Sign Classification using Local and Meta-Features

Marwan A. Mattar, Allen R. Hanson, and Erik G. Learned-Miller

Computer Vision Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
{mmattar, hanson, elm}@cs.umass.edu

Abstract

Our world is populated with visual information that a sighted person makes use of daily. Unfortunately, the visually impaired are deprived of such information, which limits their mobility in unconstrained environments. To help alleviate this we are developing a wearable system [1, 19] that is capable of detecting and recognizing signs in natural scenes. The system is composed of two main components, sign detection and recognition. The sign detector, uses a conditional maximum entropy model to find regions in an image that correspond to a sign. The sign recognizer matches the hypothesized sign regions with sign images in a database. It then uses the match scores to compute meta-features and train a classifier to decide if the most likely sign is correct or if the hypothesized sign region does not belong to a sign in the database. Our data sets encompass a wide range of variability including changes in lighting, orientation and viewing angle. In this paper, we present an overview of the system while while paying particular attention to the recognition component. Tested on 3,975 sign images from two different data sets, the recognition phase achieves accuracies of 99.5% with 35 distinct signs and 92.8% with 65 distinct signs.

1. Introduction

The development of an effective visual information system will significantly improve the degree to which the visually impaired can interact with their environment. It has been argued that a visually impaired individual seeks the same sort of cognitive information that a sighted person does [6]. For example, when a sighted person arrives at a new airport or city they navigate from signs and maps. The visually impaired would also benefit from the information provided by signs. Signs (textual or otherwise) can be seen marking buildings, streets, entrances, floors and myriad other places. In this research, a "sign" or "sign class" is defined as any physical sign, including traffic, government, public, and commercial. This wide variability of signs adds to the complexity of the problem.

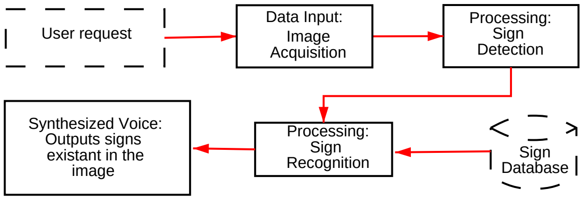
User request
Data Input: Image Acquisition
Processing: Sign Detection
Synthesized Voice: Outputs signs existant in the image
Processing: Sign Recognition
Sign Database

Figure 1. System Layout: An overview of the four modules (solid line) in our system.

The wearable system will be composed of four modules (Figure 1). The first module is a head-mounted camera used to capture an image at the users request. The second module is a sign detector, which takes in the image from the camera and finds regions that correspond to a sign. The third module is a sign recognizer which classifies each image region into one of the signs in its database. Finally, the fourth module, a speech synthesizer, outputs information about the signs found in the image.

Most of the previous techniques for recognizing signs [17, 10, 5] have been limited to standardized traffic signs, using color thresholding as the main method for detection. In other work, Silapachote, Hanson and Weiss [18] built a two-tier hierarchal system that used a color classifier and shape context matching to recognize signs in a similar domain. Several techniques for text detection have been developed [8, 9, 22]. More recently Chen and Yuille [3] developed a visual aid system for the blind that is capable of reading text on various signs.

Unlike most previous work, our system is not limited to recognizing a specific class of signs, such as text or traffic. In this application a "sign" is simply any physical object that displays information that may be helpful for the blind. This system is faced with several challenges that mainly arise from the large variability in the environment. This variability may be caused by the wide range of lighting conditions, different viewing angles, occlusion and clutter, and the broad variation in text, symbolic structure, color and shapes that signs can possess.

The recognition phase is faced with yet another challenging problem. Given that the detector is trained on specific texture features, it produces hypothesized sign regions that may not contain signs or may contain signs that are not in our database. It is the responsibility of the recognizer to ensure that a decision is only made for a specific image region if it contains a sign in the database. False positives come at a high cost for a visually impaired person using this system.

