Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018
https://doi.org/10.17979/spudc.9788497497565.0975

DETECTING TEXTUAL INFORMATION IN IMAGES FROM ONION DOMAINS USING TEXT SPOTTING

Pablo Blanco
Dept. IESA. Universidad de León, pblanm02@unileon.es

Eduardo Fidalgo
Dept. IESA. Universidad de León, eduardo.fidalgo@unileon.es

Enrique Alegre
Dept. IESA. Universidad de León, ealeg@unileon.es

Mhd Wesam Al-Nabki
Dept. IESA. Universidad de León, mnab@unileon.es

Abstract

Due to the efforts of different authorities in the fight against illegal activities in the Tor networks, the traders have developed new ways of circumventing the monitoring tools used to obtain evidence of said activities. In particular, embedding textual content into graphical objects avoids that text analysis, using Natural Language Processing (NLP) algorithms, can be used for watching such onion web contents. In this paper, we present a Text Spotting framework dedicated to detecting and recognizing textual information within images hosted in onion domains. We found that the Connectionist Text Proposal Network and Convolutional Recurrent Neural Network achieve 0.57 F-Measure when running the combined pipeline on a subset of 100 images labeled manually obtained from TOIC dataset. We also identified the parameters that have a critical influence on the Text Spotting results. The proposed technique might support tools to help the authorities in detecting these activities.

Keywords: Text detection, text recognition, cybercrime, machine learning, Tor network.

I INTRODUCTION

The Darknet is a portion of the Web that is not indexed by standard search engines, and needs special software or a proxy in order to be accessed. Dark web types include privacy networks, such as The Onion Router (Tor), which contains a great number of hidden services that cannot be search-indexed.

The onion domains are one of these domain types, which designate an anonymous hidden service reachable via the Tor network. The "onion" name refers to the technique Tor uses to grant anonymity to their users, known as onion routing. Due to this focus on anonymity, it is a common source of illegal content and media. According to [1], it is estimated that 25% of the content found in Tor network may involve potentially illegal activities, such as counterfeiting ID documents, credit cards, weapons, drug selling, and other types of illegal content. With the increase in number of these hidden services as well as the size of the available information, automated techniques are required to analyze the content and detect potential threats or illegal activities. Several works have been proposed to fight against these types of illegal activities in Tor network, including Text Classification [1] and Text Summarization [13], which are applied after crawling through the text in these networks. However, the reported methods are insensitive to the written text within the graphical content. Hence, they suffer from a significant shortage in accessing a large amount of valuable information such as a product name, brand, or even the seller name. Text Spotting (TS) comes to fill this gap, as it is capable of detecting and recognizing text in natural scene images.

The Text Spotting forms a pipeline of two phases: first, it localizes the text within an image, and second, it recognizes this text as a particular word. There are several challenges to overcome when performing this task, such as partial occlusion of the text shown, text orientation, or even the presence of different languages in the same image in order to increase the difficulty of properly detecting and recognizing text. In this paper, we present a framework for Text Spotting and apply it to onion domains.

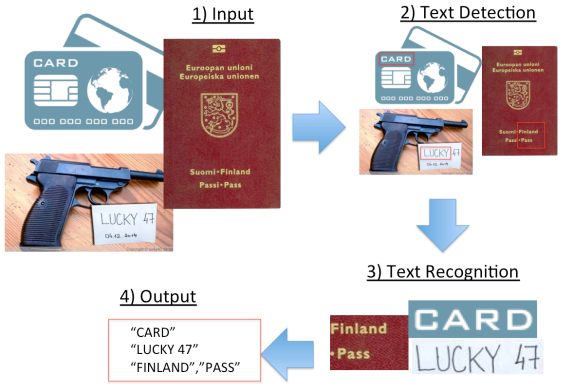

Figure 1. Proposed Framework for Text Spotting.

For this purpose, we will make use of two components: text detection using a connectionist text proposal network algorithm and text recognition using an end-to-end trainable neural network. Then, in order to test the performance of the proposed pipeline, we considered a subset of 100 images from the TOIC dataset [9], which contains five different categories of images related to different illegal activities from the Tor network (Figure 1). We set a baseline methodology by fixing the elements of the Text Spotting pipeline, which will allow future researches to compare their obtained results. The methodology we propose could be a useful tool for authorities that monitor potentially illegal activities in Tor hidden services.

