# META ANALYSIS OF NEURALLY INSPIRED FACE DETECTION ALGORITHMS*

**Ian R. Fasel**
Department of Cognitive Science
Machine Perception Laboratory
University of California, San Diego
La Jolla, CA 92093

**Javier R. Movellan**
Department of Cognitive Science
Institute for Neural Computation
University of California San Diego
La Jolla, CA 92093

## Abstract

Face detection is a crucial technology for the development of new computer systems that interact with humans in a natural manner. Rowley, Baluja, and Kanade (1998), Roth, Yang, and Ahuja (2000) and Viola and Jones (2001) are examples of state of the art face detection systems, each of which employ a wide variety of techniques. While the development of such complete systems is a an important first step it is also crucial for the advancement in the field to analyze in detail of how the different pieces of these systems contribute to their success. In this paper we present 16 different experiments designed to perform a systematic comparison of the techniques used in some of the most succesful neurally inspired face detectors Rowley et al. (1998), Roth et al. (2000), Viola and Jones (2001). We report three main findings: First, we present confirmation of SNoW's effectiveness in the face detection task and analyze how it solves the task. Second, we find that representations based on local receptive fields like the ones used in Rowley et al. consistently provide better performance than full connectivity approaches. Third, we find that the active sampling techniques such as AdaBoost and Bootstrap consistently provide significant improvements.

## 1 Introduction

Face detection is a crucial technology for aplications such as face recognition, automatic lip-reading, and facial expression recognition Pentland, Moghaddam, & Starner, 1994; Donato, Bartlett, Hager, Ekman, & Sejnowski, 1999). Despite the strong need for good face detection systems, the taks has proven difficult and is an area of active development. One aspect that has slowed down progress in this area is the lack of meta studies whose goal is not just the development of complete systems but the analysis of how the different pieces of a system contribute to its success. The goal of this study then is to perform a systematic comparison of techniques used in three of the most succesfull neurally inspired face detection systems reported in the literature Rowley et al. (1998), Roth et al. (2000). While the high-level framework for each of these face detectors is very similar, the underlying

*This research was supported by a National Science Foundation Graduate Research Fellowship and a grant from Sony Corporation.

classifiers vary in the representations used, the use of ensemble techniques, and the use of active sampliong for improving training data.

A crucial problem in machine perception tasks, including face detection, is finding a useful image representations. A wide variety of possible representations are possible, such as raw pixel values, colors, groups of pixels or responses to different filters. Neural network research has traditionally favored low dimensional, compressed representations (i.e., far fewer dimensions than pixels), since the use of high dimensional representations have long been thought to make generalization difficult. However, several recent face detectors have been tremendously successful despite their use of representations that use far more dimensions than pixels (for each $20 \times 20$ pixel image, Viola & Jones, 2001 used 45,396 features, while Roth et al., 2000 used 102,400 features). Explanations of the ability of these systems to achieve such great success despite going strongly against the intuitions of traditional neural network research are needed.

A second issue particular to face detection tasks, is that compared to the set of all possible *nonface* images is inmense when compared to the set of possible *faces*. It is thus important to evaluate the effectiveness of that attempt to automatically find representative sample of nonfaces that will be most useful for training. These methods, which include AdaBoost (in Viola & Jones, 2001) and the "Bootstrap" method (in Rowley et al., 1998; Roth et al., 2000; Viola & Jones, 2001), allow the classifier in development to improve its performance by actively focusing attention on the most informative examples with respect to the current knowledge state during training. AdaBoost does this by altering the importance of examples in the current training set, while Bootstrap seeks out and adds new examples to the training set.

## 2 Face Detection Framework and Image Database

The face detectors used throughout this paper is based on the system described in (Rowley et al., 1998). A small window is scanned across each image and a classifier is applied to each window, returning *face* or *nonface* at that location. This is repeated at multiple scales. Finally, nearby detections are supressed using the clustering and overlap removal techniques described in Rowley et al. (1998).

