# ENHANCING TEXT RECOGNITION ON TOR DARKNET IMAGES

Pablo Blanco-Medina[1,2], Eduardo Fidalgo[1,2], Enrique Alegre[1,2], Mhd Wesam Al-Nabki[1] and Deisy Chaves[1]

[1] Universidad de León, España
[2] Researcher at INCIBE (Spanish National Cybersecurity Institute), León, Spain
{pblanm, efidf, ealeg, mnab, dchas}@unileon.es

## Abstract

Text Spotting can be used as an approach to retrieve information found in images that cannot be obtained otherwise, by performing text detection first and then recognizing the located text. Examples of images to apply this task on can be found in Tor network images, which contain information that may not be found in plain text. When comparing both stages, the latter performs worse due to the low resolution of the cropped areas among other problems. Focusing on the recognition part of the pipeline, we study the performance of five recognition approaches, based on state-of-the-art neural network models, standalone OCR, and OCR enhancements. We complement them using string-matching techniques with two lexicons and compare computational time on five different datasets, including Tor network images. Our final proposal achieved 39,70% precision of text recognition in a custom dataset of images taken from Tor domains.

**Keywords:** Text Spotting, Text Recognition, OCR, Cybersecurity, Tor darknet.

## 1 Introduction

Text spotting is a pipeline of two consecutive tasks: accurate detection of text regions inside an image or video and recognition of the detected text to obtain a readable string [3, 7]. Automatizing the process of text retrieval from visual media allows to obtain a high volume of information otherwise ignored in content-based search engines [4].

The importance of text spotting is significantly evident in the Darknet network, with the onion domains of The Onion Router (Tor) network being a relevant source of illegal content. Thanks to the high level of privacy and anonymity of the Tor network, it has attracted illegal services traders to promote for their products safely, far from the monitoring tools of the authorities.

A recent study by Al-Nabki et al. (2019) [1] showed that more than 29% of the onion domains involve suspicious activities such as weapon selling, drugs trading, and counterfeiting personal identification documents. [10, 11]

The recent advance in Machine Learning and Deep Learning algorithms allowed researchers to develop intelligent tools to detect suspicious activities. In Al-Nabki et al. (2019) [1], the authors proposed an algorithm to rank the onion domains and to detect the most influential ones, while in Al-Nabki et al. (2017) [2], graph analysis is used to detect the emerging products. Other approaches have proposed image classification system to detect these activities [4].

Consequently, onion domain hosts have sorted to hiding the descriptive text of their products or services inside images. The text spotting technique comes to fill in this gap to detect and recognize this type of hidden text.


Figure 1: Example of Tor network image labelled for the task of Text Spotting

As far as we know, only one work has tackled text spotting in the onion domains [5]. However, the proposed method obtained 57% F-Measure in text detection and 0% precision in recognition. This can be attributed to issues such as partial occlusion, text orientation or the presence of multiple languages in the same image. Furthermore, several types of text can be found in the images hosted in onion domains of the Tor network, varying from hand-written based text to custom fonts and machine printed text, such as watermarks (Fig. 1).

Due to the difference in the results of each task, we split our original pipeline in order to improve their separate performance. In this paper, we focus on the text recognition phase only, in order to find the best methods suited for this task and how they can be improved.

The text recognition task can be divided into two steps, segmentation and transcription. The segmentation task consists on applying techniques that allow to extract the bounded characters from an image, separating the individual characters before their transcription. Some of the most relevant techniques in segmentation include text binarization, text line segmentation and character segmentation [23].

After the characters have been segmented, they can be transcribed into readable character sequences, by using single character or word level recognition. Words analysis is more common than sentences due to the fact that they are easier to track and transcribe.

When applied to natural-scene images, text recognizers obtain lower scores than document-based transcription. This is due to the differences in how text appears in real images when compared to the clean backgrounds and structure of document text, which may not contain multiple fonts, orientation or the noise found in regular images [21].