The rest of the paper is organized as follows: Section II presents a brief study of the state of the art. The algorithms that we selected for each phase are described in Section III. We describe our labeling process for the given TOIC images and data on Section IV alongside the details of the experiments we have carried out and the software used. We also detail the obtained results of applying the selected algorithms. Finally, Section V presents our conclusions, as well as our future lines of research.

II STATE OF THE ART

As stated by Minetto et al. (2014) [17], Text Spotting has many potential applications on real environments, such as road navigation, acquiring important geographic information, generic scene understanding and video or image indexing [7]. When learning Text Spotting, it is important not to confuse it with Optical Character Recognition (OCR). Although they might seem similar, OCR is considered a "solved" problem [27] due to its more controlled environments for recognition. Meanwhile, TS has to deal with other issues that are usually absent in OCR fields, such as text orientation or partial image occlusion.

Figure 2. Scene text (a) and Artificial Text (b)

While some of the most recently proposed text-spotting methods often achieve less than 80% in their text recognition rates, OCR usually scores about 99% when used on scanned documents [26].

According to the working environment, the TS can be either a Scene text or an Artificial one [5]. The first, Scene text, refers to the text found naturally in different media, such as a store name or a traffic signal, while the second, Artificial text, is found embedded in media files as an addition to the original image or video, e.g. a news report overlay. Figure 2 illustrates the difference between the both types.

The variability of the text found in real scenes is regarded as the most difficult problem to solve because of the many different factors that may prevent the algorithm from properly analyzing certain regions, such as scene complexity, blurring and degradation, image distortion, the use of different fonts and different aspect ratios [27]. Due to these problems, new methods and approaches that were more effective against complex backgrounds were exhaustively researched. Recently, the focus of Text Spotting has been in the use of approaches such as convolutional neural networks or unsupervised feature learning [27, 29].

There are different approaches in Text Spotting, such as methods based on character regions known as Connected Component (CC) based methods. [8,12]. This approach focuses on extracting character candidates by analyzing connected sections and then applying a grouping process in order to identify candidate regions as text. These methods obtain remarkable results in environments with constant color distribution and regular distribution.

Another relevant approach to text detection is the use of region-based methods, which are also known as sliding window methods [4, 30]. Using a binary text classifier, this approach searches for possible text areas inside a video or image with windows of multiple sizes and aspect ratio that help iterate identify and group the relevant areas into text candidates [29]. The binary classifier can use different features such as the color distribution, texture or edges and corners in order to identify and differentiate the found text from the background.

Next, the candidate regions go through a filtering process, which uses learned classifiers. This method has the advantages of being fast and performing well on low-contrast environments, but they are usually reliant on the complexity of the background. They are also commonly used alongside convolutional neural networks to train classifiers in the sliding evaluations and to perform word recognition. The other preferred method applied to detection of text is Maximally Stable Extremal Regions (MSERs) [16, 23], which has achieved up to 95,5% of detection rate in the ICDAR 2011 Dataset [28].

After successfully determining the bounding boxes of the text, the second phase, text recognition, can be applied. This stage refers to the process of properly labeling the text that was previously detected. The main methods can be divided into two categories, as segmentation-based word recognition and holistic word recognition [27,29].

The first category classifies characters individually before generating the complete word recognition. It attempts to integrate both character segmentation and recognition by using lexicons and optimization techniques [15]. Recently, most segmentation-based algorithms attempt to segment the image into smaller sections, which are then combined into character candidates in order to be classified and finally obtain an approximate word result using different techniques [3, 11]. The most common classifier in this approach is CNNs [23]. The second category attempts to extract features from the complete word before classifying it rather than extracting individual characters. These methods attempt to calculate a similarity score between the image that contains the candidate word and a query word. Some of them use label-embedding algorithms in order to increase the value of the relation between the words found on images and candidate text strings [10] [25].
Recent trends seek to integrate both text detection and text recognition into an end-to-end text recognition system that involves both steps in the same algorithm. There are different methodologies for combining both of these phases. According to [27], they can be reduced into two main approaches, the Stepwise and the Integrated methodologies. In this paper, we followed the stepwise approach in order to separate the process into well-defined stages, as the four primary steps of this methodology do. Furthermore, since some stepwise approaches use a feedback procedure from the text recognition stage in order to reduce false detections, we believe this methodology to be more suited to our problem. After exploring the TOIC dataset, we found that the images in Tor network are usually presented in a way where occlusion is not a common factor, due to the fact that the seller wants to properly identify the product being sold.

