

# Towards Component-based Car Detection

Stanley M. Bileschi<sup>1</sup>, Brian Leung<sup>1</sup>, and Ryan M. Rifkin<sup>1</sup>

Center for Biological and Computational Learning (CBCL),  
Massachusetts Institute of Technology,  
77 Massachusetts Ave  
Cambridge, MA 02139 USA  
{bileschi, leungb, rif}@mit.edu

**Abstract.** Recent results in computer vision have supported the theory that object detectors built in the statistical learning framework can benefit from a two stage learning process, first learning appropriate diagnostic features for the object being trained, and subsequently training an upper-level classifier on the excitation of these part detectors. In this study we develop a hierarchical detection architecture for automobiles. The classifier operates by first locating keypoints in the test image with a well known interest operator. These keypoints are then compared against a corpus of car-specific keypoints learned from the training data. The resulting similarity vector is input into a Support Vector Machine for classification. We compare the performance of our classifier to that of well documented learning algorithms (SVMs and k-Nearest Neighbors) on two separate databases of still images of cars. Our results suggest that part-based detection architectures indeed work well for this class of objects.

## 1 Introduction

While several systems have been developed to robustly detect instances of specific objects in natural images, the detection of generic object classes remains an unsolved problem in computer vision. Evidence from recent studies suggests that accurate and powerful object detectors can be built in a hierarchical manner, with lower level detectors first finding parts of the object, and a higher level classifier combining their outputs. In [1], a robust face detector is engineered using the outputs of detectors for the eyes, nose, mouth, etc as features to an Support Vector Machine (SVM). Similarly in [2] separate classifiers are trained for the head and other parts of a pedestrian, the outputs thereof used for the full detection. The intuition is that a part-based representation of an object is more robust to likely variations, including partial occlusions, lighting differences, and rotations in depth. In this paper, we plan to demonstrate that this hierarchical architecture advantage exists for the class of images of cars.

Toward this end, we construct a system which is capable of learning car parts directly from the data; we then leverage them to construct a robust car detector. We compare performance to a suite of more classical one-level classifiers, such as a standard nearest neighbor or SVM techniques. Furthermore, we use a multitude of image representations, including wavelet and gradient transforms, as well as the standard gray scale pixel values. The system is tested on two separate databases of images of cars (see figures 1 and 2), but is conceptually extensible to any class of objects where object specific features can reasonably be assumed to be constrained spatially. It is expected that this architecture could operate successfully on images of keyboards or faces, but probably not on amorphous objects like trees or buildings.

### 1.1 Previous Work

In the detection of any signal, finding an appropriate representation for the input is an important concern. The types of features useful for discriminating one object class may be less useful in the detection of some other object class. For instance, while wavelets are very useful for the detection of pedestrians, they are no better than simple gray scale pixel values for the detection of faces [3]. It is best to represent the image in terms of features which are discriminative, robust, and easy to compute. While it is possible to let the top level classifier learn from the raw data, performance and speed can typically be increased by dividing the learning process into two stages: one stage to learn object class specific features, and a second stage to learn how to combine the outputs of detectors for these features. Furthermore, when the data is cast into an appropriate representation, learning can be performed with far fewer training examples, perhaps even approaching the goal of 'one-shot learning' [4].

One of the first statistical component-based object detection systems was built to detect pedestrians, as described in [2]. The system used separate SVMs to detect the legs, torso, and head of a pedestrian, and a final, higher level SVM to combine the outputs of the part detectors. The disadvantage of the system

is that pedestrian parts had to be manually selected by a human. What we gleaned from these studies was that detection of certain objects can be made more robust by first learning how to detect object specific features, and then using a standard statistical learning framework to combine the outputs of these diagnostic detectors.

