

---

This is the **accepted version** of the journal article:

Mafla Delgado, Andres Patricio [et al.]. «Real-time Lexicon-free Scene Text Retrieval». *Pattern Recognition*, Vol. 110 (February 2021), art. 107656 DOI 10.1016/j.patcog.2020.107656

---

This version is available at <https://ddd.uab.cat/record/324975>

under the terms of the  license.

# Real-Time Lexicon-Free Scene Text Retrieval

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

*Computer Vision Center, Universitat Autònoma de Barcelona. Edifici O, Campus UAB, 08193  
Bellaterra (Cerdanyola) Barcelona, Spain. E-mail: {amafla,rperez,  
sdey,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 returns 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 assess the generalization capability of the model are conducted in a multilingual dataset, 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, Sounak Dey, Lluís Gómez, Marçal Rusiñol,  
Ernest Valveny, Dimosthenis Karatzas

*Computer Vision Center, Universitat Autònoma de Barcelona. Edifici O, Campus UAB, 08193  
Bellaterra (Cerdanyola) Barcelona, Spain. E-mail: {amafla,rperez,  
sdey,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 captured, stored and communicated in an explicit manner. As it is shown by several authors [1, 2], text is present in a large percentage of real-life imagery, especially in urban scenarios and documents. Adding this to the fact that there is ample availability of visual data and the importance of text, it becomes essential to develop algorithms that allow efficient information retrieval by exploiting the richness of the textual content found in images and video. Leveraging text in scene imagery provides significant boosts to tasks such as image retrieval, scene understanding, instant translation, human-computer interaction, robot navigation, assisted reading for the visually-impaired and industrial automation.

In the previous years significant advances have been accomplished, particularly since the introduction of AlexNet [3], architecture that won the ILSVRC2012 [4] 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 [5, 6]. Nonetheless, text spotting is not a trivial task and remains 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

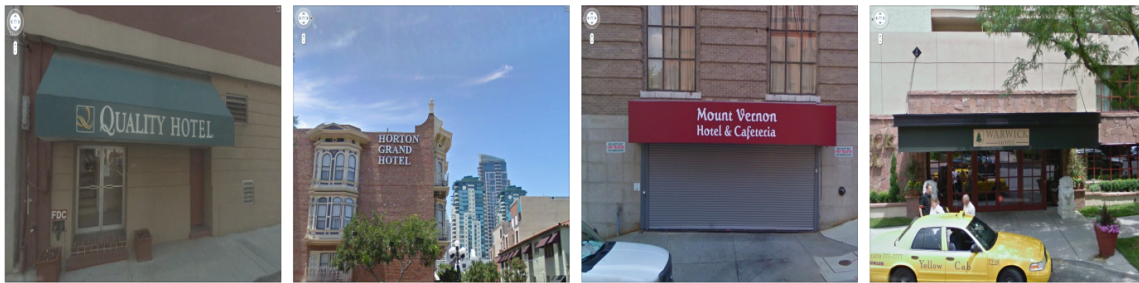


Figure 1: The visual appearance of different business places in images can be extremely variable. It seems impossible to correctly label them without reading the text in the images. Our scene text retrieval method returns all the images shown here within the top-10 ranked results among more than 10,000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

conducted by Movshovitz *et al.* [7] showed that while training a shop classifier, the proposed model ended up associating specific visual representations to textual information as the only way of differentiating between diverse businesses. The described effect is evident and addressed explicitly in later works conducted by [8, 9, 10], which focus on fine-grained classification of storefronts and bottles respectively. Additional tasks that require integration of scene text and visual information to generate a common domain knowledge have been proposed such as in [11, 12], which opens up new research paths.

Closely related to our work, Mishra *et al.* [13] 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 to out of dictionary queries never seen before. An intuitive approach to tackle such a problem is to make use of state of the art reading systems, and use their output predictions to find the closest match with the given query. However, as it has been shown by [13], 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 [14] or as it is proposed by [15, 16] 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 also worth noting that end to end reading systems usually consist of at least two clearly defined stages that employ the encoder-decoder paradigm. The pipelines comprised by these two stages, more often than not are slow at the moment of generating pre-

dictionaries of text contained in an image. These time constraints hinder the use of such algorithms in real-time scenarios or at the stage 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 efficiently encodes text in a single calculation of a Fully Convolutional Neural Network (FCNN). Inspired by backbones used for object detection, we have modified them to predict a PHOC (Pyramidal Histogram Of Characters) [17, 18] descriptor of detected scene text instances. Our experiments show that by using a custom YOLOv2 model [19] we obtain the best architecture in terms of trade-off between performance and efficiency. By employing this methodology, our model is able to perform text detection and encoding in a single calculation thus making it suitable for real time applications or for indexing large scale image collections at an unmatched speed while achieving a generalization capability to unseen text instances at training time.

The main contributions of this research, building and extending on our previous work [20], are: first, the usage of a model comprised by an end-to-end jointly trainable FCNN. Second, 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. Third, due to its design, the adoption of this method achieves an unmatched inference time of 42 images per second when processing images while at the same time achieving a new state-of-the-art.

As an extension to our previous research, in this work we explore different frameworks and backbones inspired from the most common object detection pipelines. Additionally, we present exhaustive experiments on the effect of different PHOC sizes, which yields an incremental improvement when compared to our previous model. We also analyze deeply the capacity of our model in dealing with out-of-vocabulary queries, by conducting several experiments in two multi-lingual datasets. These experiments prove that the proposed method is able to transfer knowledge acquired at training time to construct word representations of previously unseen text samples at inference time. Lastly, we propose an application of real-time text spotting on video, in which the model corroborates its robustness to noise, blur 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 [2], 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 [6, 5, 21] divide the existing methods into text detection, text recognition, and end-to-end systems. The task that this work focuses on, scene text retrieval, is also described in the upcoming sections.

## 2.1. Scene Text Detection

Deep learning methodologies usually follow a two step pipeline that comprises an end-to-end trainable detection network and a post processing step. A branch of research in text detection focuses at the pixel level of an image. In the work presented by [22], a CNN is used to predict if a given pixel belongs to a character, forms part of a text region and its orientation. Analogously, in [23] an FCN is trained to classify whether a pixel belonged to a specific character.

The work presented by [24, 25] named Textboxes, adopts a modified version of a popular object recognition model named Single Shot Detector [26]. 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 [27], which up-samples feature maps gradually and uses [28] as the network backbone, and outputs a per pixel word or text line prediction followed by a NMS step.

Based on object detection frameworks proposed by R-CNN [29, 30, 31], alternative text detection pipelines have been explored. 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 [32], rotated region proposals are presented, mostly to handle arbitrary oriented text. Analogously, R2CNN [33] the Region of Interest(ROI) pooling stage uses different fixed sizes which are concatenated for regression and classification. The work conducted by [34] 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.* [35] 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.* [36] in handwriting recognition, the same methodology has been widely used in scene text as well. The work proposed by [37] employs the CTC layer after passing the input image through a CNN that acts as the encoder and a Recurrent Neural Network (RNN) that act as the decoder.

