Single Shot Scene Text Retrieval

Lluís Gómez(✉), Andrés Mafla, Marçal Rusiñol, and Dimosthenis Karatzas

Computer Vision Center, Universitat Autònoma de Barcelona,
Edifici O, 08193 Bellaterra (Barcelona), Spain
{lgomez,andres.mafla,marcal,dimos}@cvc.uab.es

Abstract. Textual information found in scene images provides high level semantic information about the image and its context and it can be leveraged for better scene understanding. In this paper we address the problem of scene text retrieval: given a text query, the system must return all images containing the queried text. The novelty of the proposed model consists in the usage of a single shot CNN architecture that predicts at the same time bounding boxes and a compact text representation of the words in them. In this way, the text based image retrieval task can be casted as a simple nearest neighbor search of the query text representation over the outputs of the CNN over the entire image database. Our experiments demonstrate that the proposed architecture outperforms previous state-of-the-art while it offers a significant increase in processing speed.

Keywords: Image retrieval · Scene text · Word spotting
Convolutional neural networks · Region proposals networks · PHOC

1 Introduction

The world we have created is full of written information. A large percentage of everyday scene images contain text, especially in urban scenarios [1,2]. Text detection, text recognition and word spotting are important research topics which have witnessed a rapid evolution during the past few years. Despite significant advances achieved, propelled by the emergence of deep learning techniques [3], scene text understanding in unconstrained conditions remains an open problem attracting an increasing interest from the Computer Vision research community. Apart from the scientific interest, a key motivation comes by the plethora of potential applications enabled by automated scene text understanding, such as improved scene-text based image search, image geo-localization, human-computer interaction, assisted reading for the visually-impaired, robot navigation and industrial automation to mention just a few.

The textual content of scene images carries high level semantics in the form of explicit, non-trivial data, which is typically not possible to obtain from analyzing the visual information of the image alone. For example, it is very challenging,

L. Gómez and A. Mafla—Contributed equally to this work.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11218, pp. 728–744, 2018.
https://doi.org/10.1007/978-3-030-01264-9_43

even for humans, to automatically label images such as the ones illustrated in Fig. 1 as tea shops solely by their visual appearance, without actually reading the storefront signs. Recent research actually demonstrated that a shop classifier ends up automatically learning to interpret textual information, as this is the only way to distinguish between businesses [4]. In recent years, several attempts to take advantage of text contained in images have been proposed not only to achieve fine-grained image classification [5,6] but to facilitate image retrieval.

Mishra *et al.* [7] introduced the task of scene text retrieval, where, given a text query, the system must return all images that are likely to contain such text. Successfully tackling such a task entails fast word-spotting methods, able to generalize well to out-of-dictionary queries never seen before during training.

Fig. 1. The visual appearance of different tea shops' images can be extremely variable. It seems impossible to correctly label them without reading the text within them. Our scene text retrieval method returns all the images shown here within the top-10 ranked results among more than 10,000 distractors for the text query "tea".

A possible approach to implement scene text retrieval is to use an end-to-end reading system and simply look for the occurrences of the query word within its outputs. It has been shown [7] that such attempts generally yield low performance for various reasons. First, it is worth noting that end-to-end reading systems are evaluated on a different task, and optimized on different metrics, opting for high precision, and more often than not making use of explicit information about each of the images (for example, short dictionaries given for each image). In contrary, in a retrieval system, a higher number of detections can be beneficial. Secondly, end-to-end systems are generally slow in processing images, which hinders their use in real-time scenarios or for indexing large-scale collections.

In this paper we propose a real-time, high-performance word spotting method that detects and recognizes text in a single shot. We demonstrate state of the art performance in most scene text retrieval benchmarks. Moreover, we show that our scene text retrieval method yields equally good results for in-dictionary and out-of-dictionary (never before seen) text queries. Finally, we show that the resulting method is significantly faster than any state of the art approach for word spotting in scene images.

The proposed architecture is based on YOLO [8,9], a well known single shot object detector which we recast as a PHOC (Pyramidal Histogram Of Characters) [10,11] predictor, thus being able to effectively perform word detection

and recognition at the same time. The main contribution of this paper is the demonstration that using PHOC as a word representation instead of a direct word classification over a closed dictionary, provides an elegant mechanism to generalize to any text string, allowing the method to tackle efficiently out-of-dictionary queries. By learning to predict PHOC representations of words the proposed model is able to transfer the knowledge acquired from training data to represent words it has never seen before.

The remainder of this paper is organized as follows. Section 2 presents an overview of the state of the art in scene text understanding tasks, Sect. 3 describes the proposed architecture for single shot scene text retrieval. Section 4 reports the experiments and results obtained on different benchmarks for scene text based image retrieval. Finally, the conclusions and pointers to further research are given in Sect. 5.