In the following section we briefly overview the detection phase. For more details and experimental results we refer the reader to the original paper [21].

2. Detection Phase

Sign detection is an extremely challenging problem. In this application we aim to detect signs containing a broad set of fonts and color. Our overall approach [21] operates on the assumption that signs belong to a generic class of textures, and we seek to discriminate this class from the many others present in natural images.

When an image is provided to the detector, it is first divided into square patches that are the atomic units for a binary classification decision on whether the patch contains a sign or not (Figure 2). We employ a wide range of features that are based on multi-scale, oriented band-pass filters, and non-linear grating cells. These features have been shown to be effective at detecting signs in unconstrained outdoor images [21]. Once features are calculated at each patch, we classify them as being either sign or background using a conditional random field classifier. After training, classification involves checking whether the probability that an image patch is sign is above a threshold. We then create hypothesized sign regions in the image by running a connected components algorithm on the patches that were classified as sign. Figure 2 shows the results of the sign detector on sample images in the detection database [1].

3. Recognition Phase

The recognition phase is composed of two classifiers. The first classifier computes a match score between the query sign region and each sign class in the database. The second classifier trains on meta-features computed from the match scores to solve a binary classification problem of whether the class with the highest match score is the correct one or whether the query sign region does not belong to any of the classes in the database. Figure 3 shows an overview of the recognition system.

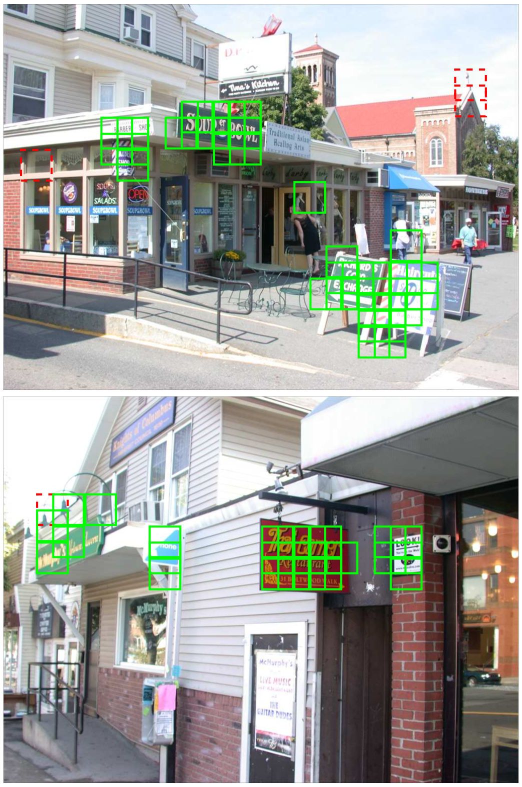

Figure 2. The detector results on two sample images.

3.1. Global and Local Image Features

Image features can be roughly grouped into two categories, local or global. Global features, such as texture descriptors, are computed over the entire image and result in one feature vector per image. On the other hand, local features are computed at multiple points in the image and describe image patches around these points. The result is a set of feature vectors for each image. All the feature vectors have the same dimensionality, but each image produces a different number of features which is dependent on the interest point detector used and image content.

Global features provide a more compact representation of an image which makes it straightforward to use them with

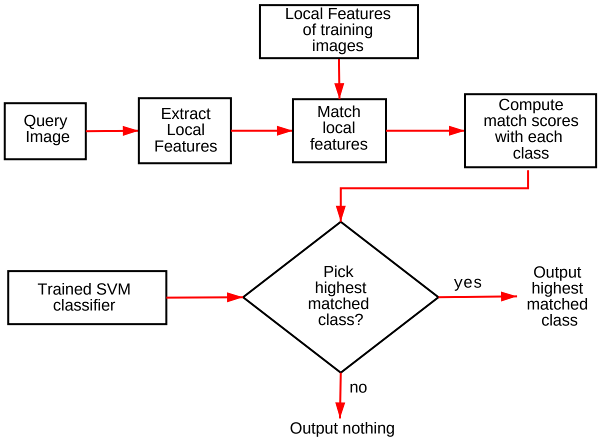
Local Features of training images
Query Image
Extract Local Features
Match local features
Compute match scores with each class
Trained SVM classifier
Pick highest matched class?
yes
Output highest matched class
no
Output nothing