Several different approaches have been proposed for the detection of diagnostic object specific features [5–11]. In [12], the face detection task is modeled as a problem of density estimation in wavelet coefficient space. By factoring the complete wavelet distribution into strongly correlated subsets of coefficients, the detection of the face becomes first a problem of detecting these face subset patterns, then combining the detections. In [1], face parts were automatically learned by building detectors for rectangular patches of synthetic faces. These patches were shrunk or grown to maximize a theoretical bound on the classifier performance. In [10, 11], Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are used, respectively, to learn a representation of the data which captures distinctive features and excludes much of the unwanted within-class variation. While one might hesitate to call these methods part-based methods, since the features used are not spatially localized, the methods undoubtedly learn face-specific features.

One of the drawbacks of using a part-based detection architecture is the computational expenditure involved in searching the entire image for each part individually. Recent research into image saliency and interest operators [13, 14] suggests that by centering each object part around a strongly salient image feature it is possible to quickly locate all part candidates without an expensive search process. This technology has been used to build very robust detectors for specific objects [13, 9]. In this paper we plan to construct a part-based car detector, and illustrate the hierarchical advantage over classifiers which do not first learn object-specific features.

## 2 Interest Operator and SIFT feature descriptor

Extracting distinctive and invariant features from images quickly is an important task for a part-based object detection system. It was decided early in the development of this system to use the interest operator discussed in [13] to make this process rapid and robust. Other options were to use those operators described in [15, 16]. The Lowe Interest Operator detects interest points by locating extrema in

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y), \quad (1)$$

where  $I(x, y)$  is the input image, and  $G(x, y, \sigma)$  is the variable scale gaussian,

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-(x^2+y^2)/2\sigma^2}. \quad (2)$$

Each interest point is parameterized by not only the x, y, and scale, but also a dominant orientation computed from the neighborhood of  $D(x, y, \sigma)$ .

At each keypoint we record the Scale Invariant Feature Transform (SIFT) of the image. The SIFT feature is a soft-bin histogram of orientations, where each bin keys into both an orientation and a spatial location within the image. Details are available in [13].

It has been shown empirically [13, 9] that this combination of keypoint detector and image feature is good for object detection.

### 3 Keypoint-based Car Detector

Our goal is to construct a hierarchical car detector by automatically learning car-specific features and a combination classifier. We chose to center our features on the keypoint detections returned by the algorithm described in [13], because they are robust, computationally efficient, and invariant to a number of common image transforms. We take the following approach. First we extract keypoints from car images in our training set. Each keypoint is a 128-dimensional vector representing a 4x4 array of orientation histograms with 8 bins in each histogram. These keypoints may come from the actual car or from the background in the car training image. We cluster the keypoints into a fixed number,  $k$ , of clusters, using  $k$ -means clustering. Let this set of cluster centroids be the set  $K$ , and let  $K_j$  denote the  $j$ th cluster centroid. Our initial hope was that these clusters would represent semantic car parts. The next step in training the classifier is representing the training data in terms of the expression strength of these parts.

In order to train an SVM classifier, we must be able to convert the image,  $I$ , of the training set into a fixed-length feature vector,  $V_I$ . Let  $T$  be the set of keypoints extracted from image  $I$ . The vector  $V_I$  has elements  $V_I(j)$ ,  $j \in [1, \dots, k]$ , such that

$$V_I(j) = \min_{r \in T} \|K_j - r\| \quad (3)$$

Our fixed-length feature vector is now a  $k$ -dimensional vector of distances between clusters from the training data and keypoints from the image. In general, since there may be more clusters than there are keypoints in an image, a particular keypoint may be associated with more than one keypoint cluster.

Typically, we expect that car images will have smaller distances to many of the clusters, and noncar images will have larger distances. We do not expect this to be a hard rule because we allow background keypoints into the clustering procedure. We do this to avoid manual labeling every keypoint as coming from a car or from the background.

With the described feature representation, we are able to train SVM classifiers. The SVM is trained on positive and negative examples – the positive examples are the same car images from which we extracted car keypoints for clustering<sup>1</sup>.