We benchmark five text recognition approaches in five different state-of-the-art datasets in order to test the performance of this task. We select three neural network based approaches; ASTER [21], FOTS [17] and CRNN [20], due to their state-of-the-art results, as well as one standalone Optical Character Recognition (OCR) approach and an enhanced OCR approach.

Two text corpora; general and Tor context-specific, as well as string-matching algorithms are used when comparing the result to the original ground truth in order to obtain better results. We also analyze the most relevant problems when recognizing text from different sources.

The rest of the paper is organized as follows. Section 2 presents a review of relevant text recognition approaches. Section 3 reports the methodology followed for comparing the text recognition methods. Section 4 details our experiments and result discussion. Finally, we present our conclusions and future lines of work in the field of text spotting.

## 2 Related Works

Several approaches have been proposed in the past few years to improve text recognition in natural images. Tencent-PRC & USTB-PRIR achieved the highest result of 43,58% precision in the ICDAR (International Conference on Document Analysis and Recognition) 2017 competition of Robust Reading Challenge on COCO-Text [12], using specific lexicons to improve word recognition.

Shi et al. [21] introduced a neural network model that combines rectification and recognition networks, based on sequence-to-sequence learning models. The first network is used to fix the orientation of the input image and the irregularities of the text, while the second performs a character sequence prediction using the newly corrected image.

Liu et al. [17] proposed a Text Spotting system that combines detection and recognition on a joint trainable system. This approach uses Convolutional Neural Networks (CNNs) shared between detection and recognition in order to perform feature extraction on a single network.

Shi et al. [20] combined Deep Convolutional Neural Networks (DCNN) and Recurrent Neural Networks (RNN) architectures to create a method that can process images of varying dimensions and predict characters or words of different lengths. The method trains both structures simultaneously with only one loss function.

Bartz et al. [3] proposed a method that uses a single neural network in order to both detect and recognize text in a semi-supervised way training both tasks jointly.

Finally, Busta et al. [7] developed a Fully Convolutional Net (FCN) based detector, combined with a Connectionist Temporal Classification (CTC) recognizer in a single, trainable framework.

## 3 Methodology

We selected five text recognition approaches, three of which are based on neural networks and two OCR approaches, in order to study their performance. For neural networks, we selected ASTER [21], FOTS [17] and CRNN [20]. For Optical Character Recognition, we chose Pytesseract4, a python wrapper for Google's Tesseract technology. We used this OCR approach under two different configurations.

We selected the neural network based models due to their state-of-the-art results. In particular, we selected ASTER as a relevant text recognizer due to its rectification feature, which can correct oriented text making the transcription more accurate.

ASTER [21] is a model that combines this rectification network alongside recognition. It predicts a character sequence from the corrected image using a bidirectional decoder, which combines the results of each decoder. Using these features,

Table 1: Precision of the text recognition methods in various datasets. Bold values indicate best results

| Method | SVT | Custom TOIC | IIT5K-Words | ICDAR 2013 | ICDAR 2015 |
|---|---|---|---|---|---|
| ASTER | **88,25%** | **28,30%** | **84,13%** | **88,29%** | **72,35%** |
| OCR (default) | 35,16% | 10,92% | 32,50% | 41,38% | 14,00% |
| OCR (set-up) | 48,41% | 18,83% | 46,89% | 60,18% | 26,19% |
| FOTS | 52,55% | 20,74% | 57,00% | 72,92% | 24,99% |
| CRNN | 80,99% | 15,70% | 80,30% | 79,32% | 58,00% |

ASTER can complement text detectors improving their accuracy by correcting text orientation. The recognition network is trained on two synthetic datasets [13], while the rectification network is trained by the gradients obtained at the end of the recognition, lacking the need for annotations.

