Learning a Rare Event Detection Cascade by Direct Feature Selection*

Jianxin Wu James M. Rehg Matthew D. Mullin
College of Computing, Georgia Institute of Technology
{wujx, rehg, mdmullin}@cc.gatech.edu

Abstract

Face detection is a canonical example of a rare event detection problem, in which target patterns occur with much lower frequency than non-targets. Out of millions of face-sized windows in an input image, for example, only a few will typically contain a face. Viola and Jones recently proposed a cascade architecture for face detection which successfully addresses the rare event nature of the task. A central part of their method is a feature selection algorithm based on AdaBoost. We present a novel cascade learning algorithm based on forward feature selection which is two orders of magnitude faster than the Viola-Jones approach and yields classifiers of similar quality. This faster method could be used for more demanding classification tasks, such as on-line learning or searching the space of classifier structures. Our experimental results highlight the dominant role of the feature set in the success of the cascade approach.

1 Introduction

Fast and robust face detection is an important computer vision problem with applications to surveillance, multimedia processing, and HCI. Face detection is often formulated as a search and classification problem: a search strategy generates potential image regions and a classifier determines whether or not they contain a face. A standard approach is brute-force search, in which the image is scanned in raster order and every $n \times n$ window of pixels over multiple image scales is classified [16, 13].

When a brute-force search strategy is used, face detection is a *rare event detection* problem, in the sense that among the millions of image regions, only very few contain faces. The resulting classifier design problem is very challenging: The detection rate must be very high in order to avoid missing any rare events. At the same time, the false positive rate must be very low (e.g. 10^{-6}) in order to dodge the flood of non-events. From the computational standpoint, huge speed-ups are possible if the sparsity of faces in the input set can be exploited. In their seminal work [18], Viola and Jones proposed a face detection method based on a cascade of classifiers, illustrated in figure 1. Each classifier node is designed to reject a portion of the nonface regions and pass all of the faces. Most image regions are rejected quickly, resulting in very fast face detection performance.

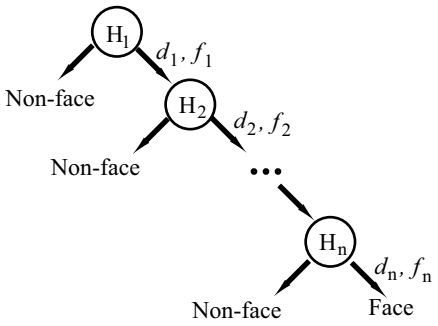


Figure 1: Illustration of the cascade architecture with n nodes.

There are three elements in the Viola-Jones framework: the cascade architecture, a rich over-complete set of rectangle features, and an algorithm based on AdaBoost for constructing ensembles of rectangle features in each classifier node. Much of the recent work on face detection following Viola-Jones has explored alternative boosting algorithms such as FloatBoost [10], GentleBoost [11], and Asymmetric AdaBoost [19]. This paper is motivated by the observation that the AdaBoost feature selection method is an *indirect* way to meet the learning goals of the cascade. It is also an expensive algorithm. For example, weeks of computation are required to produce the final cascade in [18].

In this paper we present a new cascade learning algorithm which uses direct forward feature selection to construct the ensemble classifiers in each node of the cascade. We describe two variations of this approach, a symmetric method which weights false positives and false negatives equally, and an asymmetric approach which is based on a Bayes risk criterion that assigns more weight to false negatives. We demonstrate empirically that our algorithms are two orders of magnitude faster than the Viola-Jones algorithm, and produce cascades which are very close in face detection performance. These faster methods could be used for more demanding classification tasks, such as on-line learning or searching the space of classifier structures. Our results also suggest that a large portion of the effectiveness of the Viola-Jones detector should be attributed to the cascade design and the choice of the feature set.

2 Cascade Architecture for Rare Event Detection

The learning goal for the cascade in figure 1 is the construction of a set of classifiers $\{H_i\}_{i=1}^n$. Each H_i is required to have a very high detection rate, but only a *moderate* false positive rate (e.g. 50%). An input image region is passed from H_i to H_{i+1} if it is classified as a face, otherwise it is rejected. If the $\{H_i\}$ can be constructed to produce *independent* errors, then the overall detection rate d and false positive rate f for the cascade is given by $\prod_{i=1}^n d_i$ and $\prod_{i=1}^n f_i$ respectively. In a hypothetical example, a 20 node cascade with $d_i = 0.999$ and $f_i = 0.5$ would have $d = 0.98$ and $f = 9.6e-7$.

As in [18], the overall cascade learning method in this paper is a stage-wise, greedy

feature selection process. Nodes are constructed sequentially, starting with H_1. Within a node H_i, features are added sequentially to form an ensemble. Following Viola-Jones, the training dataset is manipulated between nodes to encourage independent errors. Each node H_i is trained on all of the positive examples and a subset of the negative examples. In moving from node H_i to H_{i+1} during training, negative examples that were classified successfully by the cascade are discarded and replaced with new ones, using the standard bootstrapping approach from [16]. The difference between our method and Viola-Jones is the feature selection algorithm for the individual nodes.

