Text Detection in Stores Using a Repetition Prior

Bo Xiong Kristen Grauman
University of Texas at Austin
{bxiong,grauman} @cs.utexas.edu

Abstract

Text detection in stores has valuable applications that could transform the shopping experience, yet cluttered store environments present distinct challenges for existing techniques. We propose a strategy for text detection in stores that exploits a repetition prior. *Leveraging the fact that shops typically display multiple instances of the same product on the shelf, our approach localizes text regions with a global view of the image, preferring instances that have repeated support in the scene. On two challenging real-world datasets taken with a mobile phone and wearable camera, we demonstrate our method's substantial advantages compared to several state-of-the-art techniques in grocery store environments.*

1. Introduction

Text *detection* in natural scenes requires localizing regions in the image containing text—no matter what that text says, or what font it is written in. Text, signs, and labels are ubiquitous and informative in many natural environments. As such, with the increasing use of portable mobile and wearable computing platforms, reliable text detection is critical for many applications. For example, text detection (and a subsequent recognition process) is vital to real-world applications such as sign reading for place localization for tourists or mobile robots [19]; for assistive technology to help visually impaired users navigate the world with more independence [7, 22, 39]; and for image/video indexing and retrieval based on scene text or graphical overlays [17, 12].

Whereas early work focused on constrained scenarios, such as finding lines of text in a document, today's methods tackle text detection "in the wild" in natural scenes. Doing so requires robustness to different fonts, languages, illuminations, orientations, occlusions, and clutter. While in some cases one can assume a lexicon of known words is available, in the more general "lexicon-free" setting the method must also detect words that were never previously seen. Once a bounding box around text in the scene is localized, it can be passed to a recognition pipeline to read what the words say.

Figure 1. There are distinct challenges in finding text in store settings (right) versus street-side scenes (left). Text detection in a store is difficult due to the high density of text and text-like textures. We propose to exploit the fact that products appear in duplicate on store shelves when performing text detection, using the redundancy as a helpful prior.

Despite a surge of exciting progress in natural scene text detection, we observe that a domain of great practical interest—stores—has largely been ignored. Current methods and datasets often focus on outdoor StreetView-style settings where text may appear on storefront signs, street signs, addresses, or license plates [37, 31, 40, 16]. However, text is also abundant in indoor store environments, where text appears on the labels of products that line the shelves (e.g., grocery stores, bookstores, electronics, etc.). Detecting it would assist in identifying products, retrieving relevant product reviews, reading prices, checking online vendors, searching for relevant coupons, or helping a visually impaired user complete his shopping list. Such applications promise to revolutionize the traditional shopping experience, mixing the bricks-and- mortar environment with the online marketplace.

However, text detection in a store presents its own challenges. Images of store shelves contain many products crowded together. Even worse, many products contain design patterns that share similar texture as text, and most have a high density of text occurrences. These properties can be problematic for mainstream text detection systems using sliding window [37, 13, 3, 43] or connected components [6, 2, 24, 42, 26, 27] to find characters. While some products are rigid and have planar surfaces facing outwards, others are deformable or have more complex shapes, and,

regardless, consumers disrupt the orderliness whenever they pick and replace a product. Furthermore, whereas existing datasets often contain images purposely taken so the text is somewhat prominent within the view, imagery captured more casually and even passively (i.e., on a shopper's wearable camera) will lack helpful cues implicit in the image composition. See Figure 1.

We propose an approach to text detection that specifically targets indoor store settings. As discussed above, the high density of products on a shelf creates many nuisances for detecting text. However, that same density comes along with one helpful factor: *each product typically appears multiple times on the shelf, side by side*. Our key insight is that duplicate occurrences of text can be a valuable prior for a text detector. Intuitively, a detector primed to see multiple instances of the same word can prioritize windows that have repeated support. We call this a *repetition prior*. Enforcing this prior is non-trivial, since not only the text repeats, but so does everything else on the product label!

Our method works as follows. In contrast to a standard sliding window approach that would check each region in isolation for its "text-ness", we take a more global view of the scene and jointly detect text, with a preference for text windows that repeat. Given an image, our approach first generates a set of text bounding box proposals using low-level cues. Then, a clustering step identifies those candidates with repeated support in terms of text appearance, overlap, and scale similarity. The resulting clusters are considered to be the most trustworthy text windows. Using those high-precision windows as anchors, we expand recall via local feature matching between the clustered hypotheses and the remaining image. The output is a ranked list of detected text bounding boxes (one word per box) and their confidences.