131 The introduction of an attention mechanism was initially proposed by [38] in the  
 132 task of machine translation. This mechanism was briefly adopted in several vision  
 133 tasks, including text recognition. The work proposed by [39], namely Focus Atten-  
 134 tion Network, employs attention to supervise relevant locations for word recognition.  
 135 Jaderberg *et al.* [40] proposed the Spatial Transformer Network, which is used by [41]  
 136 to align detected text horizontally to further employ an attention based recognizer.

### 137 2.3. End-to-End Text Recognition

138 An introductory approach proposed by Jaderberg *et al.* [14] employs a sliding  
 139 window to extract proposals, which are filtered and a CNN is used to regress the  
 140 bounding boxes. Later the filtered regions that surpass a threshold are classified. In  
 141 another work, Gupta *et al.* [42] defined a Fully Convolutional Regression Network  
 142 for text detection and bounding box regression and the same classification network  
 143 proposed by [14] for text recognition, being one of the first models that were fully  
 144 trainable based on deep learning methodologies solely. In [43] a YOLO[19] based  
 145 CNN is adopted to detect text instances, which later are passed through a Con-  
 146 nectionist Temporal Classification module for recognition. These two stages are trained  
 147 separately and later connected together to form an end-to-end architecture. He *et al.*  
 148 [44] use a CNN to extract proposals, which are fed into an LSTM to refine the  
 149 bounding boxes that are later employed as input to yet another LSTM to perform  
 150 recognition that fixes misalignment between attention maps and ground truth char-  
 151 acter labels. In parallel, additional work has been conducted into the development  
 152 of multilingual scene text recognizers, such as the work of [45] which consists on  
 153 two CNNs. The first one is optimized to detect text and the second one employs  
 154 a Connectionist Temporal Classification (CTC) [36] module for recognition, while  
 155 training both in an end to end manner. In concurrent work, [46] uses EAST [27] to  
 156 obtain text regions and employs a CTC recognition module to obtain an end-to-end  
 157 reading system. Lyu *et al.* [47] use a variation of Mask R-CNN[31] to detect text in  
 158 arbitrary shapes and segment an image in different instances to recognize similar text  
 159 regions. Similarly, in order to detect and recognize oriented text, the model FOTS  
 160 was proposed by [46], which comprises of a CNN followed by a ROI Rotate operation,  
 161 which applies a transformation on oriented text to obtain axis-aligned features maps  
 162 that are further recognized by a bi-directional LSTM and a CTC decoder.

### 163 2.4. Scene Text Retrieval

164 Related to our work, the scene text retrieval problem slightly differs from classical  
 165 scene text recognition methodologies. In a retrieval scenario the user defines a textual  
 166 query which he wants to retrieve, whereas most of recognition approaches are based

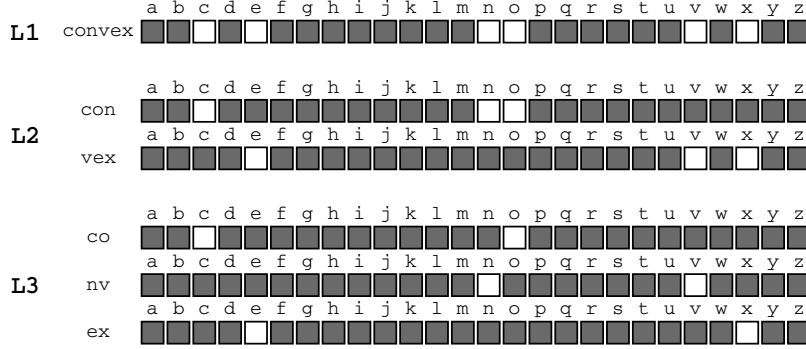


Figure 2: Pyramidal histogram of characters (PHOC) [17] of the word “convex” at levels 1, 2, and 3. The final PHOC representation is the concatenation of the partial encodings.

on employing a predefined vocabulary of the words one might come along within scene images. For instance, both Mishra *et al.*[13], who introduced the scene text retrieval task, and Jaderberg *et al.* [14], 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 [17, 48]. 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 YOLOv2 [19] that encodes found text instances in an image. We suggest that this approach brings many benefits, mostly due to the high recall and single shot calculation required to locate and recognize text contained within an image, accompanied by an unmatched processing speed.

### 3. Proposed Architecture

The proposed architecture is based on a custom-built YOLOv2 object detection model introduced by [19]. This work adapts the object detector to output a compact representation of the text instances and recast them as a PHOC [17], thus enforcing the model to learn to construct such a vectorial encoding. 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.

Batch normalization is used after every convolutional layer to help the model reach convergence by reducing the internal covariate shift. In total the model employs

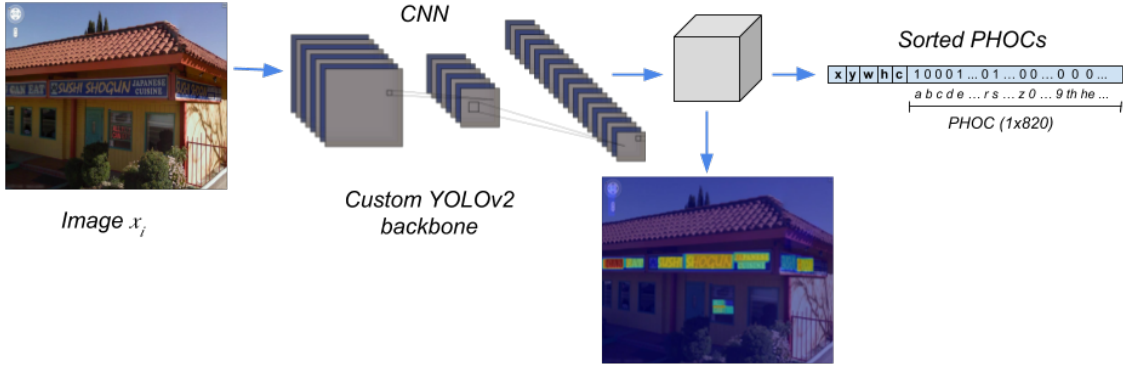


Figure 3: Our Convolutional Neural Network predicts at the same time bounding box coordinates  $x, y, w, h$ , an objectness score  $c$ , and a pyramidal histogram of characters (PHOC) of the word in each bounding box.

5 max pooling layers, which reduces the input width and height by a factor of  $2^5$ . The filter size used in convolutions is  $3 \times 3$  and the channel number is doubled after each pooling step as in previous works that adopt a VGG [49]-like backbone such as the work presented by [37]. In order to apply dimensionality reduction and decrease the computation cost, the strategy proposed by the usage of an Inception module [50] is taken, and filters of size  $1 \times 1$  are interleaved between the  $3 \times 3$  convolutional filters to obtain richer feature maps. Inspired in the Residual blocks introduced by [51], the convolutional backbone uses a pass-through layer from an earlier convolutional layer, which is concatenated and followed by a final  $1 \times 1$  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 [19]. However, the problem raises in complexity 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 [14]), 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 [17] descriptor is a multi-level vectorial representation of text strings that focuses on encoding the presence of a specific character 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 learn 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, is an improvement over our previous work [20] and employs a higher dimensional binary vector of length of 820 dimensions. This vector is formed by concatenating the L2 to the L6 unigram levels along with 2 levels of the 50 most common English language bigrams. 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. Firstly, 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 predicted PHOC vector comprises a set of numbers that satisfy the condition given by:

$$S = \{x | x \in \mathbb{R}, 0 \leq x \leq 1\} \quad (1)$$

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.