The cascade architecture in figure 1 should be suitable for other rare event problems, such as network intrusion detection in which an attack constitutes a few packets out of tens of millions. Recent work in that community has also explored a cascade approach [4].

For each node in the cascade architecture, given a training set $\{x_i, y_i\}$, the learning objective is to select a set of weak classifiers $\{h_t\}$ from a total set of F features and combine them into an ensemble H with a high detection rate d and a moderate false positive rate f. A weak classifier is formed from a rectangle feature by applying the feature to the input pattern and thresholding the result.[1] Training a weak classifier corresponds to setting its threshold.

In [18], an algorithm based on AdaBoost trains weak classifiers, adds them to the ensemble, and computes the ensemble weights. AdaBoost [14] is an iterative method for obtaining an ensemble of weak classifiers by evolving a distribution of weights, D_t, over the training data. In the Viola-Jones approach, each iteration t of boosting adds the classifier h_t with the lowest weighted error to the ensemble. After T rounds of boosting, the decision of the ensemble is defined as

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

where the α_t are the standard AdaBoost ensemble weights and θ is the threshold of the ensemble. This threshold is adjusted to meet the detection rate goal. More features are then added if necessary to meet the false positive rate goal. The flowchart for the algorithm is given in figure 2(a).

Viola and Jones compared their cascade detector with the Rowley-Baluja-Kanade detector [13], the Schneiderman-Kanade detector [15], and the Roth-Yang-Ahuja detector [22]. The cascade detector has similar performance and runs much faster.

The process of sequentially adding features which individually minimize the weighted error is at best an indirect way to meet the learning goals for the ensemble. For example, the false positive goal is relatively easy to meet, compared to the detection rate goal which is near 100%. As a consequence, the threshold θ produced by AdaBoost must be discarded in favor of a threshold computed directly from the ensemble performance. Unfortunately, the weight distribution maintained by AdaBoost requires that the complete set of weak classifiers be retrained in each iteration. This is a computationally demanding task which is in the inner loop of the feature selection algorithm.

Beyond these concerns is a more basic question about the cascade learning problem: *What is the role of boosting in forming an effective ensemble?* Our hypothesis is

[1] A feature and its corresponding classifier will be used interchangeably.

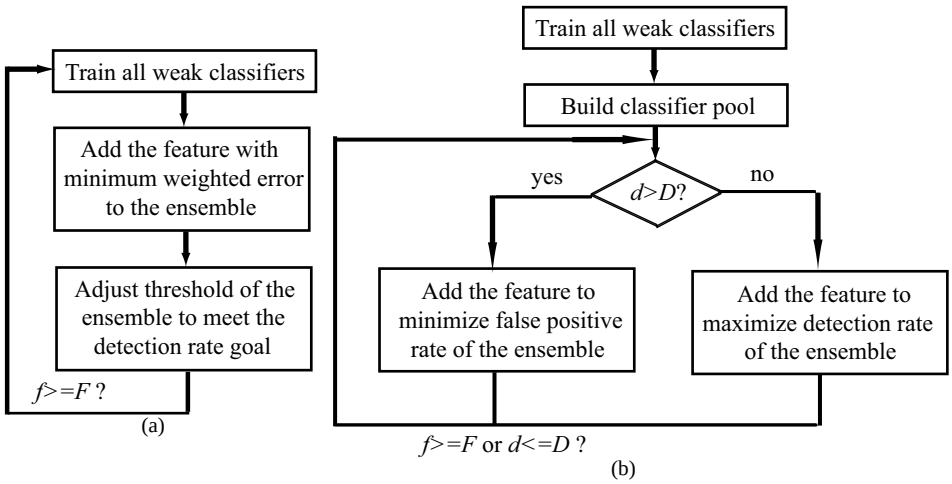


Figure 2: Diagram for training one node in the cascade architecture, (a) is for the original Viola-Jones method, and (b) is for the symmetric direct method. F and D are false positive rate and detection rate goals respectively.

that the overall success of the method depends upon having a sufficiently rich feature set, which defines the space of possible weak classifiers. From this perspective, a failure mode of the algorithm would be the inability to find sufficient features to meet the learning goal. The question then is to what extent boosting helps to avoid this problem. In the following section we describe a simple, direct feature selection algorithm that sheds some light on these issues.

3 Direct Feature Selection Method

3.1 Symmetric feature selection

We propose a new cascade learning algorithm based on forward feature selection [20]. Pseudo-code of the algorithm for building an ensemble classifier for a single node is given in table 1. The corresponding flowchart is illustrated in figure 2(b). The first step in our algorithm is to train each of the weak classifiers to meet the false positive rate goal for the ensemble. A classifier pool is formed from the weak classifiers with the highest detection rates.