For training, we randomly selected 443 frontal faces from the FERET database. Each image was manually cropped and normalized to have the same scale and position by aligning eyes, nose and center of mouth in a $20 \times 20$ window. To make the classifiers less sensitive to rotation, translation, and scale, random amounts of rotation of up to $\pm 5$ degrees, translation up to half a pixel, and scaling up to $\pm 10\%$ were added to copies of the images, expanding the training set to 8232 positive examples. For negative examples, 20,000 windows were taken from scenery images obtained through two sources: (1) random images were taken from a CCD camera of the insides and outsides of four homes, (2) nonface images randomly collected from the internet were provided by Compaq Research Laboratories. To compensate for differences in lighting and camera gains, logistic normalization (Movellan, 1995)[1] was performed on each image, with respect to an oval mask. This normalization step was also performed for each window in the detection phase.

## 3 Factors for Comparison

We constructed sixteen experimental classifiers, each using a combination of the factors used in the Rowley et al. (1998), Roth et al. (2000) and Viola and Jones (2001) face de-

[1]Using the equation $X = 1/(1 + e^{-\pi K \mu / \sqrt[3]{\sigma}})$ where $\mu$ and $\sigma$ are the mean and variance of the window, and $= 0.8$

tectors. The goal of these experiments was to clarify which particular techniques were responsible for the success of these algorithms. The techniques compared were as follows:

**Ridge Regression** [Hoerl and Kennard (1970)] This method was used for training classifiers directly on real valued pixel inputs. Ridge regression has been shown to be equivalent to weight decay (Hertz, Krogh, & Palmer, 1991) in linear networks, which can greatly improve generalization performance.

**SNoW** [Roth et al. (2000)] In contrast to using raw pixel inputs, this classifier first transforms the pixel inputs into a sparse binary representation and then uses the Winnow update rule of Littlestone (1988) for training. In effect, the resulting network performs an arbitrary function on each input pixel, then combines the function outputs linearly and applies a threshold. While this high-dimensional representation is counterintuitive to traditional neural network researchers, Roth et al. have nevertheless reported the most accurate face detector in the literature. It is thus important to replicate Roth et al.'s results in order to form a better understanding of how SNoW produces such impressive results.

**Full vs. Retinal Conectivity** [global vs. patches] Rowley et al. (1998) used a standard backpropogation network, but used retinal connections over 26 rectangular subregions inspired by Le Cun et al. (1989). The regions were 4 10x10 pixel patches, 16 5x5 pixel patches, and 6 overlapping 20x5 horizontal stripes. In each experiment, the component classifers (trained with ridge regression or SNoW) either recieved input fromm the entire image or from one of these smaller regions, which were then combined using an ensemble technique.

**Bagging** [Breiman (1996)] In this ensemble method, multiple instances of a classifier are trained on random samples from the training set. The final hypothesis of each classifier is then combined with a unity vote. This procedure has been shown to improve performance in many types of classifiers (e.g.,Breiman, 1996; Opitz & Maclin, 1999).

**Adaboost** A modification of Bagging, AdaBoost (Freund & Schapire, 1996) trains an ensemble of classifiers sequentially. For each round of boosting, a distribution over the training set is modified so that examples misclassified in previous rounds of boosting receive more emphasis in later rounds. This procedure guarantees an exponentially decreasing upper bound on training error, and in practice AdaBoost is highly resistant to overfitting(Opitz & Maclin, 1999; Schapire & Singer, 1998.

**AdaBoost and Bagging for Feature Selection** Tieu and Viola (2000) and Viola and Jones (2001) used AdaBoost as a method for selecting a few key features from a large set of possible features by constraining the weak learners to make their decision using only one feature at a time. The procedure is as follows:

1. For each feature, train a classifier that uses only that single feature as input.
2. Pick the classifier with the best performance with respect to the current distribution over the training set.
3. Using the AdaBoost equations, choose a weight for that classifier and update the distribution over the training set.
4. Remove the feature just used from the set of possible features and go back to step 1. Repeat until the generalization error of the ensemble is satisfactory, or all the features have been used. The classifiers/features are combined as in AdaBoost.