Figure 3. An overview the sign recognition phase.

a standard classification algorithm (e.g. support vector machines). However, local features possess several qualities that make them more suitable for our application. Local features are computed at multiple interest points in an image, and thus are more robust to clutter and occlusion and do not require a segmentation. Given the imperfect nature of the sign detector in its current state, we must account for errors in the outline of the sign. Also, local features have proved to be very successful in numerous object recognition applications [11, 20].

Local feature extraction consists of two components, the interest point detector, and the feature descriptor. The interest point detector finds specific image structures that are considered important. Examples of such structures include corners, which are points where the intensity surface changes in two directions; and blobs, which are patches of relatively constant intensity, distinct from the background. Typically, interest points are computed at multiple scales, and are designed to be stable under image transformations [15]. The feature descriptor produces a compact and robust representation of the image patch around the interest point. Although there are several criteria that can be used to compare detectors [15], such as repeatability and information content, the choice of a specific detector is ultimately dependent on the objects of interest. One is not restricted to a single interest point detector, but may include feature vectors from multiple detectors into the classification scheme [4].

Many interest point detectors [15] and feature descriptors [12] exist in the literature. While the detectors and descriptors are often designed together, the solutions to these problems are independent [12]. Recently, several feature descriptors including Scale Invariant Feature Transform (SIFT) [11], gradient location and orientation histogram (GLOH, extended SIFT descriptor) [12], shape context [2], and steerable filters [7], were evaluated [12]. Results showed that SIFT and GLOH obtained the highest matching accuracy on a test data set focused to test the robustness of the features. Experiments also showed that accuracy rankings for the descriptors was relatively insensitive to the interest point detector used.

3.2. Scale Invariant Feature Transform

Due to its high accuracy in other domains, we decided to use SIFT [11] local features for the recognition component. SIFT uses a Difference of Gaussians (DoG) interest point detector and a histogram of gradient orientations as the feature descriptor. The SIFT algorithm is composed of four main stages: (1) scale-space peak detection; (2) keypoint localization; (3) orientation assignment; (4) keypoint descriptor. In the first stage, potential interest points are found by searching across image location and scale. This is implemented efficiently by finding local peaks in a series of DoG functions. The second stage, fits a model to each candidate point to determine location and scale, and discards any points that are found unstable. The third stage finds the dominant orientation for each keypoint based on its local image patch. All future operations are performed on image data that has been transformed relative to the assigned orientation, location and scale to provide invariance to these transformations. The final stage computes 8 bin histograms of gradient orientations at 16 patches around the interest point, resulting in a 128 dimensional feature vector. The vectors are then normalized and any vectors with small magnitude are discarded. SIFT has been shown to be very effective in numerous object recognition problems [11, 12, 4, 13]. Also, the features are computed over gray scale images which increases their robustness to varying illumination changes, a very useful property for an outdoor sign recognition system.

3.3. Image Similarity Measure

One technique for classification with local features is to find point correspondences between two images. A feature F_A in image A corresponds or matches to a feature F_B in image B if the nearest neighbor (in feature space)of F_A in image B is F_B and the Euclidean distance (in feature space) between them falls below a threshold. The Euclidean distance (in feature space) is usually used with histogram-based descriptors, such as SIFT, while other features such as Differential features are compared using the Mahalanobis distance, because the range of values of their components differ by orders of magnitude.