We validate our approach on two challenging grocery shopping datasets taken with a mobile phone [8] and wearable Google Glass camera [29], both of which we newly annotate to support benchmarking of text detection. We demonstrate that by leveraging a text repetition prior, our method outperforms and/or enhances multiple state-of-the-art techniques. The result is a promising step toward text detection in complex real-world shopping environments.

2. Related Work

We next summarize how our idea relates to previous work in text detection, product recognition, near-duplicate image detection, and object cosegmentation.

Text detection Space does not permit a comprehensive review of the text detection literature, so we briefly summarize current trends. Please see [41, 44] for surveys. One fundamental strategy is to search for text-like regions using bottom-up grouping [6, 2, 24, 42, 4, 26, 27]. The Stroke Width Transform (SWT) [6] and Maximally Stable Extremal Regions (MSER) [24] are two widely used examples. An alternative strategy is to learn a detector to classify pixels or windows as text/non-text (or a particular character) [37, 13, 3, 43, 23, 12, 28]. For example, recently deep convolutional neural networks [38, 11, 13, 12] have been explored. We employ the method of [13] to generate our initial text proposal regions. In both major strategies, the detected characters are then grouped into words. Recent work also considers detecting whole words at once [12]. When available, a lexicon can guide the detection [1, 37]. However, like recent methods [7, 2, 42, 26, 26, 27, 43, 11, 6, 3, 38, 12, 9, 12], we aim to operate lexicon-free, so that prior knowledge about the words to be encountered is not required.

In contrast to any existing text detection work, 1) we are specifically concerned with text in store settings, 2) we propose a novel "repetition prior" suited to those settings, and 3) we treat the detection process at the scene level, as opposed to independently scanning each region for the presence of text.

Product recognition Prior vision systems involving store products focus on the product recognition problem [8, 39, 18, 35, 32], including limited work specifically for groceries [8, 39, 22]. In those systems, the task is instance recognition, where local features are matched to a database of object models. In contrast, our task is text detection, where text regions are discovered on objects for which the system has no prior model. While it may be possible to link the two ideas, our problem is distinct in important ways. First, knowing where an object (product) is falls short of extracting its text regions, which are needed if word recognition is to be done. Second, the need for a bank of known object models is restrictive; in a store setting, new products are continually added and vendors revise the product labels over time. Third, the repetitive structure we exploit in our approach is actually a confounding factor for standard instance recognition methods: spatial verification often fails when multiple similar looking things appear together [14, 34]. Finally, product recognition methods are known to fail for fine-grained differences (e.g., different flavors of the same food product), some of which may be best handled by good text detection (e.g., to distinguish "barbecue" vs. "plain" potato chips).

Detecting repeated patterns Our use of repeated patterns may bring to mind work on near-duplicate detection (NDD) (e.g., [15, 36]). In NDD, the goal is to identify similar images of the same real-world content, but with some minor alterations (like cropping, resampling, etc.). Like product recognition methods, NDD techniques largely rely on local feature (SIFT) matching. Unlike NDD, our goal is to detect text. Furthermore, in our case the repeated patterns

exist within the same image, and the number of repetitions is unknown, ranging from none to many.

The idea of performing better localization by exploiting repetition has its roots in object *cosegmentation* [30, 10]. Given two images containing the same object on two distinct backgrounds, cosegmentation methods seek a joint segmentation where the foregrounds agree in appearance. At a high level, our idea to jointly extract text bounding boxes by exploiting the fact they repeat across the scene is related in spirit. However, again, in our scenario the repeated patterns occur in the same image, and there are multiple possible patterns that repeat. Furthermore, whereas cosegmentation assumes the foreground objects rest against unrelated backgrounds, our "foreground" text regions are explicitly embedded within similar-looking backgrounds, a significant challenge.

3. Approach

Our goal is text detection in store environments. The input is an image, the output is a set of confidence-ranked bounding boxes believed to contain one word of text each. Our method does not perform text recognition.

Our "repetition prior" has value in settings where products repeat on the shelves, and they have labels with texts and/or visible price tags. This applies to places like grocery stores, bookstores, music stores, etc., but not arbitrary stores, e.g., not most clothing stores. A user may snap a photo of the shelves using a mobile device or simply walk down the aisle wearing a camera like Google Glass; we study both such scenarios in our experiments. We make no explicit assumptions about the positioning of objects on the shelves, though (due to the behavior of existing text detection and feature matching methods) our approach will fare best when they are near-planar and/or positioned such that their text regions are similarly visible. We assume no prior knowledge about a lexicon nor any prior knowledge about how many unique products appear in a single input image. Our method is also independent of any specific product.

