Learning Object Detection from a Small Number of Examples: the Importance of Good Features.

Kobi Levi and Yair Weiss
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{*kobilevi,yweiss*}*@cs.huji.ac.il*

Abstract

Face detection systems have recently achieved high detection rates[11, 8, 5] and real-time performance[11]. However, these methods usually rely on a huge training database (around $5,000$ positive examples for good performance). While such huge databases may be feasible for building a system that detects a single object, it is obviously problematic for scenarios where multiple objects (or multiple views of a single object) need to be detected. Indeed, even for multiview face detection the performance of existing systems is far from satisfactory.

In this work we focus on the problem of learning to detect objects from a small training database. We show that performance depends crucially on the features that are used to represent the objects. Specifically, we show that using local edge orientation histograms (EOH) as features can significantly improve performance compared to the standard linear features used in existing systems. For frontal faces, local orientation histograms enable state of the art performance using only a few hundred training examples. For profile view faces, local orientation histograms enable learning a system that seems to outperform the state of the art in real-time systems even with a small number of training examples.

1 Introduction

In recent years, considerable progress has been made on the problem of frontal face detection [11, 8, 5]. Existing systems achieve roughly 90% detection rate with a tolerable amount of false positives and can operate in real time [11]. One might be tempted, therefore, to declare frontal face detection a "solved" problem.

Despite this first impression, most of the frontal face detection systems require a huge training database to achieve good results. Furthermore, most of these systems can not be easily applied for other types of objects. Even the similar problem of detecting profile view faces, is found to be much harder.

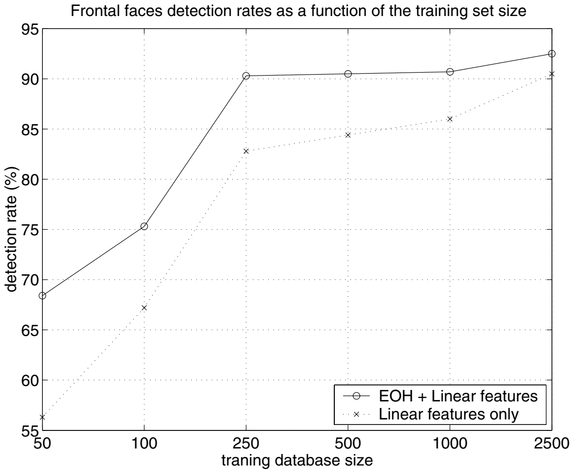


Figure 1: Frontal face detection rates as a function of the size of the training database (with about 100 false positives on the MIT-CMU test set). The two curves show systems trained using an identical learning algorithm (AdaBoost) and with the same training set but with different feature representations of the input patch. In the bottom curve, the features were the ones suggested by Viola and Jones [11], namely edge filter responses at different sizes and locations (see figure 2). The top curve shows the performance where features include the edge filter responses as well as local orientation histograms. We also tested a third type of features, the average intensity of image sub-patches. With these features we got poor results (30.2% with 100 examples and 34% with 250 examples).

In this paper we focus on achieving good results (in terms of detection rates) from small training databases. We would also like to refrain from limiting the system to the frontal face detection problem but rather to find a more generic solution, that could be applied to other types of ob-

1063-6919/04 $20.00 © 2004 **IEEE**

jects.

There are some advantages to learning from small databases. Obviously, the system's ability to achieve good results from few examples is strong evidence for its generalization abilities.

There are also some practical problems when using large databases. Although it takes only a few days to gather thousands of frontal faces, it is much harder to create such large databases for less common objects. Moreover, as mentioned above, one of the major problems in object detection is their diversified appearance from different points of view. Most studies overcome this problem by training a classifier for each point of view [12, 8, 1]. However, this creates many classifiers, and therefore makes it hard to gather large databases for each point of view.

As the size of the database increases, the running time of the training phase also increases. Usually we only require that the running time of the detection phase be reasonable. However, this is not always the case. For example, assuming we want to train a face detector on a customer's site in order to customize the system to the sites' specific conditions. In such cases, not only do we have few examples but we would also like the training phase to be as short as possible.