2 Related Work

The first attempts at recognizing text in scene images divided the problem in two distinguished steps, text detection and text recognition. For instance, in the work of Jaderberg *et al.* [12] scene text segmentation was performed by a text proposals mechanism that was later refined by a CNN that regressed the correct position of bounding boxes. Afterwards, those bounding boxes were inputed to a CNN that classified them in terms of a predefined vocabulary. Gupta *et al.* [13] followed a similar strategy by first using a Fully Convolutional Regression Network for detection and the same classification network than Jaderberg for recognition. Liao *et al.* [14,15] used a modified version of the SSD [16] object detection architecture adapted to text and then a CRNN [17] for text recognition. However, breaking the problem into two separate and independent steps presented an important drawback since detection errors might significantly hinder the further recognition step. Recently, end-to-end systems that approach the problem as a whole have gained the attention of the community. Since the segmentation and recognition tasks are highly correlated from an end to end perspective, in the sense that learned features can be used to solve both problems, researchers started to jointly train their models. Buvsta *et al.* [18] proposed to use a Fully Convolutional Neural Network for text detection and another module that employed a CTC (Connectionist Temporal Classification) for text recognition. Both modules were first trained independently and further joined together in order to make an end-to-end trainable architecture. Li *et al.* [19] proposed a pipeline that included a CNN to obtain text region proposals followed by a region feature encoding module that is the input to an LSTM to detect text. The detected regions are the input to another LSTM which outputs features to be decoded by a LSTM with attention to recognize the words. In that sense, we strongly believe that single shot object detection paradigms such as YOLO [9] can bring many benefits to the field of scene text recognition by having a unique architecture that is able to locate and recognize the desired text in an unique step.

However, the scene text retrieval problem slightly differs from classical scene text recognition applications. In a retrieval scenario the user should be able to cast whatever textual query he wants to retrieve, whereas most of recognition approaches are based on using a predefined vocabulary of the words one might find within scene images. For instance, both Mishra *et al.* [7], who introduced the scene text retrieval task, and Jaderberg *et al.* [12], use a fixed vocabulary to create an inverted index which contains the presence of a word in the image. Such approach obviously limits the user that does not have the freedom to cast out of vocabulary queries. In order to tackle such problem, text string descriptors based on n-gram frequencies, like the PHOC descriptor, have been successfully used for word spotting applications [10,20,21]. By using a vectorial codification of text strings, users can cast whatever query at processing time without being restricted to specific word sets.

3 Single Shot Word Spotting Architecture

The proposed architecture, illustrated in Fig. 2, consists in a single shot CNN model that predicts at the same time bounding boxes and a compact text representation of the words within them. To accomplish this we adapt the YOLOv2 object detection model [8,9] and recast it as a PHOC [10] predictor. Although the proposed method can be implemented on top of other object detection frameworks we opted for YOLOv2 because it can be up to $10\times$ faster than two-stage frameworks like Faster R-CNN [22], and processing time is critical for us since we aim at processing images at high resolution to correctly deal with small text.

The YOLOv2 architecture is composed of 21 convolutional layers with a leaky ReLU activation and batch normalization [7] and 5 max pooling layers. It uses 3×3 filters and double the number of channels after every pooling step as in VGG models [17], but also uses 1×1 filters interspersed between 3×3 convolutions to compress the feature maps as in [9]. The backbone includes a pass-through layer from the second convolution layer and is followed by a final 1×1 convolutional layer with a linear activation with the number of filters matching the desired output tensor size for object detection. For example, in the PASCAL VOC challenge dataset (20 object classes) it needs 125 filters to predict 5 boxes with 4 coordinates each, 1 objectness value, and 20 classes per box ($(4 + 1 + 20) \times 5 = 125$). The resulting model achieves state of the art in object detection, has a smaller number of parameters than other single shot models, and features real time object detection.

A straightforward application of the YOLOv2 architecture to the word spotting task would be to treat each possible word as an object class. This way the one hot classification vectors in the output tensor would encode the word class probability distribution among a predefined list of possible words (the dictionary) for each bounding box prediction. The downside of such an approach is that we are limited in the number of words the model can detect. For a dictionary of 20 words the model would theoretically perform as well as for the 20 object classes of the PASCAL dataset, but training for a larger dictionary (e.g.