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

$$255 \quad L(b, C, c, \hat{b}, \hat{C}, \hat{c}) = \lambda_{box} L_{box}(b, \hat{b}) + L_{obj}(C, \hat{C}, \lambda_{obj}, \lambda_{noobj}) + \lambda_{cls} L_{cls}(c, \hat{c}) \quad (2)$$

256 where  $b$  is a vector with coordinates' offsets to an anchor bounding box,  $C$  is the  
 257 probability of that bounding box containing an object,  $c$  is the one hot classification  
 258 vector, and the three terms  $L_{box}$ ,  $L_{obj}$ , and  $L_{cls}$  are respectively independent losses  
 259 for bounding box regression, objectness estimation, and classification. All the afore-  
 260 mentioned losses are essentially the sum-squared errors of ground truth  $(b, C, c)$  and  
 261 predicted  $(\hat{b}, \hat{C}, \hat{c})$  values. At the moment of predicting a PHOC,  $c$  (ground truth) is  
 262 a binary vector and  $\hat{c}$  (prediction) meets the condition stated in equation 1, therefore  
 263 a multi-label cross-entropy loss function is employed as:

$$264 \quad L_{cls}(c, \hat{c}) = \frac{-1}{N} \sum_{n=1}^N c_n \log \hat{c}_n + (1 - c_n) \log(1 - \hat{c}_n) \quad (3)$$

265 Where  $N$  represents the dimensionality of the PHOC descriptor. It is important  
 266 to note that the combination of the sum-squared errors  $L_{box}$  and  $L_{obj}$  with the cross-  
 267 entropy loss  $L_{cls}$  is controlled by the scaling parameters  $\lambda_{box}$ ,  $\lambda_{obj}$ ,  $\lambda_{noobj}$ , and  $\lambda_{cls}$ ,  
 268 which require a careful tuning strategy during training.

269 Apart from the modifications made so far on top of the original YOLOv2 architec-  
 270 ture, we also changed the number, the scales, and the aspect ratios of the pre-defined  
 271 anchor boxes used by the network to predict bounding boxes. As an enhancement  
 272 strategy when compared to our previous model, we ran a K-Means model with 13  
 273 centroids to obtain the initial anchor boxes priors that better capture the real dis-  
 274 tribution of text shapes. The centroids were obtained by gathering the width and  
 275 height of all the text instances found in the datasets used at training time. Finally,  
 276 the proposed model predicts 13 anchor boxes per grid cell as output features.

277 At test time, our model provides a total of  $W/32 \times H/32 \times 13$  bounding box  
 278 proposals, where  $W$  and  $H$  are the image input size (width and height), each one of  
 279 them with an objectness score ( $\hat{C}$ ) and a PHOC prediction ( $\hat{c}$ ). The original YOLOv2  
 280 model filters the bounding box candidates with a detection threshold  $\tau$  considering  
 281 that a bounding box is a valid detection if  $\hat{C}max(\hat{c}) \geq \tau$ . If the threshold condition  
 282 is met, a non-maximal suppression (NMS) strategy is applied in order to get rid of  
 283 overlapping detections of the same object. In our case the threshold is applied only  
 284 on the objectness score ( $\hat{C}$ ) but with a much smaller value ( $\tau = 0.0025$ ) than in the  
 285 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 per 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

The model was trained with a combination of several datasets. The first one is a modified version of the synthetic dataset of Gupta *et al.*[42]. 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.*[14], instead of the Newsgroup20 [52] dataset 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 not include 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 4 shows a set of samples of our training data. Additional 10k synthetic images proposed by [53] were added to the training data. This dataset has shown to improve scene text detectors and recognizers. Real images taken from the 1,233 training images found in the ICDAR2013 [15] and ICDAR2015 [16] datasets were also added in the training set.

We trained the improved version of the model for 10 epochs with an initial learning rate of 0.001 and a batch size of 16, however a gradient accumulation strategy was employed every 4 iterations, thus resulting in a total batch size of 64.

The training process involved the combination of two recent optimizers, RAdam [54] and the Lookahead [55] which when working in combination yielded an improved model. We initialized the weights of our model with the backbone pre-trained on Imagenet. At this stage, we employed a fixed input image size of  $448 \times 448$ . After the initial 10 epochs, we decreased the learning rate to 0.0001 and trained the model for 5 additional epochs. At this point we also adopted multi-scale training by randomly resizing input images among 13 possible sizes in the range from  $352 \times 352$  to  $768 \times 768$



Figure 4: Synthetic training data generated with a modified version of the method of Gupta *et al.* [42]. We make use of a custom dictionary with the 90K most frequent English words, and restrict the range of random rotation to 15 degrees.

to increase the generalization ability of the model towards different sized images. As a way to fine-tune the model to predict text instances in real images, a final epoch was employed, which involved the usage of images only from the ICDAR2013 and ICDAR2015 datasets. Through the training process we kept the parameter  $\lambda_{cls}$  set to 2.0 to increase the weight placed to learn accurate PHOCs, the parameter  $\lambda_{box}$  was set to 5.0 and the rest of the parameters were fixed to 1.0. During the whole training process we used a data augmentation strategy that involved randomly cropping the center of an image without affecting text instances and adding saturation and hue.

## 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 several experiments performed as a comparison with the published state-of-the-art. As an extension to our previous work [20], experiments on employing different backbones are presented, along with exhaustive exploration on the effect of different PHOC dimensions. A follow-up assessment when dealing with out-of-vocabulary words is conducted by analyzing the model with two multi-lingual datasets. 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 [13] 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

348 appears in a wide range of different fonts and styles, and from different view points,  
349 among many distractors (images without any text).

#### 350 4.1.2. *IIIT Sports-10k dataset*

351 The Sports-10k dataset [13] is another scene text retrieval dataset composed of  
352 10,000 images extracted from sports video clips. It has 10 predefined query words  
353 with their corresponding relevant images' lists. Scene text retrieval in this dataset is  
354 specially challenging because images are low resolution and often noisy, occluded or  
355 blurred, with small text generally located on advertisements signboard.

#### 356 4.1.3. *Street View Text (SVT) dataset*

357 The SVT dataset [56] is comprised of images harvested from Google Street View  
358 where text instances are present. The scene text found exhibits great variability  
359 and is usually related to business names and descriptions. It contains more than  
360 900 words annotated in 350 different images. In our experiments we use the official  
361 partition that splits the images in a train set of 100 images and a test set of 249  
362 images. This dataset also provides a lexicon of 50 words per image for recognition  
363 purposes, but we do not make use of it. For the image retrieval task we consider as  
364 queries the 427 unique words annotated on the test set.