Learning systems usually consist of two elements, the learning algorithm and the features. In this paper we show that the type of feature has great impact on the results (see figure 1). In particular, we show that using local edge orientation histograms (EOH) as features in the AdaBoost algorithm greatly improves the learning of frontal faces from a small database and enables improving the state-of-the art real-time systems for learning profile faces. We also show that the EOH features are not limited to faces and can significantly improve results on different types of objects such as chairs.

1.1 Previous work

In the last few years, major advances have been made toward a real-time, reliable and accurate face detection system. The most common approach was to use statistical learning tools, mainly from the field of supervised learning, in which the input of the algorithm is a labeled set of examples, containing images of faces as well as non-face images [7, 11, 8].

One of the earliest works on face detection was presented by Rowley, Baluja and Kanade [7]. They trained a neural network with a database of 1050 faces. They manipulated this database (mainly by applying rotations) and their final database included more than 10,000 faces. Later on, they extended their work such that it would be invariant to in-plane rotation[6].

Schneiderman and Kanade[8] use a Naive Bayes approach with wavelets coefficients or eigenvectors as the attributes. Their method achieves excellent results in terms of detection rate, yet is not applicable in real-time. Their work was the first to successfully address the problem of detecting faces from profile point of view (profile faces). Schneiderman and Kanade created a training database containing over 2000 frontal view faces (frontal faces). They extended their database by applying small changes in rotation, scale, position etc. on each of the 2000 faces. All together their final training database contained more than 80,000 examples.

Viola and Jones [11] presented the first highly accurate as well as real-time frontal face detector. In their work they presented a set of very simple features and used the AdaBoost algorithm to build a cascade of classifiers. The cascade data structure decreases the running time of the system by rejecting at the beginning of the cascade most of the areas in the image which do not contain a face. They used more than 5000 examples to train their system and their final detector achieves over 90% detection rate in real-time performance (15 frames per second).

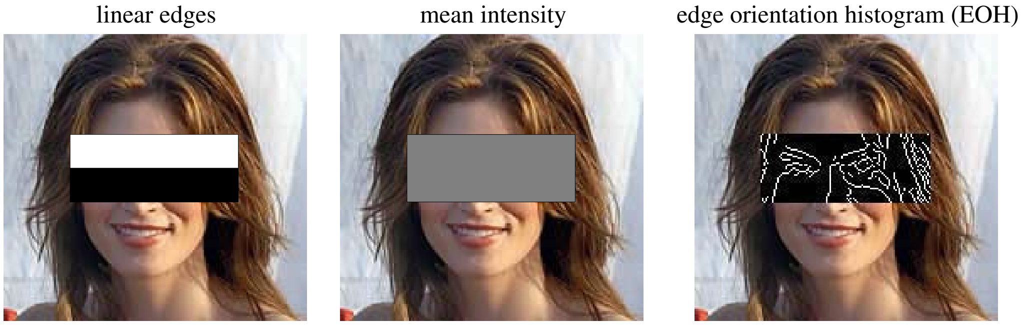
Although many studies deal with the frontal face detection problem, only few have addressed the problem of detecting profile faces [12, 8, 1]. As mentioned before, Schneiderman and Kanade [8] were the first to present a relatively accurate (but not real-time) profile face detector. Some work has been done to extend Viola and Jones' work for the profile face detection problem. Li et al. [1] presented a profile face detector but did not publish the detection rates. Lately, Viola and Jones presented small variations in their feature set and applied it on profile faces.[12]. Despite the fact that their profile detector is less accurate than Schneiderman and Kanade's it does work in real-time.

Orientation histograms have already been identified as an informative tool for various vision tasks. C.Sun and D.Si [10] used orientation histograms to find the symmetry axis in an image. W.Freeman and M.Roth [2] developed a method for hand gesture recognition based on the global orientation histogram of the image. Lowe [4] developed a recognition method which is based on local orientation histograms. However, this method is targeted in scenarios where a specific instance should be recognised rather than in generalisation to the object class.

