Words Matter: Scene Text for Image Classification and Retrieval

Sezer Karaoglu†, Ran Tao†, Theo Gevers and Arnold W. M. Smeulders

***Abstract*—Text in natural images typically adds meaning to an object or scene. In particular, text specifies which business places serve drinks (e.g. cafe, teahouse) or food (e.g. restaurant, pizzeria), and what kind of service is provided (e.g. massage, repair). The mere presence of text, its words and meaning are closely related to the semantics of the object or scene. This paper exploits textual contents in images for fine-grained business place classification and logo retrieval. There are four main contributions. First, we show that the textual cues extracted by the proposed method are effective for the two tasks. Combining the proposed textual and visual cues outperforms visual only classification and retrieval by a large margin. Second, to extract the textual cues, a generic and fully unsupervised word box proposal method is introduced. The method reaches state-of-the-art word detection recall with a limited number of proposals. Third, contrary to what is widely acknowledged in text detection literature, we demonstrate that high recall in word detection is more important than high f-score at least for both tasks considered in this work. Last, this paper provides a large annotated text detection dataset with 10K images and 27601 word boxes.**

I. Introduction

Fine-grained classification is the problem of assigning images to classes where instances from different classes differ slightly in the appearances e.g., flower types [39], bird [57] and dog species [26], and models of a product [29]. In contrast to coarse object category recognition e.g., cars, cats and airplanes, low-level visual cues are often not sufficient to make distinction between fine-grained classes. Even for human observers, fine-grained classification tasks usually require expert and domain specific knowledge. Accordingly, most recent works also integrated such domain specific knowledge into their solutions. For instance, dogs have ears, nose, body, legs etc., and the differentiation of dog species relies on the subtle differences in these parts. Different bird species have different wing and beak appearances, and such differences in local parts provide the critical information to categorize different bird types. [63], [59], [26] exploit the part information and extract features from particular parts for better birds and dogs recognition. In this paper, we focus on classification of different business places, e.g., bakery, cafe and bookstore. Various business places have subtle differences in visual appearances. For instance, on the shop windows, a pizzeria often has pizza images whereas a bakery usually shows images of bread, see Figure 1.a. In this particular problem, we make use of the domain specific knowledge of *business places*. We exploit the recognized text in images for fine-grained classification of business places. Automatic recognition and indexing of business places will be useful in many practical scenarios. For instance, it can be used to extract information from Google street view images and Google Map can use the information to provide recommendations of bakeries, restaurants close to the location of the user.

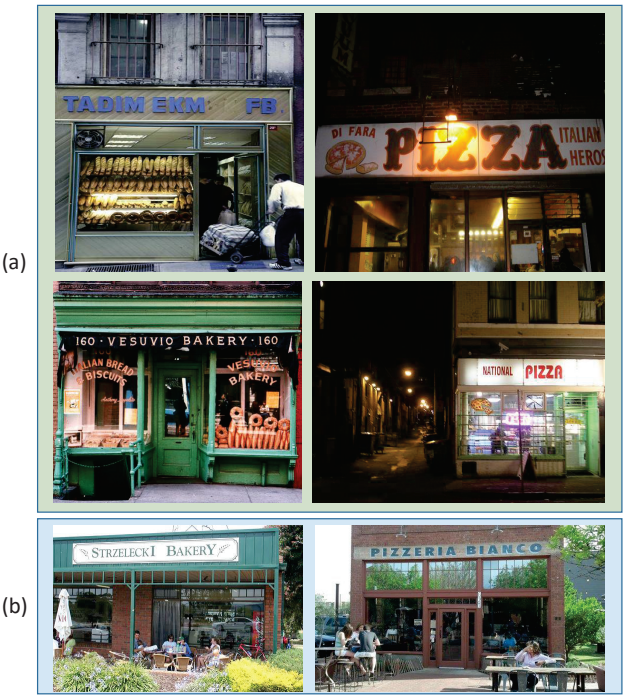


Fig. 1: Example images of *bakery* (left column) and *pizzeria* (right column). Different business places often have subtle visual differences. As an example, in (a), the main difference between *bakery* images and *pizzeria* images is that *bakery* shop windows have images of bread whereas *pizzeria* shop windows have pizza images. The information carried by scene text helps to distinguish these two types of business places. (b) shows an extremely challenging case where the two shops can hardly be distinguished unless the scene text is used.

S. Karaoglu is with the Computer Vision Lab, Amsterdam, University of Amsterdam, 1098 XH Amsterdam, The Netherlands (e-mail: s.karaoglu@uva.nl).

R. Tao is with the Intelligent Sensory Information Systems Lab, Amsterdam, University of Amsterdam, 1098 XH Amsterdam, The Netherlands (e-mail: r.tao@uva.nl).

T. Gevers is with Computer Vision Lab, Amsterdam, University of Amsterdam, 1098 XH Amsterdam, The Netherlands, and also with the Computer Vision Center, Universitat Autònoma de Barcelona, 08193 Barcelona, Spain (e-mail: th.gevers@uva.nl).

A. W. M. Smeulders is with the Intelligent Sensory Information Systems Lab, Amsterdam, University of Amsterdam, 1098 XH Amsterdam, The Netherlands (e-mail: a.w.m.smeulders@uva.nl).

† indicates equal contribution.

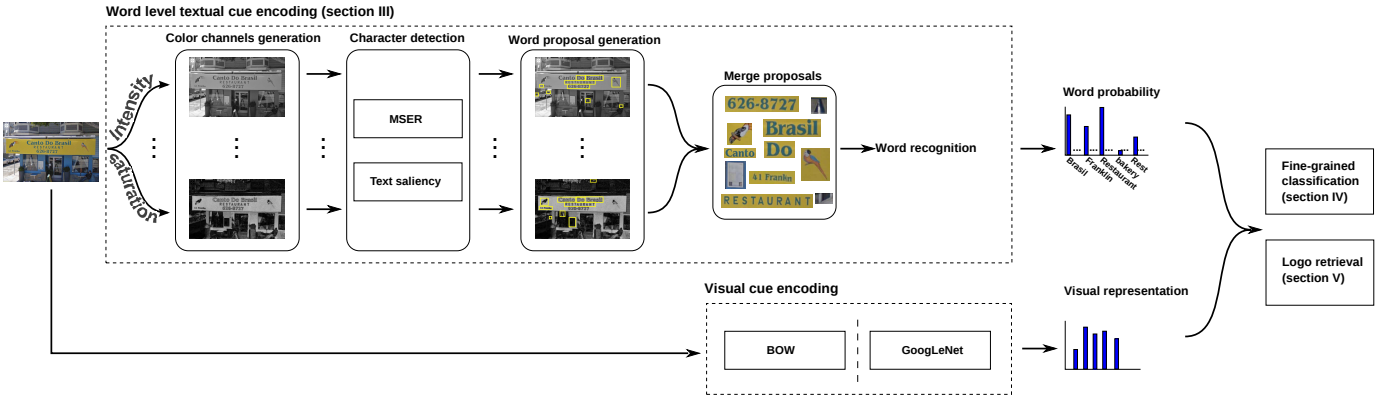


Fig. 2: Pipeline of our multimodal approach. Text is encoded at a word level and utilized for fine-grained classification and logo retrieval. A generic and fully unsupervised word box proposal method is proposed to detect words in images. The method uses different color spaces and character detection algorithms (MSER [31] and text saliency [23]). The word box candidates are used as input for a state-of-the-art word recognition method [18] to perform word-level encoding. An English vocabulary consisting of around $90k$ words is considered [18]. For the visual cues, bag-of-words (BOW) and GoogLeNet features [46] are used. The multimodal approach combines the visual and textual cues.

Most of the time, the stores use text to indicate what type of food (pizzeria, diner), drink (tea, coffee) and service (drycleaning, repair) that they provide. This text information is helpful even for human observers to understand what type of business place it is. For instance, in Figure 1.b, the images of two different business places (*pizzeria* and *bakery*) have a very similar appearance. However, they are different types of business places. It is only possible with text information to identify what type of business places these are. Moreover, text is also useful to identify similar products (logo) such as *Heineken*, *Foster* and *Carlsberg*. Therefore, we propose a multimodal approach which uses recognized text and visual cues for fine-grained classification and logo retrieval.