The output of each weak classifier on each training data item is collected in a large look-up table. The core algorithm is an exhaustive search over possible classifiers. In each iteration, we consider adding each possible classifier to the ensemble and select the one which makes the largest improvement. The selection criteria directly maximizes the learning objective for the node. The look-up table, in conjunction with majority vote rule, makes this feature search extremely fast.

The resulting algorithm is roughly 100 times faster than Viola-Jones. The key difference is that we train the weak classifiers only once per node, while in the Viola-

1. For node n, we are given the nth bootstrapped training set, the minimum detection rate d_n, and the maximum false positive rate f_n.
2. For every feature, j, train a weak classifier h_j, whose false positive rate is f_n. Sort these weak classifiers according to their detection rate and form a classifier pool P with the first s weak classifiers that have largest detection rates.
3. Initialize the ensemble H to an empty set, i.e. $H \leftarrow \phi$. $t \leftarrow 0$, $d_0 = 0.0$, $f_0 = 1.0$.
4. while $d_t < d_n$ or $f_t > f_n$
   (a) if $d_t < d_n$, then, find the feature k, such that by adding it to H, the ensemble will have largest detection rate d_{t+1}.
   (b) else, find the feature k, such that by adding it to H, the ensemble will have smallest false positive rate f_{t+1}.
   (c) $t \leftarrow t + 1$, $H \leftarrow H \cup \{h_k\}$.
5. The decision of the ensemble classifier is formed by a majority voting of weak classifiers in H, i.e.

$$H(x) = \begin{cases} 1 & \sum_{h_j \in H} h_j(x) \geq \theta \\ 0 & \text{otherwise} \end{cases},$$

   where $\theta = \frac{T}{2}$. Decrease θ if necessary.

Table 1: The symmetric direct feature selection method for building an ensemble classifier at node n in the cascade.

1. Given a training set, maintain a distribution D over it.
2. Select N features using the algorithm in table 1. These features form a set F.
3. Initialize the ensemble classifier to an empty set, i.e. $H \leftarrow \emptyset$.
4. for $i = 1 : N$
   (a) Select the feature k from F that has smallest error ϵ on the training set, weighted over the distribution D.
   (b) Update the distribution D according to the AdaBoost algorithm as in [18].
   (c) Add the feature k and it's associated weight $\alpha_k = -\log \frac{\epsilon}{1-\epsilon}$ to H. And remove the feature k from F.
5. Decision of the ensemble classifier is formed by a weighted average of weak classifiers in H. Decrease the threshold θ until the ensemble reaches the detection rate goal.

Table 2: Weight setting algorithm after feature selection.

Jones method they are trained once for each feature in the cascade. Let T be the training time for weak classifiers[2] and F be the number of features in the final cascade. The learning time for Viola-Jones is roughly FT, which in [18] was on the order of weeks. Let N be the number of nodes in the cascade. Empirically the learning time for our method is $2NT$, which is on the order of hours in our experiments. For the cascade of 32 nodes with 4297 features in [18], the difference in learning time will be dramatic.

The difficulty of the classifier design problem increases with the depth of the cascade, as the non-face patterns selected by bootstrapping become more challenging. A large number of features may be required to achieve the learning objectives when majority vote is used. In this case, a weighted ensemble could be advantageous. Once feature selection has been performed, a variant of the Viola-Jones algorithm can be used to obtain a weighted ensemble. Pseudo-code for this weight setting method is given in table 2.

3.2 Asymmetric feature selection

In the direct feature selection method described in section 3.1, an ensemble classifier with very high detection rate and moderate false positive rate is obtained by selecting features that optimize different criterion (i.e. detection rate and false positive rate) at different stages of the algorithm. The key idea is that the detection rate goal is harder to meet than the false positive rate goal. Thus, an false negative costs more than a false positive. Instead of using the above two-stage optimizing algorithm, there is a natural

[2] In our experiments, T is about 10 minutes.

1. For node n, we are given the nth bootstrapped training set, the minimum detection rate d_n, and the maximum false positive rate f_n.

2. For every feature, j, train a weak classifier h_j, whose false positive rate is f_n. Sort these weak classifiers according to their detection rate and form a classifier pool P with the first s weak classifiers that have largest detection rates.

3. Initialize the ensemble H to an empty set, i.e. $H \leftarrow \phi$. $t \leftarrow 0$, $d_0 = 0.0$, $f_0 = 1.0$.

4. while $d_t < d_n$ or $f_t > f_n$

   (a) Find the feature k, such that by adding it to H, the ensemble will have smallest asymmetric cost. The asymmetric cost of the ensemble is defined as its false positive rate plus λ times its false negative rate, in which λ is the cost ratio.

   (b) $t \leftarrow t + 1$, $H \leftarrow H \cup \{h_k\}$.

   (c) Calculate the new ensemble's detection rate and false positive rate.

5. The decision of the ensemble classifier is formed by a majority voting of weak classifiers in H, i.e.

$$H(x) = \begin{cases} 1 & \sum_{h_j \in H} h_j(x) \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

   where $\theta = \frac{T}{2}$. Decrease θ if necessary.