Figure 2 shows an overview of our approach. Given an image, we first generate candidate text bounding boxes (Sec. 3.1), and then identify similar candidate boxes by clustering with criteria specially crafted for our task (Sec. 3.2.1). Next, for each identified cluster, we expand the recall rate by matching a representative of each cluster to other visually similar regions that were ignored by the initial text detector (Sec. 3.2.2). Finally, based on the results of the clustering and matching stages, we rank each text bounding box with a confidence score (Sec. 3.2.3).

3.1. Generating text region proposals

The first step is to generate an initial set of text region proposals. Any existing text detector could be used for this step. It serves as a starting point, to be refined and expanded by our approach.

We use the state-of-the art method of Jaderberg et al. [13]. It trains a convolutional network network (CNN) to classify 24×24 pixel image patches as text or non-text. Given a novel image, the classifier is applied using a multi-scale sliding window search, producing a confidence-rated "text saliency" value at each pixel. We use the code kindly shared by the authors to obtain these saliency maps. Then, we postprocess them in a manner similar to [13] to obtain a set of candidate word box proposals. Specifically, we threshold the text saliency to find connected component regions of high probability; these likely correspond to individual characters. Then we group neighboring character regions based on their spatial distance and the differences in their heights, yielding boxes likely to correspond to words. Word bounding boxes are generated independently at each scale and then merged into a global set by non-maximal suppression.

3.2. Incorporating the repetition prior

With the set of text region proposals in hand, we next incorporate the repetition prior. The idea is to first identify similar-looking proposals and group them into clusters. As discussed above, since we expect *a priori* to see a word multiple times in the image, finding multiple similar proposals is evidence that they are more trustworthy instances. Conversely, "singleton" text region proposals, while still possibly valid and *not* discarded by our method, are trusted less under the repetition prior.

Incorporating a repetition prior is non-trivial. This is because 1) there is significant repetition of *non-text elements*, too, and 2) there is an unknown number of repeating products per input image, from zero to many. We design our clustering procedure and the subsequent matching step with these considerations in mind.

3.2.1 Connected components clustering of proposals

We pose the problem of finding multiple occurrences of the same text as finding connected components in a graph, which has several advantages. First, connected components allows us to find groups of similar proposals without hand-selecting the number of distinct words in the scene (i.e., as would be necessary with many alternative clustering methods). Furthermore, it naturally handles a mix of both singletons and repeating words. This means we can employ the prior without ignoring word occurrences that don't follow the prior. Finally, it permits a clean way to require multiple grouping criteria simultaneously via a binary adjacency matrix, as opposed to alternative clustering methods that combine multiple real-valued similarities.

We build a graph for the image, where each node is a

Figure 2. Overview of our approach. We start with candidate text bounding boxes as shown on the left (only a subset are drawn for legibility purposes), and then find similar candidates by connected components clustering (middle). Using each cluster as an anchor, we detect additional text boxes via local feature matching, as shown on the right. Original candidate boxes are drawn in solid lines and newly detected text candidate boxes are drawn in dotted lines. Best viewed on pdf.

proposal box. To define adjacency between the nodes, we consider three criteria: visual similarity, size, and overlap.

The visual similarity criterion says that two proposal nodes are connected if they look similar, meaning they are likely to contain the same text. As features we use histograms of DoG SIFT visual words, pooled in 1×4 spatial bins with a k-means vocabulary with 300 words. Let N denote the number of initial proposals, and let A^v be the $N \times N$ visual similarity adjacency matrix for the graph. For proposal nodes with descriptors x_i and x_j, we define

$$A^v_{ij} = \begin{cases} 1 & \text{if } \chi^2(x_i, x_j) < \tau \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where χ^2 is the χ^2 histogram distance, and τ is a threshold derived from the data. In particular, $\tau = \mu - 1.5\sigma$, where μ and σ are the mean and standard deviation, respectively, of the distances among all pairs of boxes in the image.

The size criterion says that two proposal nodes are connected if they are similar in size and aspect ratio. Let w_i, w_j and h_i, h_j denote the width and the height for candidate boxes i and j, respectively. We define the size adjacency matrix A^s by comparing the size ratios in each dimension:

$$A^s_{ij} = \begin{cases} 1 & \text{if } \frac{1}{\beta} < \frac{w_i}{w_j} < \beta \text{ and } \frac{1}{\beta} < \frac{h_i}{h_j} < \beta \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $\beta = \frac{4}{3}$. We set this threshold based on manually inspecting a few examples, then fixed it for all results.

