Journal of New Media
DOI: 10.32604/jnm.2022.027890
Article

Tech Science Press

Menu Text Recognition of Few-shot Learning

Xiaoyu[1,2], Tian Zhenzhen[2], Xin Zihao[2], Liu Suolan[2], Chen Fuhua[3] and Wang Hongyuan[2,*]

[1]School of Computer Science and Artificial Intelligencea, Changzhou, 213164, China
[2]Changzhou University, Changzhou, Jiangsu, 213164, China
[3]West Liberty University, 208 University Drive, West Liberty, 26074, USA
*Corresponding Author: Wang Hongyuan. Email: hywang@cczu.edu.cn

Received: 27 January 2022; Accepted: 15 April 2022

Abstract: Recent advances in OCR show that end-to-end (E2E) training pipelines including detection and identification can achieve the best results. However, many existing methods usually focus on case insensitive English characters. In this paper, we apply an E2E approach, the multiplex multilingual mask TextSpotter, which performs script recognition at the word level and uses different recognition headers to process different scripts while maintaining uniform loss, thus optimizing script recognition and multiple recognition headers simultaneously. Experiments show that this method is superior to the single-head model with similar number of parameters in end-to-end identification tasks.

Keywords: Text recognition; script identification; few-shot learning; multiple languages

1 Introduction

The end-to-end OCR approach already delivers good accuracy, and detection and recognition can help each other improve, but most of the previous approaches were for a single English language. The English language has only 26 letters and is generally case-insensitive when it comes to recognition. In addition to English, there are a lot of Western characters, special characters, script and artsy characters. These characters are very different from English, and the proportion of non-English characters is not small. Therefore, studying the end-to-end method of English alone is not enough to solve this problem. It is very important to explore an end-to-end method that can recognize multi-lingual text.

The easiest way to do text spotting for multiple languages is to simply expand our recognition categories and train our recognizer with all font candidates. There is a problem, however, that different languages, due to their different structures, require different feature extractors to achieve better results. In addition, there is another problem, the amount of training data of different languages is different, so there will be a long tail effect, and some rare fonts will appear less probability in the training, which will also lead to the recognition probability of rare fonts.

We chose to use an end-to-end multilingual text Spotter multiplex network. Specifically, we propose a single text detection module and multiple text recognition headers for each language. The multiplexer sends the detected text to the corresponding header, and the Language Prediction Network (LPN) makes an output decision. This strategy can be thought of as our perception of text. Even if you do not know the language, it's easy to locate words in most languages, only in the process of recognizing actual characters and words, it is usually necessary to recognize the language/genre first.

The design of this multiplexer is of great significance for text location systems in the real world. With language-specific text recognition headers, we can customize the design architecture to the characteristics and difficulty of each language, and share the same text detection backbone for learning. For a new language, it can be added to the system without retraining the entire model, which is very easy and will not affect the existing language. Migrating this approach to the small sample menu recognition task of this competition requires only designing three script recognition headers, namely English and Spanish and characters.

2 Related Works

Text recognition is usually divided into two sub-tasks: Text detection and text recognition. Script recognition is also necessary in multi-language scenarios, either explicitly by learning classification models or implicitly as a by-product of text recognition. In the past, these three sub-tasks were often considered separately and then linked together, but recently, the end-to-end method of seeking simultaneous learning is very popular.

2.1 Text Detection

Text detection is usually an important prerequisite for image content analysis and understanding. Early methods often consist of artificially designed features or heuristics, such as connecting components [1] or sliding windows. Some other solutions are combined with convolutional neural network [2], and convolutional neural network has been proved to be successfully used for target detection, while recent methods almost entirely use deep detection models [3]. Given that text can take on different orientations and shapes, further improvements have focused on keeping the rotation of text detection constant [4] or switching from the common rectangular bounding box method used to define the geometry of the object to a more flexible segmentation mask method [5]. The character-level annotation detection method based on weakly supervised learning has also been proved to be an effective method.