2 System description

Since our intention is to find visual attributes which will be useful in detection tasks rather than developing a new algorithm, we adopted Viola and Jones' [11] framework which proved itself to be both accurate and fast. We will briefly describe this framework.

In order to detect a face in an image we need to examine each possible sub-window and determine whether it con-



linear edges

mean intensity

edge orientation histogram (EOH)

Figure 2: The three types of features we are comparing in this paper. We show that using local orientation histograms over a subarea of the face greatly improves the performance in comparison to Viola and Jones' linear edges. On the other hand using only mean intensity features greatly decreases the performance.

tains a face or not. In a regular image of 320*240 pixels there are up to 500,000 sub-windows.

In order to reduce the total running time of the system, we need to radically bound the average time that the system spends on each sub-window. For this purpose, Viola and Jones [11] suggested using a cascade of classifiers. The idea of a cascade is based on the observation that we need very few features to create a classifier that accepts almost all (more than 99%) positive examples while rejecting many (20 - 50%) of the false examples. Linking many such classifiers one after the other will create a cascade of classifiers that separates true from false examples almost perfectly. This is done with a very low cost per tested window because most of the non-face sub-windows will be rejected in the early classifiers of the cascade. Viola and Jones use the discrete version of Adaboost [3] to select features and determine their weights. Therefore at stage t of the cascade the classifier is:

$$H_t(x) = sign(\sum_{i=1}^{n} \alpha_i h_i(x)) \tag{1}$$

where $h_i(x)$ is a weak hypothesis and α_i is its weight.

In the Viola and Jones framework, each weak hypothesis is associated with a certain feature:

$$h_j(x) = \begin{cases} 1 & \text{if } F_j(x) \geq T_j \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

where $F_j(x)$ is the value of the feature j and T_j is its corresponding learned threshold. For each such feature, we can create a second weak hypothesis by replacing the condition $F(x) \geq T$ with its dual $F(x) < T$.

We deviate slightly from the framework of Viola and Jones in that we found that we can improve detection rates of frontal faces by using a second cascade, which contains the vertical mirror image of the features in the cascade that was created. Using the original cascade and the mirrored cascade at the same time, we can gain up to 2% more detection rates with the same false detection rate.

3 Features for AdaBoost

Every weak learner in the Viola and Jones framework is a thresholded feature detector. During boosting, a subset of features is chosen from this pool of features. We compared three types of feature pools:

- Linear edge detectors as used originally by Viola and Jones. These features measure the response of linear edge detectors at different subareas of the input image.
- Average intensity detectors. These features simply measure the mean intensity at a subarea of the input image.
- Local edge orientation histograms (EOH).

Figure 2 illustrates the three types of features. While global orientation histograms have been used extensively in a wide range of vision applications (e.g. [2, 10, 9]) the use of localized orientation histograms for object detection, is to the best of our knowledge, novel.

Our reason for using local orientation histograms was our belief that they would give much better generalization than simple linear edge filters. First, the orientation histogram is largely invariant to global illumination changes. Second, local orientation histograms are capable of capturing geometric properties of faces that are difficult to capture with linear edge filters. Figure 3 shows some examples. We will now explain the calculation of the local orientation histograms in detail.

3.1 Preprocessing

We begin by performing edge detection on the image. We use Sobel masks due to their simplicity and efficiency.

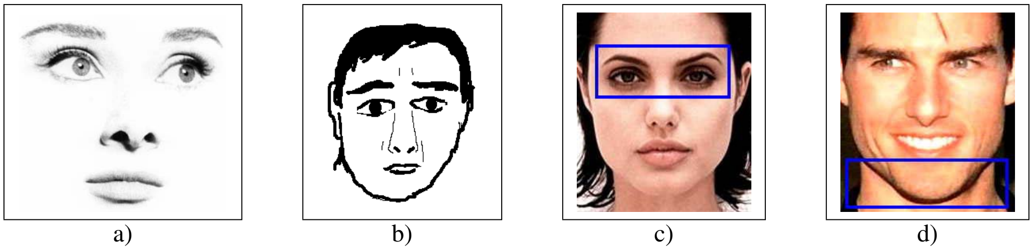


Figure 3: *Global vs. Local statistics of frontal faces. Using the orientation histograms we can gather both global statistics of the object as well as local statistics. (a,b) show some global characteristic of a face. (c,d) show important local features. a) The inner part of the face as a whole includes much more horizontal edges than vertical edges. b) The ratio between vertical and horizontal edges is bounded. c) The area of the eyes includes mainly horizontal edges. d) The chin has more or less the same number of oblique edges on both sides.*

