Enhancing Text Spotting with a Language Model and Visual Context Information

Ahmed Sabir [a,1], Francesc Moreno-Noguer [b] and Lluís Padró [a]
[a] *TALP Research Center, Universitat Politècnica de Catalunya*
[b] *Institut de Robòtica i Informàtica Industrial*

Abstract. This paper addresses the problem of detecting and recognizing text in images acquired 'in the wild'. This is a severely under-constrained problem which needs to tackle a number of challenges including large occlusions, changing lighting conditions, cluttered backgrounds and different font types and sizes. In order to address this problem we leverage on recent and successful developments in the cross-fields of machine learning and natural language understanding. In particular, we initially rely on off-the-shelf deep networks already trained with large amounts of data and that provide a series of text hypotheses per input image. The outputs of this network are then combined with different priors obtained from both the semantic interpretation of the image and from a scene-based language model. As a result of this combination, the performance of the original network is consistently boosted. We validate our approach on ICDAR'17 shared task dataset.

Keywords. Text Spotting; Deep Learning; Language Model; Semantic Visual Context; Data Fusion.

1. Introduction

Reading letters and understanding the underlying words are very important tasks in todays technological society. Written and printed texts are everywhere e.g. in form of newspapers, documents, street signs, or logos in merchandising. Machine reading has been one of the most active areas of research in computer vision for decades, and is very well represented by current Optical Character Recognition systems (OCR) which is an almost infallible technology for text reading. However, the success of OCR systems is restricted to scanned documents with relatively simple and clean backgrounds. Texts appearing in images 'in the wild' exhibit a large variability in appearances, and can prove to be challenging even for state-of-the-art OCR methods. While many computer vision algorithms are able to recognize objects in these images, understanding and recognizing the text in these images in a robust manner still remains an open problem.

The so-called 'text spotting' problem involves solving two sub-tasks: word detection and word recognition. The goal of the detection stage is to localize, within the image, the bounding box around a candidate text. Candidate words in the bounding boxes are then aimed to be recognized during the text recognition stage. In addition, there are two approaches for performing text recognition, by either using a dictionary with a fixed

[1] Corresponding Author: asabir@cs.upc.edu

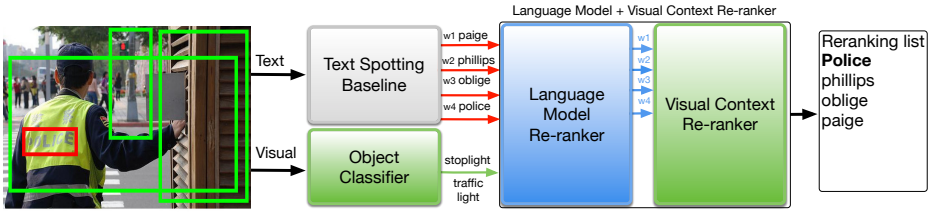

Figure 1. Overview of the proposed approach. Given the predicted output of an off-the-shelf text spotting system (baseline), we introduce priors from a language model and from the contextual visual information to re-rank the initially predicted words. In this example, the word "police" was initially ranked in the 4-th position by [1], a state-of-the-art text spotting algorithm. After considering the language and visual context priors, the word "police" is correctly ranked in the first position of the most likely predicted words.

lexicon or considering a lexicon-free strategy. In this paper, we focus on improving the text-recognition stage, in the particular case of considering a fixed lexicon.

For this purpose we propose an approach that intends to fill the gap between language and vision in scene text recognition. Most recent state-of-art focus on automatically detecting and recognizing text in natural images from a purely computer vision perspective. In this work, we tackle the same problem but leveraging also on natural language understanding techniques. Our approach seeks to integrate prior information to the text spotting pipeline. This prior information biases the initial ranking of a set of potential words, yielded by a pre-trained deep neural network. The final word re-ranking is based on the semantic relatedness between this prior information and the spotted word. As shown in the example of Figure 1, the final re-ranking word *police* is biased by the visual context information *stoplight*. This hybrid approach between deep learning and classical statistical modeling opens the possibility to produce accurate results with very simple models.

