Real-time Lexicon-Free Scene Text Retrieval

Andrés Mafla, Rubèn Tito, Lluís Gómez, Marçal Rusiñol, Ernest Valveny, Dimosthenis Karatzas

Computer Vision Center, Universitat Autonoma de Barcelona. Edifici O, Campus UAB, 08193 Bellaterra (Cerdanyola) Barcelona, Spain.
E-mail: {amafla,rperez,lgomez,marcal,ernest,dimos}@cvc.uab.cat

Abstract

In this work, we address the task of scene text retrieval: given a text query, the system must return all images containing the queried text. The proposed model uses a single shot CNN architecture that predicts bounding boxes and builds a compact representation of spotted words. In this way, this problem can be modeled as a nearest neighbor search of the textual representation of a query over the outputs of the CNN collected from the totality of an image database. Our experiments demonstrate that the proposed model outperforms previous state-of-the-art, while offering a significant increase in processing speed and unmatched expressiveness with samples never seen at training time. Several experiments to asses the robustness of the model are conducted as well as an application of real-time text spotting in videos.

Keywords: Image retrieval, Scene text detection, Scene text recognition, Word spotting, Convolutional Neural Networks, Region Proposals Networks, PHOC.

Real-time Lexicon-Free Scene Text Retrieval

Andrés Mafla, Rubèn Tito, Lluís Gómez, Marçal Rusiñol, Ernest Valveny, Dimosthenis Karatzas

Computer Vision Center, Universitat Autonoma de Barcelona. Edifici O, Campus UAB, 08193 Bellaterra (Cerdanyola) Barcelona, Spain.
E-mail: {amafla,rperez,lgomez,marcal,ernest,dimos}@cvc.uab.cat

1. Introduction

The development of language is one of the most influential inventions of humankind that allows the communication of abstract and complex ideas. Similarly, written text permits this set of complex ideas to be depicted in an explicit and semantic manner. As it is shown by several authors [1, 2, 3], there is a big percentage of media that contains text, especially in urban scenarios and documents. Adding this to the fact that there is ample availability of data and the importance of text, it becomes essential to develop and refine algorithms that exploit the richness of textual information found in images and video. Leveraging text in scene imagery allows the emergence of tasks such as image retrieval [4, 5], scene understanding [6, 7], instant translation [8, 9], human-computer interaction, robot navigation [10, 11], assisted reading for the visually-impaired [12, 13] and industrial automation [14, 15]. In the previous years significant advances have been accomplished, particularly since the introduction of AlexNet [16], architecture that won the ILSVRC2012 [17] contest by using deep learning techniques. Text spotting has been diverging from older approaches that used hand-crafted features towards current ones that employ automatic feature learning by exploiting deep learning methodologies [12, 18]. Nonetheless, text spotting is not a trivial task and remains as an open problem in the research community. Putting aside the complexity of spotting text in the wild, the importance that text encompasses is given by the high level semantic and explicit information, which can not be leveraged by using visual cues alone. For example, there is a high degree of complexity involved in labelling images without considering the text found in them, even for humans. This effect is evident in Figure 1, in which the storefronts alone can belong to a wide plethora of businesses, but the exact label can be inferred if and only if the text contained is read and leveraged appropriately. Research conducted by Movshovitz *et al.* [19] showed that while training a shop classifier, the proposed model ended up learning and interpreting textual information as the only way of differentiating between diverse businesses. The described effect is evident and addressed explicitly in later works conducted by [20, 21], which focuses on fine-grained classification of storefronts and bottles respectively. Additional tasks that require integration of textual and visual information to generate a common domain knowledge have been proposed such as in [6, 7], which opens up new research paths.

Closely related to our work, Mishra *et al.* [22] proposed the task of scene text retrieval. The input to the system is a text query, which the system must employ to return all the images that contain the queried text. This task requires systems that are robust enough to perform fast word spotting while at the same time holding the capacity of generalizing out of dictionary queries never seen before. An intuitive approach to tackle such a problem

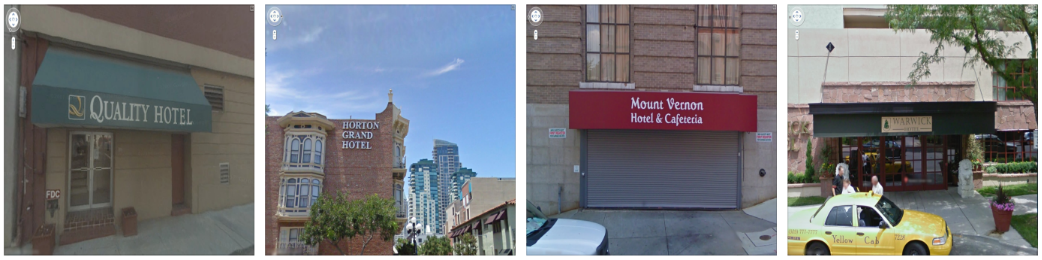


Figure 1: The visual appearance of different business places in images can be extremely variable. It seems impossible to correctly label them without reading the text within them. Our scene text retrieval method returns all the images shown here within the top-10 ranked results among more than $10,000$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

is to make use of state of the art reading systems, and use the output predictions of it to find the closest match with the given query. However, as it is shown by [22], such attempts commonly have low performance caused by limitations in end to end reading systems. On one hand, end to end reading systems are evaluated on recognition, a different task that focuses on achieving high precision scores, often using a specific language dictionary [23] or as it is proposed by [24, 13] a short dictionary per image. On the other hand, a retrieval system requires a large number of proposals (high recall) which can be beneficial at the moment of finding close matching detections when compared to a query. It is worth noting that end to end reading systems usually consist of at least two clearly defined stages that employ the encoder-decoder paradigm. The pipeline comprised by these two stages, more often that not are slow at the moment of generating predictions of the text contained in an image. This existing time constraint hinders the use of such algorithms in real-time scenarios or at the moment of indexing large scale collections of images and documents.

In order to exploit the particular requirements that need to be addressed by a retrieval system, we propose in this work a real-time, high-performance word spotting method that detects and recognizes text in a single calculation of a Fully Convolutional Neural Network (FCNN). The proposed architecture is based on the YOLO model [25, 26], a widely used single shot object detector which in our case is employed to construct a PHOC (Pyramidal Histogram Of Characters) [27, 28] predictor. By employing this methodology, our model is able to perform text detection and recognition in a single calculation thus making it suitable for real time applications or to index large scale image collections at an unmatched speed.

The main contributions of using the proposed model, as it is shown in our previous work [29] are: firstly, the usage of a layout comprised by an end-to-end jointly trainable FCNN. Secondly, the usage of the PHOC as a word representation instead of a direct word classification over a closed dictionary. Thus, providing an elegant mechanism to generalize to any text string, allowing the method to tackle efficiently out-of-dictionary queries. Lastly, due to its design, the adoption of this method achieves unmatched speed when processing images to construct a compact representations of the recognized text instances. As an extension to the preceding research, in this work we analyze deeply the capacity of dealing with out-of-vocabulary queries of our model by conducting exhaustive experiments performed in two multi-lingual datasets. These experiments prove that the proposed method is able to successfully apply knowledge transfer acquired at training time to construct word representations of previously unseen text samples at inference time. As an additional section we present supplementary experiments and provide an analysis of the system under different kinds of imperfect image conditions such as rotation, blur, occlusion and compression, experiments that confirm the robustness of the proposed architecture. Lastly, we propose an application of real-time text spotting on video, in which the model needs to confirm its robustness to noise and distortions while at the same time maintaining its characteristic high processing speed.