Using this technique, Viola and Jones (2001) were able to select about 200 features from their initial set of 45,396 to build a high performance face detector. We tested the flexibility of this technique by using the Rowley rectangular regions trained with SNoW or AdaBoost as the basic features. We also tried replacing AdaBoost with Bagging in this algorithm.

**Bootstrap for single classifiers** [Bootstrap] Rowley et al. (1998), Roth et al. (2000) and

Viola and Jones (2001) all used a "Bootstrap" technique based on Sung and Poggio (1994). The Bootstrap technique is an active sampling technique for expanding the training set of a classifier during training. Bootstrap begins by training a classifier on the full set of face examples and a random set of 8000 nonface examples. This classifier is then used in a face detector on a set of unseen scenery images, and 2000 of the false alarms are randomly selected and added back into the training set. The existing classifier is then discarded, and a new classifier is trained on this expanded training set. The process repeats until the classifier has satisfactory performance.

**Bootstrap for ensemble classifiers** [Bootstrap + Bagging] We created a novel condition in which the classifier at each round of Bootstrap is saved, and the resulting classifiers are combined with a unity vote. This is similar to Bagging, but with active sampling instead of random sampling, and makes for a fairer comparison with other ensemble techniques.

We trained different classifiers from different combinations of these methods in order to tease apart the role each method plays in the success of a face detector. Not all of possible combinations of these methods could be practically tested against all others; when we had to make choices, we focused our efforts on those areas which had the greater scientific interest. For instance, we were particularly interested in providing an intuitive example of how SNoW works, since SNoW is intuitively considered by many in the computer vision community to be an unlikely candidate for good performance in this task. In particular, we focused on the following questions: (1) Is the patch-based representation proposed by Rowley helpful? (2) Does SNoW really work? How? (3) How helpful are AdaBoost and Bagging in the face detection task? (4) How crucial is the Bootstrap method?

## 4 Results and Discussion

For each experimental classifier, our performance measure was the total error rate on a generalization set of 4200 unseen face and nonface examples; this measure seems appropriate since the classifiers were trained to minimize overall error. Table 1 shows these results for all the conditions, sorted in order of decreasing error rate. The three main findings were (1) SNoW consistently performed among the best classifiers, confirming the results of Roth et al. (2000). In addition, we found an intuitive explanation for how SNoW works, described below. (2) The Rowley rectangle inputs consistently improved performance over equivalent classifiers that used the full $20 \times 20$ input. (3) Active sampling consistently improved performance as well; AdaBoost was always superior to the equivalent network using Bagging, and Bootstrap was usually superior to the equivalent networks that didn't use Bootstrap.

### 4.1 How SNoW Works

SNoW was used in four of the best five experimental classifiers, demonstrating its strength in the face detection task. Figures 1 and 2 shows different attempts to visualize the representation learned by SNoW. The left image in Figure 1 shows the weights of the ridge regression network, and the center image shows the intensity corresponding to the peak weight in each pixel of the SNoW network. This image represents the SNoW's "favorite" face, i.e., the pattern of pixel values that maximizes the output of the SNoW model. Clearly, at the surface level, SNoW has learned a "favorite" face that is very similar to the favorite face of the linear network. The rightmost image in Figure 1 shows the sum of the weights for each pixel, which Figure 2 shows in greater detail. This image represents the importance, or attentional strength assigned by SNoW to each pixel region. From this image it is clear that the areas where SNoW has developed large weights correspond closely to recognizable facial features, while surrounding weights have been lowered close to zero. Figure 2, display the tuning curve learned by SNoW for each different pixel position. Note