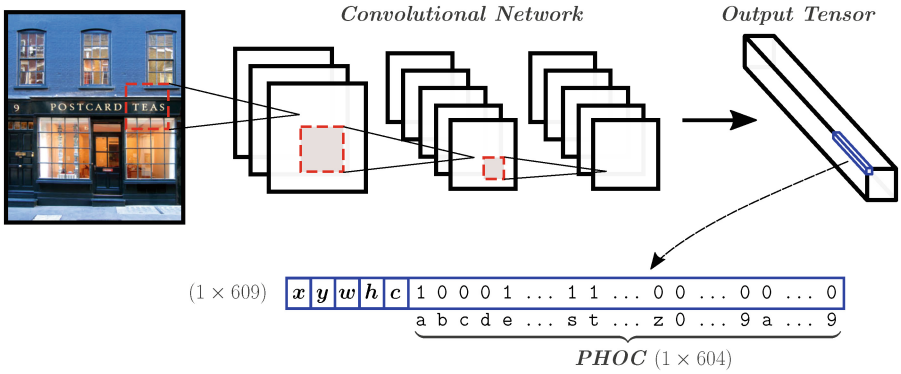

Fig. 2. Our Convolutional Neural Network predicts at the same time bounding box coordinates x, y, w, h, an objectness score c, and a pyramidal histogram of characters (PHOC) of the word in each bounding box.

the list of 100,000 most frequent words from the English vocabulary [12]) would require a final layer with 500,000 filters, and a tremendous amount of training data if we want to have enough samples for each of the 100,000 classes. Even if we could manage to train such a model, it would still be limited to the dictionary size and not able to detect any word not present on it.

Instead of the fixed vocabulary approach we would like to have a model that is able to generalize to words that were not seen at training time. This is the rationale behind casting the network as a PHOC predictor. PHOC [10] is a compact representation of text strings that encodes if a specific character appears in a particular spatial region of the string (see Fig. 3). Intuitively a model that effectively learns to predict PHOC representations will implicitly learn to identify the presence of a particular character in a particular region of the bounding box by learning character attributes independently. This way the knowledge acquired from training data can be transfered at test time for words never observed during training, because the presence of a character at a particular location of the word translates to the same information in the PHOC representation independently of the other characters in the word. Moreover, the PHOC representation offers unlimited expressiveness (it can represent any word) with a fixed length low dimensional binary vector (604 dimensions in the version we use).

In order to adapt the YOLOv2 network for PHOC prediction we need to address some particularities of this descriptor. First, since the PHOC representation is not a one hot vector we need to get rid of the softmax function used by YOLOv2 in the classification output. Second, since the PHOC is a binary representation it makes sense to squash the network output corresponding to the PHOC vector to the range 0...1. To accomplish this, a sigmoid activation function was used in the last layer. Third, we propose to modify the original YOLOv2 loss function in order to help the model through the learning process.

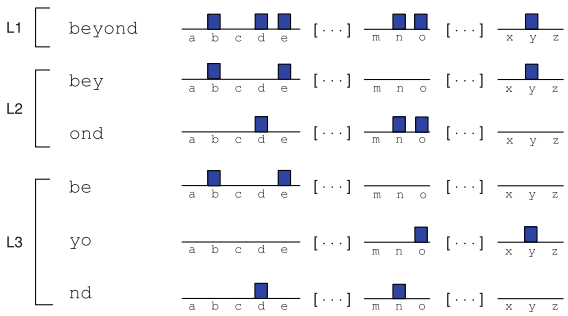

Fig. 3. Pyramidal histogram of characters (PHOC) [10] of the word "beyond" at levels 1, 2, and 3. The final PHOC representation is the concatenation of these partial histograms.

The original YOLOv2 model optimizes the following multi-part loss function:

$$L(b, C, c, \hat{b}, \hat{C}, \hat{c}) = \lambda_{box} L_{box}(b, \hat{b}) + L_{obj}(C, \hat{C}, \lambda_{obj}, \lambda_{noobj}) + \lambda_{cls} L_{cls}(c, \hat{c}) \quad (1)$$

where b is a vector with coordinates' offsets to an anchor bounding box, C is the probability of that bounding box containing an object, c is the one hot classification vector, and the three terms L_{box}, L_{obj}, and L_{cls} are respectively independent losses for bounding box regression, objectness estimation, and classification. All the aforementioned losses are essentially the sum-squared errors of ground truth (b, C, c) and predicted $(\hat{b}, \hat{C}, \hat{c})$ values. In the case of PHOC prediction, with c and $\hat{c}$ being binary vectors but with an unrestricted number of 1 values we opt for using a cross-entropy loss function in L_{cls} as in a multi-label classification task:

$$L_{cls}(c, \hat{c}) = \frac{-1}{N} \sum_{n=1}^{N} [c_n \log(\hat{c}_n) + (1 - c_n) \log(1 - \hat{c}_n)] \quad (2)$$

where N is the dimensionality of the PHOC descriptor.