The common approach to text recognition in images is to detect text first before they can be recognized [56], [17]. The state-of-the-art word detection methods [38], [55], [25], [58], [27] focus on obtaining a high f-score by balancing precision and recall. However, instead of using the f-score, our aim is to obtain a high recall. A high recall is required because textual cues that are not detected will not be considered in the next (recognition) phase of the framework. Unfortunately, there exists no single best method for detecting words with high recall due to large variations in text style, size and orientation. Therefore, we propose to combine character candidates generated by different state-of-the-art detection methods. To obtain robustness against varying imaging conditions, we use color spaces containing photometric invariant properties such as robustness against shadows, highlights and specular reflections.

The proposed method computes text lines and generates word box proposals based on the character candidates. Then, word box proposals are used as input of a state-of-the-art word recognition method [18] to yield textual cues. Finally, textual cues are combined with visual cues for fine-grained classification and logo retrieval. The proposed framework is given in Figure 2.

The paper has the following contributions. First, this paper shows that textual cues are complementary to visual cues for fine-grained classification and logo retrieval. Combining the proposed textual and visual cues outperforms visual only classification and retrieval by a large margin. The proposed method reaches state-of-the-art results on both tasks. Second, to extract the word-level textual cues, a generic, efficient and fully unsupervised word proposal method is introduced. The proposed method reaches state-of-the-art word detection recall with a limited number of proposals. Third, contrary to what is widely acknowledged in text detection literature, we experimentally show high recall in word detection is more important than high f-score at least for both applications considered in this work. Last, this paper provides a large annotated text detection dataset with 10K images with 27601 word boxes. This dataset is available at the project page [1].

II. Related Work

Word Detection. Word detection consists of computing bounding boxes of words in images. Existing word detection methods usually follow a bottom-up approach. Character candidates are computed by a connected component [9], [38], [62], [16], [60] or a sliding window approach [55], [17], [56]. Candidate character regions are further verified and combined to form word candidates. This is done by using geometric, structural and appearance properties of text and is based on hand-crafted rules [9] or learning schemes [55], [17]. State-of-the-art word detection methods [16], [25], [38], [55], [65] focus on high f-score by the trade-off between recall and precision. Strict rules are used in character detection and word formation to keep only boxes that most likely contain words. As a consequence, methods aiming for high f-score may miss a number of correct word boxes. In contrast, we propose to generate word boxes with the goal to include all words i.e. high recall. We use recall in text detection because our aim is

[1] https://staff.fnwi.uva.nl/s.karaoglu/datasetWeb/Dataset.html)

not to miss correct word boxes with the cost of introducing false detections.

Our work is similar to the recent works [19], [15] in terms of providing word box proposals. [19] combines two generic object proposal outputs, namely Edge Boxes [67] and Aggregate Channel Feature Detector [8], as preliminary word box proposals. Then, these proposals are filtered using the HOG [7] feature with a Random Forest text/non-text classifier [3]. Finally, the remaining word box proposals are processed using a convolutional neural network regressor to refine the coordinates of these word boxes. [15] performs an over-segmentation using maximally stable extremal region (MSER) algorithm with flexible parameters. Then, the segments are grouped together using distance metrics related to text (e.g. color, stroke width etc.). Finally, weak classifiers are used to obtain a text-likeliness measure for these word candidates. In contrast, our word box generator is uniquely designed to detect text in images without any training involved. Moreover, [19], [15] in the end aim at high f-score word recognition whereas this work aims only at high recall. We experimentally verify high recall is more important than high f-score for the applications considered in this work. Further, different from [19], [15] which address word recognition, the aim of this paper is to combine textual and visual cues for better fine-grained classification and logo retrieval.

Text Recognition. Text recognition approaches can be categorized into two groups: character and word based methods. Character based methods first recognize single characters, then form words [32], [33], [40]. Recent work [1], [14], [18] shows that entire-word recognition performs better than recognizing characters first and then forming words. In this work, we follow the state-of-the-art word recognition approach [18] to encode the textual cues.

Textual Cues. Mishra et al. [34] propose to use textual cues for query-by-text image retrieval. Given a query text, the method assigns scores to images based on the presence of the query characters. Additional pairwise spatial constraints between characters are used to refine the ranking. Karaoglu et al. [24] propose to use textual cues in combination with visual cues for fine-grained classification. Bi-grams are computed based on recognized characters in images. These bi-grams are used to encode the textual cues. In contrast, this work performs a word-level textual cue encoding. Moreover, the proposed method aims at high recall word detection which leads to combine state-of-the-art text detectors performed in various color spaces.

Multimodal Classification. Video captions (textual) are extensively used in combination with visual cues for video classification. An overview of these methods can be found in [28]. Textual cues are also used for document image analysis [44]. In contrast, we propose to use textual information to classify natural scene images. Extracting textual information from natural scene images is more challenging than from controlled environment such as captions overlaid on video frames and document images. To classify images, Wang et al. [54] propose to combine visual cues with corresponding user tags. In [66], the authors exploit textual information by extracting visual features around the text regions and combining them with global visual features. Different from these works, in this paper, textual information is obtained from detected and recognized text in natural scene images.

Fine-grained Classification. Many recent works in fine-grained classification exploit domain specific knowledge. Dogs and birds are composed of a number of semantic parts, such as head, body and tail. [63], [59], [64] use parts for better fine-grained recognition. [63] learns part detectors and localizes the parts to isolate the subtle differences in specific parts. [59] shows the hidden layers of a deep neural network are actually part detectors and uses the filters in the hidden layers to detect specific bird and dog parts. [64] generates multi-scale part proposals and selects useful parts. [13] presents another successful use of domain specific knowledge for bird species recognition. It exploits the fact that birds have rather fixed poses and fits an ellipse to represent the overall shape of a bird. In this work, we exploit the domain specific knowledge for business place classification. In our case, the domain knowledge is the scene text in the business place images. We propose a multimodal approach to fine-grained business place classification by fusing the textual and visual cues. A recent paper from *Google* [35] also studies the classification of different business places. [35] only considers visual cues for classification while in our paper we show that adding textual cues significantly outperforms methods that only use visual information.

III. Word-level Textual Cue Encoding

In order to extract the textual cues from the image, a two-step procedure is followed. In the first step, word box proposals are generated to locate the words in the image. In the second step, the word proposals are used as input to a word recognizer to form the word-level representation.

A. Word Box Proposals

High recall. When a word in an image is not detected or localized incorrectly, it is not possible to identify it. Our aim is to obtain high recall with the cost of false positives. To this end, the proposed method uses a complementary set of character detection algorithms and color invariant spaces.

Low computational cost. The word box proposal method needs to be efficient especially for large scale scenarios. Further, the number of possible word box candidates (i.e. proposals) should be as low as possible.

Generic. We aim for a generic word proposal method. No need for tuning the method for different alphabets or datasets.

Therefore, we propose an efficient and fully unsupervised bottom-up approach. First, characters are detected by a text-independent approach. Then, these detected characters are filtered based on geometric and appearance properties. Finally, they are grouped to generate word box proposals.

1) Character Detection: As stated earlier, there exists no single character detection algorithm that is robust against all variations in text style, size and orientation and imaging conditions. Therefore, we propose to compute character candidates using two methods with different strengths,

Fig. 3: Saliency map samples: Original images, color saliency and curvature saliency (top to down order). It is shown that text edges are detected better with color saliency for the first two images whereas curvature saliency works better for the last two images.

i.e., text saliency [23] and Maximally Stable Extremal Regions(MSERs) [31].

In [23], a text saliency map is computed using scene background. It is assumed that background pixels are uniformly colored e.g., windows, boards, roads, buildings, fences etc., and that they contrast with text regions. Accordingly, the method uses background homogeneity to form connectivity between background pixels. The method selects initial background seeds and grows these seeds iteratively until all background pixels are covered (detected). Assuming that text regions have strong contrast with the background [2], text regions will remain uncovered by the region growing algorithm. Finally, the background image is subtracted from the original image to obtain a text saliency map, which is further binarized using [12] to obtain character candidates.

Detecting background relies heavily on correctly selecting initial seeds. As text is salient [22], [5], the background that highly contrasts with text is assumed to be non-salient. Moreover, text rarely appears at image borders. Therefore, seeds are selected from non-salient image regions which are refined by image boundary information.