Table 3: The asymmetric direct feature selection method for building an ensemble classifier.

way to incorporate the difference between false negatives and false positives.

Both the original Viola-Jones algorithm and the direct feature selection algorithm in section 3.1 used a *symmetric* cost function, in which the cost associated with a false negative and a false positive are both 1. If we use an *asymmetric* cost function in which a false positive costs 1, while a false negative costs λ, the algorithm in table 1 can be further simplified. The resulting algorithm, which we call asymmetric direct feature selection, is listed in table 3.[3] The parameter λ is the cost ratio between false negatives and false positives. In [19], Viola and Jones present an alternative asymmetric version of their original feature selection method which incorporates asymmetry into the AdaBoost weights. See section 5 for a detailed comparison.

The asymmetric cost function can be viewed as a Bayes risk criterion [17] which is derived from the original cascade learning problem. It is worth noting that the sym-

[3]The algorithm in table 1 is called the symmetric direct feature selection algorithm accordingly.

metric feature selection algorithm can be treated as a special case of the asymmetric algorithm, in which λ is set to zero if the detection rate goal is met, or infinity if otherwise.

3.3 Stopping Criteria

One critical issue in the implementation of the feature selection algorithm is the choice of the target detection and false positive rates, d_n and f_n, for each node n in the cascade. As specified in the pseudocode (see tables 1 and 3), these target rates determine the stopping criteria for feature selection in a single node. Ideally, we would like to be able to set d_n and f_n in advance, resulting in a completely automatic cascade learning algorithm which yields high quality classifiers. In practice, this is not possible because the best d_n and f_n that can be achieved will vary with n due to the increasing difficulty of the learning task with increasing depth in the cascade. If the d_n and f_n goals are enforced too rigidly, the algorithm will not terminate. This is also true with the Viola-Jones AdaBoost-based algorithm.

In order to ensure that the algorithm terminates, we adopt a manual stopping criteria in our experiments. If the current targets have not been met after 201 features have been added to the ensemble, then the targets are slowly relaxed until they can be achieved. The relaxation schedule is tuned to allow the false positive target to grow more rapidly than the detection rate target.

In addition to the manual stopping criteria, we also explored a fully automatic approach in the case of asymmetric feature selection. In the automatic stopping criteria, each ensemble is trained to use 201 features where the parameters are chosen to minimize the asymmetric cost of the ensemble (as defined in Table 3). This allows an automatic trade-off between false positives and false negatives, controlled by the cost ratio λ. Since we are minimizing a cost function over a fixed number of features, the algorithm will always terminate. Experiments in section 4.3 consider the effects of different values of λ.

4 Experimental Results

4.1 Symmetric feature selection

We conducted three experiments to compare our symmetric feature selection method to the Viola-Jones algorithm. Our training set contained 5000 example face images from the data set in [18] and 5000 initial non-face examples, all of size 24x24. We used approximately 2284 million non-face patches to bootstrap the non-face examples between nodes. For testing purposes we used the MIT+CMU frontal face test set [13] in all experiments. Although many researchers use automatic procedures to evaluate their algorithm, we decided to manually count the missed faces and false positives.[4] When scanning a test image at different scales, the image is re-scaled repeatedly by a factor of 1.25. Post-processing is similar to [18].

[4] We found that the criterion for automatically finding detection errors in [11] was too loose. This criterion yielded higher detection rates and lower false positive rates than manual counting.

In the first experiment we constructed two face detection cascades and compared their performance to the classifier in [18]. One cascade used the symmetric direct feature selection method from table 1. The second cascade used the weight setting algorithm in table 2. The algorithm stopped when it exhausted the set of non-face training examples. The first cascade had 19 nodes and the second cascade had 37 nodes. ROC curves for our two cascades and the Viola-Jones method are depicted in figure 3(a). We constructed our ROC curves by removing nodes from the cascade to generate points with increasing detection and false positive rates. The Viola-Jones ROC curve came from Table 3 of [18].

These curves demonstrate that the test performance of our method is remarkably close to that of Viola-Jones. The direct feature selection algorithm comes within 2% of the Viola-Jones result. The weight setting algorithm produces superior performance for less than 50 false positives and identical performance for more than 200 false positives.