Similarly as in [8] the combination of the sum-squared errors L_{box} and L_{obj} with the cross-entropy loss L_{cls} is controlled by the scaling parameters λ_{box}, λ_{obj}, λ_{noobj}, and λ_{cls}.

Apart of the modifications made so far on top of the original YOLOv2 architecture we also changed the number, the scales, and the aspect ratios of the pre-defined anchor boxes used by the network to predict bounding boxes. Similarly as in [8] we have found the ideal set of anchor boxes B for our training dataset by requiring that for each bounding box annotation there exists at least one anchor box in B with an intersection over union of at least 0.6. Figure 4 illustrates the 13 bounding boxes found to be better suited for our training data and their difference with the ones used in object detection models.

At test time, our model provides a total of $W/32 \times H/32 \times 13$ bounding box proposals, with W and H being the image input size, each one of them with an objectness score $(\hat{C})$ and a PHOC prediction $(\hat{c})$. The original YOLOv2 model

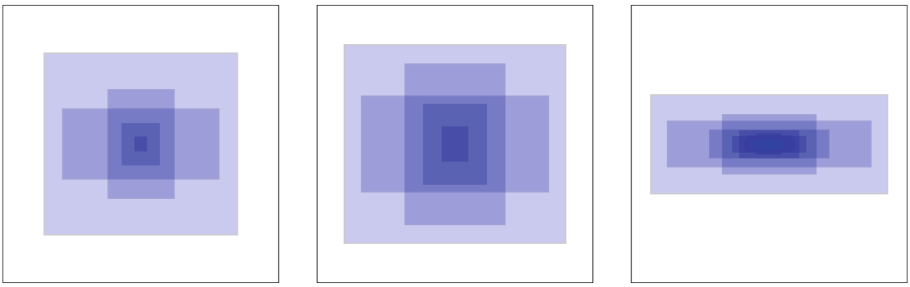

Fig. 4. Anchor boxes used in the original YOLOv2 model for object detection in COCO (a) and PASCAL (b) datasets. (c) Our set of anchor boxes for text detection.

filters the bounding box candidates with a detection threshold τ considering that a bounding box is a valid detection if $\hat{C}max(\hat{c}) \geq \tau$. If the threshold condition is met, a non-maximal suppression (NMS) strategy is applied in order to get rid of overlapping detections of the same object. In our case the threshold is applied only on the objectness score ($\hat{C}$) but with a much smaller value ($\tau = 0.0025$) than in the original model ($\tau \approx 0.2$), and we do not apply NMS. The reason is that any evidence of the presence of a word, even if it is small, it may be beneficial in terms of retrieval if its PHOC representation has a small distance to the PHOC of the queried word. With this threshold we generate an average of 60 descriptors for every image in the dataset and all of them conform our retrieval database.

In this way, the scene text retrieval of a given query word is performed with a simple nearest neighbor search of the query PHOC representation over the outputs of the CNN in the entire image database. While the distance between PHOCs is usually computed using the cosine similarity, we did not find any noticeable downside on using an Euclidean distance for the nearest neighbor search.

3.1 Implementation Details

We have trained our model in a modified version of the synthetic dataset of Gupta *et al.* [13]. First the dataset generator has been evenly modified to use a custom dictionary with the 90 K most frequent English words, as proposed by Jaderberg *et al.* [12], instead of the Newsgroup20 dataset [23] dictionary originally used by Gupta *et al.* The rationale was that in the original dataset there was no control about word occurrences, and the distribution of word instances had a large bias towards stop-words found in newsgroups' emails. Moreover, the text corpus of the Newsgroup20 dataset contains words with special characters and non ASCII strings that we do not contemplate in our PHOC representations. Finally, since the PHOC representation of a word with a strong rotation does not make sense under the pyramidal scheme employed, the dataset generator

was modified to allow rotated text up to 15°. This way we generated a dataset of 1 million images for training purposes. Figure 5 shows a set of samples of our training data.

Fig. 5. Synthetic training data generated with a modified version of the method of Gupta *et al.* [13]. We make use of a custom dictionary with the 90 K most frequent English words, and restrict the range of random rotation to 15°.

The model was trained for 30 epochs of the dataset using SGD with a batch size of 64, an initial learning rate of 0.001, a momentum of 0.9, and a decay of 0.0005. We initialize the weights of our model with the YOLOv2 backbone pre-trained on Imagenet. During the firsts 10 epochs we train the model only for word detection, without backpropagating the loss of the PHOC prediction and using a fixed input size of 448 × 448. On the following 10 epochs we start learning the PHOC prediction output with the λ_{cls} parameter set to 1.0. After that, we continue learning for 10 more epochs with a learning rate of 0.0001 and setting the parameters λ_{box} and λ_{cls} to 5.0 and 0.015 respectively. At his point we also adopted a multi-resolution training, by randomly resizing the input images among 14 possible sizes in the range from 352 × 352 to 800 × 800, and we added new samples in our training data. In particular, the added samples were the 1,233 training images of the ICDAR2013 [24] and ICDAR2015 [25] datasets. During the whole training process we used the same basic data augmentation as in [8].