2. Related Work

In the past years, several advances in Deep Learning have been accomplished due to data availability and computing power [3], allowing deep learning models to surpass several benchmarks in a wide range of tasks. The main advantage of using deep learning methodologies is the possibility of automatic feature learning, rather than hand-crafted ones. Most literature [18, 12] divide the existing methods as: text detection, text recognition, end-to-end systems. Other applications such as fine-grained classification, image understanding and image retrieval are briefly described in the upcoming sections.

2.1. Scene Text Detection

Initial deep learning methodologies employed several steps to produce proposals. In the work presented by [30], a CNN is used to predict if a given pixel belongs to a character, forms part of a text region and its orientation. Yao *et al.* [31] propose a CNN that outputs text proposals, which are filtered by separating different text instances by employing a semantic segmentation model. Later works focus on simplifying the pipeline and thus improving speed and training of models. These models usually follow a two step pipeline that comprise of an end-to-end trainable detection network and a post processing step. The work presented by [32, 33] named Textboxes, adopts a modified version of a popular object recognition model named Single Shot Detector [34]. It employs modified anchor boxes to regress the ground truth boxes followed by a non-maximum suppression step (NMS). A performance focused approach is given by EAST [35], which upsamples feature maps gradually and uses [36] as the network backbone, and outputs a per pixel word or text line prediction followed by a NMS step.

Inspired by the object detection framework proposed by R-CNN [37, 38, 39], ample research has been conducted. The common approach consists of a Region Proposal Network (RPN) that produces candidate text regions, which later are passed through a pooling layer that classifies the region as text or not text. In the model presented by [40], rotated region proposals are presented, mostly to handle arbitrary oriented text. Analogously, R2CNN [41] the Region of Interest(ROI) pooling stage uses different fixed sizes which are concatenated for regression and classification. The work conducted by [42] mainly focuses on adaptive weighted pooling in different scales to further predict and regress region proposals.

2.2. Scene Text Recognition

Initial approaches explored by Jaderberg *et al.* [43] tackle text recognition as a classification problem. After training a CNN on synthetic generated samples, the obtained features are used to predict a vector that classifies the input word over approximately 90,000 classes. After the introduction of the Connectionist Temporal Classification (CTC) by Graves *et al.* [44] in handwriting recognition, the same methodology has been widely used in scene text as well. The work proposed by [45] employs the CTC layer after passing the input image through a CNN that acts as the encoder and a RNN that act as the decoder. The introduction of an attention mechanism was initially proposed by [46] in the task of machine translation. This mechanism was briefly adopted in several vision tasks, including text recognition. The work proposed by [47], namely Focus Attention Network, employs attention to supervise relevant locations for word recognition. Bai *et al.* [48] introduce an edit probability to handle the misalignment between the ground truth string and the attention output string. Jaderberg *et al.* [49] proposed the Spatial Transformer Network, which is used by [50] to align detected text horizontally to further employ an attention based recognizer.

2.3. End-to-End Text Recognition

A commencing approach proposed by Jaderberg *et al.* [23] employs a sliding window to extract proposals, which are filtered and a CNN is used to regress the bounding boxes. Later the filtered regions that surpass a threshold are classified. In another work, Gupta *et al.* [51] defined a Fully Convolutional Regression Network for text detection and bounding box regression and the same classification network proposed by [23] for text recognition, being one of the first models that were fully trainable based on deep learning methodologies solely. In [52] a YOLO[53] based CNN is adopted to detect text instances, which later are passed through a Connectionist Temporal Classification module for recognition. These two stages are trained separately and later connected together to form and end-to-end architecture.
The research presented by [54]introduces a CNN that is used as an encoder and a Long Short-Term Memory (LSTM) along with an attention mechanism module as decoder, both employed for detection and recognition. He *et al.* [55] use a CNN to extract proposals, which are fed into an LSTM to refine

the bounding boxes that are later employed as input to yet another LSTM to perform recognition that fixes misalignment between attention maps and ground truth character labels. In more recent work, [56] uses EAST [35] to obtain text regions and employs a CTC recognition module [44] to obtain an end-to-end reading system. Lyu *et al.* [57] use a variation of Mask R-CNN[39] to detect text in arbitrary shapes and segment an image in different instances to recognize similar text regions.

2.4. Scene Text Retrieval

Closely related to our work, the scene text retrieval problem slightly differs from classical scene text recognition methodologies. In a retrieval scenario the user defines a textual query which he wants to retrieve, whereas most of recognition approaches are based on employing a predefined vocabulary of the words one might come along within scene images. For instance, both Mishra *et al.*[22], who introduced the scene text retrieval task, and Jaderberg *et al.* [23], use a fixed vocabulary to create an inverted index which contains the presence of a word in the image. These approaches limit the freedom of queries to a set of predefined vocabulary words.

To address such a problem, text string descriptors based on n-gram frequencies, like the PHOC descriptor (Figure 2), have been successfully used for word spotting applications [58, 27, 59]. By using a vectorial codification of text strings, users can query any string at inference time without being limited to a specific set of predefined vocabulary words. In this work, we make use of the PHOC descriptor along with an object detection framework based on YOLO [25, 53] that encodes found text instances. We suggest that this approach brings many benefits, mostly due to the high recall and single shot

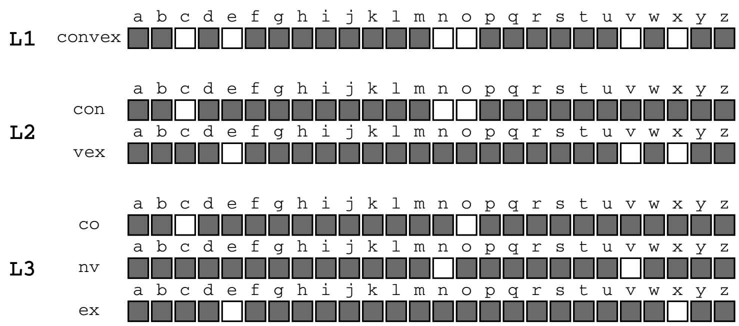


Figure 2: Pyramidal histogram of characters (PHOC) [27] of the word "convex" at levels 1, 2, and 3. The final PHOC representation is the concatenation of the partial one-hot encodings.

calculation required to locate and recognize text contained within an image, accompanied by unmatched processing speeds.