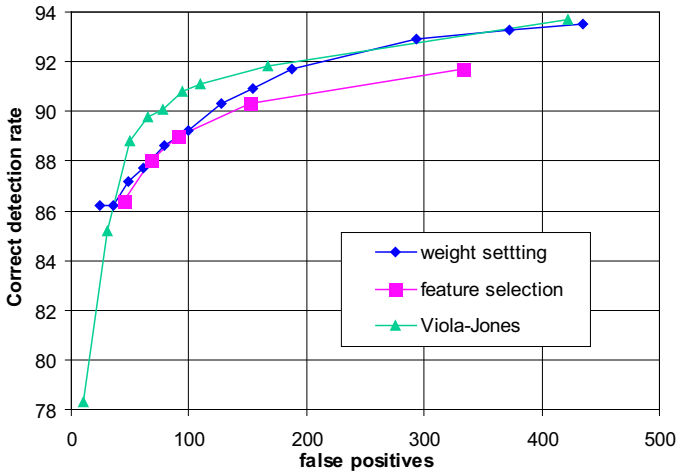
The second experiment explored the ability of the rectangle feature set to meet the detection rate goal for the ensemble on a difficult node. Figure 3(b) shows the false positive and detection rates for the ensemble as a function of the number of features that are added to the 17th node in the direct feature selection cascade from figure 3(a). Even for this difficult learning task, the algorithm can improve the detection rate from 0.6 to 0.9 using only 10 features, without any significant increase in false positive rate. This suggests that the rectangle feature set is sufficiently rich. Our hypothesis is that the strength of this feature set in the context of the cascade architecture is the key to the success of the Viola-Jones approach.

We cannot reproduce all of the details of the experiment in [18] which produced the ROC curve in figure 3(a). To address this limitation, we conducted a third experiment in which the learning conditions for both algorithms were carefully controlled. In particular, the feature set and training data were identical. The experiment focused on learning the 17th node in a cascade. Figure 4 shows ROC curves for the Viola-Jones, direct feature selection, and weight setting methods. Unlike the previous ROC curves, these curves show the performance of the node in isolation using a hold-out set of the training data. These curves reinforce the similarity in the performance of our method compared to Viola-Jones. In particular, the false positive target for this node was 0.8, and the curves are quite close at this point. It's interesting that weight setting does not improve the performance in this case.

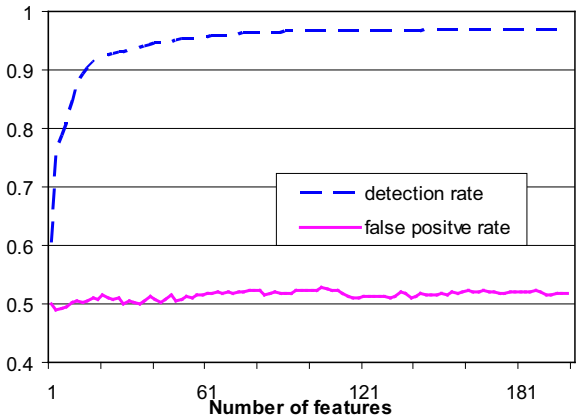
4.2 Feature set size

We conducted two experiments to examine the effect of feature set size on the detection performance. Instead of using the full feature set which contains all possible rectangle features defined in a 24 by 24 pixel window, we sampled 10% of the features.

The first experiment build a cascade with 36 nodes, using the symmetric feature selection algorithm in table 1 and the training set described in section 4.1. The features are sampled by selecting the first feature of every 10 features. The ROC curve on the MIT+CMU test set of this cascade and the 2 cascades described in section 4.1 are depicted in figure 5. The performance of the cascade trained from the 10% sub-sampled feature set is very close to those cascades trained by the full feature set.



(a)



(b)

Figure 3: Experimental Results. (a) is ROC curves on MIT+CMU of the symmetric feature selection method and the Viola-Jones method and (b) is trend of detection and false positive rates when more features are combined in one node.

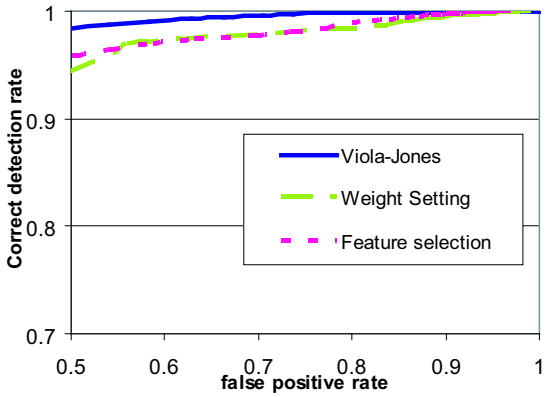


Figure 4: ROC curve of controlled experiment.

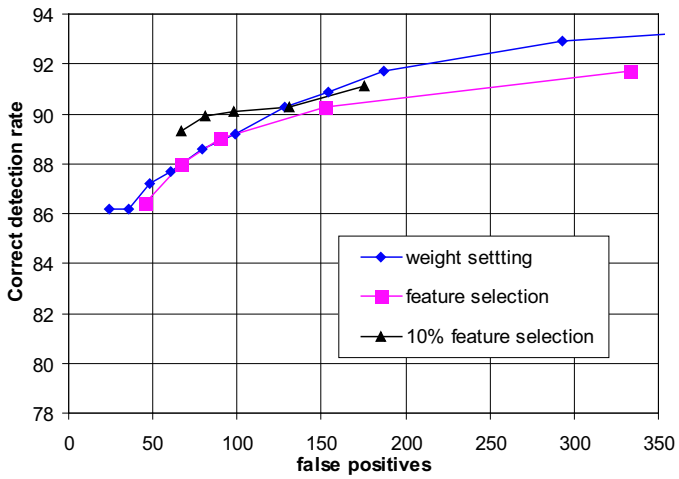


Figure 5: ROC curves on MIT+CMU comparing cascades trained on the full feature set and a 10% down-sampled feature set.