Our contributions are therefore the following: First, we introduce an independent language model into the text spotting pipeline. We show that by introducing a second dictionary, and without the need to perform additional training, we can improve the word ranking from the hypotheses done by an external deep model. In addition, we overcome the baseline limitation of false detection of short word [2]. Secondly, we show that by adding the visual information to the text spotting system, we can relate the spotted text to its visual scene. We experimentally demonstrate that by understanding the semantic relatedness between spotted text and its visual context information we can significantly boost the final recognition accuracy.

The rest of the paper is organized as follows: Sections 2 and 3 describe related work and our proposed pipeline. Sections 4 and 5 introduce the two external prior knowledge we use, unigram frequencies and visual context information. Sections 6 and 7 present experimental validation of our approach on a variety of publicly available standard datasets. Finally, Section 8 summarizes and specifies future work.

2. Related work

Text spotting (or end-to-end text recognition), refers to the problem of automatically detecting and recognizing text in images in the wild. This problem can be tackled by either a lexicon-based or a lexicon-free perspective. Lexicon-based recognition methods

use a pre-defined dictionary as a reference to guide the recognition. Lexicon free methods (or unconstrained recognition techniques), predict characters without relying on any dictionary. The first lexicon free text spotting system was proposed by [3]. The system extracted character candidates via maximally stable extremal regions (MSER) and eliminated non-textual ones through a trained classifier. The remaining candidates were fed into a character recognition module, trained using a large amount of synthetic data.

More recently, several deep learning alternatives have been proposed. For instance, PhotoOCR [4] uses a Deep Neural Network (DNN) that performs end-to-end text spotting using histograms of oriented gradients as input of the network. It is a lexicon-free system able to read characters in uncontrolled conditions. The final word re-rank is performed by means of two language models, namely a character and an N-gram language model. This approach combined two language models, a character based bi-gram model with compact 8-gram and 4-gram word-level model. Another approach employed language model for final word re-ranking [5]. The top-down integration can tolerate the error in text detection or mis-recognition.

The first attempt using Convolutional Neural Network (CNN) was proposed by [6], that pre-trained with unsupervised learning feature. The word re-ranking score is based on post-processing techniques, such as non-maximal suppression (NMS) and beam search. Another CNN based approach is that of [7], which applies a sliding window over CNN features that use a fixed-lexicon based dictionary. This is further extended in [1], through a deep architecture that allows feature sharing. In [8] the problem is addressed using a Recurrent CNN, a novel lexicon-free neural network architecture that integrates Convolutional and Recurrent Neural Networks for image based sequence recognition. Finally most recently, [9] introduce a CNN with connectionist temporal classification (CTC) [10] to generate the final label sequence without a sequence model such as RNN, LSTM. This approach use stacked convolutional to capture the dependencies of the input sequence. This algorithm can be integrated with both methods, fixed lexicon and lexicon free based recognition. Deep learning fixed lexicon based methods, however, have two drawbacks. First, they rely on large datasets to train. Secondly, they require a non-trainable fixed dictionary.

In this work we show that considering a hybrid approach that re-ranks the output of a deep learning approach based on a classical statistical model opens the possibility to overcome both these limitations, yielding simpler models with improved results.

3. General Approach and Baseline System

Text recognition approaches can be divided in two categories: (a) character level recognition methods that rely on a single character classifier plus some kind of sequence modeling (ngram models, LSTM, etc), and (b) word level recognition techniques that aim to classify the image as a whole.

In both cases, the system can be configured to produce the k most likely words given the input image. Our approach focuses on re-ranking that list using external knowledge. We will use two sources of information: (1) general language information such as word frequencies, and (2) visual context of the image in which the text was located.

The used baseline model is an off-the-shelf CNN [1] with fixed-lexicon based recognition. It uses a dictionary containing around 90K word forms. The baseline is trained

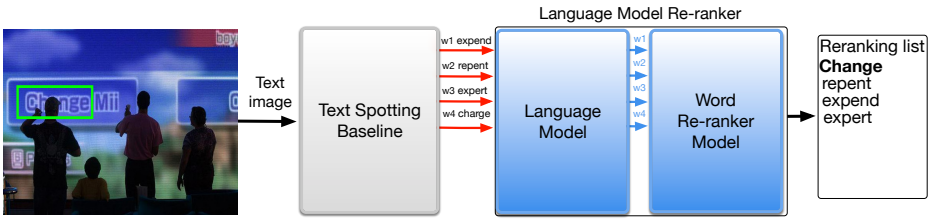

Figure 2. Introducing a language model prior. We use a language model trained on an external corpora (frequency count) to re-rank the output probabilities of the text spotting baseline algorithm. Note that the blue box corresponds to the 'Language Model Re-Ranker' box in Fig. 1.

on a synthetic dataset [11] created from this dictionary. The output of the CNN is a vector of 90K softmax probabilities, one for each word of the dictionary, from which we will extract the words with highest score and their corresponding probabilities. We next describe each of these ingredients.

