Sign Finder Application
Technical Report

Giovanni Fusco[1], Ender Tekin[2] and James M. Coughlan[1]

1 The Smith-Kettlewell Eye Research Institute, San Francisco, CA 94115

2 Waisman Center, University of Wisconsin – Madison, Madison, WI 53705

May 23, 2016

Summary

This technical report describes the algorithms we have implemented for detecting signs. We have implemented and tested the sign detection algorithms for the Exit sign and the Men's and Women's Restroom signs. The algorithms have been implemented using standard libraries. They have been tested on both desktop environments and ported to an Android tablet, using either streaming images from a remote server or from the tablet's internal backward-facing camera.

Detection Algorithm

Overview

The sign detection algorithm is based on a sliding window approach [1], in which a small window is translated (i.e., "slid") over the entire image. For each type of target sign to be detected, the corresponding sliding window has a fixed aspect ratio, and multiple scales are used to capture the signs at different apparent sizes in the image. (For the Exit sign these windows range in size from 18 x 12 to 216 x 144 pixels, while for the Restroom sign the size ranges from 12 x 32 to 120 x 320 pixels.) Each image patch is converted to a visual descriptor, [2], which is fed into a classifier that determines whether an image patch is classified as either containing a sign of interest or not. We search over multiple scales to accommodate a range of viewing distances (with adjacent scales separated by a factor of 1.5). This results in roughly ~10^5 candidate image patches for each image that must be classified as SIGN or NO SIGN.

The overall classifier for each patch is based on a cascade of filters in a boosting paradigm [3, 4], with filters in each stage removing patches from subsequent consideration if they are classified as non-signs; at each successive layer fewer image patches need to be analyzed. At the end, a more discriminative (but computationally intensive) classifier is used to make a final SIGN / NO SIGN decision on the remaining candidate image patches, typically much fewer in number (a few tens of candidates per image).

Different Types of Signs

Currently we have implemented two sign detectors, one for Exit signs and another for Men's and Women's Restroom signs. (The tablet app allows the user to select either type of sign to detect.)

Note that it is possible to modify the app to simultaneously detect multiple types of signs, if desired by the user (e.g., the user may wish to be informed whenever an Exit sign or a Restroom sign of specific gender is

detected). However, more computations would be required in this mode. By having separate modes for each sign, the computational load is reduced, enabling real-time performance and improved responsiveness, and potentially also prolonging the tablet's battery life.

First Stage Classifier

The first stage cascade uses a Gentle Adaboost [5,6] classifier using Local Binary Pattern (LBP) descriptors [7, 8] to describe an image. Implementation of the cascade classifier uses OpenCV's implementation, which uses a set of very simple decision tree classifiers as *weak classifiers*, and combines them to learn a single *strong classifier* that is trained to minimize the number of actual signs that are missed, sacrificing precision (possibly including some non-sign patches) to achieve this high recall rate. (See definition of recall and precision in Results section below.) This ensures that the SIGN of interest is not eliminated at this stage, but is passed on to the next stage which is responsible for finding it (if it exists) among the remaining detections.

Typically a single target in the image will give rise to multiple detections at similar locations and with similar sizes, since the Adaboost classifier is robust to small translations and size changes of the target in the sliding window. Since these multiple detections are redundant, we have implemented a clustering step at the end of the first stage which identifies clusters of rectangles with similar location and size and selects only a single rectangle (detection candidate) from each cluster. This reduces the number of detection candidates that have to be processed in the second stage classifier, which is more selective but also more computationally intensive.

Second Stage Classifier

At the output of the first stage cascade classifier, the number of candidates is reduced to around a few tens per image. The second layer of the cascade uses the Histogram of Oriented Gradients (HoG) [10] as a visual descriptor, which complements the LBP descriptors used in the first layer. Note that HoG is too computationally intensive to apply to all ~10^5 original image patches (which are analyzed by the first layer of the cascade), but the first layer filters out the great majority of these patches. This descriptor is used as input into a support vector machine (SVM, [11]) with an RBF Kernel [4, 12].

The SVM layer classifies all remaining patches as SIGN or NO SIGN. Each classification is also assigned a confidence value between 0 and 1 corresponding to the likelihood of the patch being a SIGN, with 1 being very likely and 0 being very unlikely. Among the patches that are classified as containing the SIGN of interest, only the ones whose likelihood exceeds a set threshold are returned. If no patch is classified as SIGN with a confidence higher than this threshold, no detection is reported. The basic Restroom sign detector responds equally to Men's and Women's signs, but an additional processing stage is used to distinguish between Men's and Women's; a second and final SVM layer is applied after a Restroom sign has been detected in order to determine whether it is a Men's or Women's sign.