| Condition | Total Error | Hit Rate | False Alarm Rate |
|---|---|---|---|
| 1) Patches + Ridge + Bagging + Bootstrap | 47.85 % | 52.75% | 48.21% |
| 2) Global + Ridge | 21.95 % | 96.00% | 32.46% |
| 3) Global + Ridge + Bagging | 21.86 % | 96.00% | 32.00% |
| 4) Global + Ridge + Bootstrap | 11.53 % | 92.75% | 14.03% |
| 5) Patches + Ridge + Bagging | 6.47 % | 99.75% | 10.09% |
| 6) Global + Ridge + Bagging + Bootstrap | 2.28 % | 97.75% | 2.30% |
| 7) Global + SNoW | 0.64 % | 98.50% | 0.15% |
| 8) Global + SNoW + Bagging | 0.46 % | 99.00% | 0.15% |
| 9) Global + SNoW + Bootstrap | 0.35 % | 99.75% | 0.40% |
| 10) Global + SNoW + Bagging + Bootstrap | 0.21 % | 99.50% | 0.04% |
| 11) Global + Ridge + AdaBoost | 0.18 % | 99.50% | 0.00% |
| 12) Global + SNoW + AdaBoost | 0.16 % | 99.75% | 0.15% |
| 13) Patches + SNoW + Bagging | 0.16 % | 99.75% | 0.11% |
| 14) Patches + SNoW + Bagging + Bootstrap | 0.12 % | 99.75% | 0.04% |
| 15) Patches + SNoW + AdaBoost | 0.12 % | 99.75% | 0.04% |
| 16) Patches + Ridge + AdaBoost | 0.09 % | 99.75% | 0.00% |

Table 1: Performance on generalization set.

that all the important pixels have have unimodal tuning functions with a range of preferred intensities. The fact that the tuning curves developed by SNoW are unimodal is interesting, because SNoW could have developed arbitrary tuning curves, such as linearly increasing or decreasing weights (which would be indentical to the ridge regression solution). This also suggests a possible architecture for an improved face detector: since the SNoW weights resemble bandpass tuning functions, a classifier that explicitly uses such tuning functions in training may be able to perform even better.

Figure 1: Left: The weights from the ridge regression network in experiment (1). Brighter pixels are more positive and darker pixels are more negative. Center: The "favorite" pixel intensities from the SNoW network in experiment (4). Notice the similarity to the ridge regression weights. Right: The sum of the weights for each pixel in SNoW. The brighter a pixel is, the more important it is for that pixel to be close to the "favorite" intensity. The high value pixels SNoW focuses on correspond to facial features such as eyes, bridge of the nose, nostrils, cheeks and forehead.

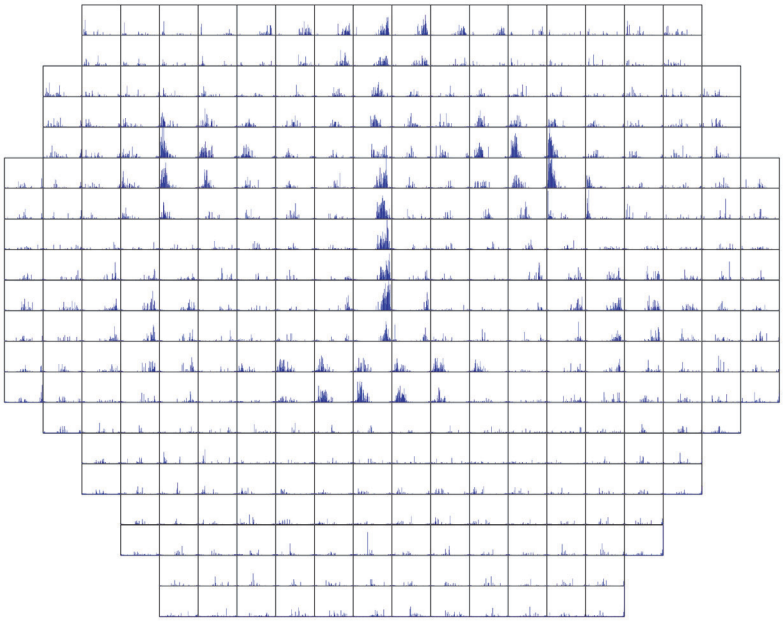

Figure 2: SNoW generated weights from experiment (4). Each box represents a pixel, showing the weights learned by SNoW for that pixel. The x axis is the intensity level and the y axis is the magnitude of the weights. For most pixels in which the weights are not close to zero, the weights resemble a tuning curve, or perhaps a bandpass filter.