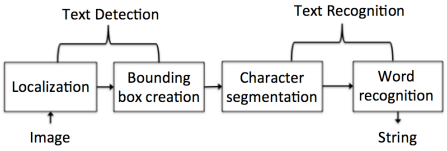

Figure 3. Text Spotting, stepwise methodology.

However, partial occlusion does appear in certain areas such as the seller's name being hidden or partially obscured by the product. Personal documents and money are usually presented oriented in order to show multiple of them in the same picture, which is why orientation will be relevant when applying TS. Regarding the type of the text found, machine-printed text seemed more prominent in our initial study rather than hand-written text.

Because of these reasons and after reviewing the state of the art, we have selected Tian. Z et al. [22] and Shi. B et al. [20] as our main approaches to extract text from images retrieved from the Tor HS due to their robustness against ambiguous text, their reliability multi-scaled and multi-language text and their focus towards generating practical models meant for real-world application scenarios.

Focusing on extracting and processing information from illegal sites of Tor, we selected a small batch of images from the TOIC dataset, in order to test the performance of the proposed pipeline. This dataset was proposed by Fidalgo et al. [9] after crawling Tor images domains. In Tor, several types of services can be found [6] but also illegal activities or hidden services, such as illegal drugs selling [2] and other as identified by Al-Nabki et al [1], which include drug selling, weapons and personal ID forgery.

III METHODOLOGY

A. Text Detection

The selected algorithm for text detection, Tian. Z et al. [22], is based on a Connectionist Text Proposal Network, which segments the image into regions and score them according to probability of having text or not. A vertical anchor mechanism is used so these text and non-text regions are considered based on their score against the proposal regions.
This approach allows creating an end-to-end trainable model, resulting in a method that explores context information within the image in order to detect ambiguous text. The Connectionist Text Proposal Network (CTPN) uses a fully convolutional network that allows to input images of varying sizes.

It detects lines of text by using a sliding window in the convolutional feature maps and obtains a sequence of text proposals, much like a regular Region Proposal Network method [19]. Thanks to a number of flexible anchor mechanisms, a single window is able to predict objects in a wide range of scales and aspect ratios. The vertical properties of the image are given a higher weight in order to reduce the search space grouping sections in this direction, using up to ten anchors for each proposal region. The sliding window takes a convolutional feature for producing the prediction using the VGG16 model [21]. Anchor locations are fixed to output the text scores and filtered with an initial threshold of over 0.7. The next processes involve side-refinement approaches trying to estimate the offset of each anchor, now focusing on the horizontal properties of said regions. Alongside being computationally efficient, this approach has obtained impressive results on popular datasets, such as ICDAR 2013 Dataset [14], with a 0.93 score of precision, 0.83 recall and 0.88 F-Measure, as well as a 0.61 score in F-Measure on the ICDAR 2015 dataset. It has achieved good results on the SWT [8] and Multilingual [18] approaches as well, with over 0.66 and 0.82 F-Measure scores respectively.

B. Text Recognition

We adopted Shi et al. model [20] as they proposed an end-to-end trainable neural network, which handles sequences of different lengths and generates an effective and small model. They used Convolutional Recurrent Neural Network (CRNN) because its capable to learn from labeled sequences directly, and only requiring height normalization of the regions in testing and training phases, achieving highly competitive performance and containing fewer parameters that a standard model. The CRNN consists of three main components, the convolutional and recurrent layers, and a transcription layer. The convolutional layers extract a feature sequence from the input image, while the recurrent layers predict a label distribution for the taken regions.

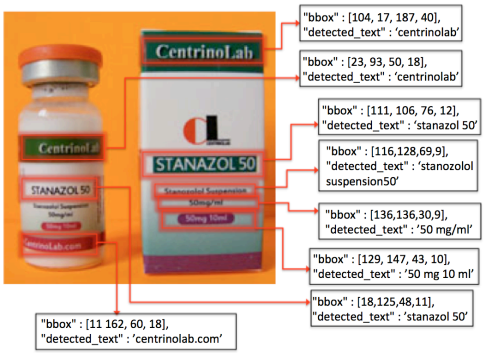

Figure 4. Text detection applied on an image from the drugs category.