2.2 Text Recognition

The purpose of text detection is to locate the text area in the image. After positioning, the area will usually be clipped and entered into the text recognition system for single character recognition and text line recognition. Traditional text recognition often adopts the k-nearest neighbor method, which has a large amount of calculation and is not suitable for the system with high real-time requirements. With the wide application of deep learning methods, many excellent recognition models based on deep learning have emerged, such as VGGNe, ResNet, Inception, DenseNet, etc [6]. The accuracy of single character recognition has also been significantly improved. For the recognition of text lines, there are mainly two methods with excellent results in recent years. One is CNN + RNN + CTC, and the other is CNN + RNN based on Attention [7].

2.3 Script Identification

For multiple language text detection, script identification is usually used to identify the language of text recognition. The traditional method is mainly to recognize the script language in a simple environment [8], which relies on manually designed features to capture the attributes of the scene text. Usually, these methods find candidate characters by extreme regional extraction or edge detection. CNN image patches (obtained through sliding windows) are used to predict text/no text score, characters and binary classes. With increasingly complex demand for text recognition, the demand for text recognition in natural scenes is becoming stronger and stronger [9]. Shi et al. [10] cut and mark the text in the image from Google Street View for text recognition in natural scenes, and then use a special multi-level pool layer to train convolutional neural network for classification. Through dense extraction of local descriptors and discriminant clustering, the accuracy is further improved. Fujii et al. [11] proposed a line by line recognition script recognition method, which converts characters into sequences to mark the problem. E2e-mlt [12] is a multilingual OCR for scene text released for the first time. An end-to-end trainable method for multilingual scene text location and recognition is proposed. The method is based on a single complete convolution network (FCN) with a shared layer for all tasks, Script recognition is not required. Or after the text recognition step, infer the language by identifying the language with the most common characters in the text [13], but this method will affect the performance of the model to a certain extent. We found that, compared with most methods, using the hidden set features of words as input and multiplexing the text recognition header, The script recognition task executed in our application language prediction network (LPN) can significantly improve the accuracy of script recognition task.

3 OCR Model

An end-to-end multi-channel OCR model is designed, which has both detection unit and recognition module, CNN feature sharing and joint training. In order to automatically select the recognition module suitable for a given script, we propose a language prediction network (LPN), which uses masked pooled features as input and adopts the integration loss function with recognition head. A training strategy that can be easily extended to new languages is designed, and a specific recognition header trained from a single script data is used to better solve the problem of data imbalance between different languages. The experimental data show that the reuse model has a good effect and is not prone to the deviation of training data distribution.

The proposed M3 TextSpotter and Mask TextSpotterV3 [14] shares the same detection and segmentation backbone, but incorporate a new language prediction network (LPN). According to the output decision obtained by the LPN, which text recognition header the multiplexer selects.

Like Mask TextSpotterV3, this multiplexer has the same detector and segmentation module. The recognition model of Mask TextSpotterV3 only the spatial attention module, but does not include the character segmentation module. We used only the spatial attention module in the model for the following reasons: (1) when the character set is expanded from 36 to 10 k, both modules are not scalable, so it has no impact; (2) the order of characters in the segmentation MAPS cannot be changed, and the character segmentation module needs character-level annotation during supervised training [15].

To extend the model from English to multiple languages, there are two directions: (1) treat all languages and characters as belonging to the same language as all characters, and use a single recognition header to process all languages and characters; (2) establish independent recognition headers to process words from different languages, and then select/combine them for prediction [16]. We chose method (2) because it is more flexible and has greater potential for scaling without worrying about data imbalance between different languages when we train the model [17].

There are two solutions to extend the model from English to multiple languages: (1) using a single recognition header to process all languages and characters, that is, all languages and characters are regarded as the same language; (2) create a separate recognition header for each language and then select/combine them for prediction. We chose method (2) because it does not have to worry about the data imbalance between different languages when training the model, and has greater expansion potential. Fig. 1 shows the overall framework of end-to-end multi-channel OCR model.