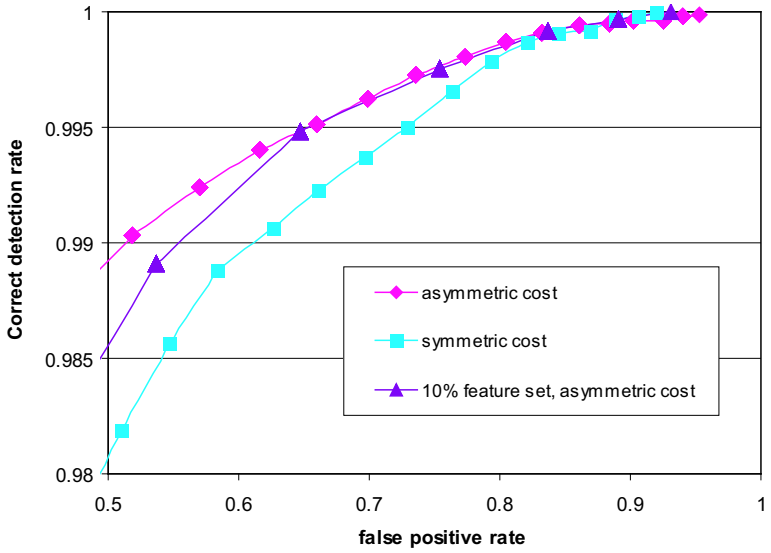


Figure 6: ROC curve on a holdout set of controlled experiment to compare the effect of down-sampled feature set and asymmetric feature selection.

The second experiment compared the abilities to achieve the learning objective in one node. We used the same training set as in figure 4 and examined the performance of different algorithms on a hold-out validation set. Figure 6 shows ROC curves for 3 different ensemble classifiers trained by the asymmetric feature selection algorithm with full feature set, the symmetric algorithm with full feature set, and the asymmetric algorithm with a 10% down-sampled feature set. The cost ratio λ is set to 10 in this experiment. Comparing the two ROC curves trained by asymmetric feature selection algorithm, it is clear that although the feature set size is greatly reduced, the performance is essentially unchanged.

These experiments show that even if the feature set is reduced to 10% of its original size, the difference in detection performance is negligible. This is more evidence for the richness and diversity of the feature set of the Viola-Jones method.

4.3 Asymmetric feature selection

The three experiments in this section were designed to examine the performance of the asymmetric feature selection algorithm.

The first experiment compared performance of symmetric and asymmetric versions of the proposed method on a single node. The experimental setup was the same as the second experiment in section 4.2 and the result is shown in figure 6. Using the full feature set, the asymmetric feature selection algorithm's performance is consistently better than the symmetric one.

In the second experiment, we trained a cascade with 35 nodes using the asymmetric

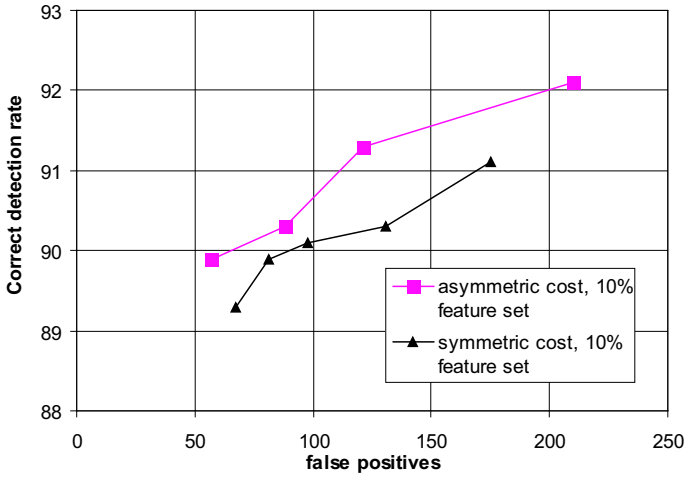


Figure 7: ROC curves of the asymmetric and symmetric feature selection algorithm on MIT+CMU, using a 10% down-sampled feature set.

feature selection method and the 10% down-sampled feature set, following the experimental setup in section 4.2. The cost ratio λ was set to 10 in this experiment. This ROC curve is tested against the symmetric ROC curve from section 4.2 on the MIT+CMU test set. The result is shown in figure 7. The performance of these two cascades are close, but the asymmetric feature selection algorithm's performance is consistently better than the symmetric one.