#### 365 4.1.4. *Multi-lingual scene text (MLT) datasets*

366 These two datasets MLT2017 [57] and MLT2019 [58] are scene text detection  
367 and recognition datasets that contain 7,200 and 10,000 images respectively in 10  
368 different languages (Chinese, Japanese, Korean, English, French, Arabic, Italian,  
369 German, Bangla and Hindi) in equal proportions, representing 7 different scripts.  
370 These datasets mostly comprises focused text in natural images, and even though the  
371 main task is text detection and recognition, we adapted it to conduct text retrieval  
372 experiments. We employ this dataset to assess the generalization power of the PHOC  
373 representation of unseen words at training time.

#### 374 4.1.5. *Text in videos (TiV) dataset*

375 The TiV dataset [59] contains 25 videos (13450 frames in total) and a test set of 24  
376 videos (14374 frames in total) recorded from 4 different cameras. We use this dataset  
377 to asses the performance at real-time of our model at the moment of retrieving a  
378 specific text query. The challenge in this dataset remains in the fact that usually  
379 video frames contain a lower quality when compared to static images. The problems  
380 of text spotting usually relate to rotation, blur and occlusion of text found on each  
381 frame due to movement and focusing issues while including loss of information at  
382 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 queried 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 consists of considering only the top- $n$  ranked images and calculating the precision at this specific cut-off point ( $P@n$ ).

In this section we show exhaustive experiments conducted to assess the best suited model for scene text retrieval. We provide results by employing different backbones inspired from the most commonly used object detection frameworks along with experiments assessing the effect of different PHOC dimensions on the textual representation.

### 4.2.1. Backbones Comparison

Several works analyzing a wide range of backbones have been proposed to tackle the object detection task [60, 30, 26, 61]. More often than not, these frameworks are used as a starting point to construct text detectors and recognizers which rather than focusing on objects, are designed to detect and recognize textual instances. The model presented in our previous work [20] employs a customized backbone inspired from YOLOv2 [19]. Additionally, we include the results of the following frameworks: YOLOv3 [62], Faster R-CNN [30] and RetinaNet [60] modified to predict a PHOC embedding given that a text instance was detected. Adapting each of the mentioned frameworks for scene text retrieval required specific design choices and training methodologies that are explained as follows.

YOLOv3 [62] is an incremental improvement on the YOLOv2 model. A significantly bigger number of convolutional layers were added and if considering additional layers such as routing, upsampling and shortcut layers, the total number increases to 106 layers. The custom YOLOv3 used in our experiments makes predictions at each grid cell through 3 different scales and employs 4 different anchors boxes per scale. The initialization of the model is given by the original YOLOv3 weights pre-trained on ImageNet [63]. Rather than predicting a class label, a PHOC prediction is performed on detected text and the most confident PHOCs are employed as scene text retrieval features. Since YOLOv3 already uses binary cross entropy to make predictions, we did not make any change to the original loss function when working with a PHOC descriptor.

420 The second framework is taken from the work by [60], which presents a model  
 421 named RetinaNet. This model leverages a Feature Pyramid Network (FPN) [64] to  
 422 extract features across different layers of a network. The resulting features are further  
 423 used to perform bounding box regression and classification by two separate branches.  
 424 A novel loss function that focuses on hard, misclassified samples while reducing the  
 425 relative loss on well classified ones is adopted. In our experiments we used as the FPN  
 426 backbone a ResNet-18 [51] pre-trained on ImageNet. Since the RetinaNet model does  
 427 not output a confidence parameter, we modified the bounding box regression branch.  
 428 The network was trained to predict a confidence value based on the IoU between the  
 429 predicted anchor boxes and the ground truth text annotations. We added a MSE  
 430 Loss term to predict this new confidence output which was added to the total function  
 431 to be optimized. The evaluated model outputs 9 anchor predictions per each FPN  
 432 level, across 5 different levels (L3-L7) readjusted to capture text shape. Similar to  
 433 the idea from [20], at inference time we use the confidence values to sort the most  
 434 confident PHOCs predicted from the scene text contained in an image.

435 The third evaluated framework is a Faster R-CNN, a multi-stage object detec-  
 436 tor [30]. The Faster R-CNN comprises of two stages, a Region Proposal Network  
 437 (RPN) and a Region Of Interest (ROI) Pooling stage used for later classification and  
 438 regression. The backbone of this model is a VGG-16 [49] which extracts common  
 439 features for the RPN and ROI layers. In the RPN layer, 9 anchors are proposed at  
 440 each sliding window, yielding coordinates and scores. The coordinates refer to the  
 441 location of regions from the convolutional features to be used as input into the ROI  
 442 stage to be later regressed and used to predict PHOCs by two different branches. The  
 443 scores indicate the presence of text and background according to the IoU between  
 444 predicted anchors and ground truth labels. We use this score to sort the PHOC  
 445 predictions at inference time accordingly.

446 It is paramount to indicate that training and finding a global optima among all  
 447 these models is not a trivial task. The additionally explored frameworks contain a  
 448 significantly larger amount of parameters compared to the original model proposed  
 449 by [20], which in turn produce a slower training procedure. Such models use several  
 450 hyper-parameters that can be further tuned but are specific to each one. Due to the  
 451 computational resources involved in fine-tuning and optimizing each hyper-parameter  
 452 of the explored models individually, we trained all models for 10 epochs. The PHOC  
 453 dimension was kept constant across all of them by using the most common size in  
 454 literature 604-d [17][20][18]. A gradient accumulation policy was employed to train  
 455 the additional frameworks in a way that the resulting gradient at back-propagation  
 456 was applied to a larger batch of samples, in a similar manner as the original YOLOv2  
 457 model. This methodology was adopted because the explored backbones contained

Table 1: Comparison of different frameworks and backbones in the scene text retrieval task. Each model predicts PHOCs of textual instances found and the most confident ones are employed as textual features for retrieval. The metric employed is the mean average precision (mAP).

| Framework    | Backbone   | Training Hours | IIIT         | Sports10K    | SVT          | FPS       |
|--------------|------------|----------------|--------------|--------------|--------------|-----------|
| RetinaNet    | ResNet 18  | 888            | 63.29        | 74.64        | 79.31        | 11        |
| Faster R-CNN | VGG 16     | 960            | 66.21        | 76.08        | 80.53        | 5         |
| YOLOv3       | Darknet 53 | 288            | 67.99        | <b>79.33</b> | 83.59        | 20        |
| YOLOv2       | Darknet 19 | <b>144</b>     | <b>69.92</b> | 78.01        | <b>85.12</b> | <b>43</b> |

a big number of parameters that could not be optimized by using a single GPU otherwise.