Color and Curvature Saliency. Color edges are useful to detect if a region belongs to the background. Color is usually homogeneous for different backgrounds such as roads, fences and skies. Furthermore, color edge responses correlate with colorful text/background transitions. To exploit this, we use the color boosting algorithm [52] to enhance the saliency of colorful text/background transitions and to select the background seeds based on non-salient regions in the color saliency map.

The color saliency measure is inappropriate for colorless edge transitions, see the right two images in Figure 3. Therefore, in addition to color, curvature (L) is used for colorless edges. Because of the text/background contrast, text regions result in high curvature even for colorless edge transitions. Non-salient regions in the curvature saliency map are used to select the background seeds.

Saliency Refinement Using Background Priors. The image regions not having strong responses in color and curvature are considered to be background pixels. However, some of the regions which have high saliency response may also belong to the background. Therefore, salient regions which are unlikely to belong to text are suppressed using background priors, *i.e.*, text is mostly located in the center of the image [23]. Hence, salient regions, which are connected to the borders, are suppressed in the color and curvature saliency maps.

The refined saliency maps are used to select the background seeds. Specifically, the refined saliency maps are normalized to $[0, 1]$ and linearly combined. Regions without any response on this combined saliency map are considered as background seeds. The background of the input image is reconstructed using morphological operators [53]. The reconstruction is performed by a conditional dilation (δ). Conditional dilation is a basic dilation which is conditioned by a mask image (i.e., the single-channel image I in our case). The conditioning is obtained by defining the output as the intersection of the dilation and I, formulated by:

$$\delta_I(J) = (J \oplus B) \wedge I \ , \tag{1}$$

where $J \oplus B$ stands for the dilation of J (the image consisting of only background seeds) and B (the structuring element), and $\wedge$ denotes the element-wise minimum.

To obtain a reconstructed background image (ρ) of image I, given the image consisting of the initial background seeds, J_0, Eqn. 1 is executed until stability is reached. That is, starting from the initial background seeds J_0 repeat $J_n = \delta_I(J_{n-1})$ until $J_n = J_{n-1}, (n = 1, 2, 3)$ and obtain ρ by $\rho = J_n$.

Text saliency computation does not require any tuning for varying text size, style and orientation, and is robust to image noise. However, due to the information loss caused by the image boundary priors and the binarization, the method may miss characters. To compensate for this, we enable MSER as another character detection algorithm. MSERs define an extremal region as a connected component of which image values remain stable within the boundary and highly contrast against boundary pixels [31]. MSER regions are widely in use for character detection [6], [38]. MSER is suited for character detection because text regions are usually designed to have uniform appearance (color). Further, they usually have high contrast with their surroundings. However, MSER has certain shortcomings for character detection such as detecting characters in blurry and noisy images [6]. Moreover, MSER is sensitive to character sizes due to the parameters used to define stable regions. In fact, the MSER and text saliency results are, to a certain extent, complementary. Figure 4 illustrates complementary properties of MSER and saliency methods.

2) Complementary Color Spaces: Images are captured under uncontrolled illumination conditions. Therefore, text regions may be influenced by different photometric changes such as shadows and specular reflections. A uniformly colored character may vary in intensity due to shadows or highlights. Hence, these shadows or highlights may negatively influence the pixel connectivity for a uniformly colored character.

To compensate for this, the proposed method computes the character candidates using a variety of color spaces containing a range of invariant properties. The two channels, (O_1, O_2),

Fig. 4: Original images (up), text saliency (middle) and MSER character detections (down) obtained. Text saliency method is robust against changes in text size and noise while MSER detects characters at image boundaries.

	I	O_1	O_2	S	H
Highlights	-	+	+	-	+
Shadows	-	-	-	+	+

TABLE I: Color spaces and their invariant properties. I is the gray scale. (O_1, O_2) are the two channels from the opponent color space [10]. Saturation (S) and Hue (H) are from the *HSV* color space [51]. '+' means invariant. In this paper, we use all these color spaces with different invariant properties to cope with the photometric changes in natural images.

from the opponent color space [10], Saturation (S) and Hue (H) from *HSV* [51], and (I) from gray scale are considered in the proposed method. The invariant properties are summarized in Table I.

3) Character Filtering: The character candidates provided in Section III-A1 may consist of non-character regions. Our method for character filtering is based on state-of-the-art text detection systems [6], [9] to filter out non-character regions efficiently.

Aspect ratio. Most of the real characters have a width-height ratio close to 1 [9]. Therefore, the proposed method limits the aspect ratio of character candidates to be a value between 0.1 and 10. These values are reported in [9] to be conservative enough to still keep characters such as 'i', 'I' or '1'. This process filters out text-like items in images such as fences and branches of trees.

Size. The proposed method limits the height of a character candidate to be greater than 5 pixels and the area to contain more than 50 pixels [9]. If the character is too small, the information it carries is limited. Therefore, it is likely that even if these regions are not eliminated, recognition on these regions would fail.

Solidity. The solidity is defined as the proportion of the number of character pixels to the convex area which covers the text candidate. It has been observed that text regions have low solidity [6]. Therefore, the proposed method eliminates character candidates which have high solidity (>0.95) and longer width than height. Longer width is to avoid removing characters like 'i' and 'l'. This process filters out brick-like image regions which have solidity close to 1. Solidity threshold is set to be conservative enough to keep characters like 'w' and 'm'.

Contrast. Pixels at character borders usually have high contrast and the contrast decreases with the distance to the borders. As a result, the box which neatly covers a character will have a higher average contrast than its slightly expanded version. Therefore, the proposed method eliminates the character candidates which do not meet this condition. Contrast (C) of an image pixel (p) is calculated by $C_p = \sqrt{I_x^2(p) + I_y^2(p) + I_x(p)I_y(p)}$, where I_x, I_y are the first order image derivatives (x and y dimensions) in intensity I.

A character candidate satisfying all these conditions is remained for further processing. This filtering step removes those obvious non-character candidates to reduce computational cost in following steps. Figure 5 shows filtered character candidates for each condition.

4) Word Box Proposal Generation: The next step is to compute word box proposals using character candidates. We consider combinations of character candidates as potential words. However, it is computationally expensive if all possible combinations are considered. And, due to the nature of text, characters within a word cannot have arbitrary positions and sizes [9], [11], [25], [30], [36], [37]. Therefore, as the first step of computing word box proposals, we generate text lines to restrict the selection of combinations by linking character candidates based on five pairwise constraints. In Figure 6, the two boxes stand for two character candidates with (x_1, y_1), $height_1$ and $width_1$ being the coordinates of the top-left corner, height and width of the box covering the first character. The box of the second character is defined likewise.

The five pairwise constraints are as follows:

(a) Distance between two character centers is smaller than 2.5 times of the longer axis of the character box [9], [11], [37]. $Distance < \max([height_1, width_1, height_2, width_2]) \times 2.5$ where $Distance = (x_2 + \frac{width_2}{2}) - (x_1 + \frac{width_1}{2})$. 2.5 is considered to allow one missed character in between.

(b) The ratio of the vertical displacement and horizontal offset is no greater than 0.2 [25], [37], formally expressed by $\frac{VD}{Distance} \leq 0.2$ where $VD = |(y_2 + \frac{height_2}{2}) - (y_1 + \frac{height_1}{2})|$ and $Distance$ as defined in (a). Text is mostly horizontally aligned.

(c) The height ratio of two characters is not greater than 2 [9], [11], [37], i.e., $0.5 \leq \frac{height_1}{height_2} \leq 2$. Two characters of a word should have similar height and 2 is considered to allow the case of a lower-case character following a capital.

(d) Two characters must not overlap more than 0.1, formally, $\frac{Area(Char_1 \cap Char_2)}{Area(Char_1 \cup Char_2)} \leq 0.1$. Characters of a word usually

Fig. 5: Samples for character candidate filtering. The green, red and yellow (colors are used to highlight boxes) boxes represent filtered character candidates after corresponding filtering condition is applied (i.e. size, aspect ratio, solidity and contrast).

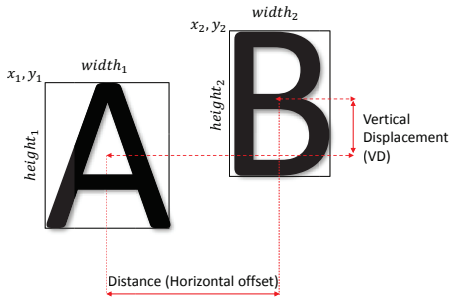


Fig. 6: An illustration on the notions of two character candidates. This illustration is used to elaborate the pairwise constraints.

do not overlap except in special cases, e.g., italic.