---

<sup>1</sup> We showed experimentally that the potential system performance improvement from splitting the training set into two subsets for the purposes of clustering car keypoints and training the SVM classifier was offset entirely by the loss in performance due to less data for training.

In order to classify a novel image,  $Z$ , as a car or a noncar, we first extract keypoints from it. We then calculate the  $V_Z$  as before. This vector of minimum distances is then passed to the SVM for classification. This process is repeated for each image in the test database. By sweeping over the SVM threshold we can generate ROC curves, enabling us to compare system performance to other methods. Also, these measures can help us examine the effect of free parameters in the system design.

There are numerous free variables in our approach. The keypoint format depends on the number of scales, orientations, and histogram bins. We chose to keep the parameters at the default values suggested by [13]. These were known to work well for detecting specific objects. We address the parameter  $k$ , the number of clusters, in our experiments.

## 4 Experiments

### 4.1 Databases



**Fig. 1.** Examples of cars and noncars from the training and test set in the Modified UIUC Image Database for Car Detection. All car images are side views of cars where the car is spatially localized in the image. The first and third row are examples of positive and negative training images. The second row contains the crops from the test images of the UIUC Image Database; these images were used as our positive test images. The last row contains examples of negative test images.

**Modified UIUC Image Database for Car Detection.** This database is constructed from the UIUC Image Database for Car Detection [6]. The UIUC Image Database has a training set with 550 side views of cars and 500 noncar images. Each of these gray scale images are  $40 \times 100$ . The cars are roughly of the same build and at the same position, and are subject to possible occlusion and differences in illumination. We used this training set unmodified. The provided test set contains cars to be located in images with a large amount of background.

For our purposes, we cropped out the first car from each of the test images to serve as our positive test set and provided our own set of negative examples. In total, we have 170 side views of cars and 4183 noncars for the test set. See Figure 1 for examples of cars and noncars in the training and test set.



**Fig. 2.** Examples of cars and noncars from the training and test set in the StreetScenes Subset Database.

**StreetScenes Subset Database.** The StreetScenes database contains high resolution (up to  $768 \times 1024$ ) images of cars in a natural environment. The database contains cars, trucks, and buses in many poses, different degrees of occlusion, as well as variations in illumination. In order to simplify the problem, we have decided to concentrate on side views of cars. We have performed semi-automated sorting of the cars in the database by pose in order to extract the side views of cars. We also extract noncars from the StreetScenes images with a similar distribution in sizes. All images are converted into gray scale, extracted at a fixed aspect ratio, and finally scaled down to a fixed image size. Figure 2 shows examples of images from this database. Our training database contains 350 images of cars and 4000 images of noncars. The test database contains 149 images of cars and 4059 images of noncars.

Car recognition on the StreetScenes database is a considerably more difficult task than on the UIUC Image Database. Because the side views of cars were semi-automatically extracted to improve the generality of our methods, errors did occur as shown in Figure 3. Also, the StreetScenes Subset database contains images of bulldozers and buses, making the classification task difficult.

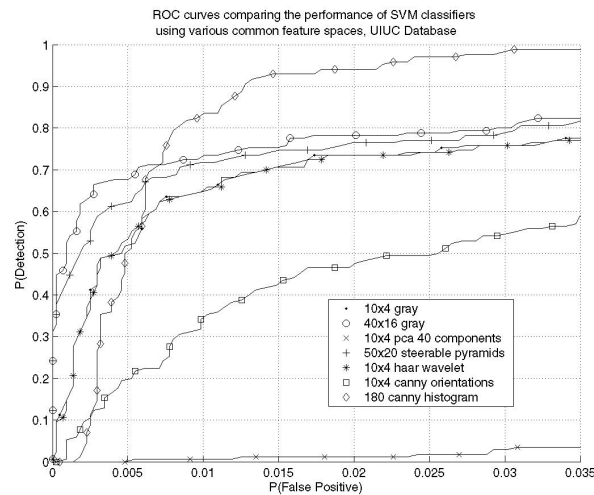
#### 4.2 Global (Non-hierarchical) SVM Classifiers - A survey of feature spaces