CRNN [20] consists of three main components integrated into a single framework. The convolutional layers, which extract feature maps from an image, the recurrent layers that predict a label for the extracted frames and the final transcription layer, which transcripts the predictions into a readable character sequence. The model is trained on synthetic datasets [13] and based on the VGG-Very Deep Architecture.

The FOTS method [17] focuses on retrieving and recognizing text regions with incidental text simultaneously. This approach shares features among the two tasks, reducing computational time and resulting in a single end-to-end trainable network. It uses synthetic datasets [13] before training the network and fine-tuning the model.

We also placed a strong emphasis on OCR approaches due to the high presence of watermarks and machine-printed text in our custom dataset. In our Tor based images, machine-printed text often appears in images with clear backgrounds and big font sizes. Copyright disclaimers are also often found in customized borders, separate from the rest of the image's content. OCR approaches peform well in these environments [23].

The chosen OCR approach can be executed with the default parameters, which performs automatic page segmentation using the legacy engine but no orientation and script detection.

This algorithm can also be adapted for the treatment of the images (as a single line, word or character) and the type of engine used to recognize the character (LSTM neural networks or the legacy engine). We set these parameters so that we treat each image as a single line of text, due to the fact that each cropped image only contains one word in the used datasets. We also used LSTM as the engine mode, creating a setup different from the default algorithm execution.

In order to test all methods, we used five different datasets. The SVT dataset [22], which contains 647 text regions, the ICDAR 2013 [15] and 2015 [14] datasets, which hold 1093 and 2096 images respectively, and the IIIT5K-Words dataset [18], with a total of 3000 images, were chosen as relevant datasets for the text recognition task. Additionally, we used a small subset of our own custom dataset [5], named "Custom TOIC", consisting of 100 images gathered from the Tor Darknet with a total of 1112 text regions. We only chose the regions labelled as "legible", reducing the quantity of images to 675 for these experiments. Fig. 2 illustrates some of the most relevant images.


Figure 2: Image examples from the used datasets

After running all the methods on the datasets, we proposed the use of three string comparison methods, Jaro-Winkler [9], Ratcliff-Obershelp [19] and Levenshtein [16] to increase the performance of the OCR approach, by matching the transcribed results against strings found in lexicons.

Each of these string methods measures word close-

Table 2: OCR precision results using lexicons on Custom TOIC

| Dictionary | Tor-specific | | | Generic | | |
|---|---|---|---|---|---|---|
| Comparison | Ratcliff | Levenshtein | Jaro | Ratcliff | Levenshtein | Jaro |
| Default | 13,40% | **16,50%** | 13,40% | 11,90% | **16,70%** | 11,90% |
| Set-up | 22,20% | **25,50%** | 22,20% | 19,50% | **21,10%** | 19,50% |
| Speed (s) | 0,42 | 2,44 | **0,30** | 32,00 | 170,03 | **21,51** |

ness with a different scoring system, resulting in an ordered list with the most similar words per method. We selected them based on these different resulting values, in order to measure their performance in a Tor-specific lexicon.

The Levensthein distance measures closeness as the minimum amount of string operations such as substitutions or additions needed to turn the first string into the second, meaning a lower number indicates a higher match.

The Ratcliff / Obershelp approach compares strings by doubling the length of matched character groups and dividing it by the summed length of both words.

Finally, the Jaro-Winkler word similarity matches characters if they are found in the compared string at a distance less than half of its length. This approach penalizes less errors at the end of the string.

We tested each approach using two dictionaries, one Tor-specific and one English-based, containing 5.570 and 471.376 words respectively and measuring the average time taken by each word match.

Afterwards, we applied ASTER's text rectification feature in all the datasets in order to enhance the performance of the algorithm, initially with no dictionaries attached. Finally, we merge this approach with the use of the Tor-specific dictionary in our custom dataset, in order to identify and analyze the hardest images and conditions to properly transcribe.