(e) The bottom of one character is below the center of the other [37], i.e., $(y_1 + height_1) \geq y_2 + \frac{height_2}{2}$ and $(y_2 + height_2) \geq y_1 + \frac{height_1}{2}$.Two consecutive characters of a word are usually well aligned for easy reading.

As the second step, we compute word box proposals by considering all possible combinations of character candidates within a text line. A combination of character candidates corresponds to the box covering the union of the character candidates. The proposed method starts with a single character candidate as a word proposal. The reason is that when the characters of a word are connected the word is covered by only one character candidate.

Word box proposals are generated from each character detection algorithm and color space independently and then combined.

B. Word Recognition and Textual Cue Encoding

Section III-A generates word box proposals. To recognize words, we employ a state-of-the-art word recognition approach [18]. [18] formulates word recognition as a multi-class classification problem, where a word from a predefined English vocabulary is treated as one class. A convolutional neural network classifier with four convolutional layers and two fully-connected layers is used to solve the classification problem. We refer to [18] for the details of the network. The network takes a word box proposal b as input and produces for each word w a probability of the word being present in the box, $P(w|b)$. The probability is modeled by the softmax scaling of the final multi-way classification layer. As a result, each word box proposal is represented by a n-dimensional feature, where n is the number of words in the vocabulary. In this work, we use the model[2] provided by the authors of [18]. The model considers a vocabulary of $88,172$ words and is trained using synthetic data. We encode the textual cues in an image by summarizing the representations of word box proposals with average pooling. Each dimension of the resulting image feature represents the probability of the corresponding word being present in the image.

IV. Fine-grained Classification

Fine-grained classification is the problem of the categorization of subordinate-level categories such as bird species [57], flower types [39] and business places [24]. The small inter-class visual differences and the large intra-class variations make fine-grained classification challenging. In this section, in addition to visual features, we exploit the textual cues in the images for fine-grained image classification.

A. Dataset and Implementation Details

Dataset. We use the *Con-Text dataset* proposed in [24]. The dataset is for fine-grained classification of business places, e.g., *Cafe*, *Bookstore* and *Pharmacy*. The dataset consists of $24,255$ images from 28 categories. The dataset is divided into three folds. Experiments are repeated three times, each time using two folds as training and the other as testing. We report the mean performance over the three runs. Average precision is used to measure the performance.

Dataset Annotation. To study the influence of precision and recall for word detection in the context of fine-grained classification, we have annotated text regions for the first 10 classes of the dataset (in alphabetical order). All the text (Latin alphabet) visible and recognizable has been annotated. The annotated dataset consists of 9131 images. 5219 of these

[2] http://www.robots.ox.ac.uk/~vgg/research/text

images contain at least one word box. In total there are 27601 word boxes annotated.

Visual baselines. Seven visual-only classification baselines are considered. All the visual baselines employ one-versus-rest SVM classifiers for classification, while the differences lie in the employed visual representations and kernel functions.

First, as in [24], we use bag of visual words representation with 3×1 and 2×2 spatial pyramid, and histogram intersection kernel. This baseline is denoted as *BOW*.

Second, as image representation, we use the $L2$ normalized output of the last average pooling layer of the ImageNet-pretrained GoogLeNet [46], given the full image as input. The network is trained on the 1000 ImageNet categories[3] [45], available in the Caffe library [20]. Linear kernel is employed. This baseline is denoted as *DEEP*.

Third, we fine-tune the pretrained GoogLeNet with a 28-way softmax classifier on the *Con-Text dataset*. After fine-tuning, the last average pooling layer output of network is used as the image representation. The details of the fine-tuning are as follows. The learning rate is initially set to be 0.001, and is decreased by a factor of 10 every 5 epochs. The network is fine-tuned for 20 epochs. The weight decay parameter equals 0.0005. The network is fine-tuned using SGD with momentum which is set to be 0.9. Again linear kernel is used. This visual baseline is denoted as *DEEP-FT*[4].

Fourth, we design a visual baseline which exploits local regional information. For each image, 100 local regions are extracted using EdgeBox [67], and represented using the pretrained GoogLeNet features. To include the local regional comparison, and inspired by [50], we adopt the kernel function expressed as $k = \frac{1}{n_1}\frac{1}{n_2}\sum_{i=1}^{100}\sum_{j=1}^{100} d_i \cdot d_j$. n_1 and n_2 are the normalization factors to ensure self-similarity to be 1. d_i, d_j are the $L2$ normalized regional features, and $d_i \cdot d_j$ is the cosine similarity. The kernel cross compares all the local regions of two images. This visual baseline is denoted as *DEEP-LOC*.

The fifth visual baseline is the same as the fourth baseline except that the fine-tuned GoogLeNet is used to extract the regional features. This baseline is denoted as *DEEP-FT-LOC*.

Sixth, to make use of both the global and local information, we derive a visual baseline by combining *DEEP* which relies on global representation and *DEEP-LOC* which focuses on local regional representation. This is denoted as *DEEP-GL*.

Likewise, we derive the last visual baseline by combining *DEEP-FT* and *DEEP-FT-LOC*, denoted by *DEEP-FT-GL*.

Text-based classification. For text-based classification, the textual cues are extracted as described in Section III. One-versus-rest SVM with linear kernel is used for classification.

Multi-modal fusion. To fuse the visual and textual cues, kernel fusion is employed. Specifically, two kernels are computed separately, one for visual and one for textual, and the two kernels are summed to derive the final kernel matrix. Kernel fusion is also used in visual baselines *DEEP-GL* and *DEEP-FT-GL* to combine global and local information.

[3]http://www.image-net.org/challenges/LSVRC/2012/

[4]In addition, we have conducted an experiment where the model is used to output directly the classification result. This end-to-end model achieves 61.1% in mAP.

In all experiments, Libsvm [4] is used and the default value for the C parameter (=1) is used without tuning.

B. The Influence of Word Detection Precision and Recall on Fine-grained Classification

We use the annotated 10 classes to analyze the effect of word detection precision and recall on fine-grained classification. We systematically change recall or precision and evaluate the classification performance. Within this section, only textual cues are used.

1) Performance on images without text: Not all images in the dataset contain text. However, the proposed method may generate candidate word proposals in non-textual regions in the image. The method uses the character detectors, MSER and text saliency. Consequently, regions of interest, other than text, may also be detected. We have evaluated the classification performance using the 'textual cues' extracted by the proposed method on images without text. Interestingly, it achieves 28.9% in mean average precision (mAP), significantly better than random guessing, although the textual cues are much more effective on images with text (67.7% in mAP). This indicates some salient non-text patterns within the same class could be consistently detected and similarly encoded. In the following analysis, we consider two cases, one with images containing text and the other considering all images.

2) The influence of word detection precision: To study the influence of word detection precision on fine-grained classification, we increase the precision while keeping the recall unchanged by removing the false positive detections (FP) from the generated word proposals. Figure 7 (left) shows that increasing the precision does not improve the classification performance.

Interestingly, this experiment has brought the following additional insight. The classification performance actually decreases when too many false positives are removed from the generated word proposals, especially when all images are considered ('All images'). There are two reasons for this. (1) The proposed word proposal method may detect salient but non-text regions. And some salient non-text patterns within the same class could be consistently detected and similarly encoded. Consequently, some false positive word proposals may contribute positively to the classification, especially for those images without text. This has been discussed in Section IV-B1. This is also the reason for the decrease in classification performance when removing too many false positives. This is more significant on 'all images' than 'images with text' as shown in Figure 7. (2) The boxes, that cover the text regions for less than 50% overlap with the ground-truth, are treated as false positives. These boxes may contain parts of words or complete words with extra background regions. Removing such boxes may have a negative influence on the classification results.