For comparison purposes, we perform the learning task on these databases using canonical statistical learning machines, vis. SVM and k-Nearest Neighbor technique. We explored system performance using various common feature spaces and a few novel ones. ROC curves comparing results of different global (non-hierarchical) SVM classifiers are provided in Figures 4 and 5. All the ROC plots



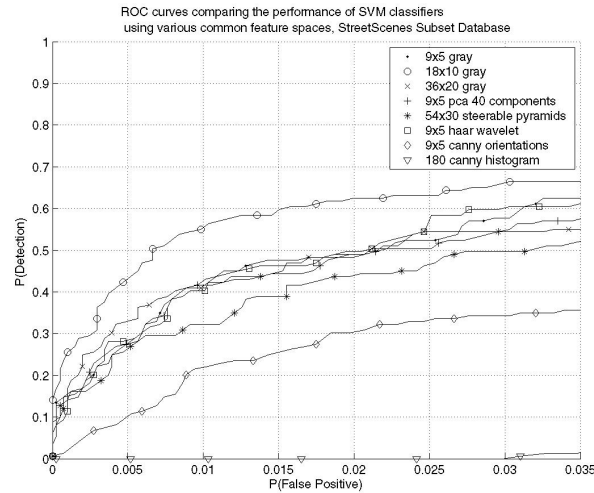
**Fig. 3.** Difficult car examples from the StreetScenes Subset database.

concentrate in the low false positive region, because this is the region believed to be important for the application of car detection in images. All generalizations made in the low false positive region extends to the high false positive region unless otherwise stated. Table 6 gives error rates comparing SVM performance to k-Nearest Neighbors for the UIUC Image Database.



**Fig. 4.** ROC plot comparing performance on the UIUC Image Database of SVM classifiers using various common feature spaces.

The first feature space we explored was the original gray pixel intensities. We applied the following pre-processing steps. First, each gray scale image is scaled down to a common size. We vary this to measure the effect on performance. Histogram equalization is then performed on each image individually to remove variations in image brightness and contrast.



**Fig. 5.** ROC plot comparing performance on the StreetScenes Subset Database of SVM classifiers using various common feature spaces.

In [10, 17], it was found that by using Principle Component Analysis (PCA) the dimensionality of the data could be reduced, while preserving system performance, or possibly even improving it. Along with gray scale features, we tested PCA features in this stage of the experiment. We tried running the experiment with 5, 10, 20, and 40 principle components.

Additional features tested include haar wavelets, and correlations with an overcomplete bank of steerable filters. We also tried the local canny edge orientation at each pixel as a feature space. Perhaps the most complicated feature space we chose to include was a 180-bin histogram of the canny edge orientations, measured at each pixel.

For the sake of comparing different feature spaces, we provide Table 6, the test set error rates for the best SVM classifiers and k-Nearest Neighbor classifiers of each feature space. The error rates for the SVMs are points on the ROC curves of Figure 4. Although ROC curves are more useful for image detection applications, we are not able to generate meaningful curves for the k-Nearest Neighbor classifiers. Nevertheless, we provide this table as part of the survey of the effectiveness of each feature space. From the table, we see that at the chosen operating point, gray value pixels and canny histogram generally work about equally well, and all other features performed worse.

### 4.3 Keypoint-based Car Detector

In this section, we show results from various experiments we performed in arriving at our current system. We show that our system outperforms our global SVMs on the databases used in our experiments.