For our recognition component, we will use the number of point correspondences between two images as our similarity measure. There are two main advantages of this measure. First, SIFT feature matching has been shown to be very robust with respect to image deformation [12]. Second, nearest neighbor search can be implemented efficiently using a k-d-b tree [14] which allows fast classification. Thus, we can define an image similarity measure that is based on the number of matches between the images. Since the number of matches between image I_1 and I_2 is different from the number of matches between image I_2 and I_1, we define our bi-directional image similarity measure as

$$M(I_1, I_2) = \frac{m(I_1, I_2) + m(I_2, I_1)}{2},$$

where $m(A, B)$ is the number of matches between A and B. We refer to this method as "Image Matching."

Sign images that belong to the same class will have similar local features since each class contains the same sign under different viewing conditions. We will use that property to increase our classification accuracy by grouping all the features that belong to the same class into one bag. Thus, we will end up with one bag of keypoints for each class. Now we can match each test image with a bag and produce a match score for each class. We define a new similarity measure between an image I and a class A that contains n images A_i as

$$S(I, A) = \sum_{i=1}^{n} M(I, A_i).$$

We refer to this method as "Feature Grouping." In Section 5 we will show that the Feature Grouping method obtains higher accuracy over the Image Matching method.

3.4. Rejecting Most Likely Class

During classification, when we are presented with a hypothesized sign region we match the region to all the classes to obtain a match score of the region with all the classes. Given the match score for each class, we train a meta-classifier to decide if the class with the highest match score is the correct class or if the test image does not belong to any of the signs in the database. We have observed that when a test image does not belong to any of the signs in the database, the match scores are relatively low and have approximately the same value. Thus, for the classifier we compute meta-features from the match scores that capture that information.

First, we sort the match scores from all the classes in descending order, then we subtract adjacent match scores to get the difference between the scores of the first and second class, the second and third class, etc. However, since the difference between lower ranked classes are insignificant we limit our differences to the top 11 classes resulting in 10 features. We also use the highest match score as another feature along with the probability of that class. We estimate a posterior probability distribution over class labels by simply normalizing the match scores. Thus, the probability that image I belongs to class C_i is defined as

$$p(C_i|I) = \frac{S(I, C_i)}{\sum_{j=1}^{k} S(I, C_j)},$$

where k is the number of classes. We also compute the entropy of the probability distribution over class labels. Entropy is an information-theoretic quantity that measures the uncertainty in a random variable. The entropy $H(X)$ of a random variable X with a probability mass function $p(x)$ is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

Using these 13 features we train a classifier to decide if the class with the highest score is the correct one.

The approach of using the output of a classifier for input to another meta-classifier is similar to an ensemble algorithm known as "stacking." Stacking [16] improves classification accuracy by combining the outputs of multiple component classifiers. It concatenates the probability distributions over class labels produced by each component classifier and uses that as input to a meta-classifier. Stacking can also be used with just one component classifier. In the case of stacking both the component classifiers and the meta-classifier are solving the same n-class problem. However, in our case we use the meta-classifier to solve a different problem than the component classifier.

We adapt the idea of stacking by using the probability distribution as the sole features for the meta-classifier. In experiment 3 of Section 5, we compare a meta-classifier using our choice of features with a meta-classifier using only the posterior probabilities.

4. Data Sets

For our experiments, we used two different data sets (available online [1]) to test the recognition phase. The images of signs were taken from downtown Amherst using a still digital camera (Nikon Coolpix 995) with the automatic white balance on. Manual +/- exposure adjustment along with spot metering was used to control the amount of light projected onto the camera sensor. The following subsections provide more information regarding each of the data sets.

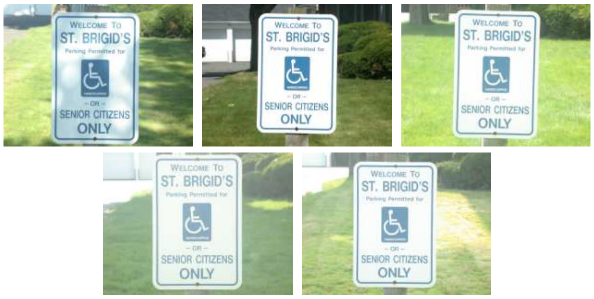

Figure 4. An example of the different lighting conditions captured by the five different images in the 35 sign data set.