# 4 Experimental Results and Discussion

## 4.1 Experimental setup

We evaluated the methods on an Intel Xeon E5 v3 computer with 128GB of RAM using an NVIDIA Titan Xp GPU. We measured text recognition performance using the precision metric, which is the percentage of fully-matched recognition results to the documented labels of each image.

## 4.2 Initial experiment

The precision results for the recognition methods, without using any dictionaries, are shown in Table 1. The obtained results show that ASTER outperforms the other methods in the state-of-the-art datasets. CRNN comes as second with slightly lower performance. For FOTS, we implemented a publicly available version at `https://github.com/WangXiaoCao/FOTS_two_stage` that is unrelated to the original authors, which is the reason for the result variation in the datasets used.

The chosen OCR approach, Pytesseract4, scored higher than FOTS under custom configurations on the ICDAR 2015 dataset and higher than CRNN on the Custom TOIC dataset, validating our initial hypothesis of OCR focused methods being relevant in these particular environments.

Our custom TOIC dataset contains multiple level annotations from character to words and sentences levels, which can cause lower results when recognizing text inside cropped images.

In contrast, the other datasets are labeled following word-based strategy only, such as SVT and IIIT5K-Words.

## 4.3 OCR and lexicons

As we observed a high volume of images containing watermarks and machine-printed text, we decided to only use OCR in order to analyze the improvements of string matching using dictionaries.

When using dictionaries, we compare the 10 highest scoring words against the transcribed sequence, trying to locate a word match. The results of applying these techniques on our own custom dataset are detailed in Table 2. The Levenshtein method obtained the best results with a 25,0% precision using the Tor-specific corpus on the setup version and a 21,10% on the generic dictionary, but with a very high computational cost. The Jaro-Winkler approach was the most computationally efficient, but scored lower than Levenshtein's. However, when large dictionaries are being used, the Jaro distance could be considered as the most relevant approach.

Using lexicons with the Levenshtein string-matching method, we improved the OCR results. The default OCR improved from a precision result of 10,90% to 16,50%, while the custom setup achieved a precision 25,50% from the original 18,83%. However, the results remain lower than ASTER's, which did not use any lexicon.

## 4.4 Rectification network experiments

Using ASTER's rectification network, we tested all the datasets once more using a different number of iterations. This network corrects an input image, predicting a set of control points from the original image and then performing a Thin-Plate-Spline [6] transformation before generating the rectified image. The new image contains a lower degree of orientation which increases the performance of the recognizers.

Table 3: ASTER precision improvements using the rectification network

| Rectifications | SVT | Custom TOIC | IIIT5K-Words | ICDAR 2013 | ICDAR 2015 |
|---|---|---|---|---|---|
| None | 88,25% | 28,30% | 84,13% | 88,29% | 72,35% |
| 1 | 90,11% | 30,52% | 85,40% | 89,84% | 74,24% |
| 2 | 90,42% | 31,70% | 86,07% | 90,39% | 75,25% |
| 5 | 90,88% | **32,74%** | 87,00% | 90,94% | 76,37% |
| 10 | **91,19%** | **32,74%** | 88,53% | 91,03% | **76,80%** |
| 20 | **91,19%** | **32,74%** | **88,93%** | **91,13%** | **76,80%** |

If the recognition was not the same as the documented label, the image would be rectified and transcribed again as many times as the iterations specified. Table 3 shows our results applying this network to five images, while the visual rectification effect in the images is shown in Fig. 3.

We improved the results up to 4% on each of the datasets before the improvements converged above 10 iterations, with subsequent rectifications not significantly altering the scores. Adding a single correction improved the results 2%.

We found that 5 iterations achieved the best results in our custom TOIC, enhancing the original results of 28,30% to 32,74%.


Figure 3: Resulting images (right) of applying ASTER's rectification network

### 4.5 Rectification network and lexicons

Lastly, we combined our selection of the Levenshtein distance, our Tor-specific lexicon, and the correction approach to improve the results in our custom dataset, as illustrated in Table 4.