The third experiment examines the effect of different values of λ. Four cascades were trained using the values 2,5,10 and 25. The ROC curves are shown in figure 8. These cascades were trained using the automatic stopping criterion for the asymmetric algorithm proposed in section 3.3. The ROC curves were generated on a holdout set consisting of 5832 face patches and 6000 non-face patches. Only ten nodes were trained for each cascade; improvements from additional nodes would occur in the region of very small false positive rate, which would not be reflected well with this validation set. We cannot directly compare the curves in this graph with the other ROC curves for cascades because of the use of the validation set, rather than manual evaluation on the MIT+CMU face set.

We can see that there is a definite correlation between the value of the cost ratio and the performance on the ROC curve. The larger values of λ result in better ROC curves, though there is some overlap in the area of low false positive rate. However, we cannot make the cost ratio too large. Experiments with $\lambda = 100$ resulted in the stopping criterion choosing ensembles with perfect detection rate and false positive rates greater than 99%. Each additional node then adds very little discriminative power, and the training sets for each successive node are nearly identical, resulting in very similar (and poor) ensembles. Perhaps using a larger training set would allow for the use of large cost ratios. Allowing the cost ratio to change from node to node, using large

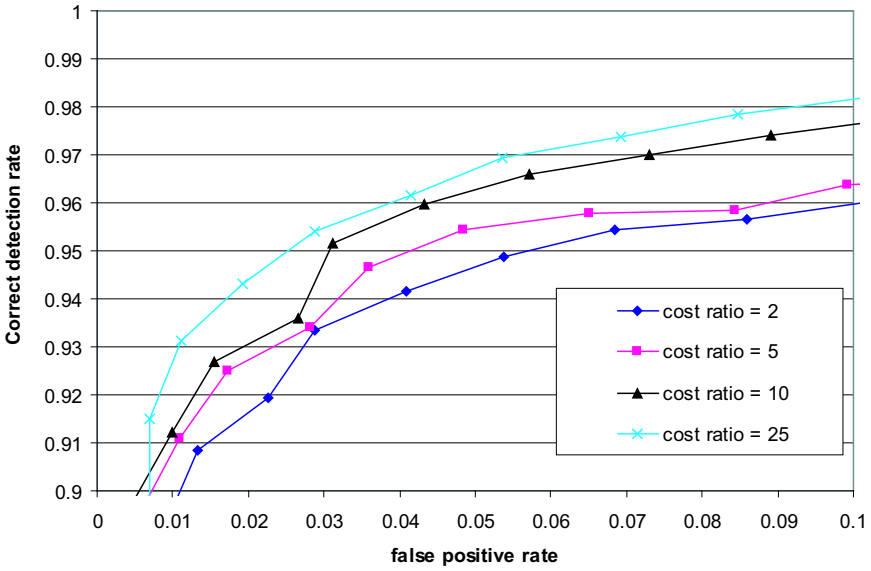


Figure 8: ROC curves of the asymmetric feature selection algorithm on a holdout set using different cost ratios.

values in the first nodes and decreasing it as the classification problem grows more difficult, may also alleviate this problem.

5 Related Work

A preliminary version of this paper was submitted to *NIPS 2003*. In comparison to this earlier version, we have introduced a new *asymmetric* version of our direct feature selection method which gives significantly better performance. In addition, we introduced an automatic stopping criterion. Finally we have conducted additional experiments which study the effect of feature set size on performance. Sections 3.2, 3.3, 4.2, and 4.3 are all new to this paper.

A survey of face detection methods can be found in [21]. We restrict our attention here to frontal face detection algorithms related to the cascade idea. The neural network-based detector of Rowley et. al. [13] incorporated a manually-designed two node cascade. Other cascade structures have been constructed for SVM classifiers. In [12], a set of reduced set vectors is calculated from the support vectors. Each reduced set vector can be interpreted as a face or anti-face template. Since these reduced set vectors are applied *sequentially* to the input pattern, they can be viewed as nodes in a cascade. An alternative cascade framework for SVM classifiers is proposed by Heisele et. al. in [5]. Based on different assumptions, Keren et al. proposed another object detection method which consists of a series of anti-face templates [7]. Carmichael and

Hebert propose a hierarchical strategy for detecting chairs at different orientations and scales [3].

Following [18], several authors have developed alternative boosting algorithms for feature selection. Li et al. incorporated floating search into the AdaBoost algorithm (FloatBoost) and proposed some new features for detecting multi-view faces [10]. Lienhart et al. [11] experimentally evaluated different boosting algorithms and different weak classifiers. Their results showed that Gentle AdaBoost and CART decision trees had the best performance. In an extension of their original work [19], Viola and Jones proposed an asymmetric AdaBoost algorithm. All of these methods explore variations in AdaBoost-based feature selection, and their training times are similar to the original Viola-Jones algorithm.