4.1. Recognition I: Lighting and Orientation

The purpose of this data set is to test the robustness of the sign recognizer with respect to various illumination changes and in plane rotations. Frontal images of signs were taken at five different times of the day, from sunrise to sunset. See Figure 4 for an example of the different lighting conditions captured in the five images. The images were manually segmented to remove the background. We then rotated each image from $-90°$ to $90°$ at $10°$ intervals, resulting in 95 synthetic images per sign. We synthesized views for 35 different signs resulting in a database of 3325 images.

4.2. Recognition II: Viewing Angle

We compiled a second recognition data set to test the robustness of the recognizer with respect to different viewing angles. This second database contains ten images each of 65 different signs under various viewing angles. Figure 5 provides sample viewing angles of nine signs in the 65-class data set. As before, all the images were manually segmented to remove any background. The different viewing angles where taken by moving the camera around the sign (i.e. the data was not synthesized).

5. Experiments and Results

We performed 3 different experiments to test the various aspects of the recognition phase. The first experiment tested the recognizer on the 35 sign database. The second tested it on the 65 sign database. Finally, the third experiment tested the recognizer on the 65 class database while omitting half of the sign classes from the training data to evaluate how well it performs on ruling out a sign image that does not belong to any of the signs in the training set. Table 1 summarizes the results of the recognizer for the different experiments. The following subsections describe the experimental set up in more detail.

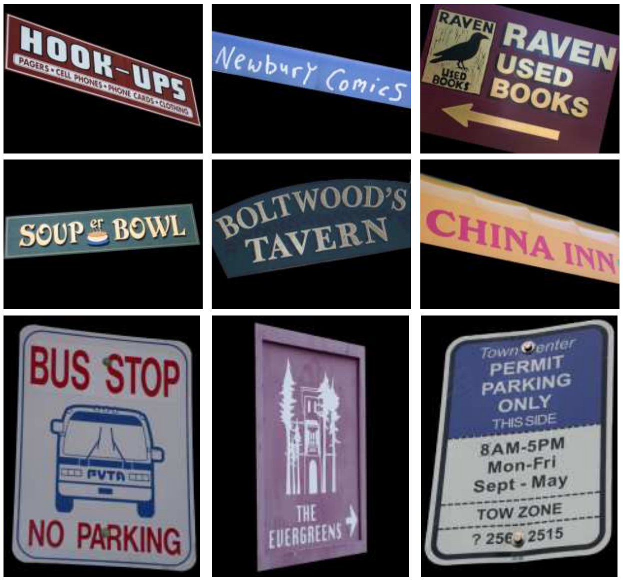

Figure 5. Nine sample images that illustrate the different signs and views in the 65 sign data set.

5.1. Recognition: 35-Class Data Set

This data set contains 3325 sign images from 35 different signs. We performed a leave-one-out experiment using 3325 test images, while using only 175 training instances. Although each sign contains 95 instances, there are only 5 unique ones since the remaining 90 correspond to the synthetic rotations. For our training set we only kept the five unique images from each sign. We compared each test image to 174 training images leaving out the one that corresponds to the rotated version of the test image.

The results of both the image matching and feature grouping were identical and extremely high, achieving a 99.5% accuracy. In fact, the two methods misclassified the same images, most of which were of very poor quality. Figure 6 shows an example of a sign that was classified incorrectly. The identical performance of the two methods implies that if a test image is very different from one training image of the correct class, it is likely to be very different from all of them. This result shows that SIFT features are very robust with respect to illumination changes and in-plane rotation present in this data set.