Tracking

No detection algorithm is perfectly reliable, which means that in some frames a valid target sign may not be detected, while spurious detections may occur in other frames. In addition, detection performance is often compromised by camera motion blur, which can occur any time the camera is moved, and is especially problematic under low light conditions (such as in indoor environments). These problems pose a challenge to

the development of an effective sign recognition system that is usable by blind and visually impaired persons, who need coherent information about the presence and location of each target of interest.

A standard way to address this problem is to apply a temporal integration stage, such as motion tracking, after the classifier stages. The basic idea behind motion tracking is to combine static appearance cues (which are obtained using the classifiers in individual video frames) with motion cues (which are obtained by integrating information over multiple video frames). We have implemented a motion tracking algorithm based on [13]. The motion of each candidate is tracked and verified via optical flow [14] through consecutive frames, and a valid SIGN is only announced after consistent detections (from the classifier) in three out of the next fifteen consecutive video frames (corresponding to roughly a half-second verification delay for a thirty frame per second video). We note that the choice of this parameter was done heuristically; a less strict criterion (e.g., require two out of every fifteen frames) will reduce delay (which may be preferable in low frame rates), and a more strict criterion (e.g. require three out of every ten frames) will reduce false positives at the expense of more delay.

The target is then tracked in subsequent frames, in which the static appearance-based criteria for selecting target candidates based on the classifiers are relaxed (which allows the possibility of tracking a target that temporarily becomes harder to resolve because of motion blur); we only require another successful validation of the SIGN once every 10 frames. If the SIGN is not validated for 10 consecutive frames of tracking, then this target is deleted from the tracker.

Note that this tracking algorithm has the effect of smoothing out the inevitable false positives (spurious detections) and false negatives (missed detections) that occur with the classifiers. It also allows for multiple targets to be tracked at the same time. Furthermore, by *locking on* to a target, a user is only alerted to each SIGN once upon detection, reducing potential confusion by a blind user (otherwise it may not be clear to the user that the detections correspond to the same object).

An example demonstrating how the tracking algorithm smooths out noisy detections is shown in a sample Exit sign video. The links to videos showing the detection process both with and without tracking turned on are provided here: link to video without tracking and link to video with tracking enabled. When tracking is turned off, there are many false negatives (missed detections) even when the Exit sign is clearly resolved in the video. By contrast, with tracking turned on, the Exit sign is detected continuously (after a very brief delay while the tracker acquires a lock on the target). The important benefit of tracking for a blind or visually impaired person is that the target is tracked continuously while it remains in view, and the accuracy of the location estimate is improved.

Results

Figures 1 and 2 show sample detections in images captured, as well as some missed and false detections. The missed detection shown in Figure 1b (blue rectangle) is an example of a sign that was correctly captured by the first classifier stage (Adaboost) but incorrectly captured by the second classifier stage (SVM). While only partial appearance-based evidence may be available for a sign in a particular image, we note that motion

continuity cues (such as the ones employed in the tracking algorithm) may boost the evidence for such a sign and result in an overall successful detection.

We have measured the algorithm performance objectively using an ROC curve that shows how precision and recall can be traded off each other. (Precision is the fraction of detections that are correct, while recall is the fraction of signs that are detected.) See Fig.'s 3 and 4, which includes results using the tracker compared with when the tracker is turned off. Note that these recall and precision calculations measure the performance of the entire detector (Restroom or Exit), using test videos that are separate from the imagery that was used to train the detectors.

Code

The Sign Finder code is available here:
https://github.com/giofusco/SignFinder

References

[1] Y. Wei; L. Tao, "Efficient histogram-based sliding window," *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 13-18 June 2010, pp.3003–3010.

[2] "Visual Descriptors," *Wikipedia*.

[3] Y. Freund and R. E. Schapire, "*A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*", Journal of Computer and System Sciences, 55(1), 1997, pp.119-139.

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2009.

[5] R. Schapire and Y. Singer. "Improved Boosting Algorithms Using Confidence-rated Predictions" in *Machine Learning*, 1999, pp. 80–91.

[6] R. Schapire and E. Robert. "The boosting approach to machine learning: An overview." *Nonlinear estimation and classification*. Springer New York, 2003, pp. 149–171.