Additionally, we study the influence of precision decrease by adding the generated word proposals (*Ours*) on top of the manually annotated word boxes (*GT*). The classification performance of *GT+Ours* (with a precision of 6.2%) is 75.7% whereas *GT* (with a precision of 100%) is 76.1%. The signifi-

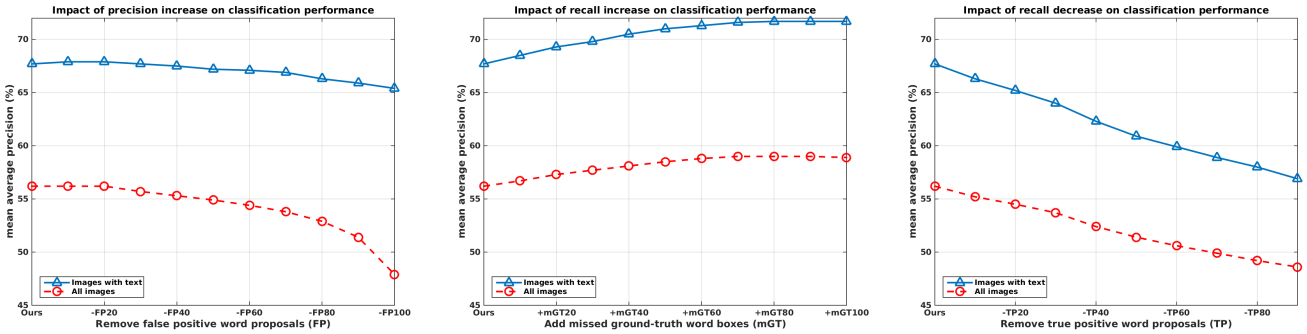


Fig. 7: The influence of the precision and recall change in word detection on fine-grained classification performance (evaluated on the 10 annotated classes). *Left*: Increasing precision by removing the false positive detections (FP) from the automatically generated set of proposals does not improve the classification performance. *'-FP20'* denotes removing 20% of the false positives. *Middle*: We systematically increase the recall by adding the missed ground-truth word boxes (mGT) on top of the automatically generated set of proposals. The classification performance keeps increasing as the word detection recall increases before it saturates. *'+mGT20'* denotes adding 20% of the missed ground-truth word boxes. *Right*: Decreasing the word detection recall by removing the true positive detections (TP) from the automatically generated set negatively influences the classification performance. *'-TP20'* denotes removing 20% of the true positives. This set of experiments show that word detection recall is more crucial than precision for the classification performance.

cant drop in precision from 100% to 6.2% results in a marginal decrease in classification performance.

These experiments indicate that the false positive word proposals generated by the proposed method do not negatively influence fine-grained classification. However, it is worth to mention that it is still desirable to produce a limited number of word proposals for memory and efficiency concerns.

3) The influence of word detection recall: First, we evaluate the influence of a recall increase on the classification rate. We systematically increase the recall by adding the missed ground-truth word boxes (mGT) on top of the automatically generated set of proposals. As shown in Figure 7 (middle), the classification performance keeps increasing as the word detection recall increases before it saturates.

Second, we decrease the recall by removing the true positive word proposals (TP) from the automatically generated set. The results in Figure 7 (right) show that decreasing the word detection recall negatively influences the classification performance.

Note that even when 90% of the true positive word proposals are removed, the classification performance is acceptable. There are two reasons for this. (1) As discussed in Section IV-B1 and IV-B2, the word proposal method is able to consistently detect a number of salient but non-text patterns which are contributing positively to the classification. (2) The boxes that cover the text regions, with less than 50% overlap with the ground-truth, are treated as false positive. Therefore, even when all true positives are removed, these boxes contribute positively to the classification result.

Additionally, we evaluate the performance only using ground-truth boxes. In the case where only images containing text are considered, the performance is 76.1%. When all images are considered, the performance is 54.2%, outperformed by *Ours* (56.2%). When using the ground-truth boxes, the performance on images with no text is random, while when using our generated word proposals, the classification on images with no text is 28.9% in mAP, much better than random guessing (as discussed in section IV-B1). This is why when all images are considered, including both images with text and images without text, the performance of using our generated boxes is slightly better than the result of using ground-truth boxes.

	Performance (mAP%)		
	Ours	*Characterness [25]*	*CTPN [48]*
Images with Text	67.7	37.8	50.6
All Images	56.2	30.6	38.1

TABLE II: Comparison to state-of-the-art text detection [25], [48]. [25], [48] aim at a high F-score. The recall, precision and F-score values of the proposed method are 64.7%, 4.7% and 8.7% while the values of [25] and [48] are 19.3%, 25.3%, 21.9% and 34.7%, 39.1%, 36.8% respectively. A high recall value is more effective than a high f-score for the fine-grained classification problem.

4) Comparison to state-of-the-art text detection: We compare the proposed word detection method with two state-of-the-art text detection approaches [25], [48]. The textual cue encoding and the classification steps are kept same. The recall, precision and f-score values of the proposed method are 64.7%, 4.7% and 8.7%. The values of [25] and [48] are 19.3%, 25.3%, 21.9% and 34.7%, 39.1%, 36.8% respectively. Compared to [25], [48], the proposed method achieves a significantly higher recall but a lower precision and f-score. In terms of fine-grained classification performance, as shown in Table II, the proposed method (*Ours*) outperforms [25], [48] by a large margin.

	Performance (mAP%)
Textual-only (full)	**38.3±0.9**
Textual-only (gray-only)	33.1±0.5
Textual-only (characterness [25])	20.2±0.6
Textual-only (CTPN [48])	27.1±0.5
Visual-only (BOW)	34.0±0.3
Visual-only (DEEP)	53.3±0.08
Visual-only (DEEP-LOC)	56.1±0.8
Visual-only (DEEP-GL)	58.4±0.3
Visual-only (DEEP-FT)	60.3±0.2
Visual-only (DEEP-FT-LOC)	59.3±0.2
Visual-only (DEEP-FT-GL)	**64.7±0.2**
Textual (full) + Visual (BOW)	55.8±1.0
Textual (full) + Visual (DEEP)	71.0±0.5
Textual (full) + Visual (DEEP-LOC)	71.6±0.7
Textual (full) + Visual (DEEP-GL)	73.5±0.6
Textual (full) + Visual (DEEP-FT)	74.5±0.8
Textual (full) + Visual (DEEP-FT-LOC)	73.9±0.7
Textual (full) + Visual (DEEP-FT-GL)	**77.3±0.7**
Textual (gray-only) + Visual (DEEP-FT-GL)	75.6±0.5
Textual (characterness [25]) + Visual (DEEP-FT-GL)	70.6±0.7
Textual (CTPN [48]) + Visual (DEEP-FT-GL)	74.0±0.8

TABLE III: Fine-grained classification performance on *ConText dataset*. The textual cue encoded by the proposed method is effective. It is complementary to the visual information. *Textual-only (full)*, *Textual-only (gray-only)*, *Textual-only (characterness* [25]) and *Textual-only (CTPN)* [48] only differ in word detection. Textual cue encoding and classification steps are kept the same. *full* outperforms *gray-only*, *characterness* [25] and *CTPN* [48] thanks to a higher recall in word detection.

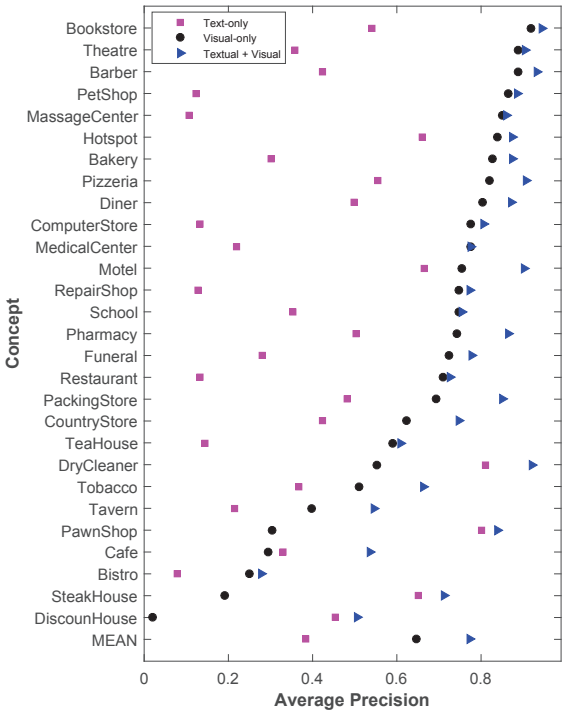


Fig. 8: Fine-grained classification performance for each class. Here the visual-only baseline is the best performing one, i.e., *DEEP-FT-GL*. Adding textual cues improves the performance on 27 classes out of 28. The proposed multimodal approach improves the best visual-only baseline from 64.7% to 77.3% in mean average precision. The multimodal approach guarantees at least 60% except four classes.