Test	SVM		1-NN		7-NN		15-NN	
	pos	neg	pos	neg	pos	neg	pos	neg
10x4 gray	90.59%	89.07%	95.88%	75.03%	94.71%	72.88%	95.29 %	70.07 %
20x8 gray	88.82%	86.98%	95.29%	91.36%	96.47%	91.57%	96.47 %	89.19 %
30x12 gray	92.35%	86.61%	95.88%	92.94%	95.88%	92.57%	96.47 %	90.91 %
40x16 gray	92.35%	90.29%	95.29%	94.47%	97.06%	94.54%	97.06 %	93.17 %
10x4 pca 40 comp	63.53%	41.40%	80.00%	53.78%	84.12%	47.73%	89.41 %	43.89 %
10x4 pca 20 comp	62.94%	42.49%	80.00%	51.91%	85.88%	45.62%	88.24 %	42.47 %
10x4 pca 10 comp	55.29%	47.22%	75.88%	51.72%	81.76%	47.84%	85.29 %	45.33 %
10x4 pca 5 comp	51.18%	54.15%	66.47%	47.16%	75.88%	42.47%	79.41 %	38.25 %
10x4 level 2 haar	91.18%	88.81%	94.71%	75.37%	94.71%	72.74%	95.29 %	69.98 %
50x20 steerable pyramid	92.35%	88.76%	94.71%	94.20%	95.88%	93.15%	97.06 %	91.59 %
10x4 canny orientations	90.00%	80.26%	82.35%	57.44%	83.53%	54.65%	81.18 %	53.39 %
180-bin canny histogram	94.12%	98.24%	88.24%	97.46%	94.71%	97.60%	97.65 %	97.51 %

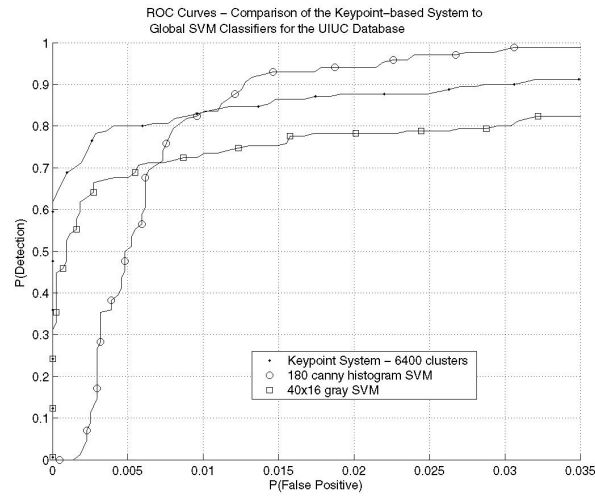
**Fig. 6.** Error rates on the UIUC Database positive and negative test sets. 170 positive examples, 4183 negative examples in test set.

**Comparison to Global SVMs.** In Figures 7 and 8, we compare our system using 6400 clusters of car keypoints to our best global SVMs for the two databases. These figures show that our classifier using the local descriptors at keypoints outperforms the best global SVM classifiers in the low false positive region. Only the global histogram of canny orientations on the UIUC Image Database beats our system in the high false positive region of the ROC curve, a region that is not particularly interesting for the application of object detection in images. Additionally, the poor performance of the global histogram of canny orientations on the Street Scenes Subset Database seems to indicate that this particular feature space is highly sensitive to the particular learning task at hand.