2.5. Other applications

Fine-grained Classification is the task of classifying visually similar objects in which subtle differences are key to find discriminative features between classes. Finding these subtle features is a challenging task which keeps this problem as an active topic in computer vision. Karaoglu *et al.* [20] tackles this task by extracting visual features by employing a GoogleNet [60] and a feature of Bag of Words to represent the text instances found in an image and further classify them. More recently, [61] uses a similar approach and extracts the visual features using a GoogleNet [60] and a combination of two models: [32] to detect text and [45] to recognize text. The recognized text instances are represented by GloVe [62], which are later used with an attention mechanism on the visual features to classify the image.
Additional work has explored other fields of scene understanding by employing textual cues. The work proposed by [6] and [7] focuses on the Visual Question Answering (VQA) [63] task. The VQA problem consists in provid-

ing an answer to a given image and question presented in natural language. Providing the correct answer is possible only if the system is capable to leverage textual information contained in the image.

3. Proposed Architecture

The proposed architecture is based on a custom-built YOLOv2 object detection model introduced by [25, 26]. This work adapts the object detector to output a compact representation of the text instances and recast them as a PHOC [27], thus enforcing the model to learn to construct such a vectorial codification. The suggested model is kept as a Fully Convolutional Neural Network, and a straightforward diagram is illustrated in Figure 3.

The convolutional neural network is composed of 22 convolutional layers with a leaky ReLu activation function after each convolution operation. The details of the proposed architecture can be seen in Table 1.

Batch normalization is used after every convolutional layer to help the model reach convergence. In total the model employs 5 max pooling layers, which reduces the input width and height by a factor of 2^5. The filter size used in convolutions is 3×3 and the channel number is doubled after each pooling step as in previous works that adopt a VGG [64] model backbone such as the work presented by [45]. In order to apply dimensionality reduction and decrease the computation cost, the strategy proposed by the usage of an Inception module [60] is taken, and filters of size 1×1 are interleaved between the 3×3 convolutional filters to obtain richer feature maps. As it is defined in YoloV2 [26] and inspired in the Residual blocks introduced by [65], the convolutional backbone uses a pass-through layer from an earlier

Table 1: Detailed description of the proposed CNN architecture considering an input image size of 608 x 608.

Layer	Type	Filters	Size/Pad/Stride	Output
0	Input		-	608 x 608 x 3
1	Conv	32	3x3/p1/1	608 x 608 x 32
2	Max Pool		2x2/p0/2	304 x 304 x 32
3	Conv	64	3x3/p1/1	304 x 304 x 64
4	Max Pool		2x2/p0/2	152 x 152 x 64
5	Conv	128	3x3/p1/1	152 x 152 x 128
6	Conv	64	1x1/p0/1	152 x 152 x 64
7	Conv	128	3x3/p1/1	152 x 152 x 128
8	Max Pool		2x2/p0/2	76 x 76 x 128
9	Conv	256	3x3/p1/1	76 x 76 x 256
10	Conv	128	1x1/p0/1	76 x 76 x 128
11	Conv	256	3x3/p1/1	76 x 76 x 256
12	Max Pool		2x2/p0/2	38 x 38 x 256
13	Conv	512	3x3/p1/1	38 x 38 x 512
14	Conv	256	1x1/p0/1	38 x 38 x 256
15	Conv	512	3x3/p1/1	38 x 38 x 512
16	Conv	256	1x1/p0/1	38 x 38 x 256
17	Conv	512	3x3/p1/1	38 x 38 x 512
18	Max Pool		2x2/p0/2	19 x 19 x 512
19	Conv	1024	3x3/p1/1	19 x 19 x 1024
20	Conv	512	1x1/p0/1	19 x 19 x 512
21	Conv	1024	3x3/p1/1	19 x 19 x 1024
22	Conv	512	1x1/p0/1	19 x 19 x 512
23	Conv	1024	3x3/p1/1	19 x 19 x 1024
24	Conv	1024	3x3/p1/1	19 x 19 x 1024
26	Conv	1024	3x3/p1/1	19 x 19 x 1024
26	Concat[16]			38 x 38 x 512
27	Conv	64	1x1/p0/1	38 x 38 x 64
28	Concat[24,27]			19 x 19 x 1280
29	Conv	1024	3x3/p1/1	19 x 19 x 1024
30	Conv	7917	1x1/p0/1	19 x 19 x 7917

convolutional layer, which is concatenated and followed by a final 1×1 convolutional filter with a linear activation with the number of filters matching the desired output tensor size to encode the PHOC descriptor.

Following the approach from the YOLOv2 model, we could define the word spotting task as a classification problem, where each detected word is a class. This one hot classification vector in the output tensor would represent the word class probability distribution among a defined list of words (fixed size dictionary) per each bounding box prediction. As simple as it sounds, such an approach limits the number of words that the model is able to recognize. In principle, if such a model requires to recognize 20 words, it would theoretically perform as well as classifying the 20 object classes from the PASCAL dataset presented in [26]. However, the problem raises in complex-

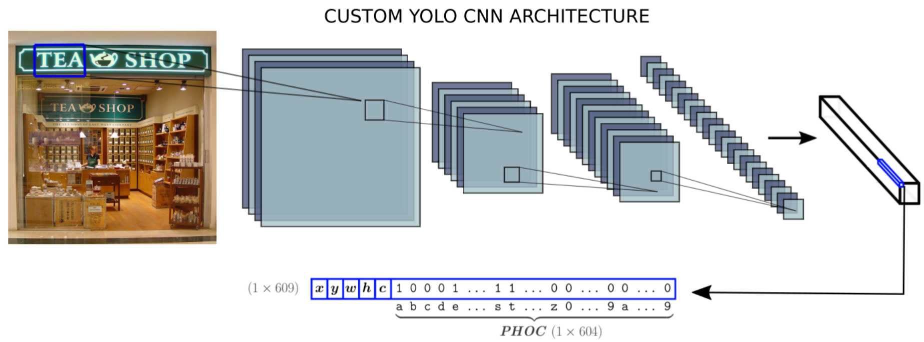


Figure 3: Our Convolutional Neural Network predicts at the same time bounding box coordinates x, y, w, h, an objectness score c, and a pyramidal histogram of characters (PHOC) of the word in each bounding box.

ity as the number of classes grow. If we consider training such a model (e.g. the list of 90,000 most frequent words from the English vocabulary [23]), the final convolutional layer would require 90,000 filters. This factor would require an immense amount of data to successfully train such a model. Even though a model with such characteristics could be designed, the limitation of only recognizing words that belong to a predefined dictionary would still be present. Recognizing out of vocabulary words would require a special treatment or simply it would be a non-viable task. Furthermore, given the number of parameters required, the model size would be too big and the real time processing speed would most likely be lost.

A way of addressing the aforementioned problems, specifically a model that is able to generalize and recognize previously unseen words, is desired. This is the main driving rationale behind casting the network as a PHOC predictor, which also permits to decrease the model's last filter size, thus allowing it to perform at real-time. The PHOC [27] descriptor is a multi-level vectorial representation of text strings that focuses on encoding if a specific character is present in a defined spatial region of a string (see Figure 2). Intuitively, a CNN based model that effectively learns to predict the PHOC representation of a detected word will inherently learns to identify the existence of a specific character in a visual region of the proposed bounding box. The model therefore will learn to construct the PHOC by automatically learning character attributes independently. Learning how to construct such a representation given the morphology of a string allows the proposed model to transfer knowledge acquired at training time and employ it at inference time to build PHOCs of unseen words. This effect is possible due to the fact that the presence of a character at a particular locality of the word translates to the same information in the PHOC representation, independently of the positioning or existence of other characters in the word. Moreover, the PHOC descriptor acts as a universal encoding scheme that offers unlimited expressiveness as it can represent any word constrained only by a language specific alphabet.