The results obtained by using different models are summarized in Table 1. We note that by employing the YOLOv3 backbone better results are obtained in the Sports10K dataset. However, the outcome of an increased performance on one dataset comes at the cost of a longer convergence time for the model at training stage and also a significant decrease at inference (50%) measured by frames per second. We should remark that Faster R-CNN and RetinaNet due to their slow processing speed are not suitable for the purpose of obtaining a real time and efficient text spotter. The complexity of these two models is significantly higher than YOLOv2 or YOLOv3, bringing in a substantial time required to achieve convergence, a key asset especially when working with limited computational resources. In order to preserve real time processing capabilities, efficient large scale indexing and faster convergence time at training we have decided to employ the YOLOv2 backbone for the rest of our experiments.

#### 4.2.2. Effect of PHOC Size

In order to assess the effect of the PHOC vector size, various retrieval experiments were conducted by incrementally adding up the pyramidal unigram levels and the redundancy of the bigrams that form the descriptor. Due to computational constraints and in order to provide a grounded comparison, all the evaluated models were trained for 10 epochs. The backbone employed was the custom variation of YOLOv2 [19] and the different loss function hyper-parameters were kept similar in all scenarios.

Table 2 shows the results obtained by employing different PHOC sizes. It is evident that using single unigram levels do not yield good results because spatial information of the characters found within a word is lost. It is important to point that higher unigram levels contain richer spatial information that is discriminative when differentiating similar words, thus achieving a higher score than employing

Table 2: Comparison of the effect of different PHOC sizes by employing different unigram and bigram levels. The backbone employed on all the experiments is a customized version of YOLOv2 [19]. The metric employed is the mean average precision (mAP) across all datasets.

| PHOC Evaluated        |               |            | Dataset      |              |              |
|-----------------------|---------------|------------|--------------|--------------|--------------|
| Unigram Levels        | Bigram Levels | Dimension  | IIIT         | Sports10K    | SVT          |
| L1                    | 0             | 36         | 47.84        | 67.31        | 70.17        |
| L2                    | 0             | 72         | 54.83        | 69.54        | 73.07        |
| L3                    | 0             | 108        | 53.94        | 69.92        | 73.67        |
| L4                    | 0             | 144        | 54.69        | 69.33        | 73.21        |
| L5                    | 0             | 180        | 56.53        | 68.44        | 72.39        |
| L6                    | 0             | 216        | 57.14        | 69.47        | 73.56        |
| L2+L3                 | 0             | 180        | 58.77        | 68.56        | 74.68        |
| L2+L3+L4              | 0             | 324        | 62.35        | 70.73        | 75.84        |
| L3+L4+L5              | 0             | 432        | 61.46        | 68.87        | 76.51        |
| L2+L3+L4+L5           | 0             | 504        | 62.19        | 72.52        | 76.98        |
| L2+L3+L4+L5+L6        | 0             | 720        | 64.57        | <b>75.29</b> | 78.16        |
| L1+L2+L3+L4+L5        | 0             | 540        | 61.29        | 72.76        | 76.19        |
| L3+L4+L5              | 1             | 482        | 62.53        | 70.79        | 75.15        |
| L3+L4+L5              | 2             | 532        | 63.35        | 73.28        | 76.72        |
| L2+L3+L4+L5           | 1             | 554        | 62.62        | 73.41        | 77.55        |
| L2+L3+L4+L5           | 2             | 604        | 62.89        | 73.88        | 77.76        |
| L2+L3+L4+L5+L6        | 1             | 770        | 63.64        | 73.31        | 78.63        |
| <b>L2+L3+L4+L5+L6</b> | <b>2</b>      | <b>820</b> | <b>66.60</b> | 74.35        | <b>79.19</b> |

486 lower unigrams. Intuitively, concatenating several unigram levels provide a much  
 487 informative vector of the positioning of characters in a word, thus resulting in a  
 488 higher precision. It is worth mentioning that combinations of PHOCs of higher levels  
 489 usually yield better results than combinations of lower unigram levels since higher  
 490 levels contain more discriminative information, particularly for longer words. Also,  
 491 we should indicate that unigram levels contain information that is more relevant for  
 492 the retrieval task than the one contained in bigrams, nonetheless adding bigram levels  
 493 contribute to improvement of the model. Lastly, by employing bigrams combined  
 494 with several unigram levels, a more informative vector is obtained at the expense  
 495 of a higher dimensional representation. As such, the best performing PHOC in our  
 496 experiments is the one that combines 5 levels (L2-L6) jointly with the concatenation  
 497 of 2 bigrams levels, resulting in a 820 dimensional vector.

Table 3: 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 [13], not by the original authors. (†) Results computed with publicly available code from the original authors.

| Method                                               | IIIT<br>(mAP)      | Sports<br>(mAP)    | SVT<br>(mAP)       | fps         |
|------------------------------------------------------|--------------------|--------------------|--------------------|-------------|
| SWT [65]+ Mishra et al. [66]                         | -                  | -                  | 19.25              |             |
| Wang <i>et al.</i> [56]                              | -                  | -                  | 21.25*             |             |
| TextSpotter [67]                                     | -                  | -                  | 23.32*             | 1.0         |
| Mishra <i>et al.</i> [13]                            | 42.7               | -                  | 56.24              | 0.1         |
| Ghosh <i>et al.</i> [68]                             | -                  | -                  | 60.91              |             |
| Mishra [69]                                          | 44.5               | -                  | 62.15              | 0.1         |
| Almazán <i>et al.</i> [17]                           | -                  | -                  | 79.65              |             |
| TextProposals [70] + DictNet [35]                    | 64.9 <sup>†</sup>  | 67.5 <sup>†</sup>  | 85.90 <sup>†</sup> | 0.4         |
| Jaderberg <i>et al.</i> [14]                         | 66.5               | 66.1               | <b>86.30</b>       | 0.3         |
| Bušta <i>et al.</i> [71] ICCV 2017                   | 62.94 <sup>†</sup> | 59.62 <sup>†</sup> | 69.37 <sup>†</sup> | 44.21       |
| Liu <i>et al.</i> [46] CVPR 2018                     | 52.82 <sup>†</sup> | 65.62 <sup>†</sup> | 68.98 <sup>†</sup> | 20.26       |
| Bušta <i>et al.</i> [45] ACCV 2018                   | 64.39 <sup>†</sup> | 68.29 <sup>†</sup> | 76.44 <sup>†</sup> | 12.2        |
| He <i>et al.</i> [44] CVPR 2018                      | 50.16 <sup>†</sup> | 50.74 <sup>†</sup> | 72.82 <sup>†</sup> | 1.25        |
| He <i>et al.</i> [44] (With dictionary)              | 66.95 <sup>†</sup> | 74.27 <sup>†</sup> | 80.54 <sup>†</sup> | 2.35        |
| He <i>et al.</i> [44] (PHOC)                         | 46.34 <sup>†</sup> | 52.04 <sup>†</sup> | 57.61 <sup>†</sup> | 2.35        |
| YOLO-PHOC 604-d ( $576 \times 576$ ) [20]            | 68.13              | 72.99              | 82.02              | <b>53.0</b> |
| YOLO-PHOC 604-d ( $608 \times 608$ ) [20]            | 69.83              | 73.75              | 83.74              | <b>43.5</b> |
| YOLO-PHOC 604-d (multi-res.) [20]                    | 71.37              | 74.67              | 85.18              | 16.1        |
| <b>YOLO-PHOC 820-d (<math>608 \times 608</math>)</b> | <b>71.67</b>       | <b>80.96</b>       | 85.74              | <b>42.2</b> |

#### 4.2.3. Comparison with SOTA

The performance of our improved model as well as the previous state of the art methods is shown in Table 3. The models are evaluated for scene text image retrieval on the IIIT-STR, Sports-10K, and SVT datasets. At inference time our model employs an input image size of ( $608 \times 608$ ), and the processing time has been calculated using a Titan X (Pascal) GPU with a batch size of 1.