C. Performance evaluation on 28 classes

In this section, we use all 28 classes for evaluation and conduct two experiments. First, we evaluate the effectiveness of the textual cues extracted by the proposed method on the 28-class classification problem. Second, we compare the classification performance of word-level and character-level textual cue encoding.

Experiment I. Four different ways to generate word box proposals are considered: (1) the proposed method using all color channels, denoted by *full*, (2) the proposed method using only the gray scale, denoted by *gray-only*,(3) the text detection approach proposed in [25] aiming at a high f-score, denoted by *characterness* and (4) a very recent text detection method [48] also aiming at high f-score, denoted by *CTPN*. We evaluate the sets of word box proposals generated by these four different ways separately while keeping the textual cue encoding and classification steps the same.

As shown in Table III, when only textual cues are considered for the classification, *full* outperforms *gray-only*, *characterness* and *CTPN* thanks to a higher recall in word detection. The same holds when textual cues and visual cues are combined. Combining the proposed textual cues with visual cues always improves the visual-only baselines, regardless of how strong the visual baselines are. It can be derived that recognized words in images contain discriminative information and that it is complementary to visual cues.

As a side note, it can be observed from Table III that the local regional information is useful for the fine-grained classification problem. When the pre-trained GoogLeNet is used, the local visual cue outperforms the global cue (*Visual-only (DEEP-LOC)* vs. *Visual-only (DEEP)*). For the fine-tuned network, the local cue is comparable with the global cue(*Visual-only (DEEP-FT-LOC)* vs. *Visual-only (DEEP-FT)*). The local cue does not outperform the global visual cue for the fine-tuned network due to the fact that the fine-tuned network is optimized based on the global image as input. In both cases, the local cue provides complementary information to the global cue (*Visual-only (DEEP-GL)* and *Visual-only (DEEP-FT-GL)*).

Figure 8 shows the per-class performance. Here the visual baseline is the best performing one, i.e., *DEEP-FT-GL*. The low performance of textual cues is due to the lack of scene text, e.g., for classes *Bistro* and *Massage Center*. However, combining visual and textual cues improves visual-only on 27 classes out of 28. The performance improvement is the highest on the classes where visual cues are not sufficient and textual cues are discriminative, e.g., *Pawn Shop*, *Steak House* and *DiscounHouse*.

Experiment II. We compare our approach with the state-of-the-art [24] which extracts the textual information at a char-

	Performance (mAP%)		
	text only	*visual only*	*text and visual*
Word Level [this paper]	38.3	34.0	**55.8**
Character Level [24]	15.6	32.9	39.0

TABLE IV: Comparison to the state-of-the-art [24] which encodes textual cues at character level. Numbers are taken from [24]. Where the visual-only performance is compatible, the proposed method outperforms [24] by a large margin. It can be derived that representing the textual information at word level is more effective than at a character level.

acter level. To ensure a fair comparison, in this experiment we use *BOW* for the visual-based classification as [24]. Table IV summarizes the results. Our method outperforms [24] by a large margin (16.8%). It shows that representing the textual information at a word level is more effective than at a character level.

V. Logo Retrieval

In logo retrieval, the objective is to retrieve all images of a specific logo from an image collection, e.g., *Heineken*, given one image example of that logo as query. Logo retrieval is useful for measuring brand exposure. Logo is a special type of objects where text can be part of the object. Examples are *Starbucks*, *Ford* and *Google*. Previous works [21], [41], [42], [47] do not consider the recognized text of the logo. These methods treat the text of the logos the same as other visual patterns. In contrast, we explicitly extract the word-level textual cues in the logos and utilize it for logo retrieval.

A. Dataset and Implementation Details

Dataset. We evaluate our approach on *FlickrLogos-32* [43]. *FlickrLogos-32* has 32 brand logos, e.g., *Google*, *Coca-cola* and *DHL*. We follow the retrieval setting of [42], which defines a set of 960 queries, 30 per logo, and a search set of 4280 images in total. The search set consists of 1280 logo images, 40 per logo, and 3000 non-logo images.

Implementation notes. The common method for logo retrieval is to use low level feature matching. In line with this paradigm, two visual baselines are considered. First, we use the available BOW representations with a visual vocabulary of 1 million visual words [42], denoted by *BOW*. Second, we implement another visual baseline based on aggregated selective match kernels [49], denoted by *ASMK*. The visual vocabulary has 20000 visual words. The kernel we use is a thresholded 4-degree polynomial kernel expressed by $\sigma(\mu) = [\mu > 0]\mu^4$, where the square bracket stands for the Iverson bracket.

For textual cues, we encode the images in the same way as in the previous fine-grained classification application, detailed in Section III. For the query images, we use the query boxes to only keep the word proposals that overlap with the query boxes. The textual representations are normalized to unit length and cosine similarity is used to rank the images.

To combine the visual and textual cues, we perform a late fusion on the similarity scores obtained from the two modalities. Both sum fusion, expressed by $S_{fusion} = S_{visual} + S_{textual}$, and product fusion, expressed by $S_{fusion} = S_{visual} * (S_{textual} + \epsilon)$ are tested. S_{fusion}, S_{visual} and $S_{textual}$ are the fused score, visual-based score and textual-based score respectively. ϵ is a small constant value added to handle cases where no text has been detected. Sum fusion requires the two scores to be roughly in the same numerical range while product fusion does not. For this reason, only the product fusion is considered for fusing with *ASMK* as the similarity scores produced by *ASMK* lie in a very different range from the scores generated based on the textual cues. The product fusion is also different from the sum fusion because the product fusion has a higher requirement than the sum fusion on the quality of both modalities to derive a decent final result. In general, the product fusion requires both modalities to be reasonably good.

	mAP%
Textual-only (full)	**32.2**
Textual-only (gray-only)	28.4
Textual-only ([25])	12.3
Textual-only ([61])	13.2
Textual-only ([56])	12.7
Visual-only (BOW)	54.8
Visual-only (ASMK)	**58.4**
Textual (full) + Visual (BOW) [*sum fusion*]	59.4
Textual (gray-only) + Visual (BOW) [*sum fusion*]	57.8
Textual ([25]) + Visual (BOW) [*sum fusion*]	56.0
Textual ([61]) + Visual (BOW) [*sum fusion*]	56.2
Textual ([56]) + Visual (BOW) [*sum fusion*]	55.9
Textual (full) + Visual (BOW) [*product fusion*]	59.5
Textual (gray-only) + Visual (BOW) [*product fusion*]	56.9
Textual ([25]) + Visual (BOW) [*product fusion*]	36.2
Textual ([61]) + Visual (BOW) [*product fusion*]	34.5
Textual ([56]) + Visual (BOW) [*product fusion*]	30.8
Textual (full) + Visual (ASMK) [*product fusion*]	**62.7**
Textual (gray-only) + Visual (ASMK) [*product fusion*]	61.0
Textual ([25]) + Visual (ASMK) [*product fusion*]	41.5
Textual ([61]) + Visual (ASMK) [*product fusion*]	40.1
Textual ([56]) + Visual (ASMK) [*product fusion*]	36.5

TABLE V: Logo retrieval performance on *FlickrLogos-32* [43]. Adding the proposed textual cues always improves the retrieval performance. The proposed textual cues are more effective than the textual cues from other text detection methods due to the focus on high recall word detection.

B. Experiments and Results

This section experimentally evaluates the proposed multi-modal approach to logo retrieval. We quantify the added value of the proposed textual cues on top of the visual baselines. Moreover, we compare with several state-of-the-art text detection methods for the purpose of logo retrieval.

Table V summarizes the results. Adding the proposed textual cues 'Textual (full)' and 'Textual (gray-only)' always improves the visual baselines. The best performance, 62.7%

Fig. 9: Example queries where adding textual cues decreases the performance. The reasons are no text (*Ferrari*), exotic font style (*Cocacola*) and vertical text (*Foster* and *Guinness*).

in mAP, is achieved by combining the proposed textual cues (full) with the visual baseline (*ASMK*) using product fusion. Interestingly, fusing the textual cues from other text detection methods with the visual baselines using the product fusion does not improve the performance because the performance of the textual part is too modest in these cases. From the experiments, it can be concluded that the proposed textual cue extraction that focuses on high recall word detection is effective, resulting in a textual cue complementary to the visual cues for logo retrieval.