When using the Tor-specific lexicon, the 10 highest score matching words are checked against the transcribed text for string comparison. Combining all approaches, we obtain an improvement of 11,40% when recognizing text from Tor-related images using the ASTER method, with each configuration increasing processing time.

Table 4: ASTER results on Custom TOIC Subset using a Tor-specific lexicon

| Lexicon | Rectifications | Precision | Time (s) |
|---|---|---|---|
| No | None | 28,30% | 75,72 |
| No | 1 | 30,52% | 106,56 |
| Yes | None | 34,81% | 330,72 |
| Yes | 1 | 37,04% | 559,50 |
| Yes | 2 | 38,22% | 753,30 |
| Yes | 5 | 39,26% | 1.257,79 |
| Yes | 10 | **39,70%** | 1.912,09 |

### 4.6 Image analysis

After finishing the enhancements on the text recognition task, we take the images that were not correctly recognized and analyze them. We identify five main problems in the text recognition task; similar characters, incorrect labelling, orientation, resolution and other factors. We found no influence due to the color of the cropped words.

Similar characters encapsulates text regions that are close to other sequences, such as common "g" and "9" mistakes. Other examples include the letter "U" being incorrectly detected as "LI", due to their analogous form. Certain font type and size deviated problems can also be associated to this category, as they can make differentiating certain characters a complex task. In most cases, the problems caused by this issue can be reduced with the use of lexicons.

Incorrect labelling refers to the wrong assignment of labels to text regions where the given sequence differs from the documented text. It is a common occurrence that the label does not include certain characters that may be difficult to appreciate at first, only to be properly recognized but not obtain a full match due to the original label not contain-

ing them. Other labelling mistakes, such as writing "Brazil" as "Brasil" or "Name" as "Nam", can also be found in most of the datasets used. While some of these errors can be solved using dictionaries, wrong labelling can reduce scores significantly if left ignored.

Orientation is one of the most relevant problems when performing text recognition. From slightly distorted text due to a particular camera angle, to curved fonts or vertically aligned text, images that contain this type of information are often incorrectly recognized. Although ASTER's rectification network can improve transcription on slightly oriented character sequences, vertical or curved text is still an issue that the proposed network fails to recognize properly. In recent years, this type of text has become more relevant, [8] as it is often missing from common datasets and competitions.

The resolution of the cropped regions is a problem of strong relevance especially in our custom dataset. As we have multiple text regions that can be found in a single image, it is a common occurrence that some of these regions are of lower dimensions than $30 \times 30$ pixels. As most of the methods often re-scale the images to a custom size before processing, text recognition is likely to fail in such images. For this purpose, superresolution techniques can be used for image enhancement.

The last category encapsulates different problems that are common in real-scene images. We found that ASTER did not perform as well in big standalone characters, long words or sentences in images that contained a significant font size difference. Other common problems were found in blurry words, high levels of brightness and certain occluded characters as well as machine printed texts, which was our main reason for using OCR approaches.

## 5 Conclusions

In this paper, we have selected five text-recognition based approaches using three neural network approaches and two OCR configurations, comparing their performance in five datasets with the goal of extracting text found inside images taken from the Tor Darknet.

We have improved the results in the OCR approach by using dictionaries and string-matching methods to enhance text recognition. We obtained a precision of $25,5\%$ in our custom dataset using Pytesseract4, concluding that the use of OCR approaches can be useful in Tor images that contain machine printed text.

In order to enhance recognition results, we used ASTER's rectification network, concluding that five iterations is the most efficient approach when considering computational efficiency. We obtained an increase of 4% precision with this configuration, which is further increased to 10,96% when combining the approach with a Tor-specific lexicon and the Levenshtein distance, which takes a high computational cost.

Combining ASTER's rectification and recognition network with the use of a Tor-based lexicon and the Levenshtein distance, we obtained a recognition precision in our custom dataset of 39,70%. When using using OCR approaches and the same lexicon, we obtained a 25,50%.