[7] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR), 1994, vol. 1, pp. 582–585.

[8] X. Wang, T. X. Han, and S. Yan. "An HOG-LBP Human Detector with Partial Occlusion Handling", *International Conference on Computer Vision (ICCV)*, 2009

[9] M. Nashvili, "Tutorial | Data Mining With Decision Trees." [Online] Available: http://decisiontrees.net/decision-trees-tutorial/

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886–893.

[11] "Support Vector Machines," *Wikipedia*.

[12] N. Cristianini and J. Shawe-Taylor, "Support Vector and Kernel Methods," in *Intelligent Data Analysis*, M. Berthold and D. J. Hand, Eds. Springer Berlin Heidelberg, 2007, pp. 169–197.

[13] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," International Conference on Pattern Recognition, 2010, pp. 23-26.

[14] R. Szeliski, **Computer vision: algorithms and applications**. Springer Science & Business Media. 2010.

Figure 1. Sample Exit sign detections. From left to right: (a) Successful detection. (b) Blue rectangle shows a false negative (missed detection), indicating a candidate detected by the first (Adaboost) classifier stage that is incorrectly rejected by the second (SVM) classifier stage. (c) False positive (spurious) detection shows a region with texture from building façade.

Figure 2. Sample Restroom sign detections. Note that the current Restroom detector captures both Men's and Women's icons; in the future we will create an additional classifier to distinguish between the two genders. The third image illustrates two false negatives, i.e., two Restroom icons that are not detected.