The new proposed method outperforms by a significant margin our previous model [20] by more than 6 points on mAP on the Sports dataset. The improvement is smaller on the other two datasets, nonetheless it is still relevant. However it

Table 4: Comparison to previous state of the art for text based image retrieval: precision at n (P@n) for the Sports-10K dataset.

| Method                       | Sports-10K (P@10) | Sports-10K (P@20) |
|------------------------------|-------------------|-------------------|
| Mishra <i>et al.</i> [13]    | 44.82             | 43.42             |
| Mishra [69]                  | 47.20             | 46.25             |
| Jaderberg <i>et al.</i> [14] | 91.00             | 92.50             |
| YOLO-PHOC 604-d              | 92.00             | 90.00             |
| <b>YOLO-PHOC 820-d</b>       | <b>93.50</b>      | <b>93.00</b>      |

is important to highlight that our model does not employ a multi-resolution strategy at inference time to predict PHOCs at different scales as in our previous work [20], and by using just a fixed input size of  $608 \times 608$  is able to surpass previously obtained results. We should mention that our method achieves a slightly slower measurement in frames per second (fps) when compared to the model that uses a PHOC of 604 dimensions at the  $608 \times 608$  input resolution, due to the usage of a bigger PHOC descriptor. Nevertheless, the previous mentioned effect is almost negligible when compared to single resolution models, specially if we consider the significant boost in mAP. The improvement comes into place due to a larger preservation of spatial information of words by employing a larger descriptor with redundant information, resulting in greater performance achieved on all three evaluated datasets. Lastly, we compare the performance to additional state of the art text recognizers, such as [43, 45, 46, 44]. We also include results from a variant of He *et al.* [44], but rather than direct string matching, the model’s results are first transformed to PHOC descriptors and the look-up is based on similarity in the PHOC space. It can be seen that the PHOC space does not offer any advantage compared to end-to-end recognition methods.

Table 4 further compares the improved method to previous state of the art by showcasing the retrieval precision at 10 (P@10) and 20 (P@20) cut-off points on the Sports-10K dataset. Due to the significant improvement on the evaluated dataset, our YOLOv2-820-d model is able to achieve a better precision and obtaining state of the art performance when compared to other methods.

In Figure 5 we provide qualitative results by showing the top-5 retrieved images for the query “adidas” on the Sports10k dataset. We achieve a 100% precision despite the fact that the images contain blurred, partially occluded and rotated text instances.

To offer additional insights, Figure 6 depicts the heat-maps of our model by



Figure 5: Top-5 retrieved images for the query “adidas”. Our model successfully retrieves partially occluded and blurred words. Best viewed in color.



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

534 calculating the closest matching PHOCs and the corresponding bounding boxes in  
 535 relation to a given query. As it can be seen on the showcased figure, several predicted  
 536 PHOCs closely match the queried word. By revisiting the implementation details  
 537 defined in the previous section, we confirm the fact that avoiding the usage of an  
 538 NMS post processing strategy is indeed a safe way to preserve high matching PHOC  
 539 proposals that could be discarded otherwise.

#### 540 4.3. Multi-Lingual Scene Text Retrieval

541 As an extension to our previous work [20], we focus on analyzing the generaliza-  
 542 tion capability of the proposed models. It becomes essential to note that designing  
 543 an algorithm that learns to construct a compact representation of a string, such as  
 544 the PHOC, paves the road to further development of models that are not constrained  
 545 to a fixed dictionary or training data samples. In order to assess the expressiveness  
 546 of our architecture, we make use of two Multi-lingual datasets 2017 [57] and 2019 [58]  
 547 in which we can easily find out-of-vocabulary words (text not seen at training time)  
 548 with different distributions and characteristics. These datasets are used by the re-  
 549 search community to perform text detection and recognition tasks, but not text based  
 550 image retrieval. Therefore, we have selected a set of 100 queries for in-vocabulary  
 551 experiments and another set of 100 queries for out-of-vocabulary experiments for

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). The metric employed is the mean average precision (mAP).

| Method                 | MLT 2017     |              | MLT 2019     |              |
|------------------------|--------------|--------------|--------------|--------------|
|                        | IV           | OOV          | IV           | OOV          |
| He et al. [44]         | 24.79        | 19.47        | 27.6         | 24.99        |
| YOLO-PHOC 604-d        | 46.52        | 46.87        | 46.41        | 46.03        |
| <b>YOLO-PHOC 820-d</b> | <b>47.61</b> | <b>47.75</b> | <b>47.55</b> | <b>47.19</b> |

Table 6: 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)

| Method                 | MLT 2017    |             |             |             |             |             | MLT 2019    |             |             |             |             |             |
|------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|                        | IV          |             |             | OOV         |             |             | IV          |             |             | OOV         |             |             |
|                        | 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.[44]          | 0.51        | 0.37        | 0.22        | 0.46        | 0.33        | 0.20        | 0.62        | 0.44        | 0.27        | 0.60        | 0.40        | 0.23        |
| YOLO-PHOC 604-d        | 0.77        | 0.57        | 0.34        | 0.78        | 0.59        | 0.34        | 0.80        | 0.64        | 0.41        | 0.80        | 0.64        | 0.40        |
| <b>YOLO-PHOC 820-d</b> | <b>0.79</b> | <b>0.59</b> | <b>0.35</b> | <b>0.80</b> | <b>0.61</b> | <b>0.36</b> | <b>0.82</b> | <b>0.66</b> | <b>0.43</b> | <b>0.83</b> | <b>0.65</b> | <b>0.41</b> |

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 as the ones obtained in the out-of-vocabulary queries. Therefore, we end up with queries that have comparable occurrences in both datasets with an average of 13.32 and 13.99 for IV and OOV queries respectively.

Tables 5 and 6 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 [44] at the text retrieval task. Additionally, the method from [44] was trained using the dictionary from [52] which contains English words, thus performing poorly when dealing with out of vocabulary words mostly belonging to different languages. We can also observe that a slight increase in performance of the proposed model is obtained when dealing with OOV queries rather than IV queries in the MLT 2017 dataset. Due to the similar occurrences of positive samples of queries both for IV and OOV words, this difference is negligible and rather proves the



Figure 7: From top to bottom, top-5 ranked images for the queries “vodafone”, “uscita”, “werden” and “parkausweis”. Although our model has not seen these words at training time it is able to achieve a 100% P@5 for all of them. Best viewed in color.

generalization and knowledge transfer capability of our model to construct PHOCs of unseen samples at training time.