The gradients at the point (x,y) in the image I can be found by convolving Sobel masks with the image.

$$G_x(x,y) = Sobel_x * I(x,y) \tag{3}$$

and

$$G_y(x,y) = Sobel_y * I(x,y) \tag{4}$$

Where $Sobel_x$ and $Sobel_y$ are the x and y Sobel masks respectively. The strength of the edge at the point (x,y)

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{5}$$

In order to ignore noise we threshold G(x,y) such that

$$G'(x,y) = \begin{cases} G(x,y) & \text{if } G(x,y) \geq T \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

A major drawback of Sobel masks is that we have to manually set the value of the threshold T. In our experiments the value of T was set to be between 80 and 110.

The orientation of the edge is

$$\theta(x,y) = \arctan(\frac{G_y(x,y)}{G_x(x,y)}) \tag{7}$$

We then divide the edges into K bins. We denote the value of the k_{th} bin to be

$$\psi_k(x,y) = \begin{cases} G'(x,y) & \text{if } \theta(x,y) \in bin_k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We found that when K values between 4 to 8 the system generalizes well and consumes only a limited amount of memory.

3.2 Edge Orientation Histogram Features

Viola and Jones introduced the *'Integral Image'* [11] and used it to calculate the sum of the pixels for any rectangle in the image at only four table lookup operations. However, the Integral Image can be used on any non negative arrays such as the ψ_k and thus we can calculate equation 9 at only four table lookup operations.

$$E_k(R) = \sum_{(x,y)\in R} \psi_k(x,y) \tag{9}$$

Where R is some sub-window in the image.

We then define a set of features, A, such that:

$$A_{k_1,k_2}(R) = \frac{E_{k_1}(R) + \epsilon}{E_{k_2}(R) + \epsilon} \tag{10}$$

For each R we have $\binom{K}{2}$ features. Assuming that our sub-window is of size $n * n$ the number of features is bounded by $O(n^4\binom{K}{2})$. We add ϵ both to the numerator and to the denominator for smoothing purposes.

Notice that $A_{k_1,k_2}(R) \in \Re$ and therefore each feature yields two potential weak hypotheses $A_{k_1,k_2}(R) \geq T$ and $A_{k_1,k_2}(R) < T$ for some threshold $T \in \Re$. For the first weak hypothesis ($A_{k_1,k_2}(R) \geq T$) these features capture R's were k_1's orientation is dominant in respect to k_2's orientation relation.

3.2.1 Dominant Orientation Features

We are sometimes interested in finding the dominant edge orientation in a specific area rather than the ratio between two different orientations. Therefore we define a slightly different set of features, which measures the ratio between a single orientation and the others, i.e.

$$B_k(R) = \frac{E_k(R) + \epsilon}{\sum_i E_i(R) + \epsilon} \tag{11}$$

The size of this feature group is bounded by $O(K * n^4)$. When there is a dominant edge orientation these features are superior to the previous set of features, A.

3.2.2 Symmetry Features

It has been suggested before (see C.Sun and D.Si[10]), that symmetry plays an important role in object recognition. We therefore define a third set of features which captures symmetry in the image. The symmetry axes are located at the center of the image.

$$Symm(R_1, R_2) = \frac{\sum_{k \in K} |E_k(R_1) - E_k(R_2)|}{sizeof(R_1)} \qquad (12)$$

Where R_1 and R_2 are rectangles of the same size and are positioned at opposite sides of the symmetry axes. The size of this group of features is bounded by $O(n^4)$. The L_1 norm between the two histograms is divided by the size of R_1 such as to preserve the scale invariance property. As for the previous types of features, the symmetry features can be used not just to find symmetry but also to find places were symmetry is absent. For example, the lower and the upper part of the face are not symmetric to each other.