Although the word 'asymmetric' is used in both the asymmetric AdaBoost method of Viola and Jones and our asymmetric feature selection algorithm, its implications are different within these two algorithms. In the asymmetric AdaBoost algorithm, a symmetric cost function was used. To achieve asymmetry, the weights of positive examples are increased after each weak classifier is added into the AdaBoost ensemble. In our asymmetric feature selection algorithm, we do not maintain weights for training examples. We use an asymmetric cost function, in which a false negative costs more than a false positive. The asymmetric cost function can be applied to different learning algorithms, including the AdaBoost algorithm.

While all of the above methods adopt a brute-force search strategy for generating input regions, there has been some interesting work on generating candidate hypotheses from more general interest operators. Three examples are [9, 1, 8].

6 Conclusions

Face detection is a canonical example of a rare event detection task, in which target patterns occur with much lower frequency than non-targets. It results in a challenging classifier design problem: The detection rate must be very high in order to avoid missing any rare events and the false positive rate must be very low to dodge the flood of non-events. A cascade classifier architecture is well-suited to rare event detection.

The Viola-Jones face detection framework consists of a cascade architecture, a rich over-complete feature set, and a learning algorithm based on AdaBoost. We have demonstrated that a simpler direct algorithm based on forward feature selection can produce cascades of similar quality with two orders of magnitude less computation. Our algorithm directly optimizes the learning criteria for the ensemble, while the AdaBoost-based method is more indirect. This is because the learning goal is a highly-skewed tradeoff between detection rate and false positive rate which does not fit naturally into the weighted error framework of AdaBoost. Our experiments suggest that the feature set and cascade structure in the Viola-Jones framework are the key elements in the success of the method.

We have described two variations of the direct feature selection approach, a symmetric method which weights false positives and false negatives equally, and an asymmetric approach which is based on a Bayes risk criterion that assigns different weights to these two types of mistakes. The asymmetric cost is more consistent with the def-

inition of the rare event detection problem and leads to improved performance in our experiments. We also describe some surprising preliminary experimental results on feature set size which suggest that using a greatly reduced feature set (10% of the original set) yields equivalent performance to the full set. This is further evidence for the richness of the rectangle feature set for face detection.

Three issues that we plan to explore in future work are: the necessary properties for feature sets, global feature selection methods, and the incorporation of search into the cascade framework. The rectangle feature set seems particularly well-suited for face detection. What general properties must a feature set possess to be successful in the cascade framework? In other rare event detection tasks where a large set of diverse features is not naturally available, methods to create such a feature set may be useful (e.g. the random subspace method proposed by Ho [6]).

In our current algorithm, both nodes and features are added sequentially and greedily to the cascade. More global techniques for forming ensembles could yield better results. Finally, the current detection method relies on a brute-force search strategy for generating candidate regions. We plan to explore the cascade architecture in conjunction with more general interest operators, such as those defined in [2, 8].

The authors are grateful to Mike Jones and Paul Viola for providing their training data, along with many valuable discussions. This work was supported by NSF grant IIS-0133779 and the Mitsubishi Electric Research Laboratories.

References

[1] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. PAMI*, 19(11):1300–1305, 1997.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[3] O. Carmichael and M. Hebert. Object recognition by a cascade of edge probes. In *British Machine Vision Conference*, volume 1, pages 103–112, September 2002.

[4] W. Fan, W. Lee, S. J. Stolfo, and M. Miller. A multiple model cost-sensitive approach for intrusion detection. In *Proc. 11th ECML*, 2000.

[5] B. Heisele, T. Serre, S. Mukherjee, and T. Poggio. Feature reduction and hierarchy of classifiers for fast object detection in video images. In *Proc. CVPR*, volume 2, pages 18–24, 2001.

[6] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[7] D. Keren, M. Osadchy, and C. Gotsman. Antifaces: A novel, fast method for image detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(7):747–761, 2001.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *Proc. CVPR*, 2003.

[9] T. Leung, M. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. Intl. Conf. Computer Vision*, pages 637–644, 1995.

[10] S.Z. Li, Z.Q. Zhang, Harry Shum, and H.J. Zhang. FloatBoost learning for classification. In S. Thrun S. Becker and K. Obermayer, editors, *NIPS 15*. MIT Press, December 2002.

[11] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Technical report, MRL, Intel Labs, 2002.

[12] S. Romdhani, P. Torr, B. Schoelkopf, and A. Blake. Computationally efficient face detection. In *Proc. Intl. Conf. Computer Vision*, pages 695–700, 2001.

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

[14] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statististics*, 26(5):1651–1686, 1998.

[15] Henry Schneiderman and Takeo Kanade. A statistical model for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2000.

[16] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[17] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*. Wiley, 1968.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.

[19] P. Viola and M. Jones. Fast and robust classification using asymmetric AdaBoost and a detector cascade. In *NIPS 14*, 2002.

[20] A. R. Webb. *Statistical Pattern Recognition*. Oxford University Press, New York, 1999.

[21] M.-H. Yang, D. J. Kriegman, and N. Ahujua. Detecting faces in images: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[22] M.-H. Yang, D. Roth, and N Ahuja. A snow-based face detector. In *NIPS 12*, pages 862–868, 1999.