4. Language model

Let us denote the baseline probabilities of the k most likely words w produced by the CNN [1] by:

$$P_0(w) = p(w|\text{CNN}) \tag{1}$$

The first modality to re-rank the output of this CNN is based on external linguistic information. In this case, we leverage on the word unigram probabilities computed from a text corpora [12]. Based on this corpus we build a unigram language model (ULM), which captures the word frequencies of the dictionary, and aims to increase the probability of most common words. We then compute the combined CNN and ULM word probability as a simple product of unary probabilities:

$$P_1(w) = p(w|\text{CNN}) \times p(w|\text{ULM}) \tag{2}$$

This hybrid approach opens the possibility of introducing higher-order trainable language models, e.g. bi-grams or tri-grams. Figure 2 illustrates the proposed N-gram language model, which is concatenated to the output of the baseline text spotting algorithm.

5. Visual context bias information

Our second prior to re-rank the CNN baseline output is based on the visual context information about the image in which the text was located. For this, we use an independent visual object classifier, and devise a strategy to reward candidate words that are more semantically related to the objects detected in the image. For instance, as shown in Figure 3, if the original CNN baseline produces several candidate words, the correct word *police* can be re-ranked to the top position by using the fact that it is semantically related to objects in the context such as *traffic signal* or *stoplight*. We next describe how this relation is learned.

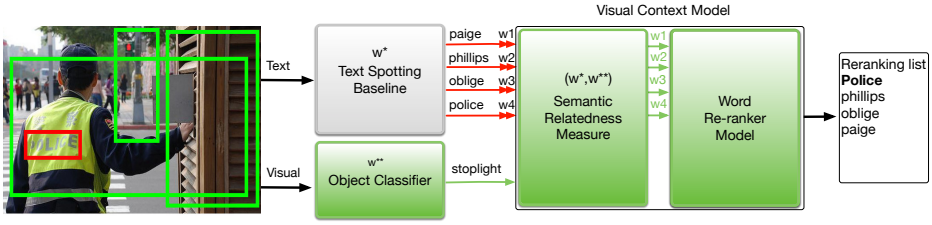

Figure 3. Introducing visual context information into the text spotting pipeline. Our approach uses the semantic relation between a word and its visual context to re-rank the most probable word provided by a baseline text spotting algorithm.

5.1. Object classifier

In order to exploit image context information we use state-of-the-art object classifiers. We considered two pre-trained CNN classifiers: ResNet [13] and GoogLeNet [14]. The output of these classifiers is a 1000-dimensional vector with the probabilities of 1000 object classes. In this work we only consider most likely object of the context, but the proposed approach can be easily extended to use more than one.

5.2. Semantic similarity

Once the objects in the image have been detected, we compute their semantic relatedness with the candidate words based on their word-embeddings [15]. Specifically, let us denote by $\vec{w}$ and $\vec{c}$ the word-embeddings of a candidate word and the name of the most likely object detected in the image, respectively. We then compute their similarity using the cosine of the embeddings:

$$sim(w,c) = \frac{\vec{w} \cdot \vec{c}}{|\vec{w}| \cdot |\vec{c}|} \tag{3}$$

We next convert the similarity score in a probability value, in order to integrate it into our re-ranking model. Following [16], we compute the conditional probability from similarity as:

$$P(w|c) = P(w)^{\alpha} \quad \text{where } \alpha = \left(\frac{1-sim(w,c)}{1+sim(w,c)}\right)^{1-P(c)} \tag{4}$$

$P(w)$ is the probability of the word in general language (obtained from the unigram model), and $P(c)$ is the probability of the context object (obtained from the object classifier).