4 Experimental results

4.1 Frontal faces

We collected about 3000 frontal faces mainly from the Internet and from a database that we obtained from Henry Schneiderman. All faces were cropped and rescaled to a size of 24*24 pixels. As false examples we randomly downloaded over 10,000 images containing more than 100,000,000 sub-windows. We used MIT-CMU test set to test our system. This test set contains 130 images with 507 frontal view faces. However, some of these faces are line drawn and they have an interesting influence on the results. As the accuracy of the detector grows, i.e. as the training database size grows, the detection rates of the line drawn faces decrease. This implies that the system learned to separate between line drawn faces and real faces. However, since most previous papers presented their results for the entire database so did we.

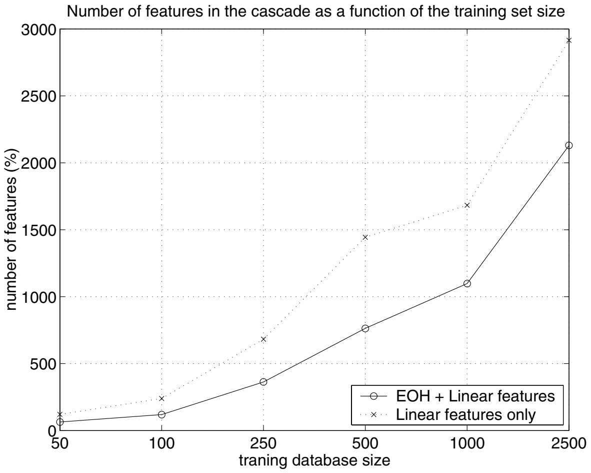


Figure 4: The number of features in the cascade increases as the training database size grows.

4.1.1 The influence of the training database size

In order to demonstrate the influence of the size of the training database we randomly created training databases each of them containing between 10 and 2500 examples. We then trained the system on each of these databases. In figure 1 we show the detection rates (with 100 false positives) as a function of the training database size. Already with only 250 positive examples we can see above 90% detection rate when using both EOH and Viola and Jones' features.

These results show that the type of features that we use has a crucial role in the ability of the system to generalize from a small number of examples. Furthermore, we can see that the difference between the detection rates of the two methods decreases as the database size grows yet it does not vanish. With 2500 features we achieved 92.5% detection with 100 false positives while using Viola and Jones' set of features the system achieved on the same database only 90.5%.

In figure 5 we show some of the features chosen by AdaBoost at the first stages (1-3) of the cascade. We can see that it chooses mainly local features (5 - 10% of the face's area) but also global features. We also see that some of these features are internal while others capture the outline of the face.

A key advantage of using small databases is that the resulting classifier, is usually shorter, and thus faster. In figure 4 we present the number of features that the final classifiers contain. We can see that the cascade created after training on 250 faces contains only 363 features, which are enough for reaching more than 90% detection rate. Our best classifier, which was trained on 2500 faces, is 10 times bigger but introduces only a moderate improvement in the detection rates. Not only does the small database reduce the size of the cascade (in terms of number of features), but moreover, using the EOH features reduces this number almost by half.

Small cascades improve the running time of the detector, as well as tremendously affect the running time of the training phase. We found that training the system on a database of 250 images, is 10 times faster than training it over 2500 images.