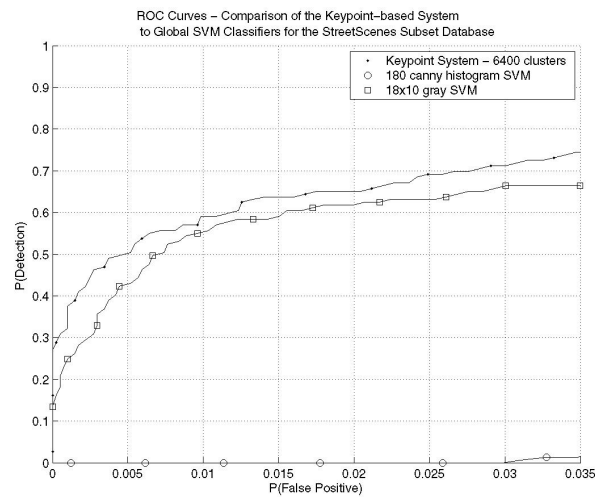
**Experiments with the Number of Clusters.** We show in Figures 9 and 10 that performance generally tends to improve as we increase the number of clusters of car keypoints. There is typically diminishing marginal returns as we increase the number of clusters. On both databases, using any more than 6400 clusters does not achieve enough performance gain to warrant the increase in computational cost, which is dominated by the distance calculations between keypoint clusters and the keypoints in a test image.

We also experimented with the use of the more powerful Gaussian Mixture Models (GMMs) instead of k-Means clustering. With the same number,  $k$ , of clusters, we did not see an improvement in performance.

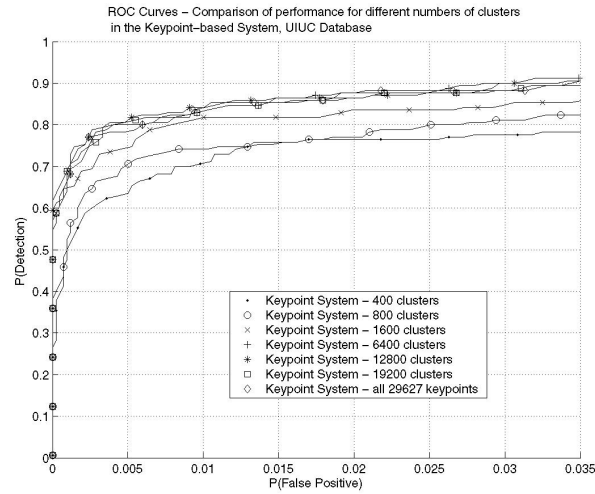
**Experiments using all car keypoints and random car keypoints** Because performance seemed to improve as we increased the number of clusters, we tried using every car keypoint as a cluster centroid. The performance of using all car keypoints is in general very close to the performance when using greater than 6400 clusters of keypoints. Additionally, we explored the expressiveness of each



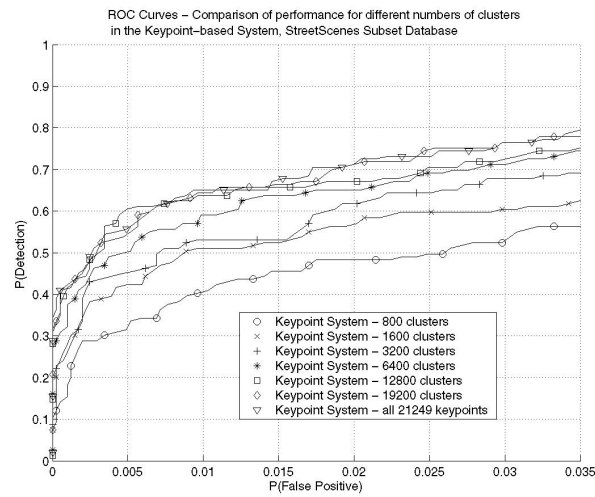
**Fig. 7.** Comparison of the Keypoint-based Car Detector to Global SVM Classifiers on the UIUC Image Database.



**Fig. 8.** Comparison of the Keypoint-based Car Detector to Global SVM Classifiers on the StreetScenes Subset Database.

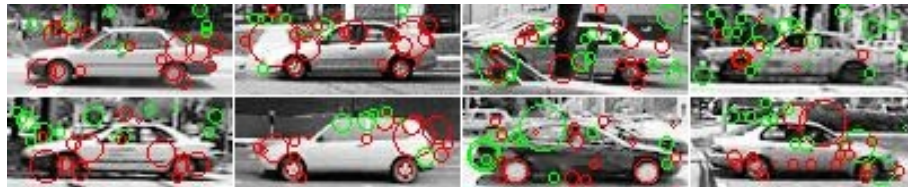


**Fig. 9.** Performance on the UIUC Image Database of the Keypoint-based Car Detector as we vary  $k$ , the number of cluster centroids used.