4 Experiments and Results

In this section we present the experiments and results obtained on different standard benchmarks for text based image retrieval. First we describe the datasets used throughout our experiments, after that we present our results and compare them with the published state-of-the-art. Finally we discuss the scalability of the proposed retrieval method.

4.1 Datasets

The IIIT Scene Text Retrieval (STR). [7] dataset is a scene text image retrieval dataset composed of 10,000 images collected from the Google image search engine and Flickr. The dataset has 50 predefined query words and for

each of them a list of 10–50 relevant images (that contain the query word) is provided. It is a challenging dataset where relevant text appears in many different fonts and styles, and from different view points, among many distractors (images without any text).

The IIIT Sports-10k Dataset. [7] is another scene text retrieval dataset composed of 10,000 images extracted from sports video clips. It has 10 predefined query words with their corresponding relevant images' lists. Scene text retrieval in this dataset is specially challenging because images are low resolution and often noisy or blurred, with small text generally located on advertisements signboards.

The Street View Text (SVT) Dataset. [26] is comprised of images harvested from Google Street View where text from business signs and names appear. It contains more than 900 words annotated in 350 different images. In our experiments we use the official partition that splits the images in a train set of 100 images and a test set of 249 images. This dataset also provides a lexicon of 50 words per image for recognition purposes, but we do not make use of it. For the image retrieval task we consider as queries the 427 unique words annotated on the test set.

4.2 Scene Text Retrieval

In the scene text retrieval task, the goal is to retrieve all images that contain instances of the query words in a dataset partition. Given a query, the database elements are sorted with respect to the probability of containing the queried word. We use the mean average precision as the accuracy measure, which is the standard measure of performance for retrieval tasks and is essentially equivalent to the area below the precision-recall curve. Notice that, since the system always returns a ranked list with all the images in the dataset, the recall is always 100%. An alternative performance measure consist in considering only the top-n ranked images and calculating the precision at this specific cut-off point ($P@n$).

Table 1 compares the proposed method to previous state of the art for text based image retrieval on the IIIT-STR, Sports-10K, and SVT datasets. We show the mean average precision (mAP) and processing speed for the same trained model using two different input sizes (576×576 and 608×608), and a multi-resolution version that combines the outputs of the model at three resolutions (544, 576 and 608). Processing time has been calculated using a Titan X (Pascal) GPU with a batch size of 1. We appreciate that our method outperforms previously published methods in two of the benchmarks while it shows a competitive performance on the SVT dataset. In order to compare with state-of-the-art end-to-end text recognition methods, we also provide a comparison with pre-trained released versions of the models of Bušta *et al.* [18] and He *et al.* [27]. For recognition-based results the look-up is performed by a direct matching between the query and the text detected by each model. Even when making use of a predefined word dictionary to filter results, our method, which

is dictionary-free, yields superior results. Last, we compared against a variant of He *et al.* [27] but this time both queries and the model's results are first transformed to PHOC descriptors and the look-up is based on similarity on PHOC space. It can be seen that the PHOC space does not offer any advantage to end-to-end recognition methods.

Table 1. Comparison to previous state of the art for text based image retrieval: mean average precision (mAP) for IIIT-STR, and Sports-10K, and SVT datasets. (*) Results reported by Mishra et al. in [7], not by the original authors. (†) Results computed with publicly available code from the original authors.

Method	STR (mAP)	Sports (mAP)	SVT (mAP)	fps
SWT [28] + Mishra et al. [29]	-	-	19.25	
Wang *et al.* [26]	-	-	21.25*	
TextSpotter [30]	-	-	23.32*	1.0
Mishra *et al.* [7]	42.7	-	56.24	0.1
Ghosh *et al.* [31]	-	-	60.91	
Mishra [32]	44.5	-	62.15	0.1
Almazán *et al.* [10]	-	-	79.65	
TextProposals [33] + DictNet [34]	64.9†	67.5†	85.90†	0.4
Jaderberg *et al.* [12]	66.5	66.1	**86.30**	0.3
Bušta *et al.* [18]	62.94†	59.62†	69.37†	44.21
He *et al.* [27]	50.16†	50.74†	72.82†	1.25
He *et al.* [27] (with dictionary)	66.95†	74.27†	80.54†	2.35
He *et al.* [27] (PHOC)	46.34†	52.04†	57.61†	2.35
Proposed (576 × 576)	**68.13**	72.99	82.02	**53.0**
Proposed (608 × 608)	**69.83**	73.75	83.74	43.5
Proposed (multi-res.)	**71.37**	**74.67**	85.18	16.1

Table 2 further compares the proposed method to previous state of the art by the precisions at 10 (P@10) and 20 (P@20) on the Sports-10K dataset.