In figure 6 we present a ROC curve of our results on a database of 2500 images compared with the results we achieved using only Viola and Jones' features. With 99

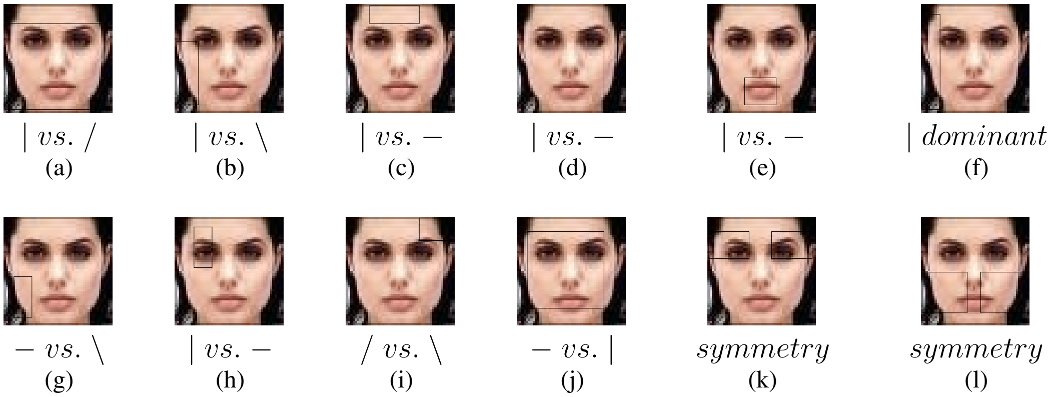


Figure 5: *Examples of the most informative features.* AdaBoost selects these features at the first stages (1-3) of the cascade. The feature in (a) determines that the average face includes more vertical edges than slant edges. In (b) the system learned that the ratio between vertical edges and slant edges is upper bounded by a threshold T. In (j) the system uses the fact that the internal part of the face contains more horizontal edges than vertical edges. Some of these features (see (a) and (c)) are useful to reject non-face images rather than to accept a face.

false detection, we achieved 92.5% detection rate, while using only Viola and Jones' features we achieved only 90.5% detection rate (with the same number of false detections). Viola and Jones [11] trained their system on 4116 face and their vertical mirror images (so that the total number of faces in the training database is 8232). They reported a detection rate of 92.1% detection (with 78 false detections).

As mentioned above, some of the images in the MIT-CMU database are line drawn. Schneiderman and Kanade's [8] results refer to the dataset excluding these images. They achieved a detection of 94.4% with 65 false detections. Our system achieves a detection rate of 92.9% with 56 false detection (on the MIT-CMU database excluding the line drawn images). However, Schneiderman and Kanade's [8] system was trained on more than 80,000 faces and is not a real-time system.

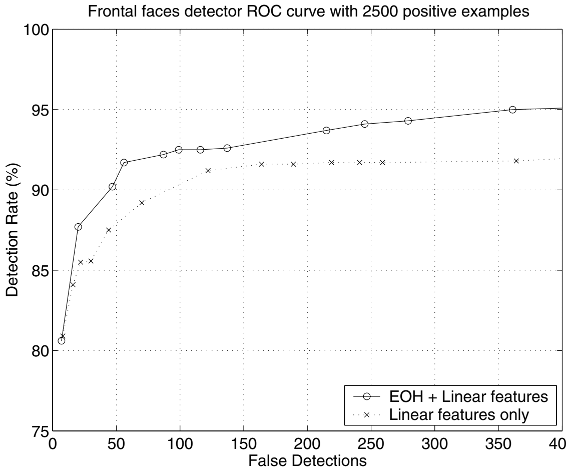


Figure 6: *ROC curves on the 2500 examples database. The advantage of using EOH does not vanish on large training databases. The results of the system when using the EOH is constantly superior to the results we achieved using only Viola and Jones' features.*

4.2 Profile Faces

Our profile faces database contains only 300 faces that were taken from the Internet and from Henry Schneiderman's training database. All faces in the training database are between 3/4 view and full profile. We manually cropped and rescaled these images to a size of 36*36 pixels. As a test set we obtained a database from Schneiderman and Kanade at CMU. This database contains 208 images with 347 faces. This test set was previously used by Schneiderman and Kanade [8] and by Viola and Jones [12] as a test set.

In figure 7 we show the ROC curve of our profile detector along with the results achieved using only Viola and Jones' features. As can be seen, the EOH features significantly improve the results. Our classifier achieves a detection rate of 84.1% with 246 false detections while using only Viola and Jones' features we achieved only 73.9% with 313 false detections.