Analysis. Adding the textual cues improves the retrieval performance on 641 queries out of 960 ('Textual (full) + Visual (ASMK)'). Text is helpful when it is in standard fonts and orientations. On the other hand, when text is not there or it is in exotic fonts or orientations, adding textual has a negative effect on the accuracy. Figure 9 shows 4 example queries where considering textual information decreases the performance of visual-only. For the query of *Ferrari*, considering textual information is not helpful because there is simply no text. The example of *Cocacola* is due to the exotic font style which makes it unrecognizable. For *Foster* and *Guinness*, the vertical text makes detection and recognition fail.

VI. Word Box Proposal Evaluation

Dataset. We evaluate the performance of our word box proposal method on the *SVT* dataset [55]. The dataset consists of 249 images downloaded from Google Street View of road-side scenes. The dataset has word-level box annotations.

Evaluation measures. The performance is measured in terms of recall, number of proposals and average maximum overlap (*AMO*) (See Table VI). We calculate the overlap between each groundtruth box and its best overlapping word box proposal. AMO is the average of these overlap values.

A. Experiments and Results

We conduct three experiments. First, we evaluate the effect of the color spaces and the character detection algorithms on the word detection performance. Second, we compare with state-of-the-art word box proposal methods [19], [15]. Third, we analyze the influence of ground-truth overlap threshold on word detection recall and word recognition accuracy.

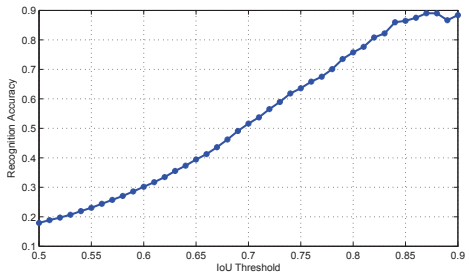


Fig. 10: The relation between the ground-truth overlap threshold (i.e., the IoU threshold) and the word recognition accuracy, evaluated on *SVT* dataset [55]. Proposals with higher IoU values are better recognizable.

Experiment I. The proposed method generates word box proposals using different color spaces and character detection algorithms. Word box proposals are generated for each color space independently and then combined. The same candidate regions may be detected for the different color spaces or character detection algorithms. To filter out these duplicate regions, non-maximum suppression is applied.

Table VI shows that adding more color spaces improves the performance in terms of recall and AMO. When a single character detection algorithm is used, the recall values for MSER and text saliency are 85.47% and 90.88% respectively, whereas the recall is 96.14% when both algorithms are considered. Hence, the use of color spaces with different invariant properties, and complementary character detection algorithms results in a high recall.

Experiment II. We compare the performance of our word proposals with the state-of-the-art word proposal methods [19], [15]. [19] uses generic object proposal methods to generate preliminary word box proposals. However, the number of boxes is prohibitively large ($> 10^4$). Therefore, [19] filters out most of these boxes using a Random Forest text/non-text classifier. As their recognition step is based on the preciseness of the word boxes, a convolutional neural network regressor is learned to refine the coordinates of the remaining word boxes. [15] uses MSER with flexible parameters and a grouping strategy to generate word proposals. These proposals are also further scored by a weak classifier for word-likeliness. Table VI shows that our method achieves a slightly higher recall than [19], [15] while requiring fewer boxes.

Experiment III. As common practice in text detection, a candidate word box is considered as a true positive if it overlaps more than 0.5 with the ground-truth word box. However, a 0.5 overlap does not guarantee a correct word recognition. In particular, not all true positives are correctly recognized. We analyze the relation between the recognition accuracy and the overlap. The lexicon word with the maximum probability returned by [18] is considered as the word recognition result for each word proposal. Given the word proposals that pass the overlap threshold, the recognition accuracy is computed as the percentage of correctly recognized proposals. Concretely,

	#proposals	recall(%)	AMO(%)
[This paper] MSER+TSAL, I	338	84.23	70.40
[This paper] MSER+TSAL, I+O_1, O_2+S	806	95.21	77.08
[This paper] MSER+TSAL, I+O_1, O_2+S, H	968	96.14	**77.54**
[This paper] MSER, I+O_1, O_2+H, S	568	85.47	70.90
[This paper] TSAL, I+O_1, O_2+H, S	500	90.88	75.12
TextProposals [15]	17358	94.00	-
Jaderberg et al. [19] without (RF+CNN-reg)	$> 10^4$	**97.00**	77.00
Jaderberg et al. [19] without CNN-reg	900	94.80	-
Jaderberg et al. [19]	900	-	-

TABLE VI: Evaluation of the word box proposals on SVT dataset. *MSER* and *TSAL* are the MSER based and text saliency based character detection algorithms. I, O_1 ,O_2, H and S are the color models. The recall increases as more color invariant models are combined because of their complementary photometric invariant properties. Using both character detection algorithms results in a higher recall than using a single algorithm. RF and CNN-reg of [19] are the Random Forest classifier for non-text box filtering and the convolutional neural network regressor for box refinement. The values for [15], [19] are taken from the references, and empty blocks are not reported in the references. Different from [15], [19] the proposed method is fully unsupervised.

for a specific overlap threshold, e.g., 0.7, we take all the word proposals that have at least 0.7 overlap with ground-truth, and compute how many of them are correctly recognized. The results are summarized in Figure 10. The results show that the candidate word boxes (proposals), which have higher overlap with ground-truth, also have higher recognition accuracy. Therefore, not only higher recall but also higher *AMO* is important for accurate textual cue extraction. Further, we vary the ground-truth overlap threshold and evaluate the word detection recall. As expected, increasing the threshold has a negative effect on the recall, see Figure 11. However, the proposed method still performs well for a threshold of (> 0.75). For this threshold value, the recall and recognition accuracy is around 70%.

In addition, we evaluated the word recognition performance of the proposed method. A common practice to improve word recognition accuracy is to make use of dataset specific dictionaries. However, we did not use the dictionaries provided for this dataset to refine the recognition results. The proposed method reaches a word recognition recall 74.65%, while the end-to-end word recognition recall reported in [19] is 59%. Both methods use the same recognition method, and the performance difference is from the detection. The performance gain shows the contribution of the proposed method to end-to-end text recognition.

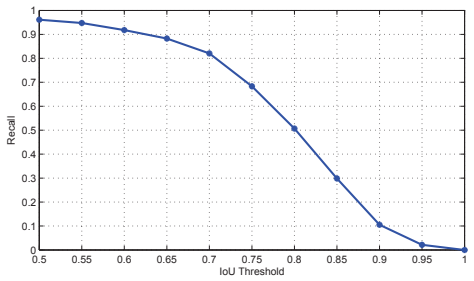


Fig. 11: The influence of the ground-truth overlap threshold (i.e., the IoU threshold) on word detection recall, evaluated on *SVT* dataset [55]. Recall decreases as IoU threshold increases.

VII. Conclusion

In this paper, we have demonstrated the effectiveness of textual cues for fine-grained business place classification and logo retrieval. In addition, we show that word-level textual cues are more effective than character-level textual cues. The proposed word-level textual cues significantly outperform the state-of-the-art [24] character-level textual cues. To extract word-level textual cues in images, a generic, efficient and fully unsupervised word proposal method is introduced. The method reaches state-of-the-art word detection recall while keeping the number of proposals limited. The main focus of text detection literature is on obtaining high f-score. In contrast, we show that high recall in word detection is more relevant than high f-score for fine-grained classification and logo retrieval. To validate the influence of the word detection recall, precision and f-score on fine-grained classification, we have annotated a large set with 10K images and 27601 word boxes.

Acknowledgment

This research is supported by the Dutch national program COMMIT/.