The PHOC version we propose in this work, contains a fixed length of 604 dimensions represented as a binary vector.

In order to adapt the YOLOv2 object detection network for single shot detection and PHOC prediction, it is necessary to define the nature of the proposed descriptor. In the first place, the PHOC descriptor does not resemble a one hot vector as in a classification scheme. To treat the PHOC as a multi-hot binary vector, the last layer does not employ a softmax function. Secondly, the prediction of a PHOC vector is comprised of a set of numbers that satisfy the condition given by:

$$S = \{x | x \in \mathbb{R}, 0 \leqslant x \leqslant 1\} \tag{1}$$

Where S represents the set of possible PHOC values. In order to have such a representation, a sigmoid activation function after the last convolutional layer is used to predict the PHOC vectors rather than the original softmax function.

Thirdly, we modify the original YOLOv2 Loss Function to facilitate the convergence and learning process of the model. As it is presented in the original YOLOv2 paper, the proposed algorithm is trained with the following multi-part loss function:

$$L(b, C, c, \hat{b}, \hat{C}, \hat{c}) = \lambda_{box} L_{box}(b, \hat{b}) + L_{obj}(C, \hat{C}, \lambda_{obj}, \lambda_{noobj}) + \lambda_{cls} L_{cls}(c, \hat{c}) \tag{2}$$

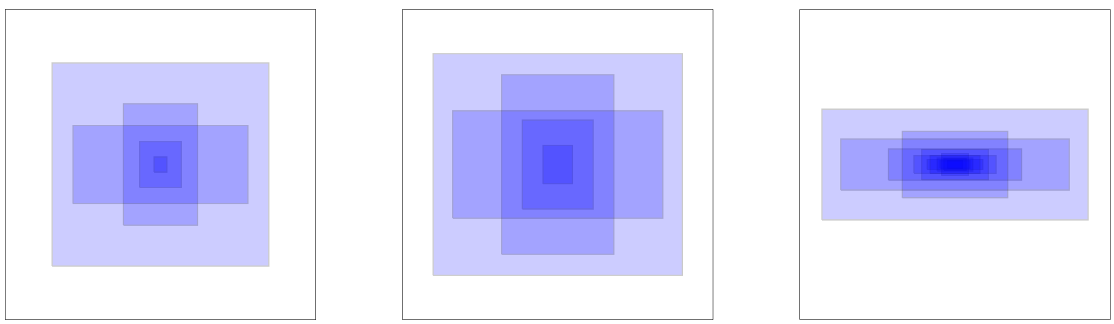
where b is a vector with coordinates' offsets to an anchor bounding box, C is the probability of that bounding box containing an object, c is the one hot classification vector, and the three terms L_{box}, L_{obj}, and L_{cls} are respectively independent losses for bounding box regression, objectness 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. At the moment of predicting a PHOC, c (the ground truth) is a binary vector and $\hat{c}$ (prediction) meets the condition stated in 1, reason to opt for cross-entropy loss function in L_{cls} as in a multi-label classification task:

$$L_{cls}(c, \hat{c}) = c \log \hat{c} + (1 - c) \log(1 - \hat{c}) \tag{3}$$

It is important to note that 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 from 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. Similar to [25], 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.



a) b) c)

Figure 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.

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 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 50 descriptors for every image in the dataset and all of them form 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. Training details

We have trained our model in a modified version of the synthetic dataset of Gupta *et al.*[51]. First the dataset generator has been evenly modified to use a custom dictionary with the 90K most frequent English words, as proposed by Jaderberg *et al.*[23], instead of the Newsgroup20 dataset [66] dictionary originally used by Gupta *et al.*. The rationale was that in the original dataset there was no control over the 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

Figure 5: Synthetic training data generated with a modified version of the method of Gupta *et al.* [51]. We make use of a custom dictionary with the 90K most frequent English words, and restrict the range of random rotation to 15 degrees.

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 degrees. This way we generated a dataset of 1 million images for training purposes. Figure 5 shows a set of samples of our training data.

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 this 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 [13] datasets. During the whole training process we used the same basic data augmentation as proposed by [25].

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 and after that, we present our results and compare them with the published state-of-the-art. As an extension to our previous work [29], an assessment when dealing with out-of-vocabulary words is conducted by analyzing the model in two multi-lingual datasets. Additionally, we conduct robustness experiments when confronted with imperfect image conditions, which further shows our models' potential. Finally, we present a real-time text spotting application in videos, only possible by the characteristic speed capability of our method.

4.1. Datasets

4.1.1. IIIT Scene Text Retrieval (STR)

The STR dataset [22] 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).

4.1.2. IIIT Sports-10k dataset

The Sports-10k dataset [22] 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.

4.1.3. Street View Text (SVT) dataset

The SVT dataset [67] is comprised of images harvested from Google Street View where advertisement signboards is present. 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.

4.1.4. Multi-lingual scene text (MLT) datasets

These two datasets MLT2017 [68] and MLT2019 [69] are scene text detection and recognition datasets that contain $7,200$ and $10,000$ images respectively in 10 different languages (Chinese, Japanese, Korean, English, French, Arabic, Italian, German, Bangla and Hindi) in equal proportions, representing 7 different scripts. These datasets mostly comprises focused text in natural images, and even though the main task is text detection and recognition, we adapted it to conduct text retrieval experiments. We employ this dataset to assess the generalization power of the PHOC representation of unseen words at training time.

4.1.5. Text in videos (TiV) dataset

The TiV dataset [70] contains 25 videos (13450 frames in total) and a test set of 24 videos (14374 frames in total) recorded from 4 different cameras. We use this dataset to asses the performance at real-time of our model at the moment of retrieving a specific text query. The challenge in this dataset remains in the fact that usually video frames contain a lower quality when compared to static images. The problems of text spotting usually relate to rotation, blur and occlusion of text found on each frame due to movement and focusing issues while including loss of information at the moment of video compression.

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 2 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 clearly outperforms previously published methods in two of the benchmarks while it shows a competitive performance on the SVT dataset. It is important to witness that our method achieves the highest measurements in frames per second (fps), leading to the the best overall trade-off between performance and processing speed in all datasets. Table 3 further compares the proposed method to previous state of the art by showcasing the precision at 10 (P@10) and 20 (P@20) on the Sports-10K dataset.

Figure 6: Bounding box heat-maps for queried words ”honda”, ”police”, ”tea” and ”sony” respectively.

In Figure 6, we depict the heat-maps of our model by calculating the closests matching PHOC and its bounding box in relation to a given query. As it can be seen on the showcased figure, several predicted PHOCs closely match the queried word. Considering the implementation details defined in the previous section, we avoid using a NMS post processing strategy to preserve high matching PHOC proposals that could be discarded otherwise.