Experiment	Mean Accuracy	STD (+/-)
35 sign: Image Matching	99.5%	N/A
35 sign: Feature Grouping	99.5%	N/A
65 sign: Image Matching	90.4%	2.75%
65 sign: Feature Grouping	92.8%	2.73%

Table 1. Summary of matching results for the sign recognizer.

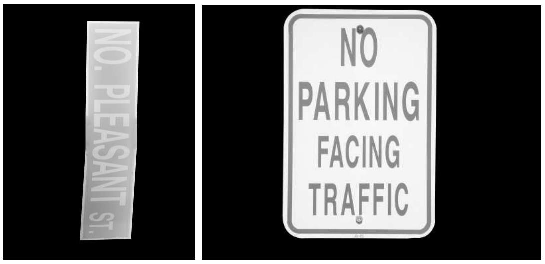

Figure 6. The image on the left is a sample sign that was misclassified in the 35 sign experiment. It was classified as a member of the sign on the right.

5.2. Recognition: 65-Class Data Set

Following the performance of the recognizer on the previous data set, we compiled a second more challenging data set that included a much larger number of sign classes and more variability in the viewing angles. We performed five fold cross validation on the 650 images. Image matching performed 90.4% accuracy, and when we grouped the features by class, the accuracy increased to 92.8%. This 25% reduction in error shows the advantage of the feature grouping method.

5.3. Recognition: 65-Class Data Set with Missing Training Classes

This experiment was intended to test the ability of the recognizer in deciding if the highest matched class is the correct one. We performed ten fold cross validation. On each fold we removed the images from a randomly selected group of 35 signs from the testing set. During training, we obtained the match scores of the classes for a specific training instance. We then computed features from the match scores and then attached a class label of 1 if the training instance belonged to a class in the new test set, 0 otherwise. We then train a Support Vector Machine (SVM) with a linear kernel and use the trained model to classify the test data.

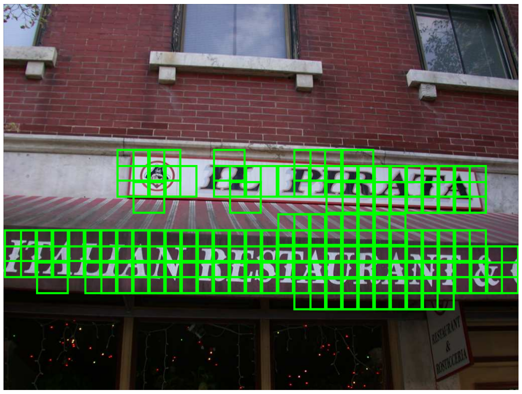

Figure 7. An example where two different signs were grouped together by the detector.

Using our 13 features, the meta-classifier achieved 90.8% accuracy, while using the probability distribution we only achieved 82.25%. These results support our choice of features and show that they contain more useful information than the probability distribution.

This is mainly because the probability distribution can be misleading with respect to the match scores. For example, assume that we have two classes in our database, and we are presented with an image that truly does not belong to either. Assume also that when we match the image with the two classes we get 1 and 0 match scores respectively. Although it is obvious that the match scores are too low for the image to belong to any of the classes, when we normalize, we obtain a 100% probability that the first class is the correct class, which is obviously incorrect. Our features capture most of the relevant information from the match scores which is important for the classification task.

6. Conclusion and Future Work

We have presented algorithms for sign detection and recognition for a wearable system to be used by the blind. The sign detector uses a wide array of features with a conditional random field classifier to find sign regions in the image. The sign recognizer matches each of the hypothesized sign regions with the sign classes in a database and then decides if the highest matched class is the correct one or if the region does not belong to any of the sign classes.

Each of the components perform well on their respective tasks. We are currently in the process of integrating the two components to obtain a complete working system. Figure 8 shows initial sample results of the two components working together. We are also working on improving the accuracy of the individual components. We plan to improve the sign detection rate by using Markov fields with ICM for fast approximate inference. Also the sign recognizer has to be extended to be able to deal with cases were a hypothesized sign region contains more than one sign in the database (Figure 7). Future work also includes adding the final two modules to the system, the head-mounted camera and the voice synthesizer.