**Fig. 10.** Performance on the StreetScenes Subset Database of the Keypoint-based Car Detector as we vary  $k$ , the number of cluster centroids used.

individual keypoint in car classification. In Figure 11, we provide training images with keypoints corresponding to negative SVM weights highlighted in red and keypoints corresponding to positive SVM weights in green. Strong expression of a red keypoint (a small distance to that keypoint) indicates the presence of a car, whereas the strong expression of a green keypoint indicates the presence of background. In the training images, keypoints close to the wheels tend to be colored red. This indicates that if a keypoint in a novel image is similar to a wheel keypoint, then it is more likely for the image to be of a car.



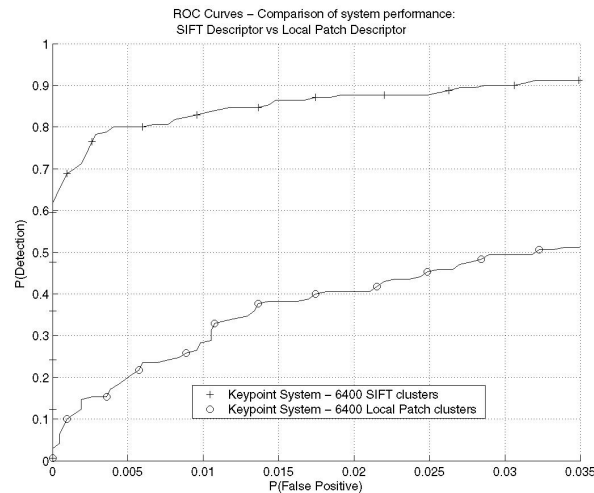
**Fig. 11.** Positive training images from the UIUC database displaying the most expressive keypoints. The size of the keypoint represents the scale at which the keypoint was taken. A keypoint is displayed in red if it is a 'car keypoint'. A keypoint is colored green if it is a 'background keypoint'.

We also tried using random subsets of the set of all car keypoints instead of the cluster centroids. With randomly selected keypoints, accuracy still tends to improve with increasing number of keypoints; however, accuracy becomes highly dependent on the random selection of keypoints. This randomness in the selection of keypoints makes it harder to make any conclusions whether it is better or worse than using clusters of keypoints.

**Local Patch Descriptors vs SIFT Descriptor.** Instead of using the SIFT descriptor, we tried using a local gray pixel descriptor at the scale of the keypoint extraction. In other words, we extract an image patch of uniform resolution at the scale of the detected keypoint and record it as our keypoint feature. We see in Figure 12 that the SIFT descriptor performs better than the local patch descriptors. These arrays of histograms of orientation features seem to better represent the differences between cars and noncars.

## 5 Discussion and Conclusions

In the course of this paper we have outlined the architecture of a hierarchical car detection algorithm, explored how the performance of this classifier is dependent on variations in the architecture, and compared the accuracy of the system to that of non-hierarchical SVMs trained on the same data. We showed that by first extracting appropriate features of the data, and subsequently training a canonical supervised learning algorithm on the expression of these features,



**Fig. 12.** Comparison of the SIFT Descriptor with a Local Patch Descriptor for the keypoint-based car detector. The experiment was performed on the UIUC Image Database.

performance can be boosted. One question the reader may ask is whether or not this system is truly a part-based car detector. For certain the SIFT features learned from the data are diagnostic for cars, as evidenced by the gain in performance over SVMs trained on other reasonable features. Also the SIFT features are spatially constrained; data from the image is only inspected at or near the key point returned by the interest operator. While the clusters we extract in the k-means step represent spatially constrained car specific features, they do not necessarily correspond to what a human would consider semantic car parts <sup>2</sup>.