The last adjacency matrix A^o captures the overlap criterion, which says two nodes cannot be connected if they overlap:

$$A^o_{ij} = \begin{cases} 0 & \text{if box } i \text{ overlaps box } j \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

This criterion is important to avoid clusters comprised of slightly different proposals surrounding the same text. It serves as a form of non-maximal suppression.

The final adjacency matrix A used for clustering combines all three adjacency matrices A^v, A^s, and A^o:

$$A_{ij} = \begin{cases} 1 & \text{if } A^v_{ij} = A^s_{ij} = A^o_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In other words, two nodes are only connected if they satisfy all three conditions.

We group vertices (candidate text boxes) into disjoint clusters by finding connected components with the adjacency matrix A. Each cluster with more than one node is hypothesized to be multiple occurrences of the same text.

3.2.2 Matching from the discovered clusters

Thus far, we have discovered plausible repeated text boxes. By ranking those boxes higher, we can expect improved precision thanks to the repetition prior. Next we show how to bootstrap from those clusters to further increase recall. This step augments the detections with regions missed by the original text detector (Sec. 3.1). In particular, we exploit the confident clustered boxes as "anchors" to search for those text regions that are difficult to find with the low-level text saliency metric alone, whether due to partial occlusions, non-frontal views, or varying illumination.

We find that due to the high rate of confuser features in the cluttered store scenes, the standard local feature-based object instance matching pipeline (e.g., [20]) is insufficient. This is because multiple occurrences of the same local features impede spatial verification and/or make many local features fail the ratio test, resulting in (seemingly) too few reliable points for matching. Thus, we devise a simple variant better suited to our setting, as follows.

First we select a proposal box for each connected component cluster from Sec. 3.2.1 to serve as its representative or "template".[1] We select the box with the minimum total

[1] We use the word "template" for simplicity, but note that it is matched via local features, not as a global template.

Figure 3. Overview of the matching step. We divide the image into grid cells and match local features from a cluster's representative text box (the "template") with those in each of the grid cells (left). For each matching, we project the template onto the image (middle). Then we refine the text box localization by matching the template with the projected region (right). See Sec. 3.2.2.

χ^2 distance to the rest of the cluster members. Then, we divide the image into overlapping grid cells, and match each template against each grid in the image. The grid adapts to the scale of each template. Namely, for a template with dimensions $w_t \times h_t$, the overlapping grid cells have width $1.5w_t$ and height $3h_t$ with horizontal step size $0.75w_t$ and vertical step size $1.5h_t$. To match a template to each grid cell, we use standard local DoG keypoint SIFT matching, followed by the second neighbor ratio test, and affine geometric verification. If more than 20 matched keypoints survive verification, we use the resulting affine transformation to project the template bounding box into the original image. Due to the robustness of local feature matching, the grid cell content may match the template even with partial occlusion.

Finally, to refine the match localization, we match the template again with the projected bounding box and then project the template to the original image with the refined affine transformation. See Figure 3. If the projected box does not overlap with any text bounding box already in the cluster, we add it as a new detection.[2]

3.2.3 Confidence ranking for text box proposals

The clustering and matching stages above leave us with 1) an expanded set of proposals relative to the initialization in Sec. 3.1, and 2) a means to refine the confidence rating associated with all proposals. Leveraging the repetition prior, we first group the proposals into two equivalence classes, based on whether they stem from a repeated pattern or not: those output by both Secs. 3.2.1 and 3.2.2 comprise one equivalence class, and the singletons comprise the other. Then, the boxes within each class are sorted by the sum of their pixels' text saliency (cf. Sec. 3.1). We use the sum, rather than the average, since otherwise small fragments of words would be favored. The output confidence ranking consists of the saliency-sorted class of repeated texts, followed by the sorted class of non-repeated texts. This entire process improves text detection precision, since we know which jointly extracted hypotheses are most trustworthy, as we will see in results.

[2]We also attempted a more complicated approach to jointly cluster and refine the boxes' localization, a la iterative methods used for weakly supervised segmentation [5, 33], but we found it to be inferior. This is likely due to the difficulty in relying on bottom-up text saliency as a "unary" potential on the boxes; it must be very strong to withstand the effects of the products' repeated non-text patterns surrounding the true text boxes.