Once we have the probability of a candidate word conditioned to the visual context objects, we define $P(w|\text{VCI}) = P(w|c)$ and use it to re-rank the output of the baseline CNN in the same way we did with the unigram model:

$$P_2(w) = P(w|\text{CNN}) \times P(w|\text{VCI}) \tag{5}$$

Finally, as shown in Figure 4, we combine the two re-rankers into a single model. Both the word-frequency model and the visual context model are combined to re-rank

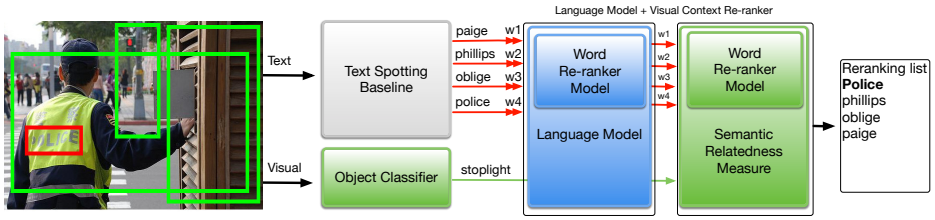

Figure 4. Illustration of the combined approaches, that exploit visual context information and a language model prior. Each model, sequentially re-ranks the work hypotheses produced by an initial Text Spotting Baseline.

the candidate words produced by the baseline CNN. The final probability of a candidate word w is computed as:

$$P_3(w) = p(w|\text{CNN}) \times p(w|\text{ULM}) \times p(w|\text{VCI}) \tag{6}$$

6. Experiments and Results

In this section we evaluate the performance of the proposed approaches on the **ICDAR-2017-Task3 (end-to-end)** dataset [17].

This dataset is based on Microsoft COCO [18] (Common Objects in Context), which consists of 63,686 images, and 145,589 text instances (annotations of the images). COCO-text was not collected with text recognition in mind, therefore, not all images contain textual annotations. The *ICDAR-2017 Task3* aims for end-to-end text spotting (i.e. both detection and recognition). Thus, this dataset includes whole images, and the texts in them may appear rotated, distorted, or partially occluded. Since we focus only on text recognition, we use the ground truth detection as a golden detector to extract the bounding boxes from the full image. The dataset consists of 43,686 full images with 145,859 text instances for training, and 10,000 images with 27,550 instances for validation.

6.1. Preliminaries

For evaluation, we did not use the standard evaluation protocol proposed by [19] which is adopted in most state-of-the-art testing benchmarks where words with less than three characters and non-alphanumerical characters are not considered. This protocol was introduced to overcome the false positives on short words that most current state-of-the-art struggle with, including our Baseline. However, we overcome this limitation by introducing the language model re-ranker. Thus, we consider all cases in the dataset, and words with less than three characters are also evaluated.

In all cases, we use the pre-trained CNN [1] as a baseline to extract the initial list of word hypotheses. Since these Baseline need to be fed with the cropped words, when evaluating on the ICDAR-2017-Task3 dataset we will use the ground truth bounding boxes of the words.

Table 1. Comparison of End to End 2- to 10-best Accuracy COCO-text (%)

Model	$k=2$		$k=3$		$k=5$		$k=10$	
	full	dic	full	dic	full	dic	full	dic
CNN [1]	*full: 21.1 dictionary: 58.6*							
CNN+LM_{3M}	21.8	60.6	22	61.3	21.6	60.1	21	58.5
CNN+$VCI_{GoogLeNet}$	22.1	61.4	22.4	62.2	22.4	62.2	22.2	61.7
CNN+$VCI_{ResNet152}$	21.1	61.5	**22.5**	**62.5**	22.4	62.4	22.2	61.7
CNN+LM_{3M}+$VCI_{GoogLeNet}$	21.8	60.8	22.2	61.7	21.7	60.2	21.3	59.2
CNN+LM_{3M}+$VCI_{ResNet152}$	21.8	61.8	22.2	61.7	21.3	60.0	21.3	59.2

Note: The baseline CNN which we use as input was able to solve 58.6% of the cases.

6.2. Experiment with Language model

6.2.1. Cropped Words Dataset

We trained two unigram models on different corpora. The first model was trained on *opensubtitles*, a database of subtitles for movies [20]. The corpus contains around 3 million word types (including numbers and other alphanumeric combinations). Secondly, we trained a larger model with the *google book n-gram* corpus, that contains around 5 million word type frequencies from american-british literature books. However since the test dataset contains numbers, the accuracy was lower than that obtained using the *opensubtitles* corpus.