Acknowledgments

The authors would particularly like to thank Dimitri Lisin for his numerous revisions of this paper. We would also like to thank Piyanuch Silapachote and Richard Weiss for their helpful suggestions and Jerod Weinman for providing the detection images. This material is based upon work supported by the National Science Foundation under Grant No. IIS-0100851.

References

[1] Vidi project web page. Univ. of Massachusetts Amherst, http://vis-www.cs.umass.edu/projects/vidi/index.html.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transaction on PAMI*, 24(4), 2002.

[3] X. Chen and A.L.Yuille. Detecting and reading text in natural scenes. In *Proceedings of IEEE CVPR, Washington, DC, June 2004*.

[4] G. Dorko and C. Schmid. Object class recognition using discriminative local features. *Submitted to IEEE Transactions on PAMI*, 2004.

[5] A. Escalera, L. Moreno, M. Salichs, and J. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44.

[6] E. T. A. N. D. for Research Working Group on Mobility Aids for the Visually Impaired and Blind. Committee on Vision, National Research Council, National Academy Press, Washington, D.C., 1986.

[7] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on PAMI*, 13(9), 1991.

[8] C. Garcia and X. Apostolidis. Text detection and segmentation in complex color images. In *Proceedings of 2000 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, June.

[9] A. Jain and S. Bhattacharjee. Text segmentation using Gabor filters for automatic document processing. *Machine Vision Applications*, 5:169–184, 1992.

[10] H. X. Liu and B. Ran. Vision-based stop sign detection and recognition system for intelligent vehicles. *Advanced Traffic Management Systems and Vehicle-Highway Automation*, pages 161–166, 2002.

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

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Submitted to IEEE Transactions on PAMI*, 2004.

[13] P. Moreels and P. Perona. Common-frame model for object recognition. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in NIPS 17*. MIT Press, Cambridge, MA, 2005.

[14] J. T. Robinson. The k-d-b-tree: A search structure for large multidimensional receptive field histograms. *Transactions of the Associateion for Computing Machinery*, 1981.

[15] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151–172, 2000.

[16] A. Seewald. *Towards Understanding Stacking - Studies of a General Ensemble Learning Scheme*. PhD thesis, Austrian Research Institute for Artificial Intelligence (FAI), 2003.

[17] L. Sekanina and J. Torresen. Detection of norwegian speed limit signs. In *Proceedings of the European Simulation Multiconference*, pages 337–340, 2002.

[18] P. Silapachote, A. Hanson, and R. Weiss. A hierarchical approach to sign recognition. In *Proc. of IEEE Workshop on Applications of Computer Vision*, pages 22–28, Breckenridge, Colorado, January 2005.

[19] P. Silapachote, J. Weinman, A. Hanson, R. Weiss, and M. Mattar. Automatic sign detection and recognition in natural scenes. In *IEEE Workshop on Computer Vision Applications for the Visually Impaired*, San Diego, June 2005.

[20] T. V. Ferrari and L. Gool. Simultaneous object recognition and segmentation by image exploration. In *Proc. 8th European Conference on Computer Vision*, 2004.

[21] J. Weinman, A. Hanson, and A. McCallum. Sign detection in natural images with conditional random fields. In *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, São Luís, Brazil, Sep. 2004.

[22] V. Wu, R. Manmatha, and E. Riseman. Finding text in images. In *Proceedings of the 2nd ACM International Conference on Digital Libraries, Images, and Multimedia*.