Not many previous works have addressed the problem of detecting profile faces. In [12] Viola and Jones extended their set of features and included also diagonal filters. Their training database includes 2868 profile faces. In figure 7 we also include their ROC curve. Despite the differences in the training database size, our results are significantly better

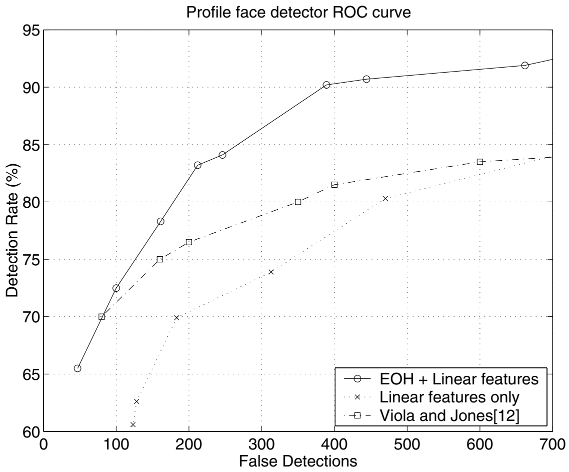


Figure 7: ROC curves of profile face detectors. We compare our detector to Viola and Jones'[12] results. We also compare to the results achieved using only the basic feature set of Viola and Jones on the same database.

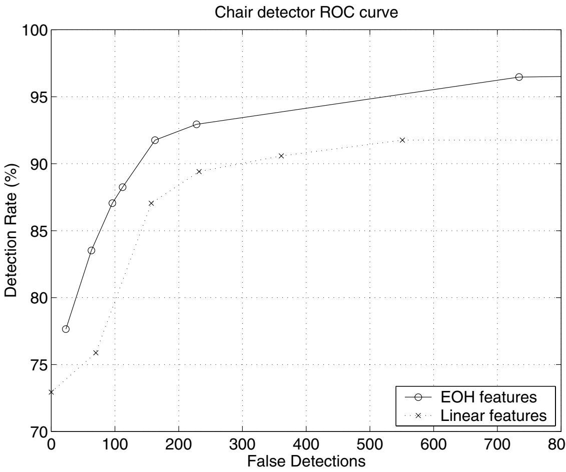


Figure 8: *ROC curves of chair detectors. the improvement achieved by the EOH features is not limited to faces. EOH features perform better also on different tasks such as the chair detector.*

than Viola and Jones' results. With 389 false detection we achieved a detection rate of 90.2% while Viola and Jones achieved a detection rate of 81.5% with 400 false detection.

Schneiderman and Kanade [8] achieved a detection rate of 92.8% with 700 false positives, 86.4% with 91 false positives and 78.6% with 12 false positives. Schneiderman and Kanade's results are slightly better than ours. However, their training database was huge and our system is real-time.

4.3 Chairs

In order to demonstrate that the EOH features are not limited to faces, we used our system to create a chair detector. We collected a set of 185 chair images, mainly from the Internet. We rescaled these images to a size of $17 * 25$ pixels and divided them into two sets: training set which contained 100 images and testing set which contained 85 images. We used the same negative examples as we used in the face detectors.

The results achieved by the chair detector are shown in figure 8. It is clear from the results that the EOH features are very efficient and show superior performance over Viola and Jones' features.

5 Discussion

Despite the impressive progress in the field of object detection, current methods still depend on huge databases to compensate for the vast variety in the appearance of objects.

In this work we have showed that learning from a small database is not only needed but also a more difficult problem. We showed that good features are crucial to the system's ability to learn from a small number of examples. Specifically, we suggested the use of local histograms of orientation as features for object detection tasks. We showed that these histograms significantly improve the ability of the system to learn from small training databases. We also showed that these histograms are well suitable for other detection tasks such as profile face and chair detection.

We achieved excellent results on frontal faces using only 250 examples. We also exceeded state of the art results on profile face detection for real time systems.

In future, we intend to extend this work by finding more visual features, e.g. corner detectors. We would also like to apply this method on other categories of objects such as animals and buildings. We would like to investigate the possibility of combining the EOH with other types of features such as color and texture features in order to create a pool of features that can cope with many types of objects.