For a further analysis of the errors made by our model we have manually

Table 2: 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 [22], not by the original authors. (†) Results computed with publicly available code from the original authors.

Method	STR (mAP)	Sports (mAP)	SVT (mAP)	fps
SWT [71]+ Mishra et al. [72]	-	-	19.25	
Wang *et al.* [67]	-	-	21.25*	
TextSpotter [73]	-	-	23.32*	1.0
Mishra *et al.* [22]	42.7	-	56.24	0.1
Ghosh *et al.* [74]	-	-	60.91	
Mishra [75]	44.5	-	62.15	0.1
Almazán *et al.* [27]	-	-	79.65	
TextProposals [76] + DictNet [43]	64.9†	67.5†	85.90†	0.4
Jaderberg *et al.* [23]	66.5	66.1	**86.30**	0.3
Bušta *et al.*[77] ICCV 2017	62.94	59.62	69.37	44.21
He *et al.*[55] CVPR 2018	50.16	50.74	72.82	1.25
He *et al.*[55] (With dictionary)	66.95	74.27	80.54	2.35
He *et al.*[55] (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

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”, ”technology” and ”sale”. 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 (non scene images) that contain the word ”sale”. Figure 7 shows some examples of those instances.

Table 3: Comparison to previous state of the art for text based image retrieval: precision at n (P@n) for Sports-10K dataset.

Method	Sport-10K (P@10)	Sport-10K (P@20)
Mishra *et al.* [22]	44.82	43.42
Mishra [75]	47.20	46.25
Jaderberg *et al.* [23]	91.00	**92.50**
Proposed (576 × 576)	91.00	90.50
Proposed (multi-res.)	**92.00**	90.00

Figure 7: Error analysis: last ranked images for queries "sale", "ibm", "indian", "institute", "technology" and "police". 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.

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 alleviated with a richer training dataset.

4.3. Multi-Lingual Scene Text Retrieval

As an extension to our previous work [29], we focus on analyzing the generalization capability of the proposed model. It becomes essential to note that designing an algorithm that learns to construct a compact representation of a string, such as the PHOC, paves the road to further development of models that are not constrained to a fixed dictionary or training data samples. In order to assess the expressiveness of our architecture, we make use of two Multi-lingual datasets 2017 [68] and 2019 [69] in which we can easily find out-of-vocabulary words (text not seen at training time) with different distributions and characteristics. These datasets are used by the research community to perform text detection and recognition tasks, but not text based image retrieval. Therefore, we have selected a set of 100 queries for in-vocabulary experiments and another set of 100 queries for out-of-vocabulary experiments for each dataset taken from the training split. Out-of-vocabulary queries are selected by choosing the latin words with most occurrences after removing stop-words and words that contain non-alphanumeric characters. For in-vocabulary queries, we also remove stop-words and words with non-alphanumeric characters before searching for latin words with similar frequencies to the out-of-vocabulary queries.

Table 4: Comparison to previous state of the art method for text based image retrieval methods when queries are words already seen during the training process (IV) or not (OOV): mean average precision (mAP)

	MLT 2017		**MLT 2019**	
Method	IV	OOV	IV	OOV
He et al. [55]	24.79	19.47	27.6	24.99
Proposed	**46.52**	**46.87**	**46.41**	**46.03**

Table 5: Comparison to previous state of the art method for text based image retrieval methods when queries are words already seen during the training process (IV) or not (OOV): precision at n (P@n)

	MLT 2017						**MLT 2019**					
	IV			**OOV**			**IV**			**OOV**		
Method	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20
He et al.	0.51	0.37	0.22	0.46	0.33	0.20	0.62	0.44	0.27	0.60	0.40	0.23
Proposed	**0.77**	**0.57**	**0.34**	**0.78**	**0.59**	**0.34**	**0.80**	**0.64**	**0.41**	**0.80**	**0.64**	**0.40**

Tables 4 and 5 show the ability for our model to perform retrieval with the same accuracy for in-vocabulary queries and out-of-vocabulary queries in both datasets. As we stated previously, this is because our model is learning how to build a PHOC from text rather than performing a classification along a fixed dictionary. It is important to note that our model performs significantly better than a state of the art reading system presented by [55] at the text retrieval task. Additionally, the method from [55] was trained using the dictionary from [66] which contains English words, thus performing poorly when dealing with out of vocabulary words mostly belonging to different languages. Figure 8 shows the top-5 ranked images for the queries "vodafone" in IIIT-STR dataset, "uscita" (italian) in MLT 2017 and "werden" (german) in MLT 2019, all of them being unseen samples at training time. In all of them our model reaches a 100% precision at 5.

Figure 8: From top to bottom, top-5 ranked images for the queries "vodafone", "uscita", "werden". Although our model has not seen these words at training time it is able to achieve a 100% P@5 for all of them.

4.4. Robustness of the Model

In the following subsection, experiments to determine the robustness of the model to imperfect conditions are performed. Experiments regarding rotation, blur, compression and occlusion are analyzed.

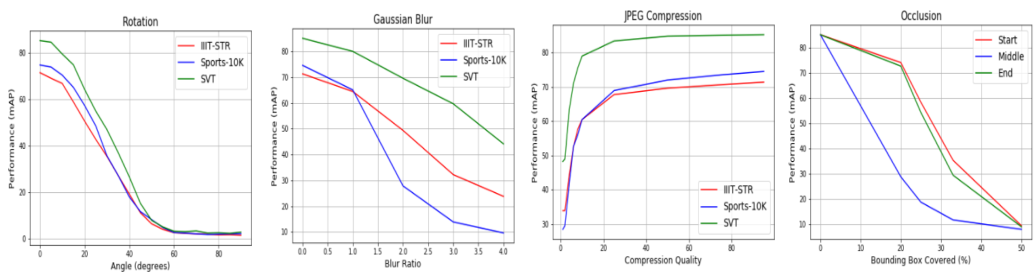


Figure 9: Robustness performance for imperfect conditions such as rotation, blur, compression and occlusion.

4.4.1. Rotation

A big difference between text found in documents and text in natural imagery is the arbitrary orientation text may have. Rotated and arbitrary shaped text instances are one of the main problems in the research community. Challenges such as the one presented in [78] remains as an open problem and an active field, in which the task is far from being a trivial one. Experiments to assess the model performance and robustness towards rotation were conducted. Each image from the analyzed datasets was rotated by a specific angle starting at 0° to 90° in steps of 5°, clockwise and counter clockwise. The images were rotated by considering the center of the image as the reference point as it is show in Figure 10. Bi-linear interpolation was used in order to avoid losing information and padding was used in order to avoid cutting-off sections that contain text in an image.

Figure 10: From left to right, qualitative rotated sample image at 0°, 20°, 40°, 60° and 80° taken from SVT dataset. Spatial positioning of characters is lost at high rotation angles, thus decreasing the model capability of constructing the PHOC representation.