While the performance of our system is better than that of the best global (non-hierarchical) SVM technique we implemented, it is also considerably more complicated in structure. One might ask if the additional performance is worth the cost. What we have seen empirically is that while a simple SVM and our technique might do equally well at detecting a car which is very similar to the majority of the training database, our method has the advantage when abnormal conditions, such as a partial occlusion or a strong lighting condition, cause part of the car to look dissimilar to what is expected.

One unexpected result in the course of this experiment was that in general it is better to use as many clusters as possible, almost all the way out to the extreme where every key point from the training database is itself its own cluster. In this way our system bears some similarity to a fragmented nearest neighbor technique. Whereas in the nearest neighbor classification regime a test image is compared via a distance metric to every car and non car image in the training

<sup>2</sup> Some clusters regularly locate the wheel of the car, in this case we would consider the cluster to be a semantic car part

data, in our technique every interesting car point is compared to every key point from the training data.

In this paper, we have presented an exploratory study comparing a hierarchical car detection system to the more conventional global approaches. It is important to note that although we use crops from the UIUC Image Database, we do not make direct comparisons to the systems implemented in [6, 18]. Instead, comparison to the score of a flat linear SVM provides a common baseline for both systems.

It is of future interest to the authors to use feature selection methods such as Fisher Linear Discriminant Analysis or Q-alpha [19] to filter out the less useful key points in an attempt to further improve performance. Also we would like to apply this architecture to the detection of cars at other poses, as well as other object classes altogether.

## References

1. Heisele, B., Poggio, T., Pontil, M.: Face detection in still gray images. A.I. memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA (2000)
2. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Volume 23. (April 2001) 349–361
3. Papageorgiou, C., Evgeniou, T., Poggio, T.: A trainable pedestrian detection system (1998)
4. Fei-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning of object categories. In: *Proc. ICCV 2003*. (2003)
5. Weber, M., Welling, W., Perona, P.: Towards automatic discovery of object categories. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. (June 2000)
6. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: *Proceedings of European Conference on Computer Vision*. (2002)
7. Ullman, S., Sali, E.: Object classification using a fragment-based representation. In: *Biologically Motivated Computer Vision* (eds. S.-W. Lee, H. Bulthoff and T. Poggio). (2000) 73–87 (Springer, New York)
8. Viola, P., Jones, M.: Robust real-time object detection. *International Journal of Computer Vision* - to appear (2002)
9. Dorkó, G., Schmid, C.: Selection of scale invariant neighborhoods for object class recognition. In: *Proceedings of the 9th International Conference on Computer Vision, Nice, France*. (2003)
10. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. (1991) 586–591
11. Belhumeur, P., Hespanha, P., Kriegman, D.: Eigenfaces vs fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19** (1997) 711–720
12. Schneiderman, H., Kanade, T.: Probabilistic modeling of local appearance and spatial relationships for object recognition. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara* (1998) 45–51

13. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the International Conference on Computer Vision ICCV, Corfu. (1999) 1150–1157
14. Itti, L., Koch, C.: Learning to detect salient objects in natural scenes using visual attention (1999)
15. Foerstner, W., Gulch, E.: A fast algorithm for detection and precise location of distinct points, corners, and circular features. In: Proc. Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken (1987) 281–305
16. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of the 4th ALVEY vision conference. (1988) 147–151
17. Heisele, B., Serre, T., Pontil, M., Poggio, T.: Component-based face detection. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. Volume 1., Hawaii (2001) 657–662
18. Garg, A., Agarwal, S., Huang, T.S.: Fusion of global and local information for object detection. In: Proceedings of International Conference on Pattern Recognition. (2002)
19. Wolf, L., Shashua, A.: Sparse spectral-based feature selection with side information (2003)