In Table 3 we show per-query mean average precision and precisions at 10 and 20 for the Sports-10K dataset. The low performance for the query "castrol" in comparison with the rest may initially be attributed to the fact that it is the only query word not seen by our model at training time. However, by visualizing the top-10 ranked images for this query, shown in Fig. 6 we can see that the dataset has many unannotated instances of "castrol". The real P@10 of our model is in fact 90% and not 50%. It appears that the annotators did not consider occluded words, while our model is able to retrieve images with partial occlusions in a consistent manner. Actually, the only retrieved image among the top-10 without the "castrol" word contains an instance of "castel". By manual inspection we have computed P@10 and P@20 to be 95.0 and 93.5 respectively.

Table 2. Comparison to previous state of the art for text based image retrieval: precision at n (P@n) for Sports-10K dataset.

Method	Sports-10K (P@10)	Sports-10K (P@20)
Mishra *et al.* [7]	44.82	43.42
Mishra [32]	47.20	46.25
Jaderberg *et al.* [12]	91.00	**92.50**
Proposed (576 × 576)	91.00	90.50
Proposed (multi-res.)	**92.00**	90.00

Table 3. Sports-10K per-query average precision (AP), P@10, and P@20 scores.

	Adidas	Castrol	Duty	Free	Hyundai	Nokia	Pakistan	Pepsi	Reliance	Sony
AP	94	16	74	61	77	75	92	70	89	89
P@10	100	50	100	90	100	80	100	90	100	90
P@20	100	55	100	85	100	85	100	95	100	90

Overall, the performance exhibited with the "castrol" query is a very important result, since it demonstrates that our model is able to generalize the PHOC prediction for words that has never seen at training time, and even to correctly retrieve them under partial occlusions. We found further support for this claim by analyzing the results for the six IIIT-STR query words that our model has not seen during training. Figure 7 shows the top-5 ranked images for the queries "apollo", "bata", "bawarchi", "maruti", "newsagency", and "vodafone". In all of them our model reaches a 100% precision at 5. In terms of mAP the results for these queries do not show a particular decrease when compared to those obtained with other words that are part of the training set, in fact in some cases they are even better. The mean average precision for the six words in question is 74.92, while for the remaining 44 queries is 69.14. To further analyze our model's ability for recognizing words it has never seen at training time, we have done an additional experiment within a multi-lingual setup. For this we manually added some images with text in different Latin script languages (French, Italian, Catalan, and Spanish) to the IIIT-STR dataset. We have observed that our model, while being trained only using English words, was always able to correctly retrieve the queried text in any of those languages.

Fig. 6. Top 10 ranked images for the query "castrol". Our model has not seen this word at training time.

In order to analyze the errors made by our model we have manually inspected the output of our model as well as the ground truth for the five queries with a lower mAP on the IIIT-STR dataset: "ibm", "indian", "institute", "sale", and "technology". In most of these queries the low accuracy of our model can be explained in terms of having only very small and blurred instances in the database. In the case of "ibm", the characteristic font type in all instances of this word tends to be ignored by our model, and the same happens for some computer generated images (*i.e.*non scene images) that contain the word "sale". Figure 8 shows some examples of those instances. All in all, the analysis indicates that while our model is able to generalize well for text strings not seen at training time it does not perform properly with text styles, fonts, sizes not seen before. Our intuition is that this problem can be easily alleviated with a richer training dataset.

4.3 Retrieval Speed Analysis

To analyze the retrieval speed of the proposed system, we have run the retrieval experiments for the IIIT-STR and Sports-10K datasets with different approximate nearest neighbor (ANN) algorithms in a standard PC with an i7 CPU and 32Gb of RAM. In Table 4 we appreciate that those ANN methods, with a search time sublinear in the number of indexed samples, reach retrieval speeds a couple of orders of magnitude faster than the exact nearest neighbor search based on ball-trees without incurring in any significant loss of retrieval accuracy.