As a result of our work, we have identified the best OCR configurations for our Tor-based text recognition goal. We have also analyzed the best string matching methods efficiency and cost-wise, being the Levenshtein and Jaro-Winkler approaches respectively.

Furthermore, we determined oriented text, character similarity and low image resolution as the most challenging conditions for text recognition. We also discovered errors in the documented labels of the dataset, which reduces the algorithm's precision results.

Our future work will be focused on vertical text recognition and low resolution images, as well as more efficient string comparison methods to improve the computational cost of using large lexicons. We will also focus on the detection section of the pipeline and how these areas can be corrected using the rectification network.

## Acknowledgements

This research is supported by the framework agreement between Universidad de León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01. We acknowledge NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This research has also been funded with support from the European Commission under the 4NSEEK project with Grant Agreement 821966. This publication reflects the views only of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein.

## References

[1] Al-Nabki, M. W., Fidalgo, E., Alegre, E., and Fernández-Robles, L. (2019). ToRank: Identifying the most influential suspicious domains in

the tor network. *Expert Systems with Applications*, 123:212 – 226.

[2] Al-Nabki, M. W., Fidalgo, E., Alegre, E., and González-Castro, V. (2017). Detecting emerging products in tor network based on k-shell graph decomposition. *III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC)*, 1:24–30.

[3] Bartz, C., Yang, H., and Meinel, C. (2017). STN-OCR: A single Neural Network for Text Detection and Text Recognition.

[4] Biswas, R., Fidalgo, E., and Alegre, E. (2017). Recognition of service domains on tor dark net using perceptual hashing and image classification techniques. *IET Conference Proceedings*, pages 7–12(5).

[5] Blanco-Medina, P., Fidalgo, E., Alegre, E., and Al-Nabki, M. W. (2018). Detecting textual information in images from onion domains using text spotting. *Actas de las XXXIX Jornadas de Automática*, pages 975–982.

[6] Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585.

[7] Busta, M., Neumann, L., and Matas, J. (2017). Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2204–2212.

[8] Ch'ng, C. K. and Chan, C. S. (2017). Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 935–942. IEEE.

[9] Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string metrics for matching names and records. In *Proc of the KDD Workshop on Data Cleaning and Object Consolidation*, volume 3, pages 73–78.

[10] Fidalgo, E., Alegre, E., González-Castro, V., and Fernández-Robles, L. (2017). 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*, pages 600–609. Springer.

[11] Gangwar, A., Fidalgo, E., Alegre, E., and González-Castro, V. (2017). Pornography and child sexual abuse detection in image and video: A comparative evaluation. In *8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017)*. IET.

[12] Gomez, R., Shi, B., Gomez, L., Numann, L., Veit, A., Matas, J., Belongie, S., and Karatzas, D. (2017). ICDAR 2017 robust reading challenge on coco-text. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1435–1443. IEEE.

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

[14] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., et al. (2015). ICDAR 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE.

[15] Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L. G., Mestre, S. R., Mas, J., Mota, D. F., Almazan, J. A., and De Las Heras, L. P. (2013). ICDAR 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE.

[16] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Insertions and Reversals. Sov*, 6.

[17] Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., and Yan, J. (2018). FOTS: Fast Oriented Text Spotting with a Unified Network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685.

[18] Mishra, A., Alahari, K., and Jawahar, C. (2012). Scene text recognition using higher order language priors. In *BMVC-British Machine Vision Conference*. BMVA.

[19] Ratcliff, J. W. and Metzener, D. E. (1988). Pattern-matching-the gestalt approach. *Dr Dobbs Journal*, 13(7):46.

[20] Shi, B., Bai, X., and 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] Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., and Bai, X. (2018). Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*.

[22] Wang, K. and Belongie, S. (2010). Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer.

[23] Ye, Q. and Doermann, D. (2015). Text detection and recognition in imagery: A survey.

*IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500.