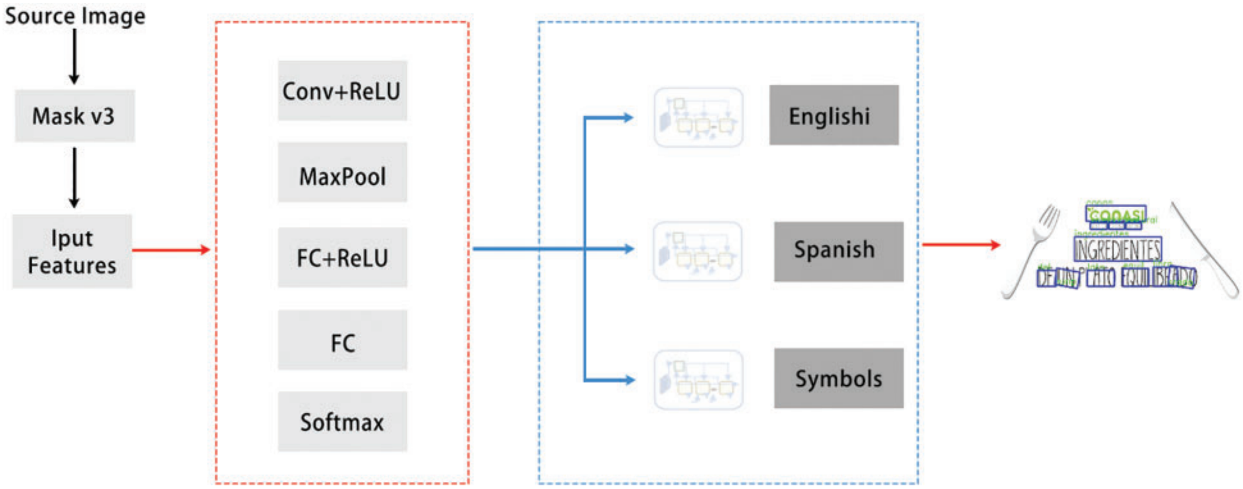

Figure 1: Flowchart of overall framework

3.1 Language Prediction Network

Language prediction network is actually a classified network. The input is masked Pooled features outputted by the detection and segmentation module [18]. The output of the network is a vector with length L, which can be converted into probability by Softmax. Where L = N_lang means that the model supports several languages. Considering the actual situation of this competition task, N_lang = 3 indicates that three scripts (English, Spanish, symbols) are supported.

We use L = N_ Lang represents the number of languages supported by the model.

3.2 Separable Loss Multiplexer

During training, LPN and recognizer head are trained in parallel, and then the loss function is added directly:

$$L_{\text{disentangled}} = \alpha_{lang} L_{lang} + \sum_{r \in R} \alpha_{seq(r)} L_{seq(r)} \tag{1}$$

L_{lang} is loss of LPN and $L_{seq(r)}$ is loss of recognizer head R. Two hyper-parameters are assigned respectively, named α_{lang} and $\alpha_{seq(r)}$. While in our menu recognition task, we set up $\alpha_{lang} = 0.02$ in the first 30 iterations to increase the number of preheated iterations and then adjust it to $\alpha_{lang} = 1$. The weight of loss of all recognizer heads is always $\alpha_{seq(r)} = 0.5$.

Language prediction is a typical N-way classification problem, so cross entropy loss can be used to calculate the language prediction loss of the following equation:

$$L_{lang} = -\sum_{l=1}^{N_{lang}} I(l = l_{gt}) \log p(l) \tag{2}$$

$I(l = l_{gt})$ is the binary indicator (0 or 1), and $p(l)$ is the probability that the word inferred by LPN belongs to language L if the language conforms to the basic facts.

Similar to the classification loss, we use negative likelihood logarithm as the loss of the recognizer:

$$L_{seq} = -\frac{1}{T}\sum_{t=1}^{T} \log p(y_t = c_t) \tag{3}$$

In addition, characters not supported by the recognizer header should be ignored:

$$L_{seq(r)} = -\frac{1}{T}\sum_{t=1}^{T} I(c_t \in C_r) \log p(y_t = c_t) \tag{4}$$

Which $I(c_t \in C_r) = 1$ indicates support, and $I(c_t \in C_r) = 0$ indicates not support.

3.3 Integrated Loss Multiplexer

Although disentangled loss can be used as a good training of initialization model, this method still has some limitations. Firstly, it is necessary to explicit annotation of the basic facts of language based on characters, which may not be accurate and not always available outside of the experimental data set. Secondly, disentangled loss can not reflect the actual prediction of the model in reasoning when there are common features among multiple recognition heads. Finally, although there is a mechanism to ignore labels, training recognition head with unsupported words to recognize the wrong language does not work well.