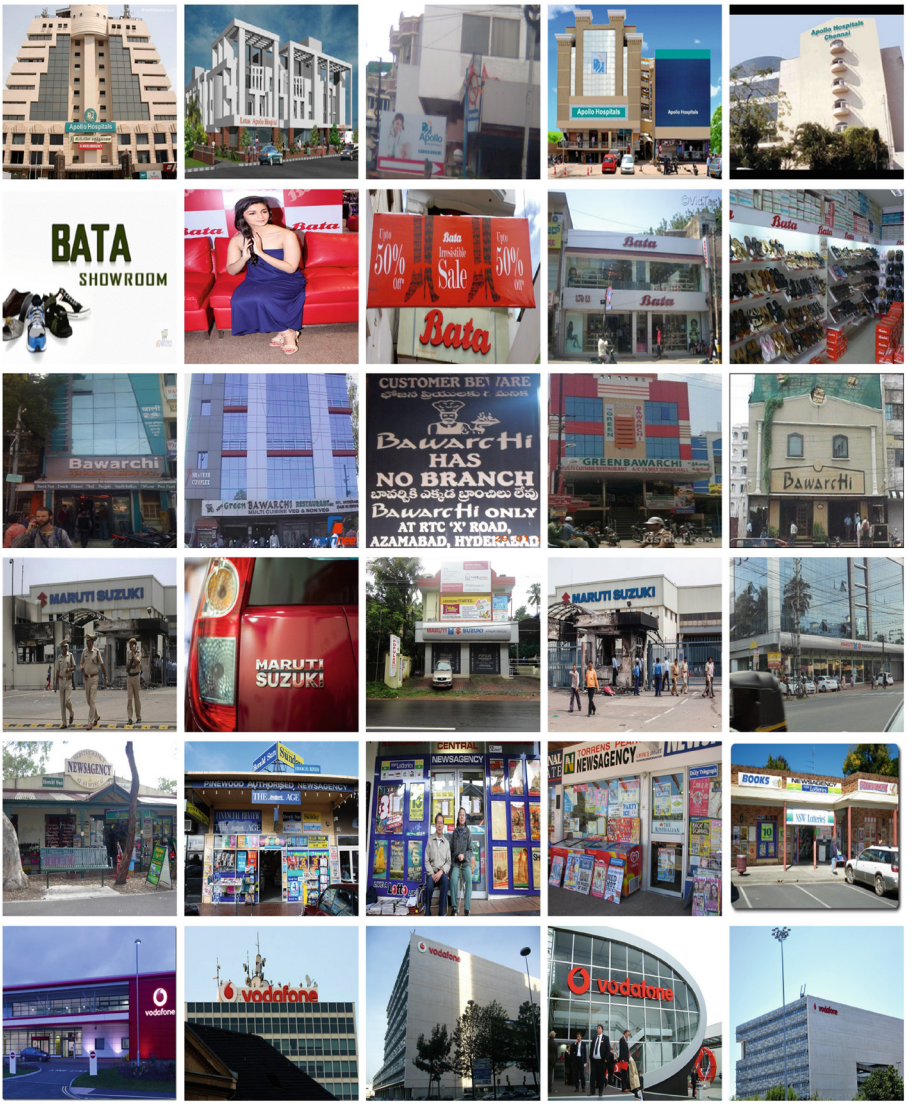

Fig. 7. From top to bottom, top-5 ranked images for the queries "apollo", "bata", "bawarchi", "maruti", "newsagency", and "vodafone". Although our model has not seen this words at training time it is able to achieve a 100% P@5 for all of them.

Fig. 8. Error analysis: most of the errors made by our model come from text instances with a particular style, font type, size, etc. that is not well represented in our training data.

Table 4. Mean Average Precision and retrieval time performance (in seconds) of different approximate nearest neighbor algorithms on the IIIT-STR and Sports datasets.

Algorithm	IIIT-STR			Sports-10K		
	mAP	Secs	#PHOCs	mAP	Secs	#PHOCs
Baseline (Ball tree)	0.6983	0.4321	620K	0.7375	0.6826	1M
Annoy (approx NN) [35]	0.6883	0.0027	620K	0.7284	0.0372	1M
HNSW (approx NN) [36]	0.6922	0.0018	620K	0.7247	0.0223	1M
Falconn LSH (approx NN) [37]	0.6903	0.0151	620K	0.7201	0.0178	1M

5 Conclusion

In this paper we detailed a real-time word spotting method, based on a simple architecture that allows it to detect and recognise text in a single shot and real-time speeds.

The proposed method significantly improves state of the art results on scene text retrieval on the IIIT-STR and Sports-10K datasets, while yielding comparable results to state of the art in the SVT dataset. Moreover, it can do so achieving faster speed compared to other state of the art methods.

Importantly, the proposed method is fully capable to deal with out-of-dictionary (never before seen) text queries, seeing its performance unaffected compared to query words previously seen in the training set.

This is due to the use of PHOC as a word representation instead of aiming for a direct word classification. It can be seen that the network is able to learn how to extract such representations efficiently, generalizing well to unseen text strings. Synthesizing training data with different characteristics could boost performance, and is one of the directions we will be exploring in the future along with investigating the use of word embeddings other than PHOC.