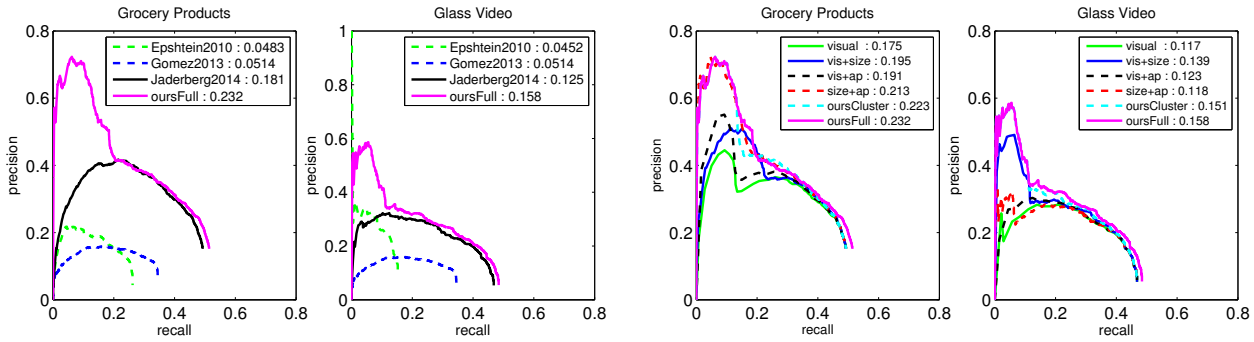
4. Results

We evaluate our approach on two challenging datasets and compare to multiple recent text detection methods. We also examine the impact of the design choices in our clustering approach.

Datasets We consider two realistic datasets containing grocery store images: GROCERY PRODUCTS [8] and GLASS VIDEO [29], both obtained from the authors.

- GROCERY PRODUCTS [8] consists of product images taken from five stores with mobile phones, with resolutions of 2448×3264 pixels or 3264×2448 pixels. The dataset has a total number of 3235 products, all of which are food and drinks, such as chocolate bar, milk bottles, cereal boxes and coffees. Each image contains a variety of products, ranging from 6 to 30 in number. For our tests, we use all 352 images containing detectable text (see below).

- GLASS VIDEO [29] consists of video frames captured with the wearable camera on GoogleGlass. It contains products similar to GROCERY PRODUCTS, but more variable viewpoints due to the wearable camera. Due to finite annotation resources, we select 139 frames from the three 8 minute videos that contain text and cover a diverse range of products. This data is even more challenging than the GROCERY PRODUCTS because 1) the frames have lower resolution (1280×720 pixels), 2) they suffer from motion blur, and 3) originating from a passive wearable camera, they con-



(a) Compare to existing methods

(b) Ablation study

Figure 4. (a) Text detection accuracy on the GROCERY PRODUCTS (left) and GLASS VIDEO (right) datasets, compared to existing methods. Our repetition prior improves the results of a state-of-the-art text detector [13], and also outperforms two existing bottom-up text finding methods [6, 9]. (b) Accuracy on the same datasets, using ablated variants of our approach. Our complete approach does best, supporting the method design choices. Numbers in legend denote mAP. Best viewed in color.

tain wider viewpoint variation compared to those photos taken purposely with a mobile phone in the other dataset.

Since both datasets were originally used for product recognition, we augment them with annotations for ground truth text bounding boxes to enable quantitative evaluation. To ensure labeling consistency, we followed the following three rules for annotation: 1) Label *each* word with a tight bounding box; 2) Do not label vertically oriented or partially visible text; 3) Do not label words of height less than 3% of the image height (80 or 20 pixels for the PRODUCTS and GLASS datasets, respectively). The last requirement is based on the minimal scale searched by the existing text saliency method [13]. This yielded a total of 2,390 and 1,222 text boxes for the two datasets[3], respectively.

Comparison to existing methods First we compare to three recent methods for lexicon-free text detection:

- STROKE WIDTH TRANSFORM, Epshtein et al. [6]: a well-known method that leverages the consistency of characters' stroke width to detect arbitrary fonts. We use the code provided at[4], applied on multiple scales to improve its results.

- MSER TEXT DETECTION, Gomez et al. [9]: uses Maximally Stable Extremal Regions [21]—a popular tool in text detection [2, 25, 26, 11, 9]—combined with a perceptual organisation framework. We use the authors' code.[5]

- DEEP TEXT SPOTTING, Jaderberg et al. [13]: a state-of-the-art method that uses multiple stages of convolutional neural networks to predict text saliency at each pixel (code publicly shared by the authors [6]), followed by the grouping stage described in Sec. 3.1 to generate boxes (implemented by us based on the authors' description in their paper).