References

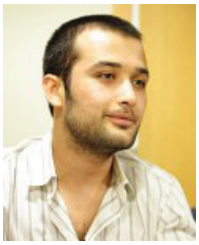
[1] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. In *TPAMI*, 2014. 3
[2] Katherine L Bouman, Golnaz Abdollahian, Mireille Boutin, and Edward J Delp. A low complexity sign detection and text localization method for mobile applications. *TMM*, 13(5):922–934, 2011. 4
[3] Leo Breiman. Random forests. *Machine Learning*, 2001. 3
[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. 7
[5] Chih-Chung Chang and Chih-Jen Lin. The attraction of visual attention to texts in real-world scenes. *Journal of Vision*, 2012. 4
[6] Huizhong Chen, Sam S. Tsai, Georg Schroth, David M. Chen, Radek Grzeszczuk, and Bernd Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *ICIP*, 2011. 4, 5

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3
[8] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *TPAMI*, 2014. 3
[9] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, 2010. 2, 5
[10] I. Everts, J. C. van Gemert, and T. Gevers. Per-patch descriptor selection using surface and scene properties. In *ECCV*, 2012. 5
[11] Nobuo Ezaki, Marius Bulacu, and Lambert Schomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *ICPR*, 2004. 5
[12] Basura Fernando, Sezer Karaoglu, and Alain Trémeau. Extreme value theory based text binarization in documents and natural scenes. In *ICMV*, 2010. 4
[13] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013. 3
[14] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In *ICDAR*, 2013. 3
[15] Lluis Gomez-Bigorda and Dimosthenis Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *arXiv preprint arXiv:1604.02619*, 2016. 3, 11, 12
[16] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV*, 2014. 2
[17] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *ECCV*, 2014. 2
[18] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 2, 3, 6, 11
[19] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016. 3, 11, 12
[20] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 7
[21] Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. In *ACM MM*, 2009. 10
[22] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *ICCV*, 2009. 4
[23] Sezer Karaoglu, Jan C van Gemert, and Theo Gevers. Object reading: Text recognition for object recognition. In *ECCV Workshops and Demonstrations*, 2012. 2, 4
[24] Sezer Karaoglu, Jan C van Gemert, and Theo Gevers. Con-text: Text detection using background connectivity for fine-grained object classification. In *ACM MM*, 2013. 3, 6, 7, 9, 10, 12
[25] Yao Li, Wenjing Jia, Chunhua Shen, and A van den Hengel. Characterness: an indicator of text in the wild. *IEEE transactions on image processing*, 23(4):1666–1677, 2014. 2, 5, 8, 9, 10
[26] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *ECCV*, 2012. 1
[27] Shijian Lu, Tao Chen, Shangxuan Tian, Joo-Hwee Lim, and Chew-Lim Tan. Scene text extraction based on edges and support vector regression. *IJDAR*, 18(2), 2015. 2
[28] Tong Lu, Shivakumara Palaiahnakote, Chew Lim Tan, and Wenyin Liu. Text detection in multimodal video analysis. In *Video Text Detection*, pages 221–246. 2014. 3
[29] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 1
[30] Junhua Mao, Houqiang Li, Wengang Zhou, Shuicheng Yan, and Qi Tian. Scale based region growing for scene text detection. 2013. 5
[31] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002. 2, 4
[32] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012. 3
[33] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *CVPR*, 2012. 3
[34] Anand Mishra, Karteek Alahari, and CV Jawahar. Image retrieval using textual cues. In *ICCV*, 2013. 3
[35] Yair Movshovitz-Attias, Qian Yu, Martin C Stumpe, Vinay Shet, Sacha Arnoud, and Liron Yatziv. Ontological supervision for fine grained classification of street view storefronts. In *CVPR*, 2015. 3
[36] Lukas Neumann and Jiri Matas. A method for text localization and recognition in real-world images. In *ACCV*. 2010. 5
[37] Lukas Neumann and Jiri Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *ICDAR*, 2011. 5, 6
[38] Lukas Neumann and Jiri Matas. Real-time scene text localization and recognition. In *CVPR*, 2012. 2, 4
[39] M-E Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 1, 6
[40] Tatiana Novikova, Olga Barinova, Pushmeet Kohli, and Victor Lempitsky. Large-lexicon attribute-consistent text recognition in natural images. In *ECCV*, 2012. 3
[41] Jerome Revaud, Matthijs Douze, and Cordelia Schmid. Correlation-based burstiness for logo retrieval. In *ACM MM*, 2012. 10
[42] Stefan Romberg and Rainer Lienhart. Bundle min-hashing for logo recognition. In *ICMR*, 2013. 10
[43] Stefan Romberg, Lluis Garcia Pueyo, Rainer Lienhart, and Roelof Van Zwol. Scalable logo recognition in real-world images. In *ICMR*, 2011. 10
[44] Marçal Rusiñol, Volkmar Frinken, Dimosthenis Karatzas, Andrew D Bagdanov, and Josep Lladós. Multimodal page classification in administrative document image streams. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(4):331–341, 2014. 3
[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 7
[46] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 2, 7
[47] Ran Tao, Efstratios Gavves, Cees G M Snoek, and Arnold W M Smeulders. Locality in generic instance search from one example. In *CVPR*, 2014. 10
[48] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 2016. 8, 9
[49] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013. 10
[50] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016. 7
[51] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 5
[52] Joost Van de Weijer, Theo Gevers, and Andrew D Bagdanov. Boosting color saliency in image feature detection. *TPAMI*, 28(1):150–156, 2006. 4
[53] Luc Vincent. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *TIP*, 2(2):176–201, 1993. 4
[54] Gang Wang, Derek Hoiem, and David Forsyth. Building text features for object image classification. In *CVPR*, pages 1367–1374, 2009. 3
[55] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, 2011. 2, 11, 12
[56] Tao Wang, David J. Wu, Adam Coates, and Andrew Y. Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, 2012. 2, 10
[57] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD birds 200. 2010. 1, 6
[58] Liang Wu, Palaiahnakote Shivakumara, Tong Lu, and Chew Tan. A new technique for multi-oriented scene text lines detection and tracking in video. *TMM*, 2015. 2
[59] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015. 1, 3
[60] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *TIP*, 23(11):4737–4749, 2014. 2
[61] Chucai Yi and YingLi Tian. Text string detection from natural scenes by structure-based partition and grouping. *TIP*, 20(9):2594–2605, 2011. 10
[62] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *TPAMI*, 36(5):970–983, 2014. 2
[63] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014. 1, 3
[64] Yu Zhang, Xiu-shen Wei, Jianxin Wu, Jianfei Cai, Jiangbo Lu, Viet-Anh Nguyen, and Minh N Do. Weakly supervised fine-grained image categorization. *arXiv preprint arXiv:1504.04943*, 2015. 3

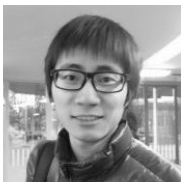
[65] Zheng Zhang, Wei Shen, Cong Yao, and Xiang Bai. Symmetry-based text line detection in natural scenes. In *CVPR*, 2015. 2

[66] Qiang Zhu, Mei-Chen Yeh, and Kwang-Ting Cheng. Multimodal fusion using learned text concepts for image categorization. In *ACMMM*, 2006. 3

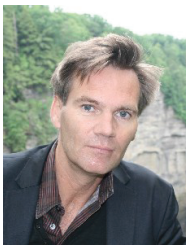
[67] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 3, 7



Sezer Karaoglu received his Ph.D. degree at Computer Vision Group, Informatics Institute, University of Amsterdam. He was selected as tuition fee scholar for a European Master degree from Color in Informatics and Media Technology (CIMET) program. He holds double master degree. His research interests are 2D-3D Computer Vision and Machine Learning. He is the CTO and Co-founder of 3DUniversum, spin-offs of the Informatics Institute of the UvA.



Ran Tao received the M.Sc. degree in Computer Science (2011) from Leiden University, The Netherlands. He is currently a Ph.D. candidate at University of Amsterdam, The Netherlands. His research interests include computer vision and machine learning, with a focus on instance search, object tracking and deep learning.



Theo Gevers is a Full Professor of Computer Vision with the University of Amsterdam (UvA), Amsterdam, The Netherlands. His main research interests are in the fundamentals of image understanding, 3-D object recognition, and color in computer vision. He is the Co-founder of Sightcorp and 3DUniversum, spin-offs of the Informatics Institute of the UvA.

Arnold W.M. Smeulders is in charge of COMMIT/, a nation-wide, very large public-private research program distributed over the Netherlands on large-scale data, content, sensing and interaction. And he is professor at the University of Amsterdam UvA for research in the theory and practice of computer vision. The groups search engines have received a top-three performance for all 14 years in the international TREC-vid competition for image categorisation. He was recipient of a Fulbright fellowship at Yale University, and visiting professor in Hong Kong, Tuskuba, Modena, Cagliari and Florida. He was co-founder of Euvision Technologies BV, a company spin-off from the UvA. He is currently director of the Qualcomm - UvA and the Bosch - UvA labs. He is associate editor of the IJCV. He is fellow of the International Association of Pattern Recognition and elected member of the Academia Europaea (AE).