Figure 7 shows the top-5 ranked images for the queries “vodafone” in IIIT-STR dataset, “uscita” (italian) and “parkausweis” (german) 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.

#### 4.4. Real-time Text Spotting in Videos

Given the high processing frame-rates that we achieve (c.f. Table 3), we can use the proposed method for spotting text in video streams in real time.

Table 7: Top 15 most frequent words with their number of occurrences and the reached F-score.

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

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 [15], 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 top 20 most occurring words in the dataset that have more than three characters in length. 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 penalize both missing frames where the query word

596 appears and false positive frames. Overall we achieved an F-score of 76.70, and we  
597 provide some results for the topmost 15 queries in Table 7. Video demos are available  
598 in our public repository<sup>1</sup>.

## 599 5. Conclusions

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

603 A comparison of different backbones was presented and results were analyzed.  
604 The proposed model achieves state of the art performance while using fewer param-  
605 eters, less computational resources and converging in a considerable shorter training  
606 time. The effect of different PHOC sizes was undertaken as well. It was shown  
607 that a bigger PHOC, with redundant unigram and bigram levels, tends to better  
608 preserve and encode spatial information of a word yielding superior retrieval results  
609 at a minor cost of performance at inference time. The newly proposed model sig-  
610 nificantly improves previous state of the art results on the scene text retrieval task  
611 on the IIIT-STR and Sports-10K dataset while obtaining comparable results in the  
612 SVT Dataset. Moreover, it can do so by achieving speeds  $50\times$  to  $150\times$  faster when  
613 compared to other methods while using a single input resolution, which opens up  
614 the possibility of employing this model for real time scenarios, such as video, and  
615 indexing large scale databases.

616 Importantly, it has been shown that the proposed method is able to construct a  
617 compact vectorial representation of out of dictionary queries at inference time, while  
618 keeping a similar performance as of words seen at training time. Achieving this result  
619 is only possible by employing the PHOC as a word representation instead of tackling  
620 the task as a direct word classification. The method showcased is able to generalize  
621 unseen samples in a powerful and efficient way, as the evidence strongly points out  
622 in experiments performed in two multilingual datasets.

623 Future work can be conducted to investigate the use of word embeddings that  
624 exploit the morphology of a word other than PHOC or diverse tasks in which mor-  
625 phology can be leveraged.

626 The code, pre-trained models, data and demo videos used in this work are publicly  
627 available at <https://github.com/AndresPMD/Pytorch-yolo-phoc>.

---

<sup>1</sup><https://github.com/AndresPMD/Pytorch-yolo-phoc>

## 628 Acknowledgments

629 This work was partially funded by the Spanish Research project TIN2017-89779-  
630 P, the grant given by the European Social Fund 2014-2020 (CCI: 2014ES05SFOP007)  
631 and Predoctoral grant (AGAUR) number 2019-FIB01233, the H2020 MarieSkodowska-  
632 Curie actions of the European Union, grant agreement No 712949(TECNIOspring  
633 PLUS), the CERCA Programme / Generalitat de Catalunya, and UAB PhD schol-  
634 arship No B18P0070.

## 635 References

- 636 [1] A. Veit, T. Matera, L. Neumann, J. Matas, S. Belongie, COCO-text: Dataset and benchmark  
637 for text detection and recognition in natural images, arXiv preprint arXiv:1601.07140.
- 638 [2] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- 639 [3] I. Sutskever, G. E. Hinton, A. Krizhevsky, Imagenet classification with deep convolutional  
640 neural networks, *Advances in neural information processing systems* (2012) 1097–1105.
- 641 [4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy,  
642 A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, *International*  
643 *journal of computer vision* 115 (3) (2015) 211–252.
- 644 [5] Y. Zhu, C. Yao, X. Bai, Scene text detection and recognition: Recent advances and future  
645 trends, *Frontiers of Computer Science* 10 (1) (2016) 19–36.
- 646 [6] S. Long, X. He, C. Ya, Scene text detection and recognition: The deep learning era, arXiv  
647 preprint arXiv:1811.04256.
- 648 [7] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet, S. Arnoud, L. Yatziv, Ontological  
649 supervision for fine grained classification of street view storefronts, in: *Proc. of the IEEE*  
650 *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1693–1702.
- 651 [8] S. Karaoglu, R. Tao, T. Gevers, A. W. Smeulders, Words matter: Scene text for image classi-  
652 fication and retrieval, *IEEE Transactions on Multimedia* 19 (5) (2017) 1063–1076.
- 653 [9] X. Bai, M. Yang, P. Lyu, Y. Xu, Integrating scene text and visual appearance for fine-grained  
654 image classification with convolutional neural networks, arXiv preprint arXiv:1704.04613.
- 655 [10] A. Mafla, S. Dey, A. F. Biten, L. Gomez, D. Karatzas, Fine-grained image classification and  
656 retrieval by combining visual and locally pooled textual features, in: *The IEEE Winter Con-*  
657 *ference on Applications of Computer Vision*, 2020, pp. 2950–2959.
- 658 [11] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, M. Rohrbach,  
659 Towards vqa models that can read, in: *Proceedings of the IEEE Conference on Computer*  
660 *Vision and Pattern Recognition* (2019), 2019, pp. 8317–8326.

- [12] A. F. Biten, R. Tito, A. Mafra, L. Gomez, M. Rusinol, E. Valveny, C. Jawahar, D. Karatzas, Scene text visual question answering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.
- [13] A. Mishra, K. Alahari, C. Jawahar, Image retrieval using textual cues, in: Proc. of the IEEE International Conference on Computer Vision, 2013, pp. 3040–3047.
- [14] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, *International Journal of Computer Vision* 116 (1) (2016) 1–20.
- [15] 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, 2013, pp. 1484–1493.
- [16] 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, 2015, pp. 1156–1160.
- [17] J. Almazán, A. Gordo, A. Fornés, E. Valveny, Word spotting and recognition with embedded attributes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (12) (2014) 2552–2566.
- [18] 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, 2016, pp. 277–282.
- [19] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
- [20] L. Gómez, A. Mafra, M. Rusinol, D. Karatzas, Single shot scene text retrieval, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 700–715.
- [21] X. Chen, L. Jin, Y. Zhu, C. Luo, T. Wang, Text recognition in the wild: A survey, arXiv preprint arXiv:2005.03492.
- [22] D. He, X. Yang, C. Liang, Z. Zhou, A. G. Ororbia, 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, 2017, pp. 3519–3528.
- [23] D. Bazazian, R. Gómez, A. Nicolaou, L. Gómez, D. Karatzas, A. D. Bagdanov, Fast: Facilitated and accurate scene text proposals through fcn guided pruning, *Pattern Recognition Letters* 119 (2019) 112–120.
- [24] 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, 2017, pp. 4161–4167.