For the first two methods [6, 9], we rank the outputs by bounding box size; the codes do not produce confidence values, so this is a sensible way to favor the more prominent detected texts, which are more often correct. For [13], we use the summed text saliency scores as done for our method (cf. Sec. 3.2.3).

Figure 4(a) shows the precision-recall results for each dataset. We follow the standard PASCAL VOC detection criterion: a detection is correct if its bounding box's Intersection over Union (IoU) score exceeds 50% overlap with the ground truth.

Overall, our method outperforms the existing methods. Our gains over the two non-learning approaches ([6, 9]) are largest, reinforcing recent findings about the power of learned character detectors that leverage large training data sets. Furthermore, we see sizeable gains over the state-of-the-art deep learning approach [13], particularly in terms of precision. This is an important empirical finding, since our method specifically builds on the output of [13], enhancing it with the repetition prior. Our method improves the CNN approach by leveraging the repetition prior to better rank the bounding boxes proposals *and* re-detect harder texts.

For all methods, the absolute accuracy is better on the GROCERY PRODUCTS dataset. This reflects the greater difficulty of the GLASS VIDEO data, as discussed above. The gap between our method and existing methods is also larger

[3] We make annotations publicly available for download at http://vision.cs.utexas.edu/projects/textdetect

[4] https://github.com/lluisgomez/DetectText

[5] https://github.com/lluisgomez/text_extraction

[6] https://bitbucket.org/jaderberg/eccv2014_textspotting

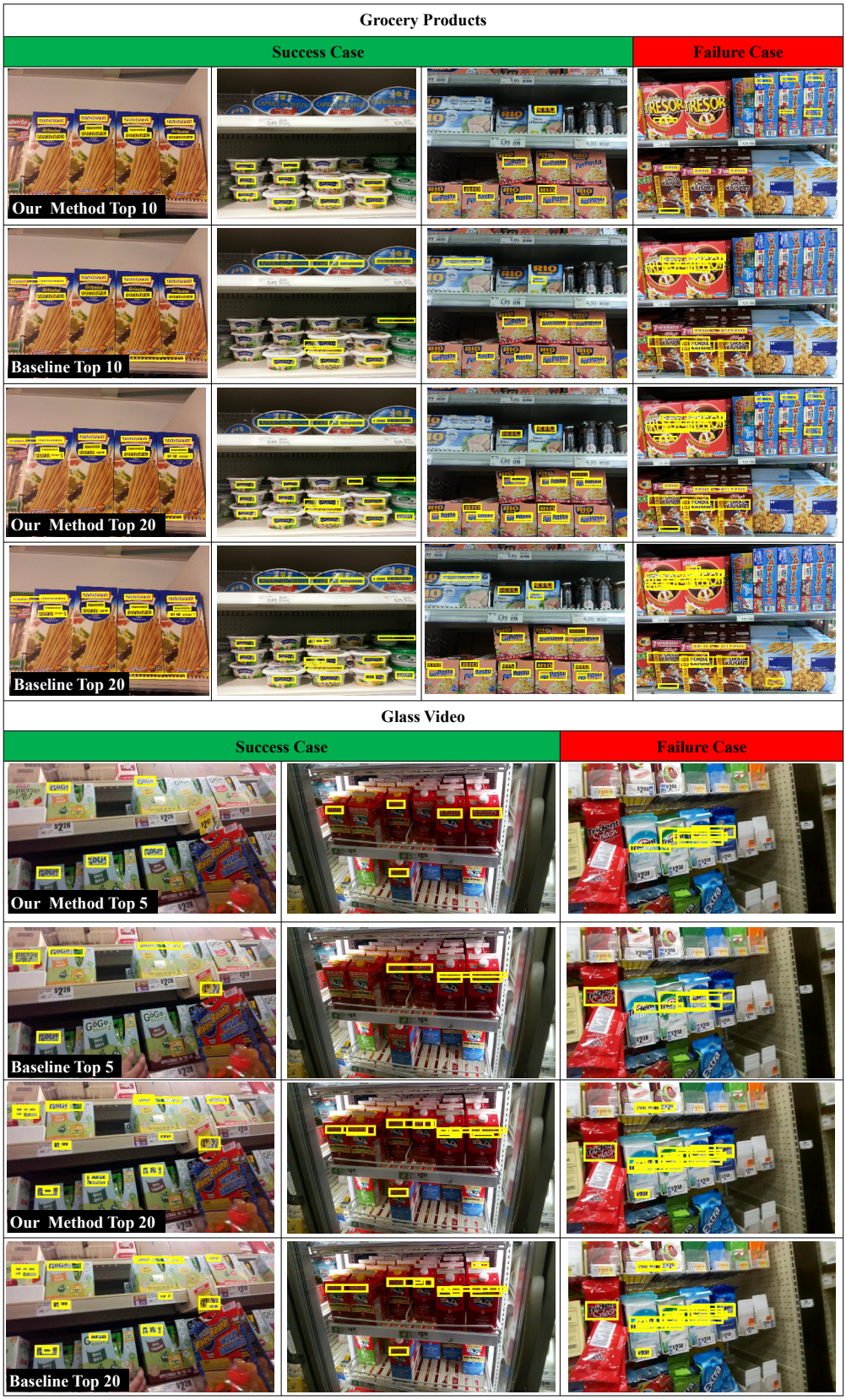


Figure 5. Example text detections for our method and the best baseline [13]. We show the top 10 and top 20 detected boxes for GROCERY PRODUCT and top 5 and top 20 detected boxes for GLASS VIDEO. We show both success (left two columns) and failure cases (rightmost column). **We stress that we show only the top-ranked detections for clearest visualization; additional boxes with lower confidence are also found in each image.** Best viewed on pdf with zoom. See text for discussion.

for the first dataset. We suspect this is because our clustering and matching tasks are correspondingly more difficult on the Glass data.

Our method takes 37 seconds to run in Matlab on one CPU per test case. Since the matching dominates the computation time, it can be easy to reduce, e.g., using k-d trees.

Ablation study Next we demonstrate the usefulness of each component of our method in an ablation study. We compare our "full" method to several variants: 1) Cluster with only visual appearance and ignore size and overlap (VISUAL); 2) Cluster with visual appearance and size but ignore overlap (VIS+SIZ); 3) Cluster with visual appearance and overlap but ignore box size (VIS+AP); 4) Cluster with only size and overlap but ignore visual appearance (SIZE+AP) and 5) Cluster with all three constraints but do not expand the detection set with matching (OURSCLUSTER).