As it can be seen in Figure 9, the greater the rotation angle is applied in the image, the performance of the model decreases. The rotation effect is amplified in the IIIT-STR dataset due to the fact that it already contains text in different orientations when compared to the more stable and horizontal text occurrences found in the remaining two datasets. It is worth noting that the proposed model was trained employing a synthetic dataset that included rotated words up to an angle of 15 degrees. This effect is perceived by noticing a significant decrease in performance (increase in gradient) whenever an image is rotated more than 25 degrees. Another fact that decreases the rotation performance in angles that approach to 90° is the shape of the predefined anchor boxes (Figure 4 c.), which possess a shape that mostly captures horizontal text. The orientation of text is key at the moment of building the PHOC representation of a word. This representation is constructed by considering spatial information of each character contained in a string, which is heavily affected by rotated words contained in an image.

4.4.2. Blur

Blur in text is a common issue in incidental images [13] as well as in video frames, specially on videos that contain rapid camera movement, fast scene transitions and not professional cameras. Different kernel sizes of Gaussian blur are employed to assess the proposed model performance, implemented by using [79]. As it can be seen in the qualitative results depicted in Figure 11, humans will not have a difficult time recognizing most of the text occurrences in blurred images. Blur is a particularly big problem in the Sports-10K dataset, which due to its nature, video frames depicted already contain blurry text. Gaussian blur augments this issue, thus a sharp decrease in performance is noted when compared to the remaining datasets, see Figure 9. Further strategies of data augmentation with blurred images or de-blurring techniques as presented by [80] can be used as an additional step before inference time.

Figure 11: Increasing Gaussian blur in a sample image taken from IIIT-STR dataset. Fine features that differentiate characters are lost, thus affecting the ability of the proposed model to recognize a word.

4.4.3. Compression

Compression in images and video can severely degrade the image quality, thus affecting subtle details that impact the performance of deep nets. In order to simulate real life compression issues, different lossy compression qualities were employed to downgrade images in the proposed datasets by using the JPEG compression algorithm. The perception in quality degradation is not linear, thus more emphasis was placed in extreme scenarios (low compression qualities). The compression method used was taken from the public implementation from [79] and different quality values were employed. As it can be seen in the qualitative results depicted in Figure 12, changes in quality above the value of 25 are barely noticeable by human perception alone. Despite this fact, as it can be seen in Figure 9, our model achieves a comparable performance with previous state of the art methods depicted in Table 2 even when the input image belongs to a low quality compression range. It is worth pointing out that for quality values of 20 and above the performance gradient tends to decrease making the performance grow slowly until achieving state of the art reported values in images with a higher compression quality. Similarly to the blur problems encountered in previous section, the Sports-10K dataset is the most susceptible to low image qualities, due to the collecting process of this dataset at the moment of extracting frames from video.

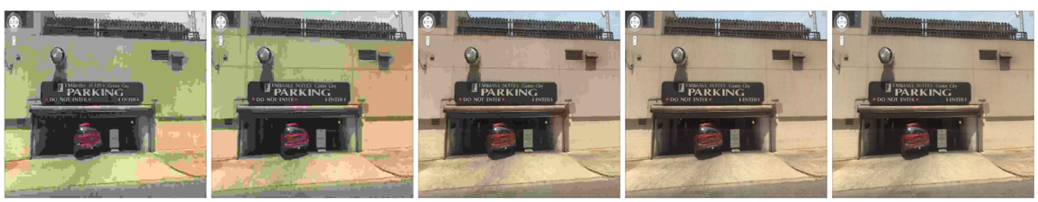


Figure 12: Increasing compression quality from left to right (1, 4, 8, 25, 75) in sample image from SVT dataset. At low qualities text at small scales is barely legible. Despite this effect, our model achieves state of the art level performance at qualities bigger than 20.

4.4.4. Occlusion

An ongoing challenge in the scene text reading community is occlusion, as it can entirely modify the morphology of spotted text. Humans are less prone to occlusion problems, due to prior knowledge of the context of an image or by the existing familiarity towards a specific language. In our experiments, three scenarios were proposed according to the position of the occlusion, namely at the beginning, middle and end of a word. These experiments were conducted only in the SVT dataset because it was the only one that already contained bounding box labels. The occlusion was generated by extreme blurring of a given percentage of the area that contains text in an image. The percentages of occlusion employed were half, one third, one fourth and one fifth of the total bounding box area, some qualitative samples can be seen in Figure 13. Not all text occurrences in a given image are occluded, because there are words that do not contain any ground truth annotations provided in the SVT dataset. As it can be seen in Figure 9, when the occlusion is located at the beginning and end of a word the model achieves a similar performance which slowly decreases as the occluded area grows. The model learns to build the PHOC of the occluded word, and successfully retrieves the closest matching representation. This outcome can be seen in Figure 14, in which the model successfully retrieves ocludded images for the query ”adidas”. However, when the occlusion affects the center of a word, the model achieves a lower performance at the moment of retrieving a specific query. This outcome can be easily explained because the detected text is treated as two different word occurrences, thus generating different proposals that actually belong to the same word.

Figure 13: Occlusion samples. From left to right: occlusion located at the beginning of the image occupying 1/3 of the total bounding box area, occlusion at the beginning involving 1/5, occlusion at the middle filling 1/3, and occlusion at the end covering 1/3 and 1/5 of the total bounding boxes area respectively.

Figure 14: Images within the top 10 ranked images for the query "adidas". Our model successfully retrieves partially occluded and blurred words.

4.5. Real-time Text Spotting in Videos

Given the high processing frame-rates that we achieve (c.f. Table 2), we can use the proposed method for spotting text in video streams in real time. Such application might be interesting in scenarios like assistance to driving systems, in order to spot certain words in the open world, or to track advertisement exposure in sports broadcasting. In such cases, the user casts a textual query that has to be sought within videos. We shall take into account that video recorded in natural scenes contain text instances that are extremely susceptible to imperfect conditions. Low quality of recording devices and rapid camera movement tends to produce blurred and rotated content. Text found in video is also vulnerable to unintended occlusions that affect several consecutive frames. In order to test the performance of the proposed method in such scenario, we have used the Text in Videos challenge dataset [24], in which the train partition consists of 25 videos, 13.450 frames in total, with their corresponding ground-truth annotation. We decided to use as queries the 20 words having more than three letters that have more occurrences in the dataset. Having set a threshold on the distance between the query PHOC representation and the closest word hypothesis in each frame, we decide whether the queried word appears or not in that frame. We evaluate the text spotting in videos task by using the F-score, so that we

Table 6: Top 15 most frequent words with their number of occurrences and the reached F-score.

Query	Occurrences	F-score
flor	539	94.05
Marie	426	83.89
Renfe	314	78.26
createurs	303	72.40
Dixan	278	87.54
FONTANEDA	261	84.44
VOTRE	257	91.01
Digestive	254	90.00
USHIP	245	75.35
ACCASTILLEUR	241	66.26
Applus	237	91.96
Rectorat	237	88.96
CONSEIL	230	83.18
mundi	230	85.24
Accastillage	199	61.41
MISTOL	186	57.51
Average	–	76.70

penalize both missing frames where the query word appears and false positive frames. Overall we achieved an F-score of 76.70, and we provide some results for the topmost 15 queries in Table 6. Video demos are available in our public repository[1].

