languages," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 252–256.

[20] A. A. Chandio, M. Asikuzzaman, M. R. Pickering, and M. Leghari, "Cursive text recognition in natural scene images using deep convolutional recurrent neural network," *IEEE Access*, vol. 10, pp. 10 062–10 078, 2022.

[21] W. Zhang, L. Zhu, L. Xu, J. Zhou, H. Sun, and X. Liu, "Deep learning based container text recognition," in *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2019, pp. 69–74.

[22] H. Nisa, J. A. Thom, V. Ciesielski, and R. Tennakoon, "A deep learning approach to handwritten text recognition in the presence of struck-out text," in *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2019, pp. 1–6.

[23] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection." arXiv, 2019. [Online]. Available: https://arxiv.org/abs/1904.01941

[24] S. Nitish, R. Darsini, G. S. Shashank, V. Tejas, and A. Arya, "Bidirectional encoder representation from transformers (bert) variants for procedural long-form answer extraction," in *2022 12th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, 2022, pp. 71–76.

[25] D. C. Bui, D. Truong, N. D. Vo, and K. Nguyen, "Mc-ocr challenge 2021: Deep learning approach for vietnamese receipts ocr," in *2021 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2021, pp. 1–6.

[26] A. Pulver and S. Lyu, "Lstm with working memory," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 845–851.

[27] Y. Wang, "A new concept using lstm neural networks for dynamic system identification," in *2017 American Control Conference (ACC)*, 2017, pp. 5324–5329.

[28] N. S. Malinovi, B. B. Predi, and M. Roganovi, "Multilayer long short-term memory (lstm) neural networks in time series analysis," in *2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, 2020, pp. 11–14.

[29] S. Liu, H. Tao, and S. Feng, "Text classification research based on bert model and bayesian network," in *2019 Chinese Automation Congress (CAC)*, 2019, pp. 5842–5846.

[30] A. Mostafa, O. Mohamed, A. Ashraf, A. Elbehery, S. Jamal, G. Khoriba, and A. S. Ghoneim, "Ocformer: A transformer-based model for arabic handwritten text recognition," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 2021, pp. 182–186.

[31] N. M. Dipu, S. A. Shohan, and K. M. A. Salam, "Bangla optical character recognition (ocr) using deep learning based image classification algorithms," in *2021 24th International Conference on Computer and Information Technology (ICCIT)*, 2021, pp. 1–5.

[32] D. Q. Nguyen and A. T. Nguyen, "Phobert: Pre-trained language models for vietnamese." arXiv, 2020, pp. 1–6. [Online]. Available: https://arxiv.org/abs/2003.00744

[33] L. Nguyen and D. Q. Nguyen, "Phonlp: A joint multi-task learning model for vietnamese part-of-speech tagging, named entity recognition and dependency parsing," Jan. 2021, pp. 1–7.

[34] D. Q. Nguyen, T. Vu, and A. Nguyen, "Bertweet: A pre-trained language model for english tweets," Jan. 2020, pp. 9–14.

[35] T.-D. H. Nguyen, D. Phung, D. T.-C. Nguyen, H. M. Tran, M. Luong, T. D. Vo, H. H. Bui, D. Phung, and D. Q. Nguyen, "A Vietnamese-English Neural Machine Translation System," in *Proc. Interspeech 2022*, 2022, pp. 5543–5544.

[36] N. Nguyen, T. Nguyen, V. Tran, M.-T. Tran, T. D. Ngo, T. Huu Nguyen, and M. Hoai, "Dictionary-guided scene text recognition," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7379–7388.

[37] P. B. C. Quoc, "Vietocr," March 2015, version 0.3.8. Retrieved from official website: https://github.com/pbcquoc/vietocr.