In order to solve the above problems, we put forward a comprehensive loss, which comprehensively considers the language prediction head results and recognition head results in the training process. In order to make the results of training and testing more reliable, we can use hard Integrated Loss:

$$L_{hard-integrated} = \alpha_{seq(r)} L_{seq(arg\ max_{1 \leq l \leq N_{rec}} p(l))} \tag{5}$$

In the case of Hard Integrated Loss, we select an accurate recognition head for each word, and select and use the loss head with the greatest probability predicted by the LPN. This loss can avoid using the irrelevant recognition head in the training process, and greatly match the operation of the text recognition system in the reasoning process. Ablation studies show that it is superior to the alternative soft integration loss. The soft integral loss is to directly multiply the probability of each language and the recognition head loss of each language, and then add it to obtain the following soft integral loss function:

$$L_{soft-integrated} = \sum_{r=1}^{N_{rec}} p(r) \cdot \alpha_{seq(r)} L_{seq(r)} \tag{6}$$

Hard integral loss can be treated as a special case of soft integral loss, where only one $p(r)$ is 1 and all the other $p(r)$ is 0. It can be seen from the experiment that under the same number of iterations, the loss of using hard integral is about 0.1 better than that of using soft integral.

4 Experiments and Results

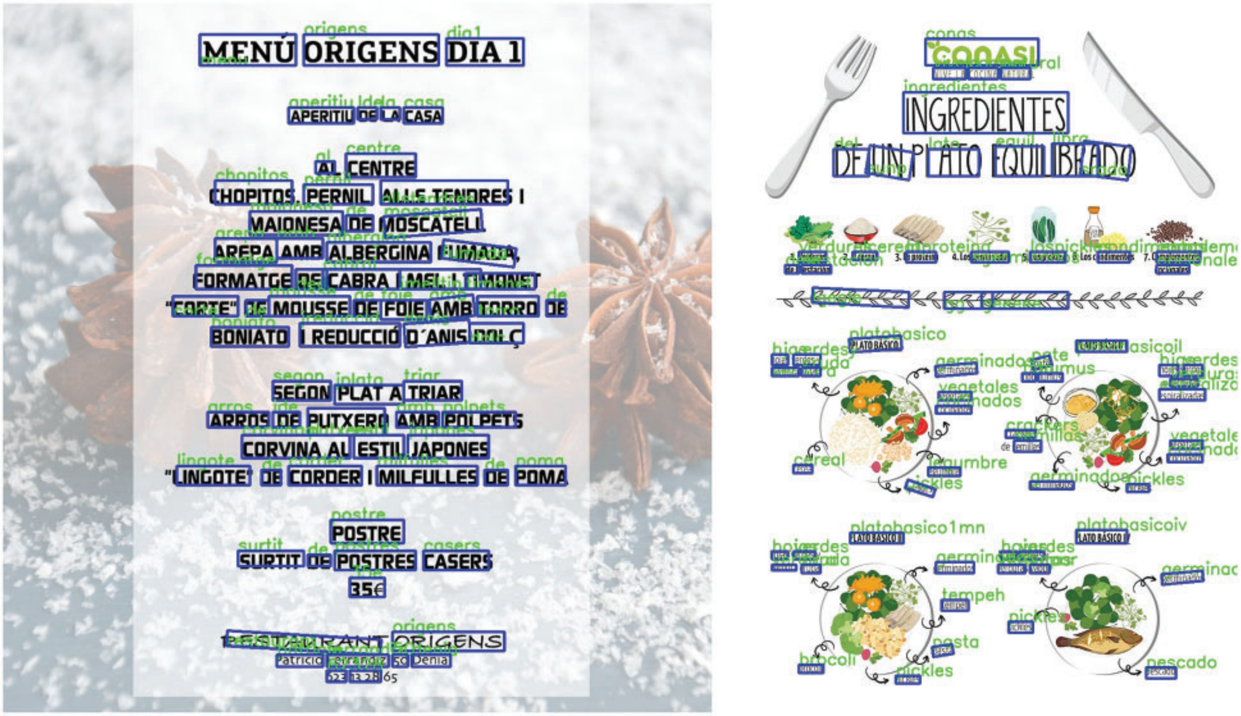

Figure 2: Experiments and results

5 Summary

In the task of small sample menu recognition in the field of image text recognition, we need to use a small sample learning model to solve the problem of multi-language, artistic characters and other complex natural scenes. To solve this problem, an end-to-end multilingual recognition multiplexer is used in this paper to make decisions through LPN network according to different language features, select matching language recognition headers, and then output text recognition results combined with spatial attention module. It can be seen from the experimental results in Fig. 2, both anchor recognition and text recognition have achieved good results.