Figure 5: Three samples (columns) for each of the five categories (rows) of TOIC dataset.

Finally, the transcription layer creates the final label sequence, by using the highest probability contained in the region-based transcriptions. The architecture of the convolutional layers is based on the VGG-VeryDeep architectures [21], but with some adjustments being made with the focus of recognizing English words. As both the deep convolutional and recurrent layers are hard to train, the batch normalization techniques are used after specific convolutional layers in order to improve the speed of the training process.

IV RESULTS AND DISCUSSION

A. Data used for the experimentation

Due to the lack of labeled datasets for the illegal activities in Tor network, we took a subset of 100 images from the TOIC dataset, such that each category holds 20 images, and labeled them inspired by ICDAR 2017 and COCO-TEXT [24] datasets, manually identifying the bounding boxes, as well as the transcribed text inside the bounding boxes in order test the performance of the proposed TS pipeline (Figure 4). We obtain a total of 1112 text regions from the 100 images analyzed (Figure 5).

B. Experimental Setup

The selected models, Tian et al. [22] and Shi. B et al. [20], were implemented using TensorFlow under Python3, while the OpenCV library was used for preprocessing the images. We test these algorithms on a Intel Xeon E5 v3 computer with 128GB of RAM using a Nvidia GPU.

In order to adapt the text detection model for our problem, we explored three different proposals by modifying the default parameters of the original model. We represent these proposals as P1, P2 and P3. The first proposal, P1, refers to the default parameters proposed by their authors based on the VGG16 model. In the second one, P2, we adjusted the anchor scales parameter for the text detection algorithm to double their original values.

Finally, for the third proposal, P3, we increased the minimum of top scoring boxes before applying non-maximum-suppression to the region proposals, as well as doubling again the original anchor scales parameter from P2. Once we have the bounding boxes for the detected regions, we apply text recognition to these areas, maintaining the TR model parameters in their default values.

To filter and evaluate the regions considered as text, we define a minimum of Intersection-over-Union (IoU) when we compare them to the ground truth of our labeled set of images. If the detected box shares a higher percentage than this IoU, they are correctly classified as containing text. Since the majority of our dataset images are small, with a mean dimension of 500x300, this threshold is set between a minimum of 0.4 and a maximum of 0.7. Otherwise, the bounding box is ignored. If there are ground truths, which are not associated to any of the detected boxes, these are labeled as false negatives. Similarly, for text recognition, we only consider that a word has been recognized correctly if the entire word matches the labeled region as described before.

C. Results Discussion.

Table 1: Performance measured on the three proposals, with the best results marked in bold.

	Threshold	Precision	Recall	F-Measure
P1	0.4	0.72	0.31	0.43
	0.5	0.71	0.30	0.42
	0.6	0.68	0.29	0.41
	0.7	0.66	0.28	0.39
P2	**0.4**	**0.84**	**0.44**	**0.57**
	0.5	0.81	0.41	0.54
	0.6	0.76	0.39	0.51
	0.7	0.72	0.37	0.49
P3	0.4	0.67	0.16	0.24
	0.5	0.63	0.13	0.22
	0.6	0.60	0.11	0.21
	0.7	0.58	0.10	0.20

Table 1 shows that the best results are achieved with the doubling of the anchor scales, outperforming both the other proposals even with the highest minimum threshold. Figure 6 helps show our observations of the regions detected in all three of these proposals by adjusting the anchor values and the effect of the threshold. Out of 1112 regions, P1 could only detect 510, while P2 was capable of detecting 610 bounding boxes correctly. However, P3 could detect only 300 due to both the further increase of the vertical threshold as well as the minimum of top scoring boxes, which had the opposite effect of the expected result.

Figure 6. Illustrating the three scenarios for the pipeline test.