References

[1] S. Z. L. et al. Statistical learning of multi-view face detection. *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, May 2002.

[2] W. Freeman and M. Roth. Orientation histogram for hand gesture recognition. In *Int'l Workshop on Automatic Face- and Gesture-Recognition*, 1995.

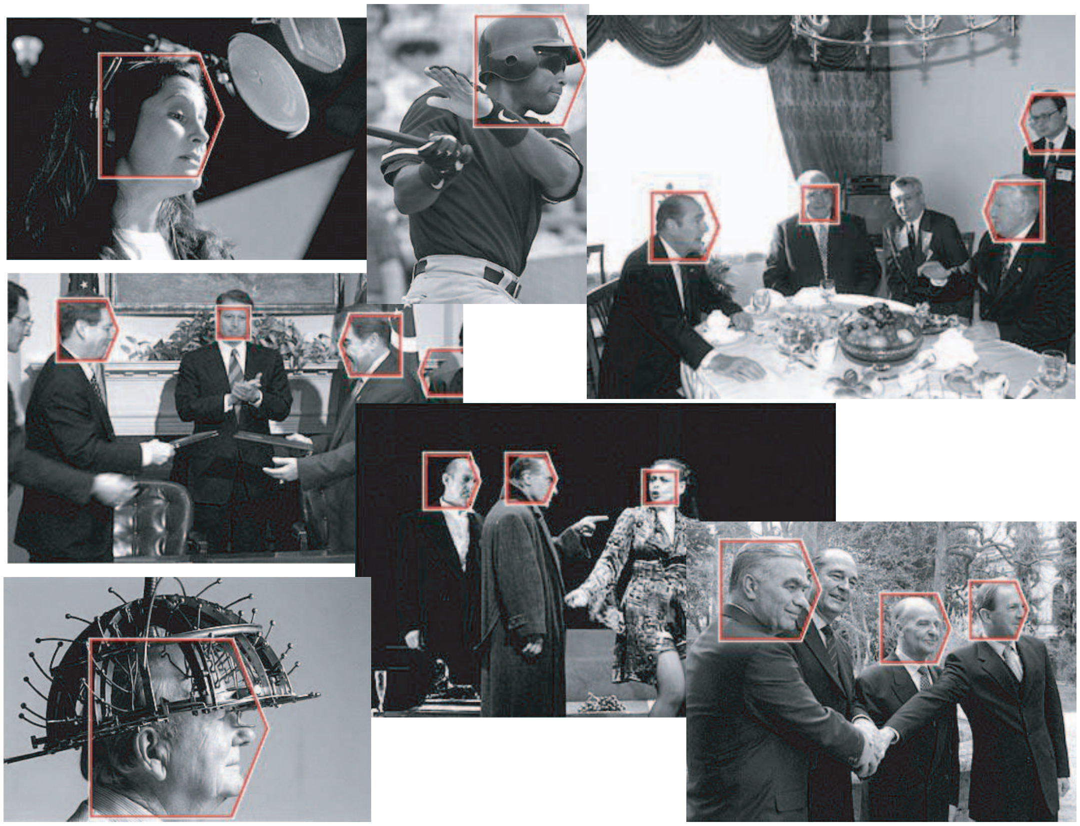


Figure 9: Examples of our frontal and profile detectors

[3] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[4] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150. IEEE Computer Society, 1999.

[5] D. Roth, M. Yang, and N. Ahuja. A SNoW-based face detector. In *NIPS(12)*, pages 855–861, 2000.

[6] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38–44, 1998.

[7] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[8] H. Schneiderman and T. Kanade. A statistical approach to 3d object detection applied to faces and cars. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 746–751, June 2000.

[9] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 750–757, 2003.

[10] C. Sun and D. Si. Fast reflectional symmetry detection using orientation histograms. *Real-Time Imaging*, 5:63–74, 1999.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[12] P. Viola and M. Jones. Fast multi-view face detection. Technical Report TR2003-96, Mitsubishi Electric Research Laboratories., 2003.