5. Conclusions

In this work, we presented a real-time performing word spotting method, based on a fully convolutional neural network that allows to detect and recognize text in a single calculation which yields real-time processing capa-

[1] `https://github.com/lluisgomez/single-shot-str`

bility. The introduced model significantly improves previous state of the art results on the scene text retrieval task on the IIIT-STR and Sports-10K dataset while obtaining comparable results to the state of the art in the SVT Dataset. Moreover, it can do so achieving speeds 50$\times$ to 150$\times$ speed compared to other state of the art methods, which opens up the possibility of employing this model for real time scenarios, such as video, and indexing large scale databases.

Importantly, it has been shown that the proposed method is able to construct a compact vectorial representation of out of dictionary queries at inference time, while keeping the performance at words previously seen at training. Achieving this result is possible by employing the PHOC as a word representation instead of tackling the task as a direct word classification. The method showcased is able to generalize unseen samples in a robust and efficient way, as the evidence strongly points out in experiments performed in a multilingual dataset. Additionally, the model proves to be robust at dealing with highly compressed images and text samples with occlusions at the beginning and at the end of a word. However, large rotation angles still present a problem which can be tackled by synthesizing training data with different characteristics and by using different priors when defining anchor boxes. Additional future work can be conducted to investigate the use of word embeddings that exploit the morphology of a word other than PHOC.

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

Funding

This work was partially funded by the Spanish Research project TIN2017-89779-P, the grant given by the European Social Fund 2014-2020 (CCI: 2014ES05SFOP007) and Predoctoral grant (AGAUR) number 2019-FIB01233, the H2020 MarieSkodowska-Curie actions of the European Union, grant agreement No 712949(TECNIOspring PLUS), and UAB PhD scholarship No B18P0070.

6. Bibliography

References

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Proc. of the European Conference on Computer Vision, Springer, pp. 740–755.

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

[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[4] S. S. Tsai, H. Chen, D. Chen, G. Schroth, R. Grzeszczuk, B. Girod, Mobile visual search on printed documents using text and low bit-rate features, in: 2011 18th IEEE International Conference on Image Processing, IEEE, pp. 2601–2604.

[5] G. Schroth, S. Hilsenbeck, R. Huitl, F. Schweiger, E. Steinbach, Exploiting text-related features for content-based image retrieval, in: 2011 IEEE International Symposium on Multimedia, IEEE, pp. 77–84.

[6] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, M. Rohrbach, Towards vqa models that can read, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] D. Karatzas, Ll. Gómez, M. Rusiñol, A. Biten, A. Mafla, R. Tito, E. Valveny, C. Jawahar, M. Mathew, ICDAR 2019 Robust Reading Challenge on Scene Text Visual Question Answering, http://http://rrc.cvc.uab.es/?ch=11, 2019. [Online, accessed 22-April-2019].

[8] Y. Dvorin, U. E. Havosha, Method and device for instant translation, 2009. US Patent App. 11/998,931.

[9] C. Parkinson, J. J. Jacobsen, D. B. Ferguson, S. A. Pombo, Instant translation system, 2016. US Patent 9,507,772.

[10] X. Liu, J. K. Samarabandu, A simple and fast text localization algorithm for indoor mobile robot navigation, in: Image Processing: Algorithms and Systems IV, volume 5672, International Society for Optics and Photonics, pp. 139–151.

[11] X. Liu, J. Samarabandu, An edge-based text region extraction algorithm for indoor mobile robot navigation, in: IEEE International Conference Mechatronics and Automation, 2005, volume 2, IEEE, pp. 701–706.

[12] Y. Zhu, C. Yao, X. Bai, Scene text detection and recognition: Recent advances and future trends, Frontiers of Computer Science 10 (2016) 19–36.

[13] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al., ICDAR 2015 competition on robust reading, in: Proc. of the IEEE International Conference on Document Analysis and Recognition, pp. 1156–1160.

[14] Z. He, J. Liu, H. Ma, P. Li, A new automatic extraction method of container identity codes, IEEE Transactions on intelligent transportation systems 6 (2005) 72–78.

[15] M. A. Chowdhury, K. Deb, Extracting and segmenting container name from container images, International Journal of Computer Applications 74 (2013).

[16] I. Sutskever, G. E. Hinton, A. Krizhevsky, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems (2012) 1097–1105.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (2015) 211–252.

[18] S. Long, X. He, C. Ya, Scene text detection and recognition: The deep learning era, arXiv preprint arXiv:1811.04256 (2018).

[19] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet, S. Arnoud, L. Yatziv, Ontological supervision for fine grained classification of street view storefronts, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1693–1702.

[20] S. Karaoglu, R. Tao, T. Gevers, A. W. Smeulders, Words matter: Scene text for image classification and retrieval, IEEE Transactions on Multimedia 19 (2017) 1063–1076.

[21] X. Bai, M. Yang, P. Lyu, Y. Xu, Integrating scene text and visual appearance for fine-grained image classification with convolutional neural networks, arXiv preprint arXiv:1704.04613 (2017).

[22] A. Mishra, K. Alahari, C. Jawahar, Image retrieval using textual cues, in: Proc. of the IEEE International Conference on Computer Vision, pp. 3040–3047.

[23] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, International Journal of Computer Vision 116 (2016) 1–20.

[24] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. De Las Heras, ICDAR 2013 robust reading competition, in: Proc. of the IEEE International Conference on Document Analysis and Recognition, pp. 1484–1493.

[25] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.

[26] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, arXiv preprint arXiv:1612.08242 (2016).

[27] J. Almazán, A. Gordo, A. Fornés, E. Valveny, Word spotting and recognition with embedded attributes, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 2552–2566.

[28] S. Sudholt, G. A. Fink, Phocnet: A deep convolutional neural network for word spotting in handwritten documents, in: Proc. of the IEEE International Conference on Frontiers in Handwriting Recognition, pp. 277–282.

[29] L. Gómez, A. Mafla, M. Rusinol, D. Karatzas, Single shot scene text retrieval, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 700–715.

[30] D. He, X. Yang, C. Liang, Z. Zhou, A. G. Ororbi, D. Kifer, C. Lee Giles, Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3519–3528.

[31] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, Z. Cao, Scene text detection via holistic, multi-channel prediction, arXiv preprint arXiv:1606.09002 (2016).

[32] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: A fast text detector with a single deep neural network, in: Proc. of the AAAI Conference on Artificial Intelligence, pp. 4161–4167.

[33] M. Liao, B. Shi, X. Bai, Textboxes++: A single-shot oriented scene text detector, arXiv preprint arXiv:1801.02765 (2018).

[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single shot multibox detector, in: Proc. of the European Conference on Computer Vision, Springer, pp. 21–37.

[35] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 2642–2651.

[36] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, M. Park, Pvanet: deep but lightweight neural networks for real-time object detection, arXiv preprint arXiv:1608.08021 (2016).

[37] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448.

[38] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proc. of the International Conference on Neural Information Processing Systems, pp. 91–99.

[39] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969.