We can see in both Table 1 and Figure 6 that the best result is achieved with the P2 proposal, noticing the relevance of the vertical threshold for the anchors that was described initially, which we will then select to apply text recognition via the rest of our pipeline. Despite the robustness of the text detection algorithm against the orientation, the performance drops sharply when exposed to images with complex backgrounds or a higher curve degree. Furthermore, some regions of text, which are often considered as one unit, can be split into different sections, increasing the difficulty of the text recognition phase and the retrieval of performance measurements such as precision or recall due to having to determine how to consider these regions. This particular problem is most likely due to the algorithm's vertical approach to text detection. In Figure 7, we can see that the text detection algorithm obtains good results on images with partial orientation, but does not work well with occlusion due to omitting the number found below the gun.

Figure 7. Samples of text regions properly detected, as well as wrong regions.

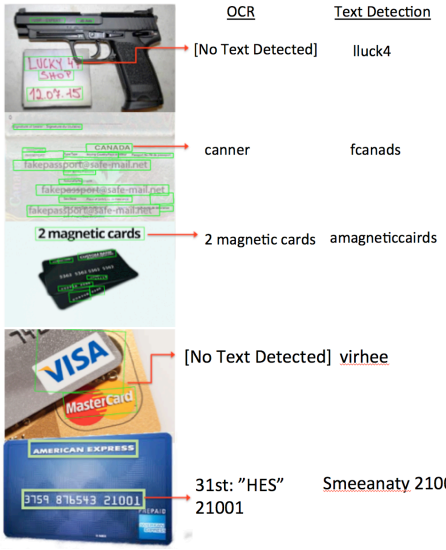

Figure 8. Text detected on a region with OCR and the Shi et al. model [20] algorithm evaluated.

Both the credit card and money ticket images (Figure 7, top section) show a significant amount of bounding boxes, being a good reference of the algorithm's performance even though some of these bounding boxes can be a bit imprecise, especially in the case of the magnetic cards (Figure 7, bottom section).

Regarding the text recognition phase, it did not achieve results as relevant as the previous phase. Figure 8 shows the result of text recognition applied to the batch of images taken from the TOIC dataset, also including the result obtained when applying OCR to the same regions in order to compare the performance of both approaches to text recognition. While both results differ slightly from the documented strings, this can be attributed to the size that the images are scaled to during detection, since recognition is applied directly to these subsections. Therefore, the smaller the regions of the detected text are the lower the accuracy for the region will be. OCR seems to yield slightly similar results, not outperforming our chosen algorithm except in areas related to artificial text (credit card examples in Figure 8), which is not the main focus of our work.

V CONCLUSIONS AND FUTURE WORK

In this paper, we have reviewed the literature of Text Spotting, and through the stepwise methodology, we propose a framework to apply it to the images of Tor onion domains. In particular, our intention was to apply TS to the images found in hidden domains that contain illegal activities, proposing a pipeline of a text detection algorithm, which feeds the bounding boxes found inside the image to a text recognition network that identifies the string inside them.

We found that the best text detection method for Tor images is based on a connectionist neural network and for text recognition, while the text and end-to-end trainable neural network. To evaluate the performance, we labeled 100 images from the TOIC dataset, where we include the bounding box and detected string for each text region present, which are used as our main data source. We found that the Connectionist Text Proposal Network algorithm could be labeled as the best algorithm for our field of work, as it achieved an F1 score of 57% in the evaluated images, which is obtained by doubling the anchor features, noticing the relevance of the vertical threshold when applied to our particular field. However, certain uniform regions of text can be divided into different sections, which increase the difficulty of the text recognition phase. This particular problem is due to the vertical approach of the algorithm, as well as the smaller than average dimensions that our images have.

Since text recognition is heavily dependent on text detection, further refinement of this section of the pipeline will be a task to focus on in order to define more of these bounding boxes. The cropped sizes of the detected boxes may also pose a problem if the end result is an image of even lower resolution. When compared against OCR, our chosen algorithm performs slightly better in hand-written, oriented text, although in categories such as credit cards the results are slightly closer between the two algorithms, as can be seen in Figure 8.

Due to the successful results in the text detection, in future works, we will focus on improving the text recognition, by training the current model with a bigger dataset and focusing on the anchor scales parameter in order to obtain as many relevant regions as possible for their successive analysis.

Acknowledgements

This research is supported by the INCIBE grant "INCIBEI 2015-27359" corresponding to the "Ayudas para la Excelencia de los Equipos de Investigación avanzada en ciberseguridad" and also by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 22. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] Al Nabki, M. W., Fidalgo, E., Alegre, E., & de Paz, I. (2017). Classifying illegal activities on TOR network based on web textual contents. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (Vol. 1, pp. 35-43).