The code, pre-trained models, and data used in this work are made publicly available at https://github.com/lluisgomez/single-shot-str.

Acknowledgement. This work has been partially supported by the Spanish research project TIN2014-52072-P, the CERCA Programme/Generalitat de Catalunya, the H2020 Marie Skłodowska-Curie actions of the European Union, grant agreement No 712949 (TECNIOspring PLUS), the Agency for Business Competitiveness of the Government of Catalonia (ACCIO), CEFIPRA Project 5302-1 and the project "aBSINTHE - AYUDAS FUNDACIÓN BBVA A EQUIPOS DE INVESTIGACION CIENTIFICA 2017. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

1. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
2. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: COCO-text: dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016)
3. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (2015)
4. Movshovitz-Attias, Y., Yu, Q., Stumpe, M.C., Shet, V., Arnoud, S., Yatziv, L.: Ontological supervision for fine grained classification of street view storefronts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1693–1702 (2015)
5. Karaoglu, S., Tao, R., van Gemert, J.C., Gevers, T.: Con-text: text detection for fine-grained object classification. IEEE Trans. Image Process. **26**(8), 3965–3980 (2017)
6. Bai, X., Yang, M., Lyu, P., Xu, Y.: Integrating scene text and visual appearance for fine-grained image classification with convolutional neural networks. arXiv preprint arXiv:1704.04613 (2017)
7. Mishra, A., Alahari, K., Jawahar, C.: Image retrieval using textual cues. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3040–3047 (2013)
8. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
9. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
10. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. IEEE Trans. Pattern Anal. Mach. Intell. **36**(12), 2552–2566 (2014)
11. Sudholt, S., Fink, G.A.: PHOCNET: a deep convolutional neural network for word spotting in handwritten documents. In: Proceedings of the IEEE International Conference on Frontiers in Handwriting Recognition, pp. 277–282 (2016)
12. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. Int. J. Comput. Vis. **116**(1), 1–20 (2016)
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324 (2016)
14. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: a fast text detector with a single deep neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4161–4167 (2017)

15. Liao, M., Shi, B., Bai, X.: TextBoxes++: a single-shot oriented scene text detector. arXiv preprint arXiv:1801.02765 (2018)
16. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
17. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11) (2017)
18. Buvsta, M., Neumann, L., Matas, J.: Deep TextSpotter: an end-to-end trainable scene text localization and recognition framework. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2204–2212 (2017)
19. Li, H., Wang, P., Shen, C.: Towards end-to-end text spotting with convolutional recurrent neural networks. arXiv preprint arXiv:1707.03985 (2017)
20. Aldavert, D., Rusiñol, M., Toledo, R., Lladós, J.: Integrating visual and textual cues for query-by-string word spotting. In: Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 511–515 (2013)
21. Ghosh, S.K., Valveny, E.: Query by string word spotting based on character bi-gram indexing. In: Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 881–885 (2015)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of the International Conference on Neural Information Processing Systems, pp. 91–99 (2015)
23. Lang, K., Mitchell, T.: Newsgroup 20 dataset (1999)
24. Karatzas, D., et al.: ICDAR 2013 robust reading competition. In: Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 1484–1493 (2013)
25. Karatzas, D., et al.: ICDAR 2015 competition on robust reading. In: Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 1156–1160 (2015)
26. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1457–1464 (2011)
27. He, T., Tian, Z., Huang, W., Shen, C., Qiao, Y., Sun, C.: An end-to-end TextSpotter with explicit alignment and attention. In: CVPR (2018)
28. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2963–2970 (2010)
29. Mishra, A., Alahari, K., Jawahar, C.: Top-down and bottom-up cues for scene text recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2687–2694 (2012)
30. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2012)
31. Ghosh, S.K., Gomez, L., Karatzas, D., Valveny, E.: Efficient indexing for query by string text retrieval. In: Proceedings of the IEEE International Conference on Document Analysis and Recognition, pp. 1236–1240 (2015)
32. Mishra, A.: Understanding Text in Scene Images. Ph.D. thesis, International Institute of Information Technology Hyderabad (2016)
33. Gómez, L., Karatzas, D.: TextProposals: a text-specific selective search algorithm for word spotting in the wild. Pattern Recogn. **70**, 60–74 (2017)

34. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
35. Bernhardsson, E.: ANNOY: approximate nearest neighbors in C++/Python optimized for memory usage and loading/saving to disk (2013)
36. Malkov, Y.A., Yashunin, D.: Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. arXiv:1603.09320 (2016)
37. Andoni, A., Indyk, P., Laarhoven, T., Razenshteyn, I., Schmidt, L.: Practical and optimal LSH for angular distance. In: NIPS (2015)