### 4.2 Local Connectivity is Better than Full Connectivity

Ensemble classifiers that split the input into the Rowley et al. patches typically performed better than classifiers that used full connectivity for every sub classifier. The four best performing classifiers all used local patches, while the classifier in experiment (8) had more than 300% lower error rate than the corresponding classifier in experiment (2), which used the global input but was otherwise identical. The conclusion we can draw from this is that the face detection task truly does benefit from the use of local receptive fields, like the ones used in Rowley. Adding retinal connectivity to the ridge regression based ensemble classifiers improved performance over the globally connected but otherwise identical networks enough to produce the best overall classifier in the study. In addition, the representation developed by SNoW could be considered the ultimate localist representation, i.e., each individual pixel goes through a non-linearity (the equivalent of a hidden unit) before is integrated linearly by the output layer.

### 4.3 Active Sampling

The active sampling done by AdaBoost and the Bootstrap method improved performance over their random sampling counterparts in all but one condition. AdaBoost was a factor in the two best classifiers using patches and the two best classifiers using global inputs, Interestingly, while AdaBoost helped in all cases, it was only slightly better than Bagging when used on SNoW. It seems that SNoW is able to account for most of the variation in the training set on the first round of Boosting, so that the impact of the active sampling done by AdaBoost is minimal. In contrast, AdaBoost provided huge benefits to the ridge regression classifiers, which ultimately performed as well as or better than the SNoW classifiers.

Adding Bootstrap to a classifier also improved performance, although to a slightly lesser degree than AdaBoost. The one exception to this was Bootstrap with ridge regression classifiers using patches. Our hypothesis is that the changing number of training examples as Bootstrap progresses interacts negatively with the optimal choice of the regularization parameter in the ridge regression. Online adjustment of the regularization term during training may alleviate this problem.

## 5 Conclusions and Future Work

This study provides clear evidence of the usefulness of some of the techniques used in face detection and suggests several areas for future improvements. First, we found that SNoW is indeed a good classifier for face detection. The analysis of the way SNoW solved the problem suggests that a powerful face detectors may be built using explicit intensity tuning functions. Second, the superiority of sparse local representations, especially when used with the AdaBoost feature selection method, supports the exploration of other localist representations. Finally, the improvements provided by active sampling methods, like Bootstrap has exciting implications for the role of active sampling in otger machine perception tasks.

## References


Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123-140.

Donato, G., Bartlett, M., Hager, J., Ekman, P., & Sejnowski, T. (1999). Classifying facial actions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *21*(10), 974–989.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. 13th international conference on machine learning* (p. 148-146). Morgan Kaufmann.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural computation.* Redwood City, CA: Addison-Wesley Publishing Company, Inc.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55-67.

Le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Backpropogation applied to handwritten zip code recognition. *Neural Computation*, *1*, 541–551.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linearthreshold algorithm. *Machine Learning*, *2*, 285-318.

Movellan, J. R. (1995). Visual speech recognition with stocastic networks. In . T. G. Tesauro, D. Toruetzky (Ed.), *Advances in neural information processing systems* (Vol. 7). MIT Press.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169-198.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and modular eigenspaces for face recognition. In *IEEE conference on computer vision and pattern recognition.*

Roth, D., Yang, M., & Ahuja, N. (2000). A snow-based face detector. *In NIPS-12. To Appear.*

Rowley, H., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *1*(20), 23-28.

Schapire, & Singer. (1998). Improved boosting algorithms using confidence-rated predictions. In *COLT: Proceedings of the workshop on computational learning theory.* Morgan Kaufmann.

Sung, K. K., & Poggio, T. (1994). *Example based learning for view-based human face detection* (Tech. Rep. No. AIM-1521).

Tieu, K., & Viola, P. (2000). Boosting image retrieval. In *Proceedings ieee conf. on computer vision and pattern recognition.*

Viola, P., & Jones, M. (2001). *Robust real-time object detection* (Tech. Rep. No. CRL 20001/01). Cambridge ResearchLaboratory.