[2] Al Nabki, M. W., Fidalgo, E., Alegre, E., and Gonzalez-Castro, V. "Detecting Emerging Products in Tor Network Based on K-Shell Graph Decomposition," III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), vol. 1.

[3] Alsharif, O., & Pineau, J. (2013). End-to-end text recognition with hybrid HMM maxout models. *arXiv preprint arXiv:1310.1811*.

[4] Anthimopoulos, M., Gatos, B., & Pratikakis, I. (2008, September). A hybrid system for text detection in video frames. In *Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on* (pp. 286-292). IEEE.

[5] Anthimopoulos, M., Gatos, B., & Pratikakis, I. (2013). Detection of artificial and scene text in images and video frames. *Pattern Analysis and Applications*, *16*(3), 431-446.

[6] Biswas, R., Fidalgo, E., & Alegre, E. (2017). Recognition of service domains on TOR dark net using perceptual hashing and image classification techniques.

[7] Chen, D., Odobez, J. M., & Bourlard, H. (2004). Text detection and recognition in images and video frames. *Pattern recognition*, *37*(3), 595-608.

[8] Epshtein, B., Ofek, E., & Wexler, Y. (2010, June). Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2963-2970). IEEE.

[9] Fidalgo, E., Alegre, E., González-Castro, V., & Fernández-Robles, L. (2017, September). Illegal Activity Categorisation in DarkNet Based on Image Classification Using CREIC Method. In *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding* (pp. 600-609). Springer, Cham.

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

[11] Jaderberg, M., Vedaldi, A., & Zisserman, A. (2014, September). Deep features for text spotting. In *European conference on computer vision* (pp. 512-528). Springer, Cham.

[12] Jain, A. K., & Yu, B. (1998). Automatic text location in images and video frames. *Pattern recognition*, *31*(12), 2055-2076.

[13] Joshi, A., Fidalgo, E., Alegre, E., & Nabki, M. W. A. Extractive Text Summarization in Dark Web: A Preliminary Study.

[14] Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L. G., Mestre, S. R., ... & De Las Heras, L. P. (2013, August). ICDAR 2013 robust reading competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on* (pp. 1484-1493). IEEE.

[15] Manmatha, R., Han, C., & Riseman, E. M. (1996, June). Word spotting: A new approach to indexing handwriting. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on* (pp. 631-637). IEEE.

[16] Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, *22*(10), 761-767.

[17] Minetto, R., Thome, N., Cord, M., Leite, N. J., & Stolfi, J. (2014). SnooperText: A text detection system for automatic indexing of urban scenes. *Computer Vision and Image Understanding*, *122*, 92-104.

[18] Pan, Y. F., Hou, X., & Liu, C. L. (2011). A hybrid approach to detect and localize texts in natural scene images. *IEEE Transactions on Image Processing*, *20*(3), 800-813.

[19] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

[20] Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, *39*(11), 2298-2304.

[21] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[22] Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016, October). Detecting text in natural image with connectionist text proposal network. In *European Conference on Computer Vision* (pp. 56-72). Springer, Cham.

[23] Turki, H., Halima, M. B., & Alimi, A. M. (2017, November). Text Detection based on MSER and CNN Features. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on* (Vol. 1, pp. 949-954). IEEE.

[24] Veit, A., Matera, T., Neumann, L., Matas, J., & Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.

[25] Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012, November). End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 3304-3308). IEEE.

[26] Weinman, J. J., Learned-Miller, E., & Hanson, A. R. (2009). Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, *31*(10), 1733-1746.

[27] Ye, Q., & Doermann, D. (2015). Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *37*(7), 1480-1500.

[28] Yin, X. C., Pei, W. Y., Zhang, J., & Hao, H. W. (2015). Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), 1930-1937.

[29] Yin, X. C., Zuo, Z. Y., Tian, S., & Liu, C. L. (2016). Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Transactions on Image Processing*, *25*(6), 2752-2773.

[30] Yin, X., Yin, X. C., Hao, H. W., & Iqbal, K. (2012, November). Effective text localization in natural scene images with MSER, geometry-based grouping and AdaBoost. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (pp. 725-728). IEEE.