Figure 4(b) shows the results. We see that our full method achieves the best performance. The connected components clustering (OURSCLUSTER) that considers all three proposed constraints (appearance, size, and overlap) outperforms clustering algorithms that use only one or two constraints. This demonstrates that visual appearance alone is not enough to find reliable repetitions. Size is more useful on GROCERY PRODUCTS than in GLASS VIDEO since products in GROCERY PRODUCTS are often fronto-parallel and therefore most duplicate occurrences of text have similar sizes. Considering the impact of our matching stage (OURSFULL vs. OURSCLUSTER), we see the intended improvement in recall at the tail of each plot. The re-detection step finds texts that are ignored by the original text detector. In short, both the grouping and matching stages contribute towards a better ranked and higher recall set of true text detections.

Qualitative examples Finally, we present example text detections in Figure 5 for both datasets. In each part, the first and third rows are our method and the second and fourth rows are the best competing baseline [13]. We show both success cases (marked in green) and failures cases (marked in red).

These image examples help illustrate where and how our repetition prior helps. For example, in the leftmost image of the first row, our method is able to find three repeating words and considers them more confident than the non-repeating candidates. These identified repeating text candidates are in fact true text and are properly localized. On the other hand, the baseline misclassifies part of a product shelf as text, possibly due to its similar appearance with the letter i or l. Such non-text regions are less likely to repeat, and therefore our prior helps disregard that error. Products in GLASS VIDEO (See Figure 5) are often not fronto-parallel. Our method can also handle 3D transformation since SIFT matching tolerates 3D rotations about 30 degrees to 50 degrees.

Overall, our method focuses attention on valid repeating texts and can ignore spurious proposals.

These images also help us analyze our method's failure modes. The rightmost image in the first row contains visually similar text bounding boxes that are poorly localized on vertically oriented texts. Since the ground truth does not include vertical text boxes, those are considered false detections. Unfortunately, these repeating false detections cause our repetition prior to fail. Then the matching step introduces more errors by finding more similar non-text regions. The rightmost image in the third row contains many overlapping text regions, most of which are poorly localized. Although we do not want text bounding boxes to overlap, clustering based on connected components does not guarantee that all pairs of boxes in the same cluster do not overlap; it only ensures each box does not overlap with at least another box in the cluster.