- 698 [25] M. Liao, B. Shi, X. Bai, Textboxes++: A single-shot oriented scene text detector, IEEE  
699 transactions on image processing 27 (8) (2018) 3676–3690.
- 700 [26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single shot  
701 multibox detector, in: Proc. of the European Conference on Computer Vision, Springer, 2016,  
702 pp. 21–37.
- 703 [27] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate  
704 scene text detector, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition  
705 (CVPR), IEEE, 2017, pp. 2642–2651.
- 706 [28] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, M. Park, Pvanet: deep but lightweight neural networks  
707 for real-time object detection, arXiv preprint arXiv:1608.08021.
- 708 [29] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer  
709 vision, 2015, pp. 1440–1448.
- 710 [30] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with  
711 region proposal networks, in: Proc. of the International Conference on Neural Information  
712 Processing Systems, 2015, pp. 91–99.
- 713 [31] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE interna-  
714 tional conference on computer vision, 2017, pp. 2961–2969.
- 715 [32] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text  
716 detection via rotation proposals, IEEE Transactions on Multimedia 20 (11) (2018) 3111–3122.
- 717 [33] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, Z. Luo, R2cnn: Rotational region  
718 cnn for orientation robust scene text detection, arXiv preprint arXiv:1706.09579.
- 719 [34] S. Zhang, Y. Liu, L. Jin, C. Luo, Feature enhancement network: A refined scene text detector,  
720 in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- 721 [35] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Synthetic data and artificial neural  
722 networks for natural scene text recognition, arXiv preprint arXiv:1406.2227.
- 723 [36] A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification:  
724 labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the  
725 23rd international conference on Machine learning, ACM, 2006, pp. 369–376.
- 726 [37] B. Shi, X. Bai, C. Yao, An end-to-end trainable neural network for image-based sequence recog-  
727 nition and its application to scene text recognition, IEEE Transactions on Pattern Analysis  
728 and Machine Intelligence 39 (11) (2017) 2298–2304.
- 729 [38] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and  
730 translate, arXiv preprint arXiv:1409.0473.

- 731 [39] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, S. Zhou, Focusing attention: Towards accurate  
732 text recognition in natural images, in: Proceedings of the IEEE International Conference on  
733 Computer Vision, 2017, pp. 5076–5084.
- 734 [40] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, in: Advances  
735 in neural information processing systems, 2015, pp. 2017–2025.
- 736 [41] B. Shi, X. Wang, P. Lyu, C. Yao, X. Bai, Robust scene text recognition with automatic  
737 rectification, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,  
738 2016, pp. 4168–4176.
- 739 [42] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images,  
740 in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp.  
741 2315–2324.
- 742 [43] M. Busta, L. Neumann, J. Matas, Deep textspotter: An end-to-end trainable scene text local-  
743 ization and recognition framework, in: Proceedings of the IEEE International Conference on  
744 Computer Vision, 2017, pp. 2204–2212.
- 745 [44] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, C. Sun, An end-to-end textspotter with explicit  
746 alignment and attention, in: Proceedings of the IEEE Conference on Computer Vision and  
747 Pattern Recognition, 2018, pp. 5020–5029.
- 748 [45] M. Bušta, Y. Patel, J. Matas, E2e-mlt-an unconstrained end-to-end method for multi-language  
749 scene text, in: Asian Conference on Computer Vision, Springer, 2018, pp. 127–143.
- 750 [46] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, J. Yan, Fots: Fast oriented text spotting with  
751 a unified network, in: Proceedings of the IEEE conference on computer vision and pattern  
752 recognition, 2018, pp. 5676–5685.
- 753 [47] P. Lyu, M. Liao, C. Yao, W. Wu, X. Bai, Mask textspotter: An end-to-end trainable neural  
754 network for spotting text with arbitrary shapes, in: Proceedings of the European Conference  
755 on Computer Vision (ECCV), 2018, pp. 67–83.
- 756 [48] S. K. Ghosh, E. Valveny, Query by string word spotting based on character bi-gram indexing,  
757 in: Proc. of the IEEE International Conference on Document Analysis and Recognition, 2015,  
758 pp. 881–885.
- 759 [49] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recogni-  
760 tion, arXiv preprint arXiv:1409.1556.
- 761 [50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke,  
762 A. Rabinovich, Going deeper with convolutions, in: Proc. of the IEEE Conference on Computer  
763 Vision and Pattern Recognition, 2015, pp. 1–9.
- 764 [51] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. of  
765 the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- 766 [52] K. Lang, T. Mitchell, Newsgroup 20 dataset.

- [53] F. Zhan, S. Lu, C. Xue, Verisimilar image synthesis for accurate detection and recognition of texts in scenes, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 249–266.
- [54] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, arXiv preprint arXiv:1908.03265.
- [55] M. Zhang, J. Lucas, J. Ba, G. E. Hinton, Lookahead optimizer: k steps forward, 1 step back, in: Advances in Neural Information Processing Systems, 2019, pp. 9593–9604.
- [56] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: Proc. of the IEEE International Conference on Computer Vision, 2011, pp. 1457–1464.
- [57] I. Bizid, J. Chazalon, H. Choi, Y. Feng, D. Karatzas, W. Khelif, 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://rrc.cvc.uab.es/?ch=8>, [Online, accessed 22-April-2019] (2017).
- [58] M. Buřta, D. Karatzas, W. Khelif, J. Matas, N. Nayef, U. Pal, Y. Patel, ICDAR 2019 Robust Reading Challenge on Multi-lingual scene text detection and recognition, <http://rrc.cvc.uab.es/?ch=11>, [Online, accessed 30-April-2019] (2019).
- [59] M. Iwamura, L. Gomez, D. Karatzas, Robust Reading Challenge on Text in videos 2013-2015, <http://rrc.cvc.uab.es/?ch=11>, [Online, accessed 22-April-2019] (2019).
- [60] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
- [61] 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(2016), 2016, pp. 779–788.
- [62] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767.
- [63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
- [64] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
- [65] 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, 2010, pp. 2963–2970.

- [66] A. Mishra, K. Alahari, C. V. Jawahar, Top-down and bottom-up cues for scene text recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2687–2694.
- [67] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3538–3545.
- [68] 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, 2015, pp. 1236–1240.
- [69] A. Mishra, Understanding text in scene images, Ph.D. thesis, International Institute of Information Technology Hyderabad (2016).
- [70] L. Gómez, D. Karatzas, Textproposals: a text-specific selective search algorithm for word spotting in the wild, Pattern Recognition 70 (2017) 60–74.
- [71] 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, 2017, pp. 2204–2212.
- 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.
- Sounak Dey** received his M.S. degree in Computer Vision in 2016 and is currently a Ph.D. candidate in Computer Vision Center at the Universitat Autònoma de Barcelona. His research interests are in the areas of sketch based image retrieval, domain generalization, historical document analysis and neural networks.
- 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.

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

845

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