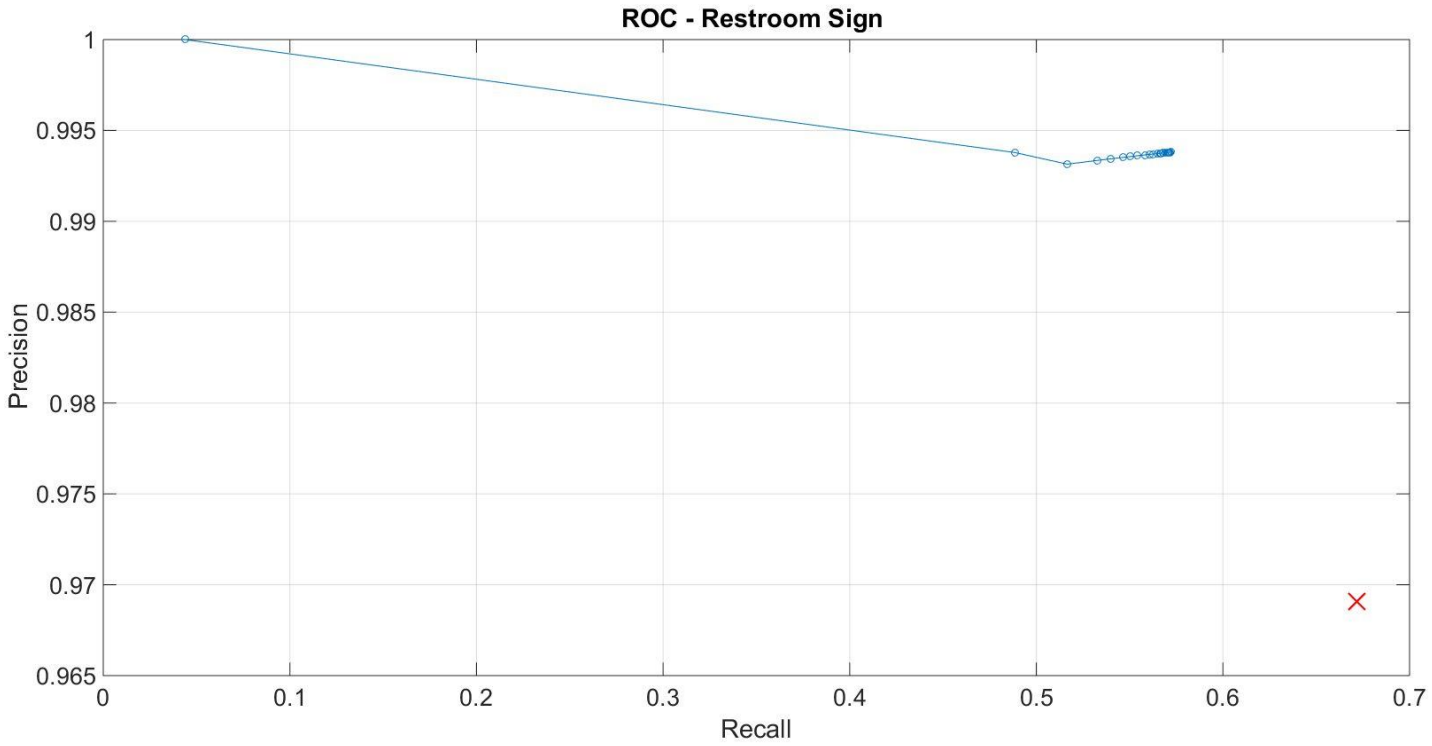

Figure 3. Receiver operating characteristics (precision vs recall) curve for Restroom sign detector: blue curve shows results without tracking, red X shows result with tracking. Note that tracking increases the recall with only a modest decrease in precision.

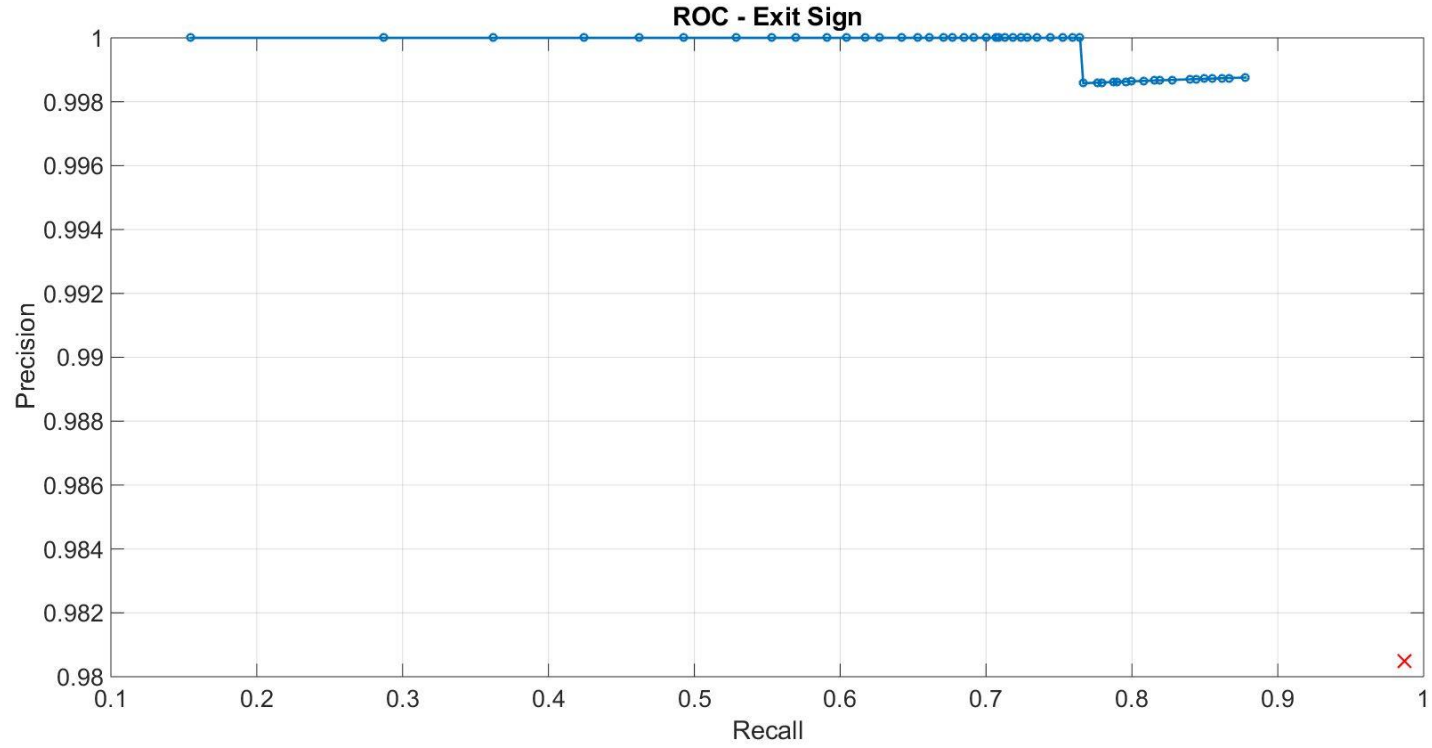

Figure 4. Receiver operating characteristics (precision vs recall) curve for Exit sign detector. See caption for Fig. 3.