5. Conclusions

Text detection is vital for various real-world applications, but it remains a challenging computer vision problem due to the great variety of scene text and background complexity. In this paper, we study text detection in stores due to both its great potential applicability and complexity. We propose to leverage a repetition prior to improve text detection, and we demonstrate its utility on two challenging datasets of grocery products. The result is a promising step toward text detection in more challenging real-world environments.

Ultimately, we envision a reliable text detection system running in real-time on either mobile or wearable computing platforms, providing users with revolutionary experiences in many real-world environments. In the future, we will explore ways to reduce computation in the text detection pipeline by focusing on the important regions where a user pays attention, leveraging video and other sensory data, such as gaze.

Acknowledgements

This research is supported in part by ONR PECASE N00014-15-1-2291 and a gift from Intel. We thank Swati Rallapalli and Lili Qiu for kindly sharing the GLASS VIDEO dataset.

References

[1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In *ICCV*, 2013.

[2] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with

edge-enhanced maximally stable extremal regions. In *ICIP*, 2011.

[3] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *ICDAR*, 2011.

[4] D. Crandall and R. Kasturi. Robust detection of stylized text events in digital video. In *ICDAR*, 2001.

[5] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 2012.

[6] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, 2010.

[7] N. Ezaki, M. Bulacu, and L. Schomaker. Text detection from natural scene images: towards a system for visually impaired persons. In *ICPR*, 2004.

[8] M. George and C. Floerkemeier. Recognizing products: A per-exemplar multi-label image classification approach. In *ECCV*. 2014.

[9] L. Gómez and D. Karatzas. Multi-script text extraction from natural scenes. In *ICDAR*, 2013.

[10] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[11] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *ECCV*. 2014.

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *arXiv preprint arXiv:1412.1842*, 2014.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *ECCV*. 2014.

[14] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[15] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar. Efficient near-duplicate detection and sub-image retrieval. In *ACM Multimedia*, 2004.

[16] S. Lee, M. S. Cho, K. Jung, and J. H. Kim. Scene text extraction with edge constraint and text collinearity. In *ICPR*, 2010.

[17] R. Lienhart and W. Effelsberg. Automatic text segmentation and text recognition for video indexing. *Multimedia systems*, 2000.

[18] X. Lin, B. Gokturk, B. Sumengen, and D. Vu. Visual search engine for product images. In *Electronic Imaging 2008*, 2008.

[19] X. Liu and J. Samarabandu. An edge-based text region extraction algorithm for indoor mobile robot navigation. In *ICMA*, 2005.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002.

[22] M. Merler, C. Galleguillos, and S. Belongie. Recognizing groceries in situ using in vitro training data. In *CVPR*, 2007.

[23] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

[24] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *ACCV*. 2011.

[25] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *ICDAR*, 2011.

[26] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *CVPR*, 2012.

[27] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *ICCV*, 2013.

[28] P. X. Nguyen, K. Wang, and S. Belongie. Video text detection and recognition: Dataset and benchmark. In *WACV*, 2014.

[29] S. Rallapalli, A. Ganesan, K. Chintalapudi, V. N. Padmanabhan, and L. Qiu. Enabling physical analytics in retail stores using smart glasses. In *ACM MobiCom*, 2014.

[30] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *CVPR*, 2006.

[31] A. Shahab, F. Shafait, and A. Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *ICDAR*, 2011.

[32] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Mobile product image search by automatic query object extraction. In *ECCV*, 2012.

[33] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014.

[34] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi. Visual place recognition with repetitive structures. In *CVPR*, 2013.

[35] S. S. Tsai, D. Chen, V. Chandrasekhar, G. Takacs, N.-M. Cheung, R. Vedantham, R. Grzeszczuk, and B. Girod. Mobile product recognition. In *ACM MM*, 2010.

[36] B. Wang, Z. Li, M. Li, and W.-Y. Ma. Large-scale duplicate detection for web image search. In *ICME*, 2006.

[37] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

[38] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, 2012.

[39] T. Winlock, E. Christiansen, and S. Belongie. Toward real-time grocery detection for the visually impaired. In *CVPRW*, 2010.

[40] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, 2012.

[41] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. *Pattern Analysis and Machine Intelligence*, 2014.

[42] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Transactions on Image Processing*, 2011.

[43] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *CVPR*, 2015.

[44] Y. Zhu, C. Yao, and X. Bai. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, 2015.