In this experiment, we extract the $k = 2 \ldots 10$ most likely words with their respective probabilities from the CNN baseline [1], and re-rank them using the unigram frequencies acquired from the corpus. It is worth remarking that the unigram model is fast and easy to train, and that it can be tuned to specific application domains by simply selecting a domain corpora as training data. In this way, the baseline CNN output can be biased towards the most suitable interpretation for the target application.

We evaluated our model on the "end-to-end COCO-Text ICDAR2017" competition dataset. However, since our baseline works on cropped words and we are not evaluating a whole end-to-end system but only the influence of adding external knowledge, we extract ground-truth bounding boxes and use them as input to the baseline.

The first two rows in Table 1 report the results on this dataset. We present two different accuracy metrics: *full* columns correspond to the accuracy on the whole dataset, while *dictionary* columns correspond to the accuracy over the solvable cases (i.e. those where the target word is among the 90K-words of the dictionary used to train the baseline CNN, which corresponds to 43.3% of the whole dataset). We provide the results using different amounts of the k-best candidates from the baseline CNN output ($k = 2, 3, 5, 10$).

In this case, the language model improves the accuracy of the baseline. The best results are obtained by considering $k = 3$ which improve the baseline 0.9%. Detailed analysis shows that the language model helps the system to overcome recognition errors in numbers and common short words.

6.3. Experiment with Visual Context Information

We next re-rank the most probable word based on visual context information. We use the official "COCO-Text ICDAR 2017 end-to-end reading robust competition" dataset. As

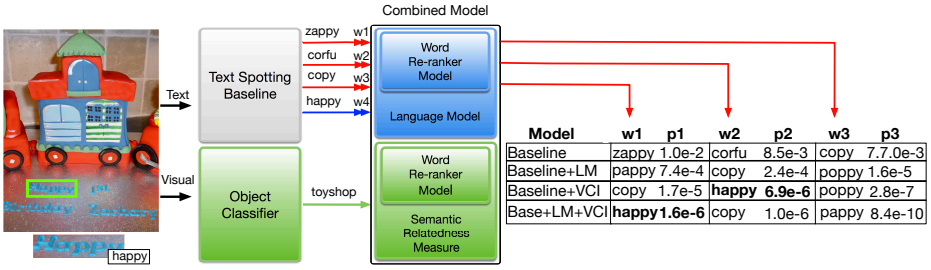

Figure 5. Illustration of the final re-ranking result of each model. The combined model re-ranks the candidate word, and fixes the individual modalities.

in the language model experiment, we used ground-truth bounding boxes as input to the CNN. However, in this case, the whole image is used as input to the object classifier.

In order to extract the visual context information we consider two different pre-trained object classifiers: Resnet [13] and GoogLenet [14]. The object classifier produces a number of hypotheses words but we only use the most probable one, though the approach can be easily extended to use more.

Summarizing, in this experiment we re-rank the k-best candidate words based on their semantic similarity with the most likely word produced by the object classifier.

The visual context information yields a remarkable accuracy improvement. Results for the visual context bias approach are better than the baseline CNN and than the combination of the baseline with the language model, as shown in third and fourth rows of Table 1.

For instance, in Figure 6 top-right example, the system correctly re-ranks the word according to the semantic similarity between the true candidate word *pay* (instead of *posy*) thanks to the detected visual *parking*.

We evaluated the model from 2 to 10-best words output from the CNN. The baseline was improved in all cases up to $k = 10$, but we achieved the best result with $k = 3$ due to the fact that the majority of right candidate words were in that range.

6.4. Experiment with Combined Model

The combined approach between the language model and visual context information has a positive impact on the accuracy, though not as large as we expected. As shown in Table 1, results for the combined approach are lower than only with the visual re-ranker. The reason is that the language model re-ranker contribution is strong enough to turn the correct decision of the visual re-ranker. This problem can be solved by adjusting the influence of each re-ranker, which we plan to explore in our future work. Figure 5 shows a successful case of the combined model.

7. Discussion

The visual context information re-ranks potential candidate words based on the semantic relatedness. However, there are some cases when there is no direct semantic relation between the visual context and candidate word. Thus, the final re-ranking score is based on the certainty of each model or both combined.