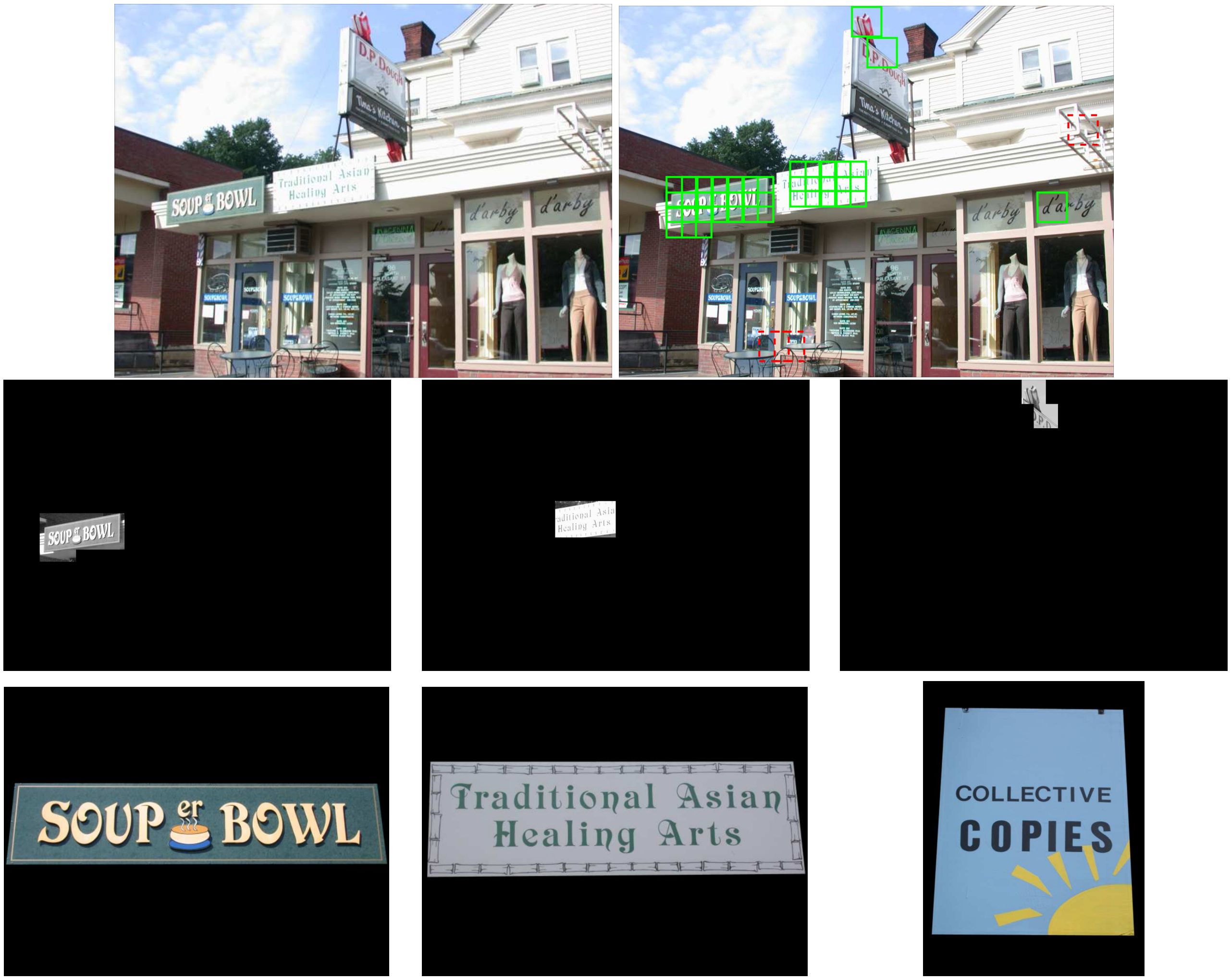

Figure 8. A sample result after integrating the detector and recognizer. The first row contains the initial image and the result of the detector. The second row shows sample results of three connected components and their respective segmentation. The third row shows the result of matching each connected component with the sign classes in the 65 sign data set. The third sign was classified incorrectly because the image region does not belong to any of the signs in the database. However, our trained meta-classifier successfully classified the image region as a negative instance, meaning that it does not belong to any of the classes in the database.