Acknowledgement: We would like to thank Professor Wang for his technical support during the preparation of this manuscript.

Funding Statement: This research was supported by the Advanced Training Project of the Professional Leaders in Jiangsu Higher Vocational Colleges(2020GRFX006).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. K. Jain, Y. Bin, “Automatic text location in images and video frames,” *Pattern Recogintion*, vol. 31, no. 12, pp. 2055–2076, 1998.

[2] W. Huang, Q. Yu, X. Tang, “Robust scene text detection with convolution neural network induced MSER trees,” in *Proc. of the Computer Vision–ECCV 2014, F*, Switzerland, Zurich, 2014.

[3] T. Zhi, W. Huang, H. Tong *et al.,* “Detecting text in natural image with connectionist text proposal network,” in *Proc. of the European Conf. on Computer Vision, F*, Amsterdam, The Netherlands, 2016.

[4] Y. Jiang, X. Y. Zhu, X. B. Wang *et al.,* “R2CNN: Rotational region CNN for orientation robust scene text detection,” 2017. [Online]. Available: https://blog.csdn.net/chengyq116/article/details/96904368.

[5] P. Lyu, M. Liao, C. Yao *et al.,* “Mask TextSpotter: An End-to-end trainable neural network for spotting text with arbitrary shapes,” in *Proc. of the European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 67–83, 2018.

[6] W. Feng, F. Yin, X. Y. Zhang *et al.,* “Semantic-aware video text detection,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Online, pp. 1695–1705, 2021.

[7] D. Bahdanau, K. Cho, Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Int. Conf. on Learning Representations (ICLR),* Santiago, Chile, 2015.

[8] T. Tan, “Rotation invariant texture features and their use in automatic script identification,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 20, no. 7, pp. 751–6, 1998.

[9] T. Wang, Y. Zhu, L. Jin *et al.,* “Implicit feature alignment: Learn to convert text recognizer to text spotter,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Online, pp. 5973–5982, 2021.

[10] B. Shi, X. Wang, P. Lyu *et al.,* “Robust scene text recognition with automatic rectification,” in *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016.

[11] Y. Fujii, K. Driesen, J. Baccash *et al.,* “Sequence-to-label script identification for multilingual OCR,” in *2017 14th IAPR Int. Conf. on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 2018.

[12] M. Bušta, Y. Patel, J. Matas, “E2e-MLT-an unconstrained End-to-end method for multi-language scene text,” in *Asian Conf. on Computer Vision*, Springer, Cham, 2018.

[13] Y. Baek, S. Shin, J. Baek *et al.,* “Character region attention for text spotting,” in *2020 European Conf. on Computer Vision*, Springer, Cham, 2020.

[14] M. Liao, G. Pang, J. Huang *et al.,* “Mask TextSpotter v3: Segmentation proposal network for robust scene text spotting,” in *European Conf. on Computer Vision*, Online, 2020.

[15] J. Huang, G. Pang *et al.,* “A multiplexed network for End-to-end, multilingual OCR,” in *2021 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Online, 2021.

[16] M. Liao, G. Pang, J. Huang *et al.,* “Mask textspotter v3: Segmentation proposal network for robust scene text spotting,” in *Computer Vision–ECCV 2020: 16th European Conf.*, Glasgow, UK, Proceedings, pp. 706–722, 2020.

[17] X. Xu, Z. Zhang, Z. Wang *et al.,* “Rethinking text segmentation: A novel dataset and a text-specific refinement approach,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Online, pp. 12045–12055, 2021.

[18] M. He, M. Liao, Z. Yang *et al.,* “MOST: A multi-oriented scene text detector with localization refinement,” in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Online, pp. 8813–8822, 2021.