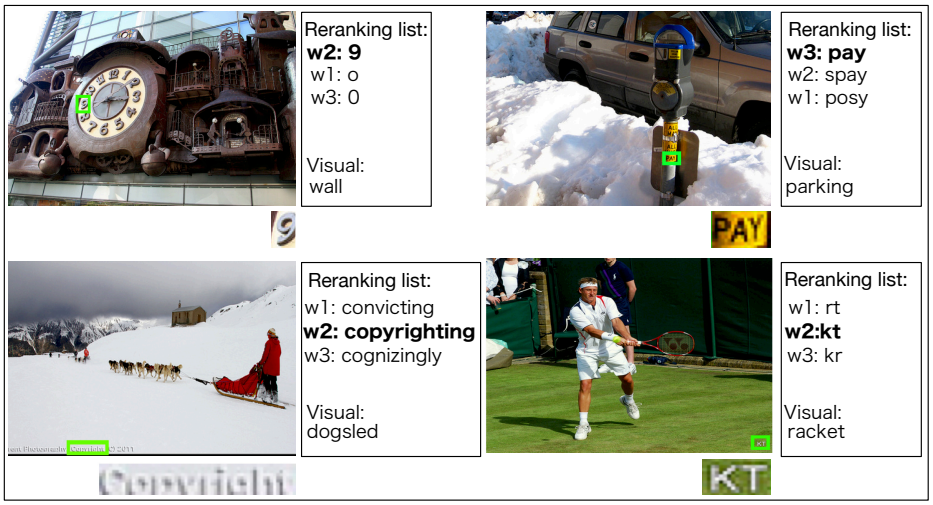

Figure 6. Some examples of visual context re-ranker. The top-two examples are successful results of the visual context re-ranker. The top two examples are a re-ranking results based on semantic relatedness between the text image and its visual. The bottom two cases are examples of words has no semantic correlation with the visual information.(Bold font words indicate the ground truth)

For instance, if the re-ranked candidate word by the language model is *ken*, which is semantically not related with the visual context *monitor*. The visual re-ranker re-ranks the candidate word to the most semantically related to the visual context information. In this example, the candidate word *key* is more semantically related to *monitor*.

Another example, as shown in Figure 5, the combined re-ranker model is not only able to re-rank the right word, but also to correct the mis-recognized word *copy* from the visual model. The combined re-ranker balances the biases between the two models, as in this case where there is no direct semantic relation between the language and visual content.

One limitation of this approach is that when the candidate word has no semantic relation with the visual context, the VCI contribution may be misleading. This could be tackled by training the word embeddings on the training data instead of (or in addition to) general text.

8. Conclusion

In this paper we have proposed a simple scheme to improve the accuracy of pre-trained text spotting algorithms. Using priors based on a language model trained over an independent corpus and on the visual context semantic information of the image, we have shown that the accuracy of a state of the art deep architecture of [1], can be boosted up to 4 percentage-points on standard benchmarks. The proposed approach is intended to be used as a drop-in replacement for any text-spotting algorithm that ranks the output words [7, 9, 21–23]. In this work, the fusion of the different modalities is made by a simple product of probabilities. In the future, we plan to explore more elaborate fusion schemes that can automatically discover the more reliable prior.

References

[1] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, 2016.

[2] M. Jaderberg, "Deep learning for text spotting," Ph.D. dissertation, University of Oxford, 2015.

[3] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference on Computer Vision*. Springer, 2010.

[4] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[5] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[6] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012.

[7] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting." in *ECCV (4)*, 2014.

[8] B. Shi, X. Bai, and C. Yao, "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*, 2016.

[9] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Reading scene text with attention convolutional sequence modeling," *arXiv preprint arXiv:1709.04303*, 2017.

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

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

[12] G. A. Miller and J. A. Selfridge, "Verbal context and the recall of meaningful material," *The American journal of psychology*, vol. 63, no. 2, 1950.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013.

[16] S. Blok, D. Medin, and D. Osherson, "Probability from similarity," in *AAAI Spring Symposium on Logical Formalization of Commonsense Reasoning*, 2003.

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

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014.

[19] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[20] P. Lison and J. Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," 2016.

[21] S. K. Ghosh, E. Valveny, and A. D. Bagdanov, "Visual attention models for scene text recognition," *arXiv preprint arXiv:1706.01487*, 2017.

[22] B. Shi, X. Bai, and C. Yao, "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*, vol. 39, no. 11, 2017.

[23] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.