[40] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE Transactions on Multimedia 20 (2018) 3111–3122.

[41] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, Z. Luo, R2cnn: Rotational region cnn for orientation robust scene text detection, arXiv preprint arXiv:1706.09579 (2017).

[42] S. Zhang, Y. Liu, L. Jin, C. Luo, Feature enhancement network: A refined scene text detector, in: Thirty-Second AAAI Conference on Artificial Intelligence.

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

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

[45] B. Shi, X. Bai, 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 39 (2017) 2298–2304.

[46] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[47] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, S. Zhou, Focusing attention: Towards accurate text recognition in natural images, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 5076–5084.

[48] F. Bai, Z. Cheng, Y. Niu, S. Pu, S. Zhou, Edit probability for scene text recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1508–1516.

[49] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, in: Advances in neural information processing systems, pp. 2017–2025.

[50] B. Shi, X. Wang, P. Lyu, C. Yao, X. Bai, Robust scene text recognition with automatic rectification, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4168–4176.

[51] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2315–2324.

[52] M. Busta, L. Neumann, J. Matas, 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.

[53] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, arXiv preprint (2017).

[54] H. Li, P. Wang, C. Shen, Towards end-to-end text spotting with convolutional recurrent neural networks, arXiv preprint arXiv:1707.03985 (2017).

[55] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, C. Sun, An end-to-end textspotter with explicit alignment and attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5020–5029.

[56] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, J. Yan, Fots: Fast oriented text spotting with a unified network, arXiv preprint arXiv:1801.01671 (2018).

[57] P. Lyu, M. Liao, C. Yao, W. Wu, X. Bai, Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes, in: Proceedings of the European Conference on Computer Vision (ECCV), pp. 67–83.

[58] D. Aldavert, M. Rusiñol, R. Toledo, J. Lladós, Integrating visual and textual cues for query-by-string word spotting, in: Proc. of the IEEE International Conference on Document Analysis and Recognition, pp. 511–515.

[59] S. K. Ghosh, E. Valveny, Query by string word spotting based on character bi-gram indexing, in: Proc. of the IEEE International Conference on Document Analysis and Recognition, pp. 881–885.

[60] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

[61] X. Bai, M. Yang, P. Lyu, Y. Xu, J. Luo, Integrating scene text and visual appearance for fine-grained image classification, IEEE Access 6 (2018) 66322–66335.

[62] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

[63] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, D. Parikh, Vqa: Visual question answering, in: Proceedings of the IEEE international conference on computer vision, pp. 2425–2433.

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

[65] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

[66] K. Lang, T. Mitchell, Newsgroup 20 dataset (1999).

[67] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: Proc. of the IEEE International Conference on Computer Vision, pp. 1457–1464.

[68] I. Bizid, J. Chazalon, H. Choi, Y. Feng, D. Karatzas, W. Khlif, Z. Luo, M. Luqman, N. Nayef, U. Pal, C. Rigaud, F. Yin J. Matas, N. Nayef, U. Pal, Y. Patel, ICDAR2017 Competition on Multi-lingual scene text detection and script identification, `http://http://rrc.cvc.uab.es/?ch=8`, 2017. [Online, accessed 22-April-2019].

[69] M. Bušta, D. Karatzas, W. Khlif, J. Matas, N. Nayef, U. Pal, Y. Patel, ICDAR 2019 Robust Reading Challenge on Multi-lingual scene text detection and recognition, `http://http://rrc.cvc.uab.es/?ch=11`, 2019. [Online, accessed 30-April-2019].

[70] M. Iwamura, L. Gomez, D. Karatzas, Robust Reading Challenge on Text in videos 2013-2015, `http://http://rrc.cvc.uab.es/?ch=11`, 2019. [Online, accessed 22-April-2019].

[71] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2963–2970.

[72] A. Mishra, K. Alahari, C. Jawahar, Top-down and bottom-up cues for scene text recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2687–2694.

[73] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3538–3545.

[74] S. K. Ghosh, L. Gomez, D. Karatzas, E. Valveny, Efficient indexing for query by string text retrieval, in: Proc. of the IEEE International Conference on Document Analysis and Recognition, pp. 1236–1240.

[75] A. Mishra, Understanding Text in Scene Images, Ph.D. thesis, International Institute of Information Technology Hyderabad, 2016.

[76] L. Gómez, D. Karatzas, Textproposals: a text-specific selective search algorithm for word spotting in the wild, Pattern Recognition 70 (2017) 60–74.

[77] M. Buvsta, L. Neumann, J. Matas, Deep textspotter: An end-to-end trainable scene text localization and recognition framework, in: Proc. of the IEEE International Conference on Computer Vision, pp. 2204–2212.

[78] Y. Sun, C Kheng, C. Chet, Y. Liu, C. Luo, Z. Ni, D. Karatzas, S. Zhang, J. Han, E. Ding, C. Semg, L. Jin, ICDAR2019 Robust Reading Challenge on Arbitrary-Shaped Text, `http://rrc.cvc.uab.es/?ch=14&com=introduction`, 2019. [Online, accessed 22-April-2019].

[79] A. Clark, et al., PIL: Python imaging library, 2010–2019. [Online; accessed 10-April-2019].

[80] X. Cao, W. Ren, W. Zuo, X. Guo, H. Foroosh, Scene text deblurring using text-specific multiscale dictionaries, IEEE Transactions on Image Processing 24 (2015) 1302–1314.

Andrés Mafla obtained a M.Sc. degree in Computer Engineering from Universitat Autònoma de Barcelona in 2018. Currently, he is a PhD student at the Computer Vision Center under the supervision of Dr. Dimosthenis Karatzas. His research interests includes text detection and recognition, scene text image retrieval, multi-modal embeddings and scene understanding.

Rubèn Tito received his Computer Engineering M.Sc. in 2018 from the Universitat Autònoma de Barcelona, and now he is doing his PhD under the supervision of Dr. Marçal Rossinyol and Dr. Ernest Valveny at the Computer Vision Center. His main research interests include text recognition, word spotting and multi-modal embeddings.

Lluís Gómez obtained his PhD in 2016 at Universitat Autònoma de Barcelona. Currently he is a TECNIOspring Research Fellow (H2020 Marie Skłodowska-Curie actions of the European Union) at the Computer Vision Center, Universitat Autònoma de Barcelona. His research interests include a variety of different topics in machine learning and computer vision.

Marçal Rusiñol received his PhD in 2009 from the Universitat Autònoma de Barcelona. He is an associate researcher at the Computer Vision Center. His main research interests include reading systems, information retrieval and performance evaluation.

Ernest Valveny received the PhD degree in 1999 from the Universitat Autònoma de Barcelona, where he is an associate professor and member of the Computer Vision Center. His research interests are on document analysis and pattern recognition, including robust reading, text recognition and retrieval, document classification and graph matching.

Dimosthenis Karatzas received his PhD in 2003 from the University of Liverpool. He is associate professor at the Universitat Autònoma de Barcelona and associate director of the Computer Vision Centre where he leads the vision and language research line. His research interests include reading systems, multi-modal embeddings, and image captioning.